Deployment Guide

Sun™ ONE Portal Server

Version 6.2

816-6760-10
March 2004

Contents

List of Figures

List of Tables

List of Procedures

About This Guide

This guide explains how to plan for and deploy Sun™ ONE Portal Server 6.2 software. 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
- Document Conventions Used in This Guide
- 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™ ONE Identity Server
- Sun™ ONE Directory Server
- Sun™ ONE 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, "Overview of Sun ONE Portal Server"

  This chapter describes the basic ideas you need to understand before designing your portal.

- Chapter 2, "Sun ONE Portal Server Architecture"

  This chapter describes the architecture, protocols, interfaces, directory structure, deployment, and customization of the Portal Server 6.2 product.

- Chapter 3, "Sun ONE 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 4, "Analyzing Your Portal Requirements"

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

- Chapter 5, "Sizing Your Portal"

  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 6, "Understanding the Portal Deployment Life Cycle"

  This chapter provides an overview of the portal deployment process.

- Chapter 7, "Creating Your Portal Design"

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

- Chapter 8, "Monitoring and Tuning Your Portal"

  This chapter describes how to monitor and tune your portal.

- Appendix A, "Troubleshooting Your Portal Deployment"

  This appendix describes how to troubleshoot Portal Server and Secure Remote Access.

- Appendix B, "Portal Deployment Worksheets"

  This appendix provides various worksheets to help in the deployment process.

- Appendix C, "Portal Server and Application Servers"

  This appendix provides an overview of Portal Server and its support for various application servers.

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:

- *Sun ONE Portal Server 6.2 Administrator's Guide*
- *Sun ONE Portal Server 6.2 Developer's Guide*
- *Sun ONE Portal Server 6.2 Installation Guide*
- *Sun ONE Portal Server 6.2 Migration Guide*
- *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide*

Where to Find This Guide Online

You can find the *Sun ONE Portal Server 6.2 Deployment Guide* online in PDF and HTML formats. This book can be found at the following URL:

http://docs.sun.com

Chapter 1

Overview of Sun ONE Portal Server

The Sun™ ONE 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™ ONE Portal Server, Secure Remote Access (SRA) product provides additional secure remote access capabilities to access web- and non-web enabled resources.

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.

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.

This chapter describes the basic ideas you need to understand before designing your portal. This chapter contains the following sections:

- Understanding Portal Server
- Independent Software Vendor Integrations with Portal Server
- Types of Portal Deployments
- Portal Deployment Architecture
- Establishing Quality Goals

Understanding Portal Server

To begin understanding Portal Server, and how it fits in your organization, use this section to gather the necessary background information on the Portal Server product and lifecycle.

What Is a Portal?

A portal is an entry point to a set of resources that an enterprise wants to make available to the portal's users. For some consumer portals, the set of resources includes the entire World-Wide Web. For most enterprise portals, the set of resources includes information, applications, and other resources that are specific to the relationship between the user and the enterprise. For service providers, the portal provides a point of entry to customer service applications as well as a controlled content aggregation service.

In general, a portal enables users to:

- Customize the available data content
- Change how channels are generated
- Control how data is displayed
- Create and manage links to other allowable web sites

Resources can include the use of provider applications and utilities such as mail, file management, and storage facilities.

Overview of a Portal Server 6.2 Deployment

A Portal Server 6.2 deployment consists of:

- Sun™ ONE Portal Server
- Sun™ ONE Directory Server
- Sun™ ONE Identity Server
- Sun™ ONE Web Server, or an application server, such as Sun™ ONE Application Server, BEA WebLogic Server™, or IBM WebSphere® Application Server

The combination of the above software 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

Examples of How Portal Server 6.2 Satisfies Business Needs

Depending on your organization's requirements and business needs, your portal deployment will vary. The following section provides a high-level look at how three organizations have deployed Portal Server. Use the information in this section to generate ideas that will help you more effectively deploy Portal Server in your organization.

Business-to-Employee Portal

This multinational company, which manufactures a wide range of products, has hundreds of thousands of employees located around the world, grouped in hundreds of business units. Thus, the company has a highly-distributed computing environment.

Previously, the company relied on a static portal for employee communications, which proved inefficient in meeting its business needs. The company decided to move to a dynamic portal where employees could get personalized access to company information. The portal needed to be configured to support multiple organizations and user roles, and to provide access to internal sites from the company's intranet.

Table 1-1 summarizes this organization's goals and presents the Portal Server feature or capability that meets this goal.

Table 1-1 Business-to-Employee Portal Server Example Goals

Goals	Portal Server Feature or Benefit That Meets This Goal
Improve information delivery to the company's distributed workforce	With technology changing at an ever-increasing speed, the company needed a solution that would facilitate knowledge sharing and collaboration among its employees. Divisions within the company are spread across disparate geographic locations and time zones, and so the portal aggregated content from disparate sources. The portal provided employees with a single point for accessing all their applications, content, and services.
Enable employees to authenticate just once	Portal Server's single sign-on (SSO) and authentication were implemented with an existing Secure ID infrastructure for security and authentication, both to the portal and other existing internal sites. Users only need sign on one time and are authenticated to all appropriate internal sites and applications.
Improve employee-to-employee communication	By bridging together different sources of information in their native form, workers were able to easily share information and collaborate in real time by using web-based email and calendaring.
Provide a Portal Desktop for a variety of purposes	The Portal Desktop was configured in such a way to use color to indicate channel function (personal, corporate, or business-related). The Portal Desktop also included mandatory channels required for all users, which are not removable by users; default channels, which are automatically selected by an administrator; and available channels, which are created by global or delegated administrators, that users can select from.
Provide for high-availability and reliable access across geographic sites	To meet these requirements, this company developed a portal architecture that included three separate geographic installations, failover between these main locations, and load balancing within each installation. LDAP replicas are placed at each installation, obtaining their information from a master LDAP at the main data center, which is also configured for HA failover.
Implement a centralized administrative model that also enables delegated administration	Identity Server 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. For example, within business units, delegated administrators can add or remove channels for their particular line of business.

Business-to-Consumer Portal

This travel company markets various vacation destinations and ancillary businesses, and needed a business-to-consumer portal to help develop a direct relationship with end customers. The portal needed to serve as the mechanism to execute a strategy moving away from travel agencies to a direct consumer relationship.

Table 1-2 summarizes this organization's goals and presents the Portal Server feature or capability that meets this goal. In this table, the first column gives the goal. The second column describes the Portal Server feature or benefit that meets this goal.

Table 1-2 Business-to-Consumer Portal Server Example Goals

Goals	Portal Server Feature or Benefit That Meets This Goal
Understand customers better to make it easier to do business with them (serve their needs)	Portal Server enabled the consolidation of numerous applications, providing single-point access for both customers and employees. The company saw immediate benefits including strengthened customer service relationships due to this consolidation.
Save money	Because the company was spending large amounts of money on IT, with no slowdown foreseen in the future, the company needed to find a way to lower expenses. The portal provided a single point of system management, helping to reduce the company's IT complexity and cost. In addition, rather than manage multiple portal projects that might later need to be combined using expensive consulting services that will duplicate effort and infrastructure, Portal Server empowers and enables IT administrators to learn and manage a single platform.
Offer new products or services to customers	Customers access the company's offerings through a web services portal, which provides access to a broad range of internal and external tools including market research, personalized feeds, pricing and configuration, product data, and presentations. This centralized services portal boosts productivity by giving customers a single point of access for all data, services, and online resources.
Communicate new information consistently and quickly	The portal solution deploys technology that enabled this company to lower operating costs while delivering personalized content to its customers. Users were able to customize their portals to deliver the most important and relevant information to them. The portal also provides extended features (such as customer support) to registered customers.

Internet Service Provider Portal

This company provides a full range of telecommunications services and products, is a leading service provider in its country, and needed to quickly transition from a telephone services company to a communications service provider. In response to new customer demands, as well as strong competitive threats, the company decided to build a solution that would include an affordable Internet access as well as small business services such as fax and email, supplementing and eventually replacing standard telephone services.

The company developed a solution that was a comprehensive, end-to-end portal framework to delivery Internet service and applications.

Table 1-3 summarizes this organization's goals and presents the Portal Server feature or capability that meets this goal.

Table 1-3 Internet Service Provider Portal Server Example Goals

Goals	Portal Server Feature or Benefit That Meets This Goal
Ease of development	J2EE™, LDAP, and XML standards enabled the organization to leverage existing investments. In addition, Portal Server provided a reliable foundation with integrated transaction management, load balancing, and failover capabilities for the delivery of J2EE-technology-based applications.
Ready adaptability of system components	The flexible design enabled the organization to create a system blueprint, allowing the architecture to be applied to multiple industries and customers.
Scalability and availability of IT architecture	The portal architecture easily scaled both horizontally and vertically without sacrificing performance. The company has the capacity of growing and handling a customer base of hundreds of thousands of subscribers in conjunction with thousands of concurrent users. As the company adds more servers to its infrastructure, it will have the capacity of supporting more than one million users. Other major benefits include availability and reliability. The company is now available 99.99 percent of the time.
Secure Web-enabled transactions	The organization's assets and confidentiality are protected by transacting vital information securely over the Web.
More intuitive system management	The portal architecture was easy to administer, making it both organized and controllable. The delegated administration feature enabled the service provider to share directory administration with its customers. As a result, customers were able to achieve the flexibility necessary to manage their own directories privately and securely.
Faster time to market	Because there was an integration methodology in place, implementing best-of-breed commercial off-the-shelf products now requires less time, and generates greater return on investment (ROI).

Portal Server Life Cycle

The previous section illustrates an important point in deploying Portal Server, namely the Portal Server product life cycle. In general, any product deployment can be broken down into the following sequence of events, or life cycle:

- **Planning**—In the planning phase, your organization and your Sun ONE representatives work to understand your business and its needs, establish business objectives, and scope and collect requirements.
- **Developing**—In this phase, your organization develops an overall portal design based on the requirements you have established and your deployment estimates.

- **Designing, Building, and Testing**—Once you have arrived at an overall architecture, you can begin designing, building, and testing your portal.
- **Deploying**—In this phase, you install a server instance as a trial and test whether your portal can handle your user load. If the portal is not adequate as it is, you then adjust your design and test the trial again. Adjust your trial design until you have a robust portal that you can confidently introduce to your organization.
- **Production**—Once you have put your portal through a trial run and tuned the portal, you need to develop and execute a plan for taking the portal from trial to production.

This guide attempts to use this life cycle to ensure the success of your portal deployment. See Chapter 6, "Understanding the Portal Deployment Life Cycle" for more information on managing a portal project.

Portal Server Resources

This section provides general information about Portal Server resources. See Chapter 2, "Sun ONE Portal Server Architecture" for a complete architectural description.

JavaServer Pages Technology

To generate the rendered Portal Desktop user interface (what the industry refers to as the "presentation"), Portal Server makes use of either JavaServer Pages™ (JSP™) technology or template files (HTML). JSP technology is preferred because it enables a much easier customization process without having to change the provider Java™ classes. JSP technology also provides a way to enable a strict separation of business and presentation logic. Specifically, this means having the business logic in the provider classes and presentation logic in JSP technology.

In JavaServer Pages technology, actions are elements that can create and access programming language objects and affect the output stream. JSP technology supports reusable modules called custom actions. You invoke a custom action by using a custom tag in a JSP file. A tag library is a collection of custom tags. The Portal Desktop custom tag library contains tags that you use to perform Desktop operations for JSP code.

Before tag libraries, JSP code was difficult to maintain because you were forced to use JavaBeans™ components and scriptlets as the main mechanism for performing tasks. Custom actions, that is, a tag library, alleviate this problem by bringing the benefits of another level of componentization to JSP code. A tag library encapsulates recurring tasks so that they can be reused across more than one application.

The Portal Server Desktop tag library consists of six parts:

- Core tags that can be used on any provider or container that implement the Provider Application Programming Interface (PAPI)
- Tags that can be used to operate on a provider or container that support the `ProviderContext` and `ContainerProviderContext` interfaces
- Tags that operate on specific container building-block providers (such as `SingleContainer`, `TableContainer`, `TabContainer`)
- JSP Standard tag libraries from Apache
- Tags that support the Search function
- Tags that provide theme support in the Portal Desktop

See the *Sun ONE Portal Server 6.0 Desktop Customization Guide* for more information on JSP technology and Portal Server.

Portal Desktop Content

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. You can edit the display profile and other Portal Desktop service data through the Identity Server administration console.

The Portal Desktop displays portlets which are pluggable web components that process requests and generate content within the context of a portal. In the Sun ONE Portal Server software, portlets are managed by the Portlet Container. Conceptually, portlets are equivalent to the Providers. Sun ONE portlets are JSR 168 compliant.

Configuration Data

As an Identity Server application, Portal Server defines services that are managed using the Identity Server service management system. Generally, any service-related data that is not server-specific is stored in the directory service. Server-specific data can be stored in properties files that are local to the specific server.

In addition, Portal Server uses certain files to manage the configuration of the Portal Desktop and Search services. The Portal Desktop configuration file, `desktopconfig.properties`, defines server-specific parameters.

The Search service uses the following configuration files: `classification.conf`, `filter.conf`, `filterrules.conf`, and `robot.conf` files. The `convert.conf` and `import.conf` files are generated by the Search server. Do not manually edit these files. The `search.conf` file lists all the specific search values you have set.

At installation time, you are given the option of defining values or using the default values for the base directory (`/opt`), the deployment URI (`/portal`) and the deploy instance (`cate.sesta.com`).

See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information on product configuration files.

Application Data

Portal Server stores certain data in the user's profile that is passed to back-end applications. For example, the User Preference channel stores NetMail service data (user preferences for using NetMail). Application data also includes Rewriter rulesets.

Site Data

Portal Server uses the local file system to store data specific to a particular instance or node. Site data includes the *identity-server-install-root*`/SUNWam/lib/AMConfig.properties` file and the `/etc/opt/SUNWps/desktop/desktopconfig.properties file` file.

Portal Server, Secure Remote Access

The Portal Server, Secure Remote Access (SRA) offers browser-based secure remote access to portal content and services from any remote browser. SRA is a cost-effective, secure access solution that is accessible to users from any browser enabled with Java technology. SRA eliminates the need for additional client software. Because SRA is integrated with Portal Server, users receive secure encrypted access to the content and services that they have permission to access.

Using SRA, you can install your portal in *secure mode*. Secure mode provides users with secure remote access to required intranet file systems and applications.

Secure mode uses the SRA gateway, which typically 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 Portal Server services such as Session, Authentication, and the Portal Desktop reside behind the DMZ in the secured intranet. Communication from the client browser to the gateway is encrypted using HTTPS (over Secure Sockets Layer). Communication from the gateway to the server and intranet resources can be either HTTP or HTTPS.

See Chapter 3, “Sun ONE Portal Server, Secure Remote Access Architecture” for more information.

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.

Migrating to a New Version of Portal Server

Migrating from Portal Server 3.0 to Portal Server 6.2 requires a different set of deployment requirements that are outside the scope of this document. Several new features in Portal Server 6.2 require format changes in the data store of Portal Server 3.0 because of the new access layer and Identity Server APIs that Portal Server now uses.

The Portal Server 3.0 Data Migration Tool Suite provided with Portal Server 6.2 enables you to migrate the following:

- LDAP data
- Templates

- JavaServer Pages code
- Resource bundles
- Custom authentication module data
- Properties files from Portal Server 3.0 to Portal Server 6.2 or to Identity Server 6.1 as necessary
- Certificates (with the exception of gateway certificates)

  The Portal Server 3.0 Data Migration Tool Suite does not migrate gateway certificates. See the *Sun ONE Portal Server 6.2 Migration Guide* for more information.

Independent Software Vendor Integrations with Portal Server

This section provides an overview of some of the independent software vendor (ISV) integrations that exist for Portal Server.

Integration Types

Listed below are some types of Portal Server integration.

- **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 includes 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™ ONE 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 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 and no applets will appear.

The following sections provide a look at some of the ISVs by category.

Collaboration and Application Emulation ISVs

ISVs in this category include:

- **Citrix NFuse**—Solves access-related problems by delivering a suite of software products and services that provide access to any device, over any network to any application or information source. Portal Server integrates with Citrix's NFuse through a portal channel, the NFuse provider. NFuse enables web-based access to applications running on MetaFrame servers in a Citrix server farm. Portal Server's integration works for both open and secure portal modes. The NFuse provider is an icon-based interface with features that can be personalized.
- **Tarantella**—Provides browser-based access to back-end applications via the portal, without any code modifications. Tarantella immediately web-enables UNIX®, Linux, Windows, mainframe, AS/400, and Java applications without rewriting code, changing the architecture, or touching the infrastructure. All of these capabilities can be displayed through Portal Server.

  In conjunction with SRA, the Tarantella Integration Pack for Portal Server is designed to enable Tarantella traffic to tunnel through the secure channel provided by the Netlet. The Tarantella Integration Pack provides single sign-on authentication for the portal user and channel integration. Users are authenticated once, when they Login to the portal, and all permitted applications are available to them immediately.

Content and Document Management ISVs

Most portals provide some support for content management. However, in general, analysts agree that portals need to be supplemented by a dedicated content management system (CMS). While portals usually handle content through search and display functions, they generally do not provide for creating and adding portal content. This is where a content management system comes in.

ISVs in this category include:

- **Documentum**—Provides automated processes for production, review, approval, and publishing of document assets within an organization. Documentum enables existing applications (for example, ERP/CRM) with enterprise content, managing business critical documentation.
- **FatWire**—As a portal content management system, FatWire Spark pCM installs and runs within the Portal Server system. It provides portlets (channels) for all three content management processes:

  - CM Home Channel—Provides the business interfaces for creating and managing content with a wide array of data types.
  - CM Control Center Channel—Provides tools for administrative functions, such as managing users and groups, creating content folders, and selecting workflows.
  - Content-rich display Channels—SparkpCM is delivered with a selection of portlets for content display. SparkpCM also includes developer interfaces and integration for rapidly building additional portlets (channels) within the portal framework.

- **Interwoven**—Provides content management software and services for the enterprise web. TeamSite from Interwoven is usually installed on separate machines. The providers are installed on the machine hosting Portal Server.

  Generally, TeamSite users have a TeamSite login and password. The first time users log in to Portal Server they must set their login information and TeamSite workarea into TeamSiteInfo provider. This information is used by all the TeamSite providers to authenticate with the TeamSite server. TeamSite channels are not initially configured, and display an error status the first time users access them. Once authentication is successful, the sample providers appear without any error status.

- **Vignette**—Provides content management across multiple web sites, content inheritance by child sites from parent sites, personalization of content, and role- and policy-based access via Portal Server. Provides various prebuilt portlets (channels) for distribution within Portal Server, including:

  - Vignette Content Contribution Channel—Displays content types that can be created by the authorized user
  - Vignette Content Inbox Channel—Manages the workflow and tasks assigned to the authorized user, and enables the user to preview the content and manage the content's status
  - Vignette Site and Channel Management Channel—Displays the site and channel management console, and enables the user to manage content across multiple sites and multiple channels

Content Syndication ISVs

ISVs in this category include:

- **YellowBrix—**Provides authoritative, real time, industry specific information used by global organizations to strengthen business decision making, gain competitive advantage, improve employee productivity, expand customer relationships, identify new opportunities, and drive new revenue streams.
- **Pinnacor (formerly Screaming Media)—**Delivers syndicated content to portal channels, including news stories from nearly 3,000 premium newswires, newspapers, magazine and trade journals, categorized and delivered to your specifications. Pinnacor content providers leverage all the functionality of Portal Server to deliver a high return on your portal investment. Modular content provider offerings deploy rapidly, run efficiently with the existing portal framework, and require minimal maintenance with no additional hardware. Using Pinnacor, you can create a targeted experience with single sign-on and multilevel personalization, without extraneous advertising or link-outs.

Enterprise Applications ISVs

ISVs in this category include:

- **PeopleSoft**—The PeopleSoft CRM connector for Portal Server enables portal users to access the PeopleSoft CRM application in Portal Server channels with single sign-on. The PeopleSoft CRM functionality exposed through the connector includes Customer Contact Information management and Opportunity Management.
- **SAP**—The integration with SAP can be achieved by using the Java Connector Architecture (JCA). Several vendors write these connectors (for example, Altio, Insevo). True SSO is not possible at this point since SAP is not Identity Server aware.
- **Siebel**—The Siebel eService and ERM Portal Connectors for Portal Server enable users to access their Siebel ERM (Employee Relationship Management) and eService applications through Portal Server channels.

  The eService connector pack installs on Portal Server and provides single sign-on with the back-end Siebel eService application. The eService channels enable users to submit and check status of service requests, view online knowledgebases, and provide communication services that organize email, postal mail, and structured feedback forms.

  The ERM connector provides access to Employee Relationship Management applications including Online Helpdesk, Employee opportunities and projects management, Performance Management, and Internal News Administration.

Personalization, Business Intelligence, and Analysis ISV

The ISV in this category is:

- **Fair, Isaac**—Provides an integration module that enables Blaze Advisor to provide rules-based personalization and other services to Portal Server. Using this module, you can take advantage of the Blaze Advisor rule syntax and sophisticated design tools while giving portal administrators and users the power to change their personalization rules using simple web-based interfaces. The scope of the integration is focussed on the following:
    - Personalization based on the user profile (both internal and external). The internal profile is the profile stored on Portal Server. The external profile is stored as part of some other data store to which the Blaze rules engine and Portal Server have access to. This kind of personalization determines the content to be displayed on individual channels and can also be used to determine the combination of channels to be displayed, based on the user.
    - Personalization based on preferences of other users with similar profiles or preferences.
    - Personalization based on the click stream analysis and history of usage.

Rapid Portlet and Web Services Development ISVs

ISVs in this category include:

- **Altio**—AltioLive provides a browser-based visual development environment to create channels with greater interactivity than HTML. AltioLive can combine data from multiple systems into the development environment. Developers can link disparate data sources for display in a single portlet (channel).
- **Cysive, Inc**.—is a provider of Interaction Server software that allows enterprise customers to interact with customers, partners, and employees over multiple communications channels.

Types of Portal Deployments

Three general types of portals are in use today: business-to-employee (B2E), business-to-consumer (B2C), and business-to-business (B2B). Each type has its own special needs, and Portal Server has features to support each type.

NOTE	Another type of portal that deserves mention is business-to-everyone, usually implemented by carriers and ISPs.

The following sections describe the various types of portals.

Business-to-Employee Portal (B2E)

B2E portals provide a collection of information and applications from the company’s internal network. These portal services are accessed by employees in their offices as well as by remote, travelling, and telecommuting employees from any web-enabled browser on the Internet. B2E portals include features such as:

- Access to applications, including mail, calendaring, and file browsing
- Server access through X Window System, Citrix, and Telnet protocols
- Access to up-to-date company information, including press releases
- News feeds from outside the company

- User-specified features such as local weather reports

Portal Server enables companies to establish secure employee portals using existing enterprise authentication mechanisms and additional one-time password and certificate-based authentication for Internet-based access. Furthermore, Portal Server is capable of presenting employee portals on the intranet using only standard HTTP port 80, and on the Internet using only secure HTTPS on port 443.

NOTE When deploying a B2E portal, you can use SRA to install a gateway, if desired. However, most often a B2E portal is only accessible behind a firewall, so SRA is not required.

Business-to-Consumer Portal (B2C)

B2C portals generally grant access to anyone on the Internet, without using secure authentication and encrypted communication. These portals typically sell products and services to anyone visiting the site. B2C portals often provide extended features (such as customer support) to registered customers, who also might or might not be paying customers. It is well known that the longer a user visits a site the more likely it is for a purchase to be made. Thus, many portals have increased their “stickiness” through the addition of syndicated content that helps to prolong site visits.

The Portal Server architecture enables companies to build B2C portals by extending Sun ONE or third-party commerce applications to customers on any web-enabled browser. Portal Server’s membership management services can be used to help build user communities through self-administered membership modules. Management services can also enforce policy-based access so that enhanced services are only provided to customers who have paid for them. You incorporate applications and content into B2C portals through channels that can be configured both by the hosting company and by individuals. Giving users power to control their portal experience increases the likelihood of return visits. To further increase site stickiness, you can configure search engines and syndicated content (such as news feeds) for user access.

NOTE Open anonymous mode is a good example of how B2C portals enable non-personalized (non-profiled) access.

Business-to-Business Portal (B2B)

B2B portals establish extranet connections through which companies and their suppliers and partners can more effectively communicate and collaborate. Suppliers can better match supplies to demand when they have direct access to ERP systems that handle the sales and production process. Consultants can be more effective when they have direct access to engineering specifications and diagrams. And company accountants can more quickly meet tax deadlines when they can get data directly from company accounting systems. Because B2B portals are designed for sharing business-critical information with third parties, security is of paramount importance. B2B portals must provide the means to authenticate the identity of their visitors, and once access is authorized, securely encrypt the data as it passes between the portal and the authorized users.

When used to support B2B portals, Portal Server can be configured to use strong authentication techniques ranging from one-time passwords to unforgeable X.509 certificates. Even before the authentication process is initiated, connections to Portal Server can be encrypted with HTTPS sessions with keys up to 128 bits in length. Once users are authorized, Portal Server can provide access to company information based on the user’s identity, group, or organization. User access can be as fine-grained as is necessary for your site.

NOTE Because security is so important for B2B portals, you need to deploy a secure portal running SRA for HTTPS sessions. See Chapter 3, “Sun ONE Portal Server, Secure Remote Access Architecture” for more information.

Portal Server can provide access to just about any kind of information that business partners need. Access to UNIX and Microsoft Windows applications is provided through Citrix technologies. Applications using Java technology applets and even proprietary protocols can be supported through SRA Netlet software. Terminal emulation is also available, giving partners access to command-line interfaces ranging from standard Telnet to mainframe applications.

Portal Deployment 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 node**—The server upon which you install the Portal Server, Sun ONE Web Server or other web container, and Identity Server software. You can also install the Search component on this node if desired.
- **Search node**—Optional. The server you use for the Portal Server Search component. You can install the Portal Server Search component on its own server for performance, scalability, and availability reasons.
- **Gateway node**—Optional. The server upon which you install the SRA gateway component. 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.
- **Reverse Proxy node**—The server that receives a request, checks the session validity and forwards the request to the server as specified by the HTTP header. Upon receiving a response from the server, the Reverse proxy translates the response so that all intranet links within the response work on the extranet.
- **Directory server**—The server running Sun ONE Directory Server software. You can install Directory Server on a separate node (a non-Portal Server node).
- **Other servers**—These servers, such as mail, file, and legacy servers, provide backend support, data, and applications to portal users.

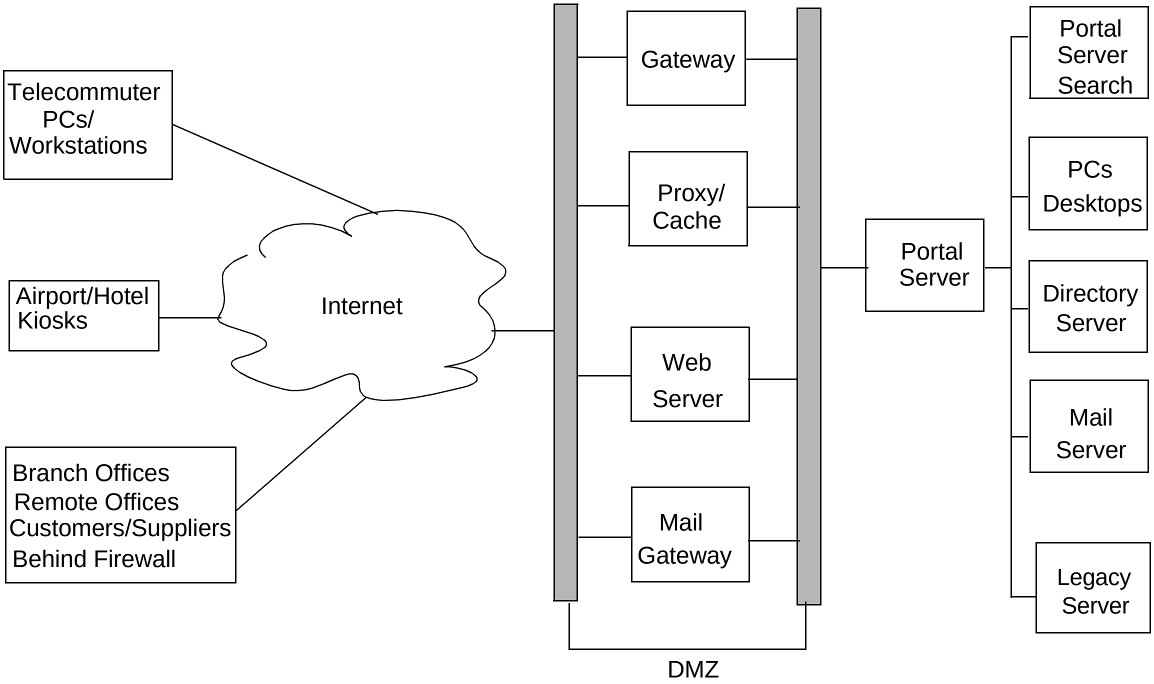
Figure 1-1 on page 41 shows the high-level architecture of a typical installation at a company site for a business-to-employee portal. In this figure, the gateway is hosted in the company's DMZ along with other systems accessible from the Internet, including web servers, proxy/cache servers, and mail 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 ONE representative. Portal Server requires customizations to provide ISP hosting functionality.

In Figure 1-1 on page 41, users on the Internet access the gateway from a web-enabled browser and connect to the gateway at 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 1-1 on page 41 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 7, "Creating Your Portal Design", for more detailed information on portal design.

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



Establishing Quality Goals

When deploying Portal Server, think about the quality goals you want to establish for your organization. Some of these goals might include:

- Upgrading all existing Portal Server servers and users to Portal Server 6.2 within a certain timeframe, say 12 months.
- If you are upgrading from previous versions of Portal Server, your organization’s IT group must maintain existing portal services during the migration.
- Setting performance and reliability expectations. You will need to establish baseline measurements and then continue tracking as you move to a production environment.
- Maintaining a completely functioning network infrastructure throughout the transition period from trial to production.
- Eliminating single points of failure for the portal system by developing an architecture that includes redundant portal servers, gateways, and directory replicas and masters at various service layers.

Chapter 2

Sun ONE Portal Server Architecture

This chapter describes the architecture, protocols, interfaces, directory structure, deployment, and customization of the Sun™ ONE Portal Server 6.2 product.

This chapter contains the following sections:

- Portal Server Components
- Java Enterprise System Software Interfaces
- Portal Server Configuration Files and Directory Structure
- Portal Server Software Deployment
- The Portal Desktop is the presentation of the portal. It is the logical component consisting of the Desktop servlet, provider APIs, channels, and various other support APIs and utilities. The Desktop is constructed of a set of channels that can be easily replaced. The Desktop also uses a proprietary templating mechanism used by many Desktop providers to separate static content from compiled Java code.
- Portal Server Customization
- Portal Server Availability and Fault Tolerance
- Portal Server Security, Encryption, and Authentication

Portal Server Components

This section describes the Portal Server components, first in terms of the platform itself and individual components, then in terms of the portal services. See Chapter 3, “Sun ONE Portal Server, Secure Remote Access Architecture” for details on the Sun™ ONE Portal Server, Secure Remote Access (SRA) product.

Deployment Platform

Portal Server is part of the Sun ONE architecture. Within the Sun ONE architecture, Portal Server provides technologies that locate, connect, aggregate, present, communicate, personalize, notify, and deliver content.

Java Enterprise System ships with Sun™ ONE Web Server and Sun™ ONE Application Server web application containers.

In addition, the following application servers can be used as its web application container, in place of the Sun™ ONE Web Server and Sun™ ONE Application Server software.

- BEA WebLogic Server™
- IBM WebSphere® Application Server

See the *Sun ONE Portal Server 6.2 Installation Guide* for information on deploying Portal Server in various web containers.

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. Portal Server add-on products include the additional software that is needed for that product. You must install Portal Server before installing an add-on product.

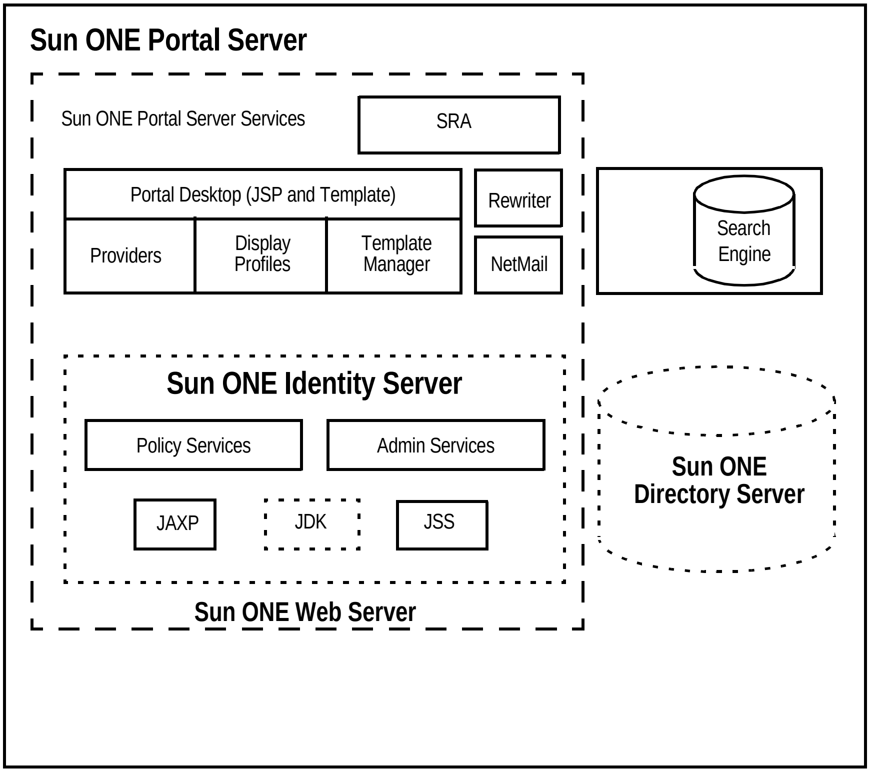
Software Components

Figure 2-1 on page 45 shows the software components that comprise Portal Server. (This figure shows Sun™ ONE Web Server software as the web container. It could just as well use one of the application servers previously mentioned.) The software components are arranged in a hierarchy.

The bottom layer is Sun™ ONE Identity Server software. Within it are the following core components: the Java™ API for XML Processing (JAXP), Java™ Development Kit (JDK™) Network Security Services for Java™ (JSS), Sun ONE Web Server, and Sun™ ONE Directory Server software.

The next layer is the Sun ONE Portal Server. Within it are the following internal components (services): Portal Desktop, NetMail, Rewriter, and Search Engine.

Figure 2-1 Portal Server Software Components



Throughout the figure, the line type in which a component is drawn indicates the following:

- Dotted lines indicate components that can use their own copies of a contained component or share copies with other components. Other components can directly use the interfaces from contained components. In addition, contained components can be updated independently from a component. For Portal Server, these components include Identity Server, Java Development Kit, and Directory Server.

- Dashed lines indicate components that have one or more characteristics from each of the other two categories. For Portal Server, this component is Sun ONE Web Server.

- Solid lines indicate components that use their own copies of the contained component. Other components are not allowed to share the contained component or directly use the interfaces from the contained component. In Portal Server, these components are the add-on components, Portal Server itself, the Search Engine, Portal Desktop, NetMail, and Rewriter , and JAXP and JSS components.

The following sections describe the software components identified in Figure 2-1.

Sun ONE Web Server, Sun ONE Application Server, BEA , and IBM Application Servers

Sun ONE Application Server is included with the Java System Enterprise software.

Sun ONE Portal Server uses Sun ONE Web Server, or one of the supported application servers, as the web application container for Sun ONE Portal Server and Sun ONE Portal Server add-on applications. Components within an instance communicate through the JVM™ using Java APIs.

See *Sun ONE Portal Server 6.2 Installation Guide* for information on deploying Portal Server in various web containers.

Sun ONE Directory Server

Sun ONE 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.

Sun ONE Identity Server

Sun ONE 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.

Java Development Kit

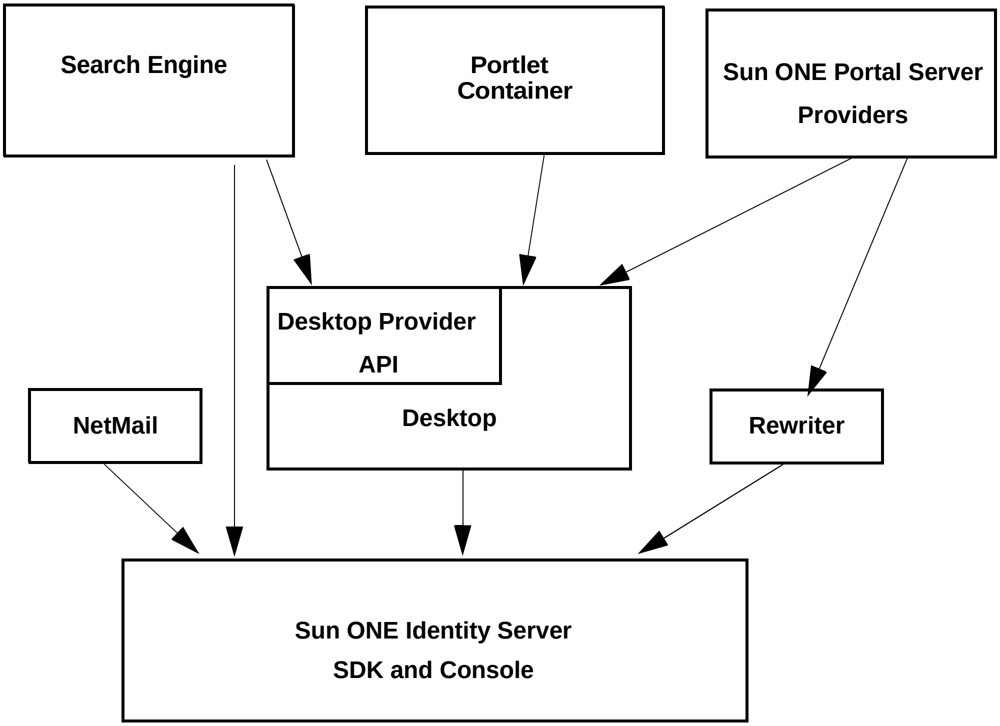
Java Development Kit provides the Java run-time environment for all Java software in Portal Server and its underlying components. Portal Server depends on the JDK of the web container.

NOTE See the *Sun ONE Portal Server 6.2 Release Notes* for specific versions of products supported by Sun ONE Portal Server 6.2.

Services Used by Portal Server

This section provides general information about Portal Server components that integrate external components into a system that is easier to install and use, provide additional functionality to external components, and provide backward compatibility for old interfaces. The relationships and interfaces associated with these components are shown in Figure 2-2.

Figure 2-2 Services Used by Portal Server



Portal Desktop

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. You can edit the display profile and other Portal Desktop service data with the Identity Server administration console.

Portlet Container

The Portal Desktop displays portlets which are pluggable web components that generate content within the context of a portal. Sun ONE portlets are Java Specification Request (JSR) 168 compliant.

The Portlet Container manages and dispatches requests to portlets. The Portlet Container collects and sends back the content through a provider.

To create a portlet, you can can use the sample portlet shipped with Portal Server as an example. For information on developing and deploying a portlet, see the *Sun ONE Portal Server 6.2 Developer's Guide*.

Sun ONE Portal Server Providers

A provider is a Java™ class responsible for converting the content in a file, or the output of an application or service into the proper format for a channel.

Portal Server implements several content providers as part of the Portal Server product rather than in the Portal Desktop component because of dependencies on other system components. For a list of providers and detailed information, see the *Sun ONE Portal Server 6.2 Desktop Customization Guide*.

Examples of providers that are part of the Portal Server product include:

- `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.

Portal Server Channels

To the end user, a *channel* is a distinct unit of content in the Portal Desktop, usually (but not always) set off with a border and header row of icons that enables users to configure the channel to their preference.

Once a portlet is deployed, Portal Server is aware of a portlet defined in an application. You can create channels based on a portlet. For information on creating channels, see *Sun ONE Portal Server 6.2 Administrator's Guide*.

Some of the channels provided by Portal Server 6.2 are:

- Login
- UserInfo
- Bookmark
- App
- SampleJSP
- Search
- Bookmark2
- SampleRSS SampleXML
- SampleURLScraper
- Notes
- PersonalNotes
- MailCheck
- SampleSimpleWebService
- SampleSimpleWebServiceConfigurable
- Subscriptions
- Discussions
- DiscussionLite
- Calendar
- AddressBook
- Mail
- IMChannel

NetMail

The NetMail component implements the NetMail (based on Java technology) and NetMail Lite email clients. These clients work with standard IMAP and SMTP servers. You can edit NetMail service data with the Identity Server administration console.

Rewriter

The Rewriter provides a Java class library for rewriting URL references in various web languages such as HTML, JavaScript™, and WML, and in HTTP Location headers (redirections). The Rewriter defines an Identity Server service for storing rules that define how rewriting is to be done and the data to be rewritten. You can edit Rewriter rules with the Identity Server administration console.

Search Engine

The Search Engine service provides basic and advanced search and browse channels for the Portal Desktop. It uses a robot to create resource descriptions for documents that are available in the intranet, and stores these resource descriptions in an indexed database. Resource Descriptions (RDs) can also be imported from another server or from a backup Summary Object Interchange Format (SOIF) file. The Search Engine includes Java and C APIs for submitting resource descriptions and for searching the database. The Search Engine database can also be used for storing other arbitrary content, for example, a shared content cache for other content providers. You can edit Search Engine service data with the Identity Server administration console.

The Search Engine service is used in the Subscription channel to summarize the number of hits (relevant information) that match each profile entry defined by the user for categorized documents and discussions.

Additionally, the Search Engine service is used in the Discussion channel to individually search contents and rate the importance for comments.

Portal Server, Secure Remote Access

SRA enables remote users to securely access their organization’s network and its services over the Internet. Additionally, it gives your organization a secure Internet portal, providing access to content, applications, and data to any targeted audience—employees, business partners, or the general public.

See Chapter 3, “Sun ONE Portal Server, Secure Remote Access Architecture” for more information.

Service Configuration

As a Sun ONE Identity Server application, Sun ONE Portal Server defines services that are managed using the Identity Server service management system. Generally, any service-related data that is not server-specific is stored in the LDAP directory. Server-specific data can be stored in properties files that are local to the specific server. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for information on these files.

Portal Server registers its services into the Identity Server Services Management Services framework. This occurs during the pre-installation of Portal Server and post-installation for Identity Server.

Services Management System provides a mechanism for services to define and manage their configuration data by using an XML file that adheres to the Services Management System Document Type Definition (DTD). The definition of the configuration parameters through the XML file is called the *schema* for the service.

The service configuration schema and the service configuration data are stored in the directory server using the LDAP Directory Information Tree (DIT) and schema defined by the product. Each Portal Server service (listed below) has its own XML and properties files for presenting and modifying service specific data.

Configuration data for a service can be classified as global, dynamic, organization, user, and policy. In general, configuration data that is global and not instance-specific is stored under the root node as `ou=service`. Configuration information that is specific to an organization is stored under the organization's node as `ou=services`. Each organization has its own configuration for Portal Desktop services.

Portal Server defines services within the Identity Server framework:

- **Portal Desktop**—includes data associated with the Desktop component, including the display profile and other configuration parameters associated with the Desktop. The Identity Server service name is `SunPortalDesktopService` .
- **Search Engine**—the search engine for each Portal Server instance. Defines at least one service, but can use multiple backend databases and search engine instances. The Identity Server service name is `SunPortalSearchService`.
- **NetMail**—includes data associated with the NetMail application primarily consisting of the user's preferences. The Identity Server service name is `SunPortalNetMailService` .

- **Rewriter**—includes data associated with the Rewriter component, including the named rule sets that control the rewriting operation. The Rewriter API makes reference to the named rulesets that are stored in the directory. The Identity Server service name is `SunPortalRewriterService`.
- **SRA—**includes the following services: Access List, Gateway, NetFile, and Netlet. The Identity Server service name is `SunPortalSRAPService.`
- **Subscriptions**—includes dynamic and user attributes. Values applied to the dynamic attributes are applied across the Sun ONE Identity Server configuration.The Identity Server service name is `SunPortalSubscriptionService`.

You administer Portal Server services (as well as the Identity Server services) through the Identity Server administration console. For more information, see the *Sun ONE Portal Server 6.2 Administrator's Guide*.

Java Enterprise System Software Interfaces

The Java Enterprise System software has the following interfaces:

- Front-end Interface—enables users to access enterprise resources from the Internet.
- Back-end Interfaces—used by Portal Server to access those resources and to provide the administrative interface.
- Customer and Third-Party Software Interface—used by developers to add functionality to the Portal Server system.

Front-end Interface

The front-end interface uses the HTTP or HTTPS protocol with markup languages (such as HTML), JavaScript functions, and Java applets, depending on the application. All of these are standard protocols supported by the most commonly used browser software. When Java applets, which are bundled with Portal Server, are downloaded into the browser, the applets use proprietary protocols layered on top of the protocols listed above to communicate with other components within Portal Server. However, since the applet is considered part of the Portal Server system, that communication happens within the Portal Server system rather than external to it.

Back-end Interfaces

The back-end interfaces provided by Identity Server include:

- **Enterprise resource access protocols**—Mail (for example, SMTP, IMAP4, and LDAP); file access (for example, FTP, NFS, and SMB); web browsing and information services (HTTP and HTTPS); and calendar (`rpc.cmsd` and IETF calendar protocol). Additional protocols might be used if you add applications to the system.
- **Administration console protocols**—HTTP with HTML and other web languages as described in the front-end interface.

Customer and Third-Party Software Interface

The customer and third-party software interface consists of extension APIs and protocols that are used to extend the Portal Server system. For more information, see the *Sun ONE Portal Server 6.2 Developer's Guide*.

Users of the Interfaces

The three classes of human interfaces to the Portal Server system correspond to the three types of people who use it:

- **End-users**—End users interact with the end-user interface, which consists of several web applications that are accessed by a web browser. The Portal Desktop application is the primary portal interface, providing web pages that consist of a collection of channels. Each channel provides an access point into some function or information. Users can configure the set of channels that is displayed and specific characteristics of each channel. Other web applications in the end-user interface provide access to specific resources, such as mail, files, and calendar.
- **Administrators**—Administrators use the Identity Server administration console, and Identity Server and Portal Server command-line utilities, to configure, administer, and maintain the system. A Portal Server system can have many administrators, each delegated with a specific responsibility. Many administrative tasks can be accomplished by using the Identity Server administration console, which is a web application accessed using a web browser. Command-line tools for administration are also available to facilitate scripting and batch execution.

- **Developers**—Developers use the programming APIs to extend the Portal Server system. These APIs provide for developing enterprise resource applications, authentication modules, and Portal Desktop channel providers.

Public Interfaces in Sun ONE Portal Server

Sun™ ONE Portal Server provides public interfaces that developers can use for to extend Portal Server software. See the *Sun ONE Portal Server 6.2 Developer's Guide* for information on various APIs.

This section lists exported interfaces and the components they apply to. .

Table 2-1 Portal Server Interfaces - Portal Desktop

Exported Interface	Description
Portal Desktop Service Definition	Defines the Identity Server configuration attributes for the Portal Desktop service. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.
Portal Desktop Display Profile XML DTD	Defines the display configuration for the Portal Desktop by defining provider and channel objects, and their properties. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.
Portal Desktop SDK (PAPI)	Supplies provider interfaces, base classes, context, and exceptions. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information.
Leaf Building-Block Providers	Supplies the URLScraper, XML, and JSP providers. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information.
Container Building-Block Providers	Supplies the JSP, single, table, tab, and tab container providers, and exceptions. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information.
Portal Desktop Command-Line Interface	Supplies the `dpadmin` and `par` command utilities for product administration. See the *Sun ONE Portal Server 6.2 Administrator's Guidee* for more information.
Portal Desktop Graphical User Interface	Provides the primary end-user interface and a mechanism for extensible content aggregation through the Provider Application Programming Interface (PAPI).
Portal Desktop Servlet	Routes client requests for content and processing and passes them on to the specific provider object. See the Portal Server Javadoc™ for more information.
Portal Desktop Template File Format	The Portal Desktop HTML templates were used in Sun ONE Portal Server 3.0 and are included for backward compatibility only. See the *Sun ONE Portal Server 6.2 Desktop Customization Guide* for more information.
Portal Desktop JSP Tag Libraries	Supplies the tag library descriptor (TLD) files that can be used on any provider or container that implement the PAPI interface, that operate on a provider or container that support the `ProviderContext` and `ContainerProviderContext` interfaces, and that operate on specific container providers (`SingleContainer`, `TableContainer`, `TabContainer`). See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information.

Table 2-1 Portal Server Interfaces - Portal Desktop *(Continued)*

Exported Interface	Description
Portal Desktop Admin Console Module	Supplies the means by which you manage Portal Server services in the Identity Server framework. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.

Table 2-2 Portal Server Interfaces - Search

Exported Interface	Description
Search Service Definition	Defines the Identity Server configuration attributes for the Search service. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.
Search SDK	Supplies the C API for customizing the way the robot crawls URLs and generates resource descriptions; the Java APIs for searching the database, for submitting data, and for manipulating SOIF objects, such as RDs (RDM and SOIF APIs); and the Search provider tag library and helper beans that enable you to write customized search JSPs. See the *Sun ONE Portal Server 6.2 Developer's Guide* for more information.
Search Provider	Supplies the search function using the Portal Server Search Engine.
Search CLI	Supplies the `rdmgr`, `sendrdm`, and `StartRobot` command-line utilities for product administration. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.

Table 2-3 Portal Server Interfaces - Rewriter

Exported Interface	Description
Rewriter Service Definition	Defines the Identity Server configuration attributes for the Rewriter service. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.
Rewriter Rules XML DTD	See the *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide* for more information.
Rewriter CLI	Supplies the `rwadmin` command-line utility for product administration. See the *Sun ONE Portal Server 6.2 Administrator's Guide* for more information.

Portal Server Configuration Files and Directory Structure

This section describes the Sun ONE Portal Server directory structure and properties files used to store configuration and operational data.

Directories Installed for Portal Server

Table 2-4 shows the platform-specific directory structures that are installed for Portal Server.

Table 2-4 Portal Server Directories

Description	Location
Default installation directory	*portal-server-install-root*/`SUNWps`
Default installation directory for configuration information	`/etc/`*portal-server-install-root*/`SUNWps`
Default installation directory for SDK	*portal-server-install-root*/`SUNWps/sdk`
Temporary files	`/usr/tmp`
Debug files	`/var/`*portal-server-install-root*/`SUNWam/debug`
Log files	`/var/`*portal-server-install-root*/`SUNWam/log` `/var/`*portal-server-install-root/*`SUNWps`*/instance-directory*
Search Engine logging, configuration, and data directories	`/var/`*portal-server-install-root*/`SUNWps/`*instance-directory*/*log-directory*
Container and channel display profile	*portal-server-install-root*/`SUNWps/samples/desktop/dp-org.xml`
Provider display profile	*portal-server-install-root*/`SUNWps/samples/desktop/dp-providers.xml`
HTML template files	`/etc/`*portal-server-install-root*/`SUNWps/desktop/default/`*channelname*`.template`
JSP template files	`/etc/`*portal-server-install-root*/`SUNWps/desktop/default/`*JSPchannelname*
Command-line utilities	*portal-server-install-root*/`SUNWps/bin/`
Tag library definitions	`/etc/`*portal-server-install-root*/`SUNWps/desktop/default/tld/*.tld`
Display profile DTD	*portal-server-install-root*/`SUNWps/dtd/psdp.dtd`
Java properties files	*portal-server-install-root*/`SUNWam/locale`

Configuration Files

All Portal Server configuration data is stored using the Identity Server services management function. Identity Server provides the bootstrap configuration file that is needed to find the directory server.

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 ONE 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 Java™ software falls into three categories:

- Applets
- Web applications
- Stand-alone Java processes

Applets used in Portal Server are compatible with Java 1.1, which is supported by most browsers.

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 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.

Portal Server Desktop

The Portal Desktop is the presentation of the portal. It is the logical component consisting of the Desktop servlet, provider APIs, channels, and various other support APIs and utilities. The Desktop is constructed of a set of channels that can be easily replaced. The Desktop also uses a proprietary templating mechanism used by many Desktop providers to separate static content from compiled Java code.

The Portal Desktop is composed of the following entities:

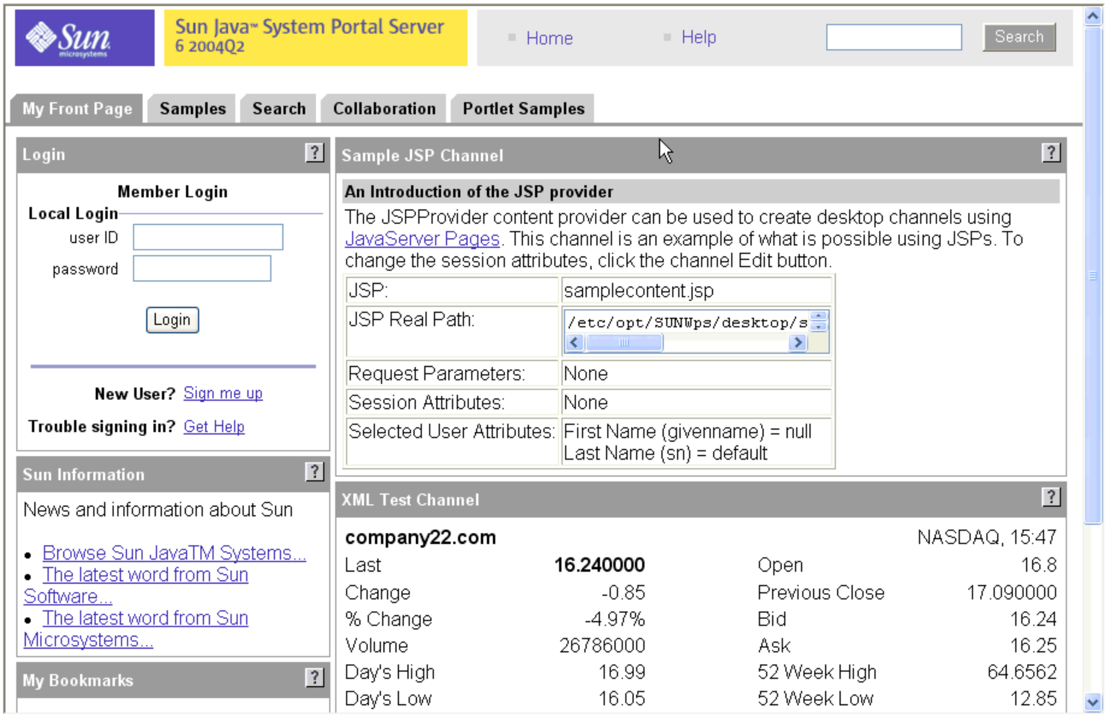
- **Content provider**—The programmatic entity responsible for the generation of content. Generated content can consist of entire pages, frames, or channels; any markup.
- **Channel**—A unit of content, usually (but not necessarily) arranged in rows and columns. A provider generates a channel.
- **Display profile**—An XML document describing container management and properties for channels.
- **Portlets**—Pluggable web components that process requests and generate content within the context of a portal. In Sun ONE Portal Server software, portlets are managed by the Portlet Container. Conceptually, portlets are equivalent to the Providers.

See the *Sun ONE Portal Server 6.2 Administrator's Guide* for Portal Desktop administration tasks. See the *Sun ONE Portal Server 6.2 Desktop Customization Guide* for tasks on how to customize the Desktop's look and feel.

User Experience with the Portal Desktop

Figure 2-3 on page 60 shows a sample of the out-of-the-box Portal Desktop front page from Sun ONE Portal Server 6.2.

Figure 2-3 Portal Server Sample Portal Desktop



New User? Sign me up

Trouble signing in? Get Help

An Introduction of the JSP provider

The JSPProvider content provider can be used to create desktop channels using JavaServer Pages. This channel is an example of what is possible using JSPs. To change the session attributes, click the channel Edit button.

JSP:	samplecontent.jsp
JSP Real Path:	/etc/opt/SUNWps/desktop/s
Request Parameters:	None
Session Attributes:	None
Selected User Attributes:	First Name (givenname) = null Last Name (sn) = default

News and information about Sun

- Browse Sun JavaTM Systems...
- The latest word from Sun Software...
- The latest word from Sun Microsystems...

company22.com			NASDAQ, 15:47
Last	**16.240000**	Open	16.8
Change	-0.85	Previous Close	17.090000
% Change	-4.97%	Bid	16.24
Volume	26786000	Ask	16.25
Day's High	16.99	52 Week High	64.6562
Day's Low	16.05	52 Week Low	12.85

After the user is authenticated through the Identity Server Authentication service, the user is directed to the Portal Server Desktop. From there, the user can access a variety of services and applications. These services and applications can be categorized as follows:

- **Portal Desktop channel applications**—Applications based entirely on one or more Portal Server Desktop channels. For example, Portal Server includes a bookmark channel that enables users to save bookmarks and use those bookmarks from any browser that has access to the portal.

- **Stand-alone web applications**—Applications for which the Portal Server Desktop provides a link to the web application. This link helps the user start the application, but there is no application-specific functionality provided on the Desktop itself. An example of this type of application is NetFile in SRA, which provides access to files in intranet file systems.
- **Web applications with front-end channel**—Applications in which Portal Server provides one or more channels as an entry point into the web application. In this context, a web application is any application whose interface is delivered through a web browser, whether it uses HTML, JavaScript functions, Java applets, plugins, or some other markup language.

User Session

Figure 2-4 on page 62 represents a typical Portal Server user session. Session exit is either by an explicit Portal Desktop log out or by an implicit session time out event. The horizontal line is a Portal Server activity time line. The activities of a single user's session is represented. Session activities proceed from left to right and are labeled from A to I as follows:

A: User submits request to home page.

B: Portal Server returns the authentication menu.

C: User submits request to authentication module.

D: Portal Server returns authentication form.

E: User submits request login credentials.

F: Portal Server returns initial Desktop display.

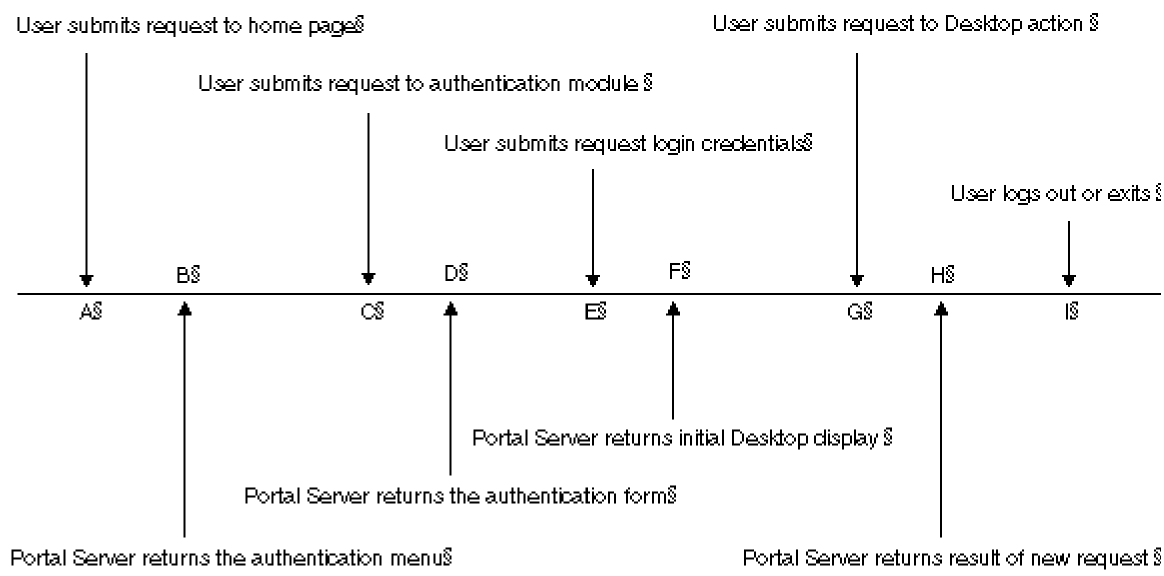
G: User submits request to Desktop action.

H: Portal Server returns result of new request.

I: User logs out or exits.

NOTE Items B and C are valid only if more than one authentication mechanism is enabled. Most organizations use a single authentication mechanism, and hence will not see the authentication menu.

Figure 2-4 Portal Server Users Session


User submits request to home page§
User submits request to Desktop action §
User submits request to authentication module §
User submits request login credentials§
User logs out or exits §
B§
D§
F§
H§
A§
C§
E§
G§
I§
Portal Server returns initial Desktop display §
Portal Server returns the authentication form§
Portal Server returns the authentication menu§
Portal Server returns result of new request §

During this session:

- From point A to B, Portal Server processes the user's request to download Portal Server's home page.
- From point B to C, the user views the result of the request and decides which authentication method to use.
- From point C to point D, the server computes and returns the authentication page for the method that the user selected.
- From point D to point E, the user, in think mode, enters authentication credentials.
- From point E to point F, Portal Server computes and returns the initial Portal Desktop display.
- From point F to point G, the user browses sites referenced by the Portal Desktop. To Portal Server, this is equivalent to think time.
- From point G to point H, Portal Server executes a new user request.

Portal Server Customization

The Sun™ ONE Portal Server user interface is fully customizable and extensible by the customer or third-parties. This section describes the various customizations you can perform on Portal Server.

The methods for customizing Portal Server include:

- Modifying the look and feel of the user interface by using JSP and template files
- Defining additional content channels using built-in content providers
- Writing custom content providers to be used in defining new channels
- Writing custom authentication modules
- Writing custom service administration modules

Customization is provided through templates (JSP or other template languages) that can be edited to modify branding or other look-and-feel characteristics. Extension is possible through the creation of applications and services that use any of these user interface models.

In addition, you can customize the system by using the capabilities of the underlying components such as Identity Server and the web container. These types of customizations include:

- Defining new services, including new data for the directory
- Writing custom web applications (servlet, JSP, and EJB™ applications)

See the *Sun ONE Portal Server 6.2 Desktop Customization Guide* and the *Sun ONE Portal Server 6.2 Developer's Guide* for information on how to customize and develop applications for Portal Server. See the *Sun ONE Identity Server Programmer's Guide* for information on defining new services and writing custom web applications.

Portal Server Availability and Fault Tolerance

Portal Server achieves high availability and fault tolerance through software replication. You can configure Portal Server to run multiple instances of each web application, thereby providing a backup if one of the instances fails. In addition, Portal Server uses Identity Server services for session management and non-local data access. Therefore, the portal system inherits all the benefits and constraints of

Identity Server with respect to high availability and fault tolerance. The Identity Server 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. See the Identity Server documentation for more information.

Within the Portal Server web applications, state is not shared among instances. This means that a failure causes the application to be restarted. Usually, end users do not notice that this has happened because the state information that is associated with the Portal Server applications can be restored by reading the user's profile and using information in the request. (This refers to the case where HTTP session replication provided by the application sever is being used, so that re-authentication is not necessary.)

Replication eliminates single points of failure in the system. For Sun ONE Directory Server, this is provided by using a multiple master configuration. However, this solution does not completely address all fault tolerant aspects of the system. A data loss can still occur due to a crash during the process of data synchronization among masters. See the Directory Server documentation for more information.

See Chapter 7, "Creating Your Portal Design", for details on creating your portal design to include high availability.

The high availability features described above are transparent to the client of those services. Portal Server components address high availability natively to different extent. There is a different level of recovery for different components. For details, check the corresponding Portal Server deployment product documentation.

Portal Server 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. The first layer of 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.

The second layer of security is provided by SRA. This product provides a gateway that resides in the DMZ and provides a single secure access point to all intranet URLs and applications. It uses HTTPS by default for connecting the browser to the intranet. The gateway includes a reverse proxy that uses the Rewriter, which enables 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 depends on the authentication service provided by 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.

Chapter 3

Sun ONE Portal Server, Secure Remote Access Architecture

This chapter describes the Sun™ ONE Portal Server, Secure Remote Access (SRA) architecture. It describes the key components of SRA with respect to their role in providing secure remote access to corporate intranet resources from outside the intranet (for example, through the Internet).

This chapter contains the following sections:

- Overview of Sun ONE Portal Server, Secure Remote Access
- SRA Components
- SRA Authentication
- SRA Configuration Files and Directory Structure

Overview of Sun ONE Portal Server, Secure Remote Access

SRA offers browser-based secure remote access to portal content and services from any remote browser. 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 Sun™ ONE 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 architecture is well suited to these types of portals. The gateway, NetFile, and Netlet features of SRA enable users to securely access intranet resources through the Internet without exposing these resources to the Internet.

The gateway, residing in the Demilitarized Zone (DMZ), provides a single secure access point to all intranet URLs, file systems, and applications. (A DMZ is a small protected network between the public Internet and a private intranet, usually demarcated with firewalls on both ends.) All services such as Session, Authentication, and the Desktop reside behind the DMZ in the secured intranet. Communication from the client browser to the gateway is encrypted using HTTPS. Communication from the gateway to the server and intranet resources can be either HTTP or HTTPS.

SRA also provides Netlet technology, which enables a secure connection between a client running a Java enabled browser and a TCP/IP application running on a server in the corporate intranet. For remote file access, SRA provides NetFile technology. The Netlet and NetFile applets are downloaded to the client machine.

Relationship Between Portal Server and SRA

Portal Server can function in two modes—open and secure—as explained in the following sections.

Open Mode

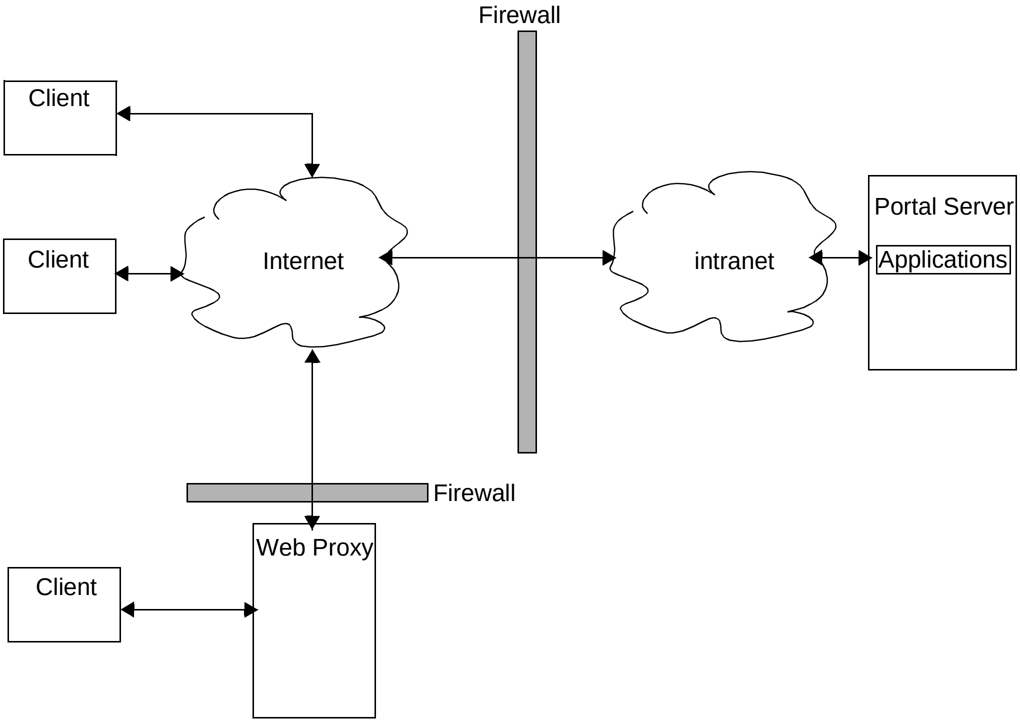
In open mode, you install Portal Server without SRA. The typical public portal, such as `my.yahoo`, runs without secure access using only the HTTP protocol. Although you can configure Portal Server to use the HTTPS protocol in open mode (either during or after installation), secure remote access is not 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 as compared to the secure mode.

Figure 3-1 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 Web proxy server that itself sits behind a firewall.

Figure 3-1 Portal Server in Open Mode



Secure Mode

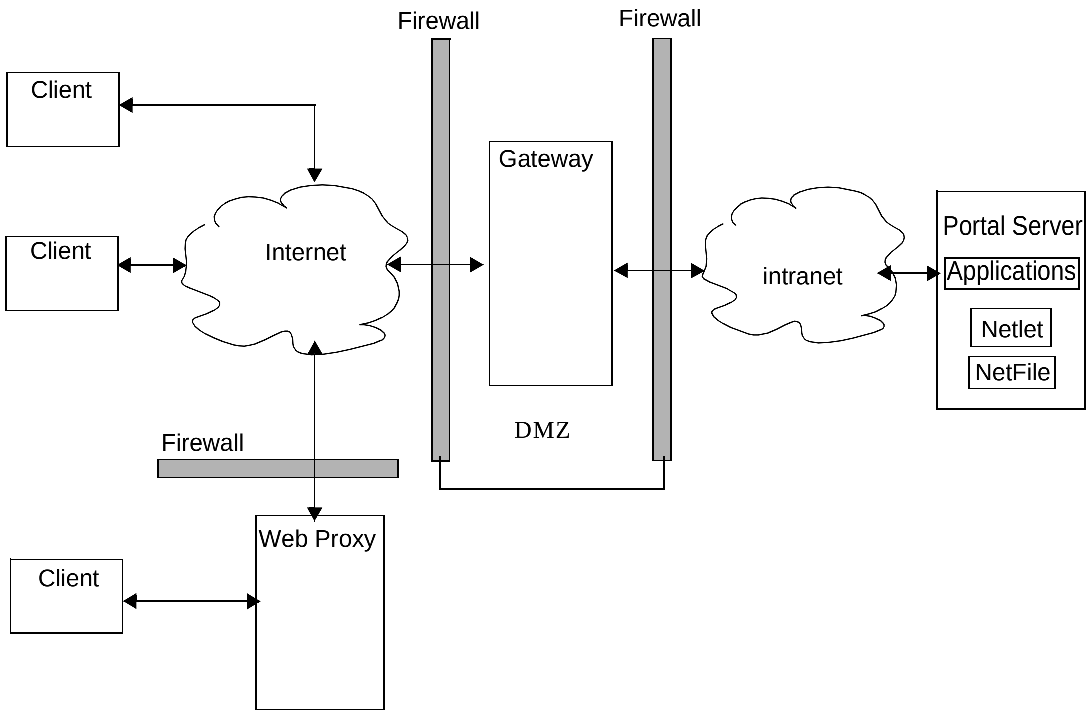
In secure mode, you install Portal Server with SRA. Secure mode provides users with secure remote access to required intranet file systems and applications.

The main advantage of SRA is that only the IP address of the gateway gets 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 result is that your organization publishes a single IP address and provides secure access to all your web servers and applications that you integrate with Netlet, Netfile, and Rewriter.

In this case, 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 ONE 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 3-2 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 3-2 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.

SRA Components

This section describes the SRA components.

To provide the services discussed in "Overview of Sun ONE Portal Server, Secure Remote Access" on page 67, SRA comprises the following components:

- SRA Gateway
- Netlet
- Netlet Proxy
- NetFile
- Rewriter
- Rewriter Proxy

You can install SRA components on the Sun ONE Portal Server node, or on any other non-portal node (referred to as a separate node). Table 3-1 on page 71 lists the various installable components and the nodes that they can be installed on.

Table 3-1 SRA Components and Nodes

Component	Node	Description
Gateway	Sun ONE™ Portal Server, separate Server	The gateway provides the interface and security barrier between remote user sessions originating from the Internet and your corporate intranet.
Secure Remote Access Support	Portal Server node	This component has three parts: • Gateway Support—Controls communication between the Portal Server and the various gateway instances. • NetFile file manager application—Enables remote access and operation of file systems and directories. NetFile comprises NetFile Java™ technology, a Java-based user interface. This is available for Java 1 and Java 2. • Netlet—Ensures communication between the Netlet applet on the client browser, the gateway, and the application servers.

Table 3-1 SRA Components and Nodes *(Continued)*

Component	Node	Description
Netlet Proxy	Portal Server node, separate node	Netlet Proxy is an optional component. You can choose not to install it, or install it later. Netlet Proxy extends the secure tunnel from the client, through the gateway to the Netlet Proxy that resides in the intranet. This restricts the number of open ports in a firewall between the DMZ and the intranet. Netlet Proxy cannot be installed on a gateway node.
Rewriter Proxy	Portal Server node, separate node	Install the Rewriter Proxy to redirect HTTP requests to the Rewriter Proxy instead of directly to the destination host. The Rewriter Proxy in turn sends the request to the destination host.

SRA uses Sun™ ONE Identity Server software to store all its configuration information, except for information about the machines (such as host name, IP address), on the host where gateway is installed.

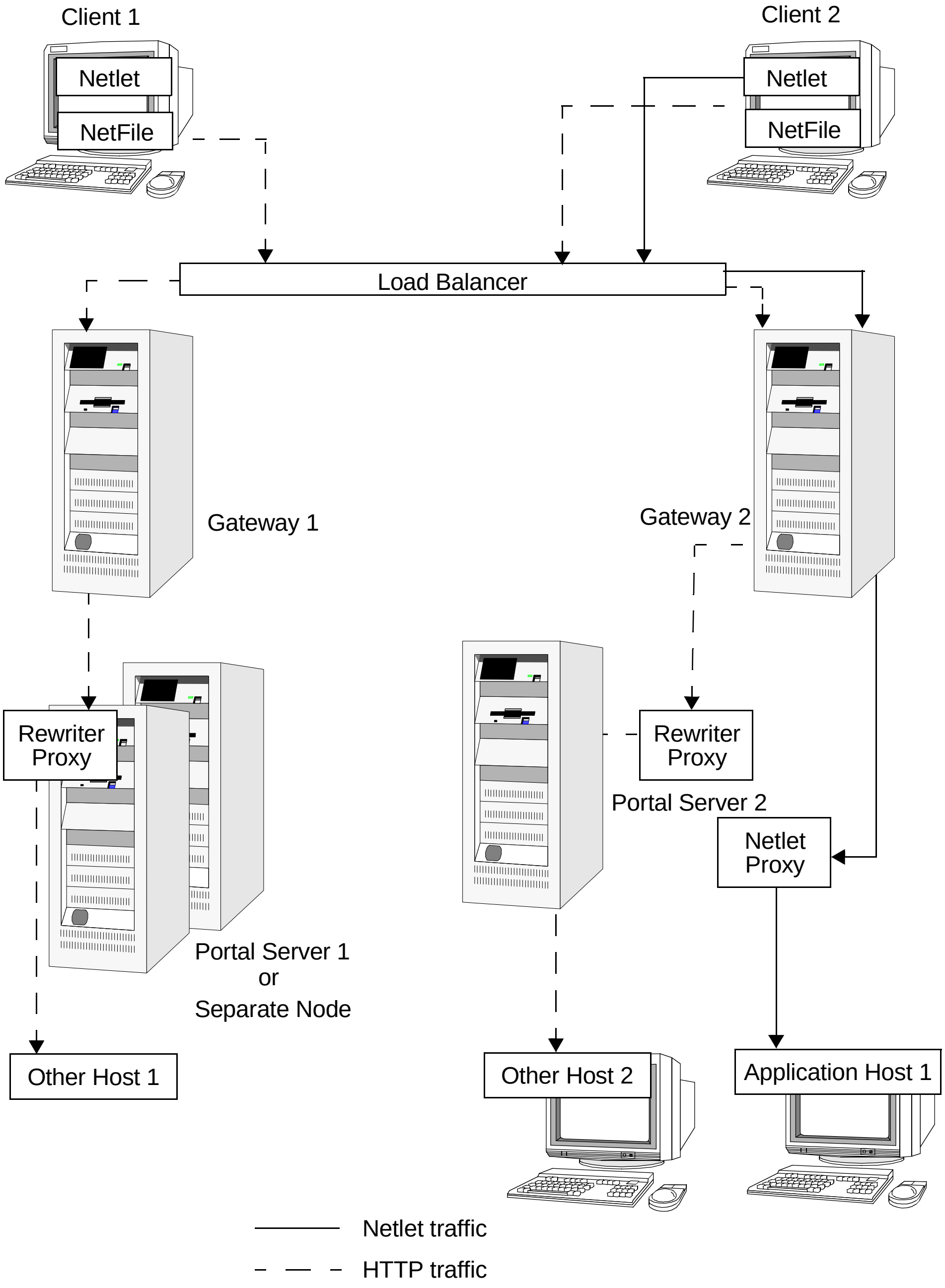
You administer the configuration information stored in Identity Server through the console modules of the respective components. These components are installed as part of the Identity Server administration console. The administration console provides a single point of administration for all SRA components.

Figure 3-3 on page 73 shows an installation consisting of all the SRA components.

In this figure, two client browsers are redirected by a load balancer, which sits in the DMZ, to one of two gateways, also located in the DMZ. Client 1 is performing a NetFile transaction. The NetFile traffic is routed by Gateway 1 to Portal Server 1, whose Rewriter Proxy directs the traffic to Other Host 1. Client 2 is performing both Netlet and NetFile transactions. Client 2’s Netlet and NetFile requests are handled by Gateway 2, which routes the traffic to Portal Server 2. The Rewriter Proxy on this host directs the NetFile traffic to Other Host 2. The Netlet Proxy on this host directs the Netlet request to Application Host 1.

NOTE This sample configuration illustrates the various components that you can use. There can be multiple installations and instances of the gateway and the Netlet Proxy.

Figure 3-3 SRA Installation with Multiple Gateways



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 the header information to determine the type of request. The gateway consists of Eproxy (Encrypted proxy) and Rproxy (a reverse proxy) subcomponents. Eproxy is responsible for handling Netlet components. Rproxy is responsible for all non-Netlet components. Depending on the type of request, the gateway performs the following:

- **Netlet request**—Eproxy checks session validity and routes the request (traffic) to the requested server. The destination of the Netlet traffic is determined by the Netlet rule that the user clicked in the Portal Desktop. See "Netlet" on page 81 for more information**.**
- **HTTP(S) traffic**—Eproxy forwards the request to the reverse proxy. When the reverse proxy receives the request, it checks the session validity and forwards the request to the server as specified by the HTTP header. Upon receiving a response from the server, the reverse proxy translates the response so that all intranet links within the response will work on the extranet.

If you do not require Netlet functionality, you can disable Eproxy. In Figure 3-4 on page 78, the requests are directly received and fetched by the reverse proxy.

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.

Multiple Gateway Instances

If desired, you can run multiple gateway instances on a single machine. 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.

When deploying the SRA gateway, you need to decide whether to have multiple instances on the same machine or on multiple machines.

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 should be stored as part of the gateway profile that the gateway uses automatically while connecting to the proxy. The Rewriter components also use the proxy information to ensure that any URL that contains the host, or DNS subdomain or domain, specified in the proxy information, is rewritten to ensure that they are routed through the gateway.

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 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 while running in HTTPS mode. If necessary, you can enable or disable SSL v2 support. You use the gateway service in the Identity Server administration console, to enable or disable specific ciphers. The gateway also supports Transport Layer Security (TLS).

NOTE You can configure the gateway to use or not use SSL. Likewise, you can also configure the web server to use or not use SSL. See the *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide* for information on configuring the gateway to use SSL.

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.

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. In case Netlet Proxy is enabled, the gateway checks for session validity and forwards it to the 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 URL Allow and URL Deny lists. Even when URL access is allowed, the gateway checks the validly 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 Allow and Deny lists. Entries in the URL Deny list take precedence over entries in the URL Allow 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

Because the entire traffic comes through the gateway for all the services provided by Portal Desktop, Netlet, and NetFile, you can monitor the complete user behavior by enabling logging on the gateway. The gateway uses the Identity Server logging API for creating logs.

Reverse Proxy (Rproxy)

The SRA Rewriter can also be used as a *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. The Rewriter component of Portal Server achieves the functionality of rewriting the web content by reverse proxy.

Figure 3-4 Reverse Proxies

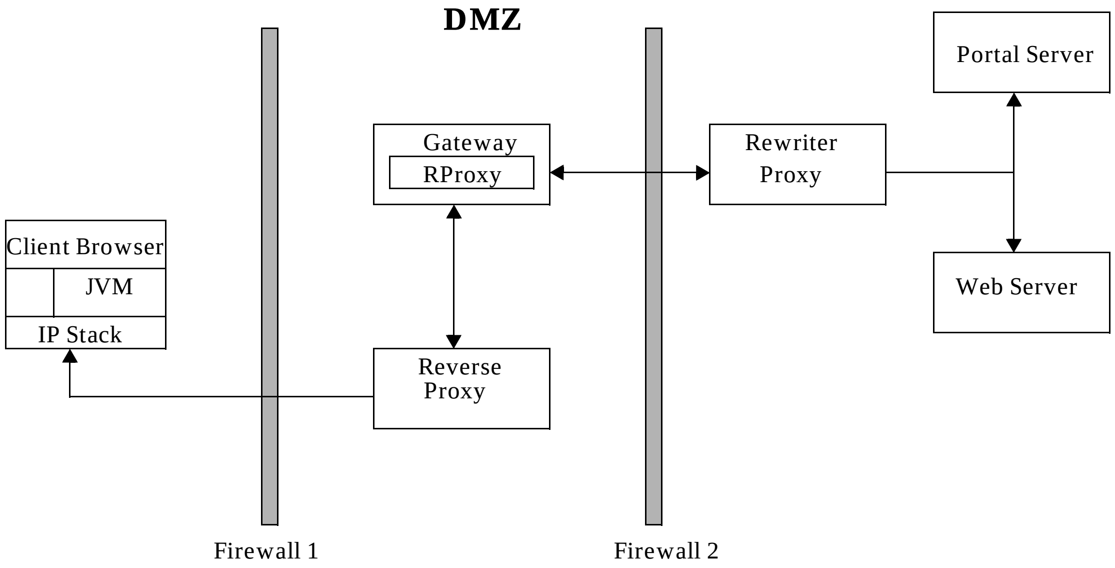


Figure 3-4 illustrates how you can configure the SRA gateway as a reverse proxy or you can put a reverse proxy in front of the gateway to serve both Internet and intranet content to authorized users. Whenever SRA serves web content, it needs to ensure that all subsequent browser requests based on this content are routed through SRA. This is achieved by identifying all URLs in this content and rewriting as appropriate.

The Rewriter component of SRA performs this rewriting. 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 in the administration console 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. All the identified URIs are sent through the URI Translator, a subcomponent of the Rewriter.
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.

The URI Translator handles the following types of URIs:

- Relative URI, for example, `<a href='/abc.html/'>`
- Absolute URI, for example. `<a href='http://sesta.com/abc.html'/>`
- JavaScript™ function call that returns a URI on evaluation, for example, `window.open(jsVarHome+ 'index.html');`

The following example shows how a simple URI is rewritten. Here the identified URI is a fully qualified URI (absolute URI):

- Identified URI: `http://intranet.sesta.com/abc.html`
- SRA URI: `https://sra.company22.com`
- Rewritten URI: `https://sra.company22.com/http://intranet.sesta.com/abc.html`

Rewriter Rulesets

A ruleset is an XML file that contains rules for all types of web content, such as HTML rules to identify URIs in HTML content, JavaScript rules to identify URIs in JavaScript, and XML rules to identify URIs in XML content.

SRA defines four types of HTML rules (Attribute, JSToken, Applet, and Form), two types of JavaScript rules (Variable and Function), and two types of XML rules (Attribute and TagText), for a total of eight types of rules to identify URIs in any web content.

NOTE The number of rules has been reduced considerably in Sun ONE™ Portal Server 6.2 from iPlanet™ Portal Server 3.0. You can also use the regular expression (*) in all rules, as opposed to limited support for certain rules in previous releases.

You can assign different rulesets for pages belonging to different domains. This helps to better manage rulesets as opposed to putting all rules for all domain pages in a single ruleset. Additionally, this domain-based assignment of rules increases portal performance, as specific rulesets are smaller in size.

You manage rulesets either through the Identity Server administration console or by using the `rwadmin` command-line interface.

For more information on managing rulesets, see the *Sun ONE Portal Server 6.2 Administrator's Guide* and the *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide*.

SRA provides three rulesets:

- `default_ruleset`: Used by the URLScraper provider to convert all URIs to absolute. SRA does not use this ruleset.
- `default_gateway_ruleset`: Contains rules required to handle the Portal Desktop and gateway modules in the Identity Server administration console.
- `generic_ruleset`: This is the most generalized ruleset and can handle web pages of simple and medium complexity. Complexity here is defined in terms the amount of JavaScript code present in a web page.

NOTE Start by adding rules to the generic ruleset. Even if an application has a complex use of JavaScript code, the generic ruleset is sufficient to handle all the rewriting in most cases.

Rewriter and MIME Types

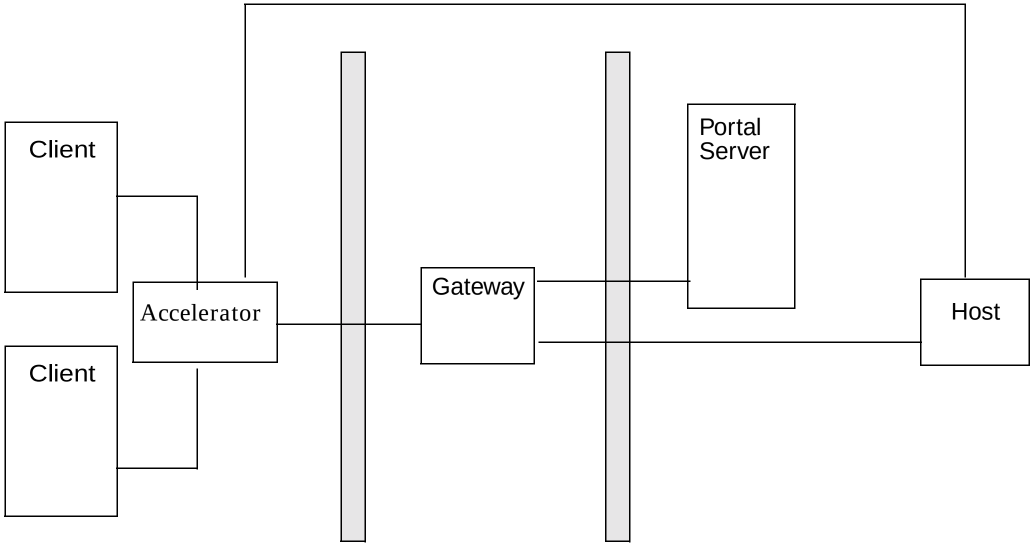
The Rewriter is rule-based and also decides on which content to rewrite based on the MIME type for the content. Out of the box, the Rewriter is configured to rewrite `text/html`, `text/htm`, `application/javascript`, and `text/xml`. You can modify this to handle different MIME types.

Using Accelerators with the Gateway

You can configure Crypto 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.

You can also configure an external SSL device to run in front of Secure Remote Access in open mode. It provides the SSL link between the client and Secure Remote Access. For information on accelerators, see the *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide*.

Figure 3-5 SRA Gateway with External Accelerator



Netlet

Netlet enables a secure connection between an arbitrary client on a system that is running a Java-enabled browser, and a network resource behind a corporate firewall. The client can be behind a remote firewall and SSL proxy, or directly connected to the Internet. Netlet can provide secure access to fixed port applications and some dynamic port applications that are available on the intranet from outside the intranet. All the secure connections made from outside the intranet to the intranet applications through the Netlet are controlled by Netlet rules.

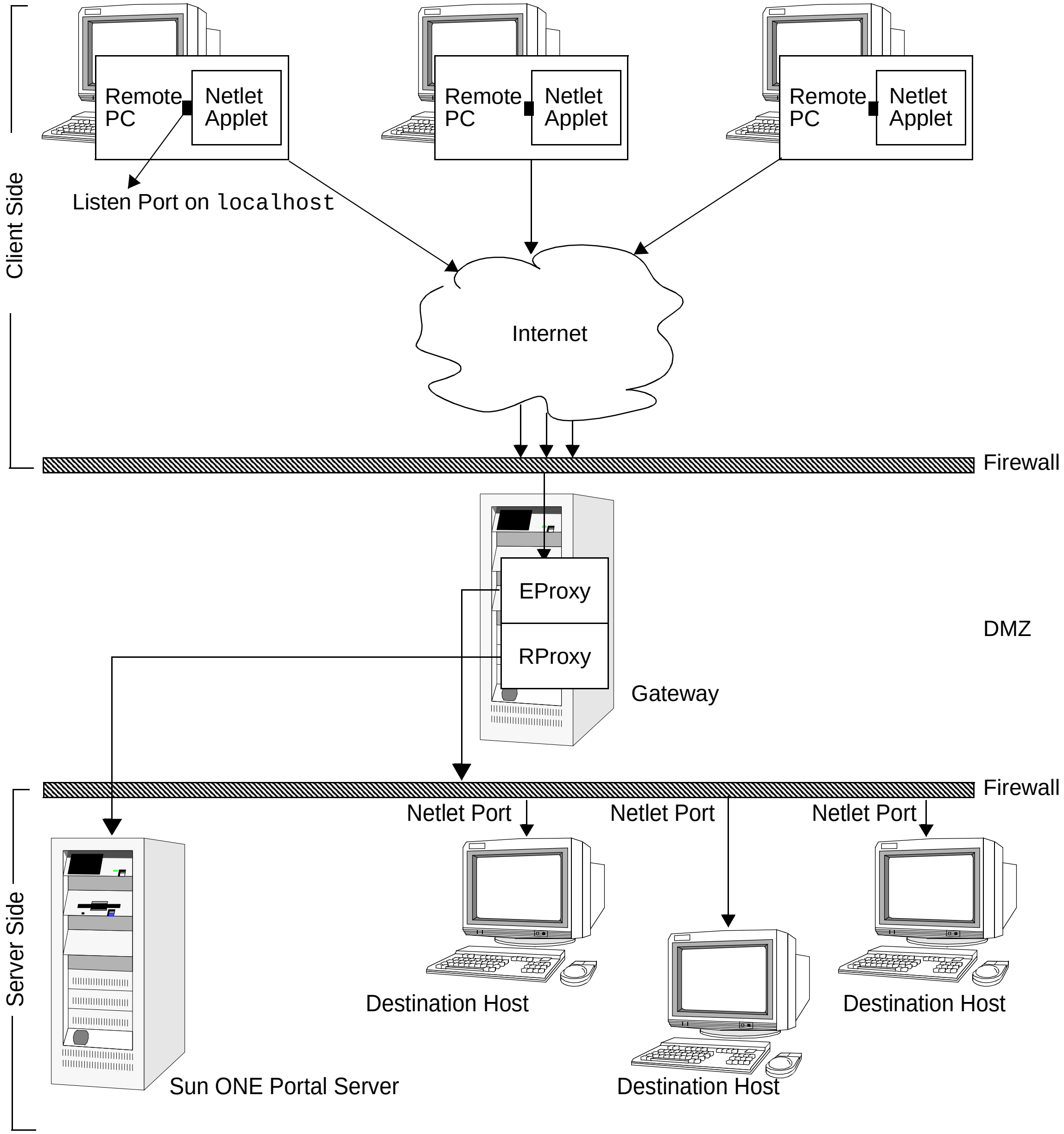
Netlet is an applet that runs on the browser. Netlet listens to and accepts connections on preconfigured ports, and routes both incoming and outgoing traffic between the client and the destination server. Both incoming and outgoing traffic is encrypted using an encryption algorithm selected by the user, 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.

To provide the required service, SRA includes the following components:

- Netlet applet
- Eproxy (Encryption proxy)
- Netlet Proxy

Figure 3-6 on page 83 shows how Netlet fits into the SRA platform. In this figure, each Netlet connection that is made to the destination host is described by a rule. The rule also describes the encryption algorithm to be used for that particular connection. The Listen Port on the `localhost` is the client machine port on which the Netlet applet listens. The Netlet applet sets up an encrypted TCP/IP tunnel between the remote client machine and intranet applications on the remote hosts. The applet encrypts the packets and sends them to the gateway, and decrypts the response packets from the gateway and sends them to the local application. All client requests are routed through the EProxy. EProxy handles only Netlet requests and passes any other request to the RProxy. EProxy parses the Netlet requests and passes them to the Netlet Proxy (if it is enabled) or directly to the destination host.

Figure 3-6 Netlet and the Sun ONE Portal Server, SRA Platform



How Netlet Works

The Netlet contains two server-side components and one client-side component. The server side components are:

- Eproxy
- Netlet Proxy

The client component is the Netlet applet.

Users invoke Netlet through the Netlet channel on the Portal Desktop. When a user clicks the appropriate link in a Netlet channel, Portal Server receives a request to download the Netlet applet. The Netlet applet is downloaded along with a preconfigured rule, which consists of the following information:

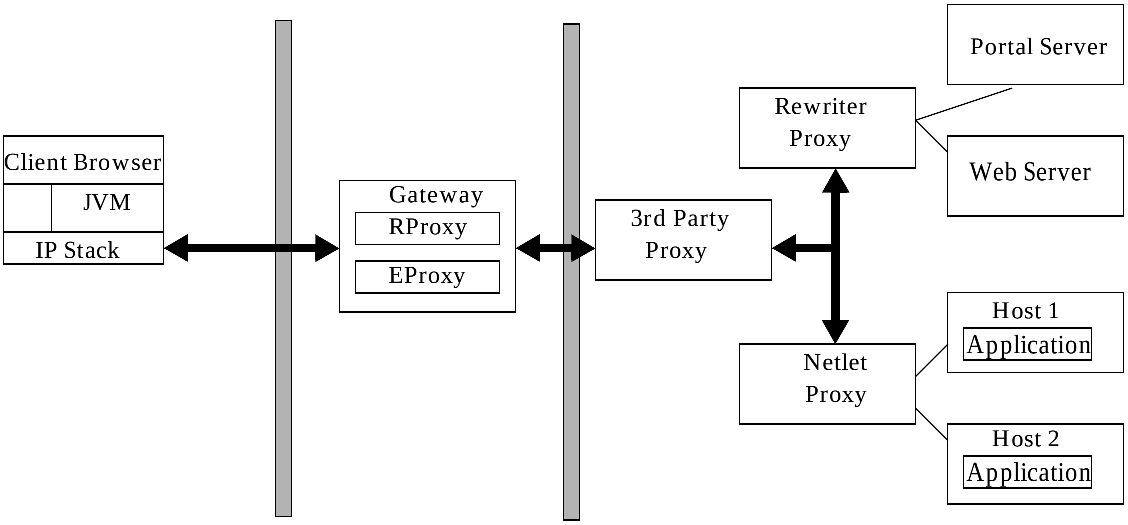
- Port to listen to on the client side
- Destination server host name
- Destination server port
- Encryption algorithm

See "Netlet Rules" on page 88 for more information.

After obtaining the appropriate permissions from the user, the Netlet applet starts listening for connections on the preconfigured ports. Next, it accepts the incoming connections, and routes the traffic through the Eproxy to the destination server after encryption. Upon receiving Netlet traffic, the Eproxy decrypts and routes the traffic to the appropriate destination server. Instead of the Eproxy decrypting and routing the traffic to the appropriate destination server, you can designate a Netlet Proxy between the Eproxy and the destination server. In this scenario, the Eproxy simply forwards the traffic to the Netlet Proxy, which decrypts and routes the traffic to the destination servers.

Figure 3-7 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 3-7 Netlet Tunnel via Proxies



Netlet and Authentication

Though Netlet does not authenticate users, it needs a valid SSO token for carrying out its operations. Netlet passes the SSO token with every request that is sent from the client to the Eproxy. Upon successful validation of the SSO token, the traffic is routed to the destination server. The Netlet applet also displays a warning message before accepting any new connections (on the machine where the applet is running). The user must acknowledge this message and type the Netlet password before accepting new connections. You can configure this message to be a check box, or you can force the user to type a password before the connection is allowed.

Static and Dynamic Port Applications

Static port applications run on known or static ports at which they can be contacted by clients. Examples include IMAP and POP servers, and Telnet daemons. In the case of static port applications, the Netlet rule includes the destination server port so that requests can be routed directly to their destinations.

Dynamic port applications agree upon a port for communication as part of the handshake. For dynamic applications, 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 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.

Encryption Algorithms

An encryption algorithm is used to encrypt/decrpt the traffic that is routed by the Netlet. These algorithms are selected by the user or administrator from the list of available algorithms. Netlet uses standard SSL algorithms for encrypting traffic. Netlet currently uses two different types of SSL implementations based on their availability of JSSE at the client side (browser).

The Netlet algorithms to be used for encryption and decryption are specified on a per rule basis. This increases the flexibility and security for all the Netlet connections between the client and the destination server.

The Netlet Applet automatically detects the capability of the client side JVM and downloads appropriate jar files for its functioning. For example if the Netlet detects a Java 2 VM with JSSE capability, only SSL algorithms are downloaded, not KSSL algorithms, hence reducing download time.

Further if the gateway is running in HTTP then the Netlet does not use any encryption-- it is considered to be plain and none of the encryption libraries will be downloaded with the Netlet applet.

At the server side, both Eproxy and Netlet Proxy use the standard JSS/NSS SSL library.

Note : For these algorithms to be successful as the part of SSL handshake, the corresponding gateway instance must have these algorithms enabled in its profile.

Algorithms Used When JSSE is Available

If the Netlet Applet is running on a browser that has Java 2 VM installed and if that version of JVM supports JSSE, then the Netlet Applet uses JSSE as its SSL implementation taking advantage of the SSL implementation that's available at the client side.

In this case, the Netlet Applet uses the following algorithms to connect to the gateway:

SSL_RSA_WITH_3DES_EDE_CBC_SHA

SSL_RSA_WITH_RC4_128_MD5

SSL_RSA_WITH_RC4_128_SHA

SSL_RSA_EXPORT_WITH_RC4_40_MD5

SSL_RSA_WITH_DES_CBC_SHA

SSL_RSA_WITH_NULL_MD5

Algorithms Used When JSSE is not Available

If the Netlet Applet is running on a browser that has Java 1 or Java 2 VM where JSSE is not available, the Netlet Applet defaults to its own custom SSL implementation called KSSL.

In this case, the Netlet Applet uses the following algorithms to connect to the gateway:

KSSL_SSL3_RSA_WITH_3DES_EDE_CBC_SHA

KSSL_SSL3_RSA_WITH_RC4_128_MD5

KSSL_SSL3_RSA_WITH_RC4_128_SHA

KSSL_SSL3_RSA_EXPORT_WITH_RC4_40_MD5

KSSL_SSL3_RSA_WITH_DES_CBC_SHA

Netlet Rules

A Netlet rule contains all the information that is needed for a Netlet connection between the client and the destination server. A Netlet rule consists of the following fields:

- Rule name
- Encryption algorithms
- URL
- Download applet
- Extend session
- Client port
- Target host(s)
- Target port(s)

See the *Sun ONE Portal Server Administrator's Guide* for more information.

Netlet rules are based on how you specify the destination host. The two types of Netlet rules are:

- **Static rule**—Specifies the destination host as part of the rule. In a static rule, the user cannot specify or change the destination host. For example, you could have an IMAP rule that enables connecting to a particular IMAP server, but the user could not change the IMAP server name.
- **Dynamic rule**—The destination host is not specified as part of the rule. The user can specify the destination host in the Netlet provider. For example, an administrator defines a Telnet rule in which the user can specify the particular host for Telnet connection.

Netlet Provider

The Netlet provider is a Portal Server Portal Desktop provider that supplies a channel with links for initiating Netlet connections as defined by the rules. The channel displays one link for every Netlet connection that can be initiated by a user. The Netlet provider also supplies the user with an interface to edit dynamic rules. Using dynamic rules, users can add or remove destination hosts as required. Users can also set the password for validating themselves at the time of invoking a new Netlet connection.

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.

Netlet and Split Tunneling

Split tunneling is when a VPN client can connect secure sites (for example, by using Netlet) and non-secure sites, without having to connect or disconnect the VPN—in this case, 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:

- 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.
- 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.
- 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. In the current release, this can be a simple check box or can require the user to enter a password. 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.

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.
- HTTP traffic is now secure between the gateway and the intranet even if the destination server only supports HTTP protocol (no HTTPS).

In case of intense traffic, the server that is running Sun ONE Portal Server software may become overloaded and the response time may be slow. To counter this, the Rewriter Proxy can be configured on a separate node.

Figure 3-8 on page 91 and Figure 3-9 on page 92 show typical deployments that include the Rewriter Proxy. In Figure 3-10, the Rewriter Proxy is installed on the same machine that is running Sun ONE™ Portal Server. In Figure 3-11, the Rewriter Proxy is installed on a separate node.

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

When the Gateway is configured with multiple instances of the Rewriter Proxy, the Gateway automatically load balances across different Rewriter Proxies. The Gateway keeps a list of active Rewriter Proxies to avoid routing a user request to an inactive Rewriter Proxy. This mechanism also avoids the single point of failure in the system. Also the Gateway maintains user request stickiness to a particular Rewriter Proxy. This improves SSL performance and a fail over occurs only when a particular Rewriter Proxy instance goes down.

Figure 3-8 Rewriter Proxy Installed on a Portal Server Node

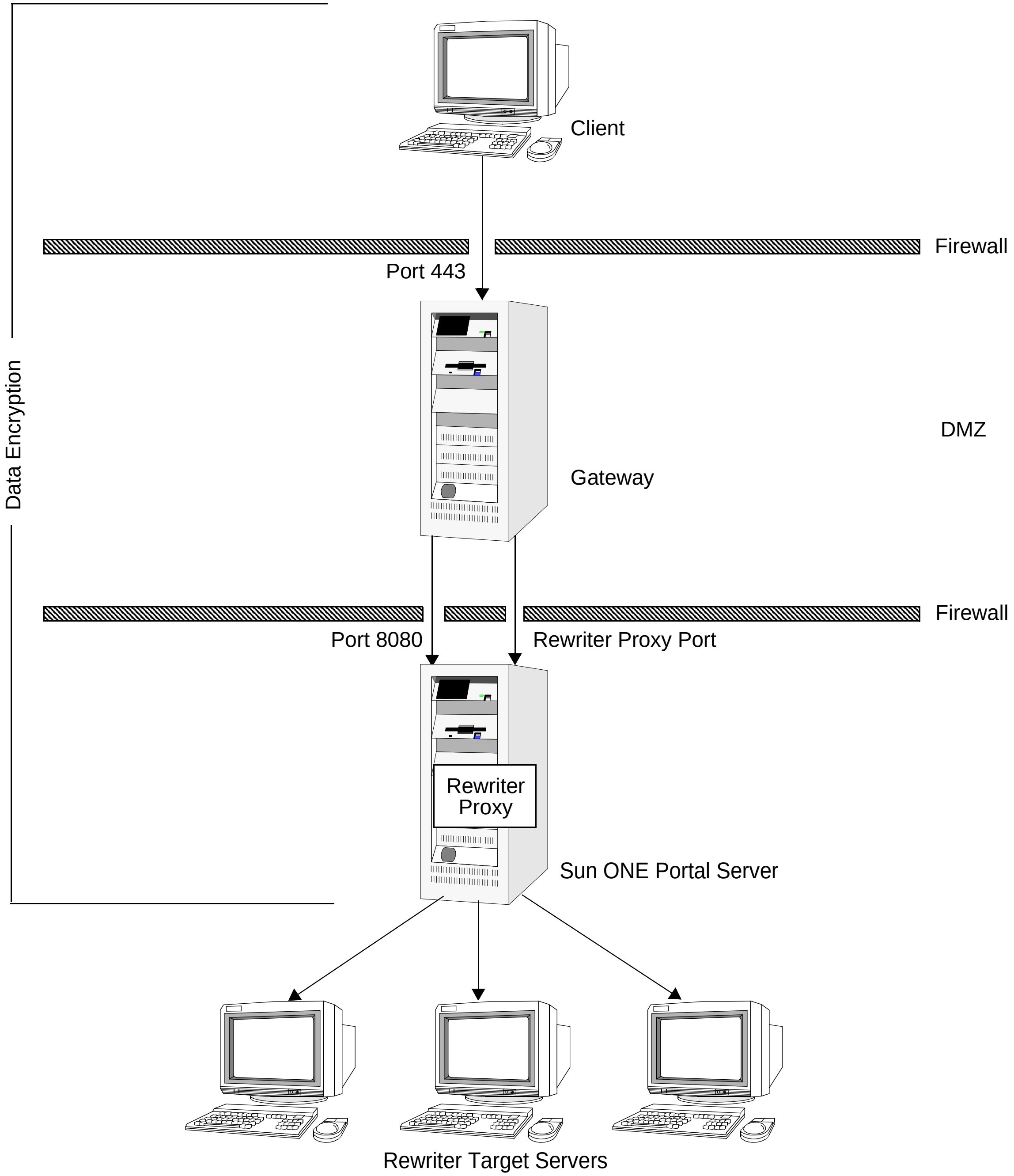
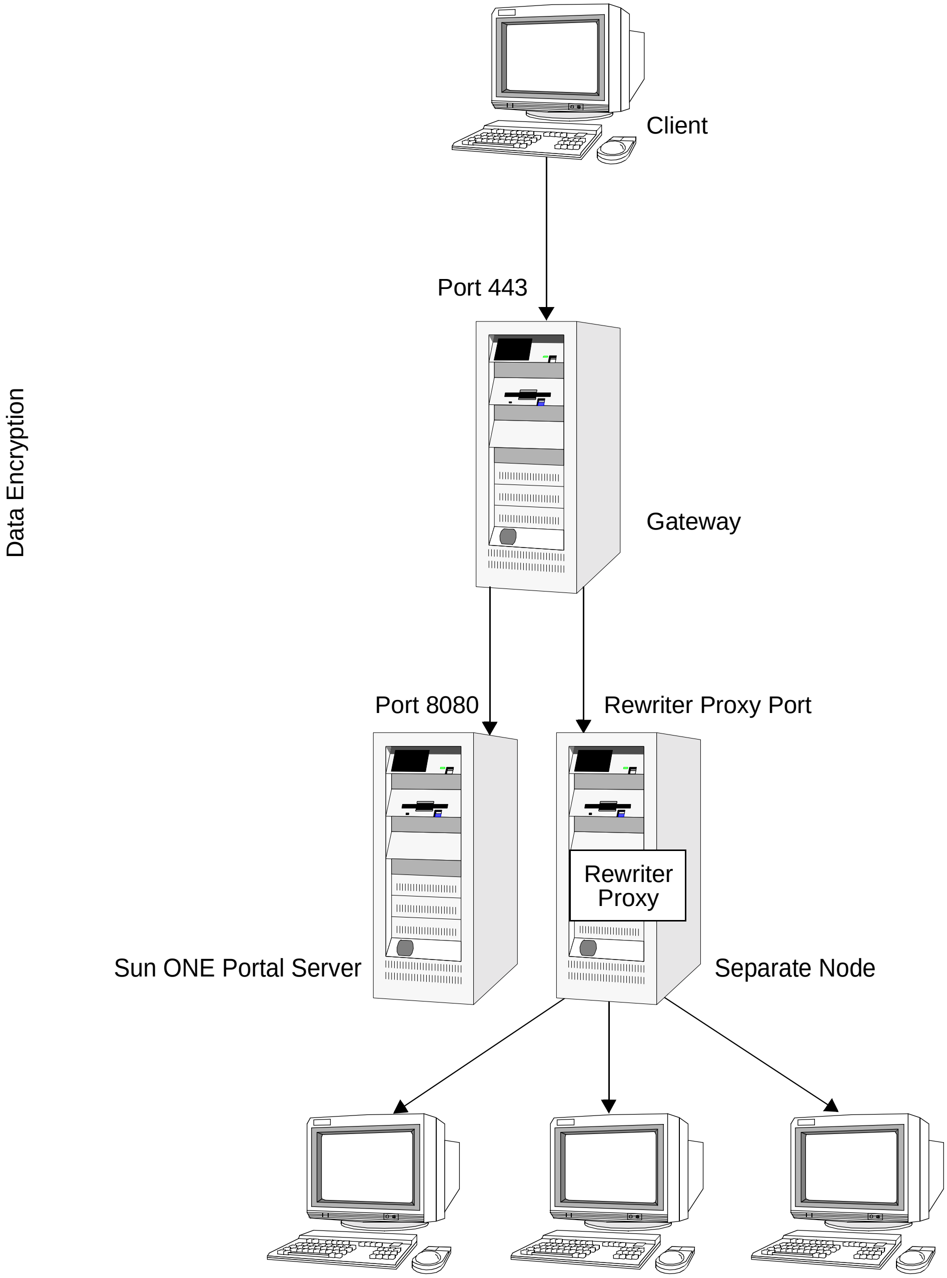


Figure 3-9 Rewriter Proxy Installed on a Separate Node



Netlet Proxy

Much like the Rewriter Proxy, Netlet Proxy also helps reduce the number of ports that you need to open in your firewall for connecting the Eproxy 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 Eproxy is in a DMZ. If Eproxy routes the traffic to all the destination servers, you would need to open a large number of ports in the second firewall. To alleviate this problem, you can use a Netlet Proxy behind the second firewall and configure Eproxy 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.

Figure 3-10 on page 95 and Figure 3-11 on page 96 show typical deployments that include the Netlet Proxy. In Figure 3-10, the Netlet Proxy is installed on the same machine that is running Sun ONE™ Portal Server. In Figure 3-11, the Netlet Proxy is installed on a separate node.

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

In both figures, the gateway ensures a secure tunnel between the remote client machine and the gateway. Netlet packets are decrypted at the gateway and sent to the destination servers. However, the gateway needs to access all the Netlet target hosts through the firewall between the demilitarized zone (DMZ) and the intranet. Conceptually, this requires opening a large number of ports in the firewall. However, the Netlet Proxy enhances security between the gateway and the intranet by extending the secure tunnel from the client, through the gateway to the Netlet Proxy that resides in the intranet. With the proxy, the Netlet packets are decrypted by the proxy and then sent to the destination server. This reduces the number of ports required to be opened in the firewall. Thus, in both figures, only one port is opened in the firewall for Netlet traffic. (Another port is opened for Portal Server traffic.)

NOTE You can run multiple Netlet Proxies to avoid a single point of failure and achieve load balancing.

When the Gateway is configured with multiple instances of the Netlet Proxy, the Gateway automatically load balances across different Netlet Proxies. The Gateway keeps a list of active Netlet Proxies to avoid routing a user request to an inactive Netlet Proxy. This mechanism also avoids the single point of failure in the system. Also the Gateway maintains user request stickiness to a particular Netlet Proxy. This improves SSL performance and a fail over occurs only when a particular Netlet Proxy instance goes down.

Figure 3-10 Netlet Proxy Installed on a Portal Server Node

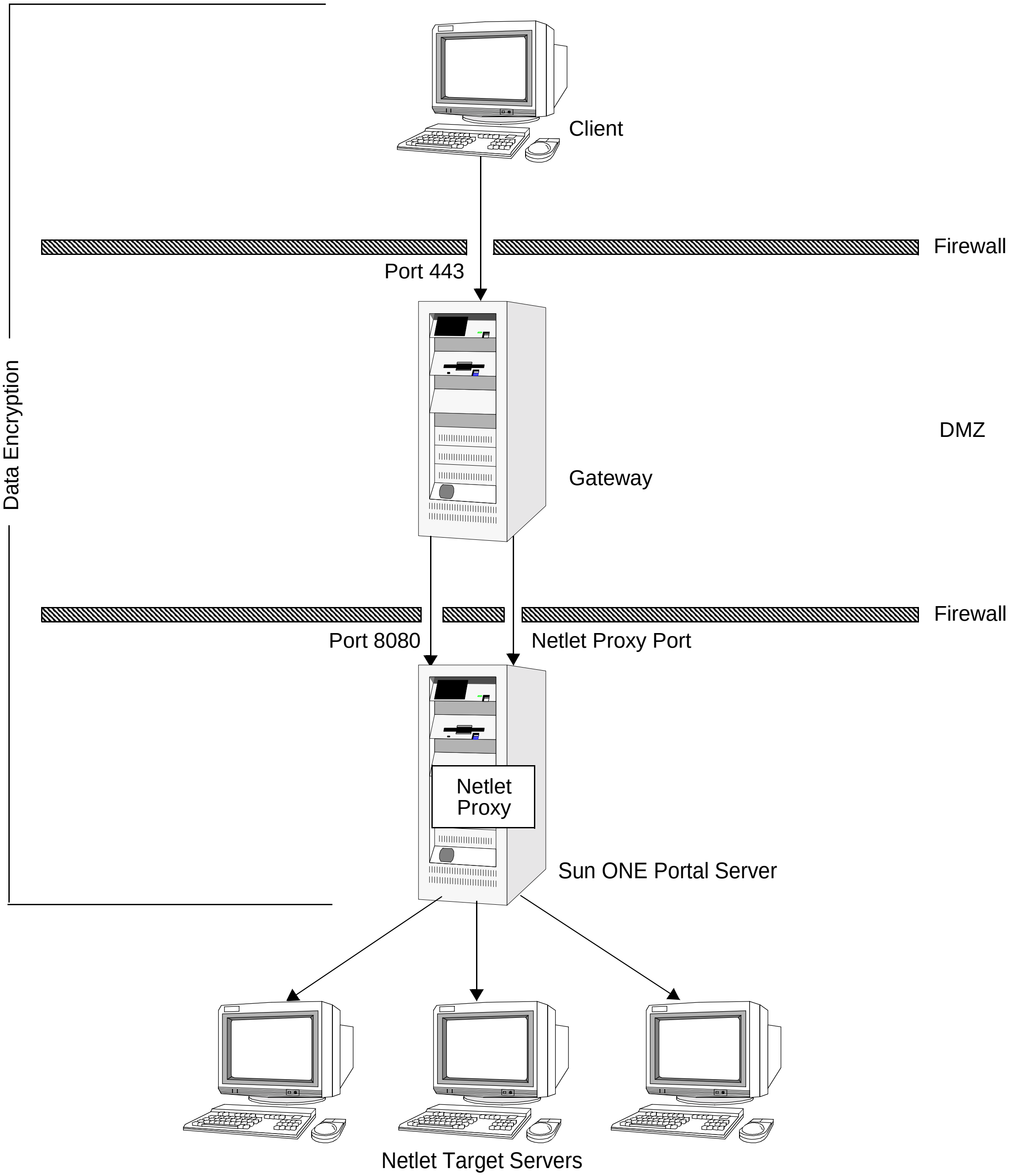
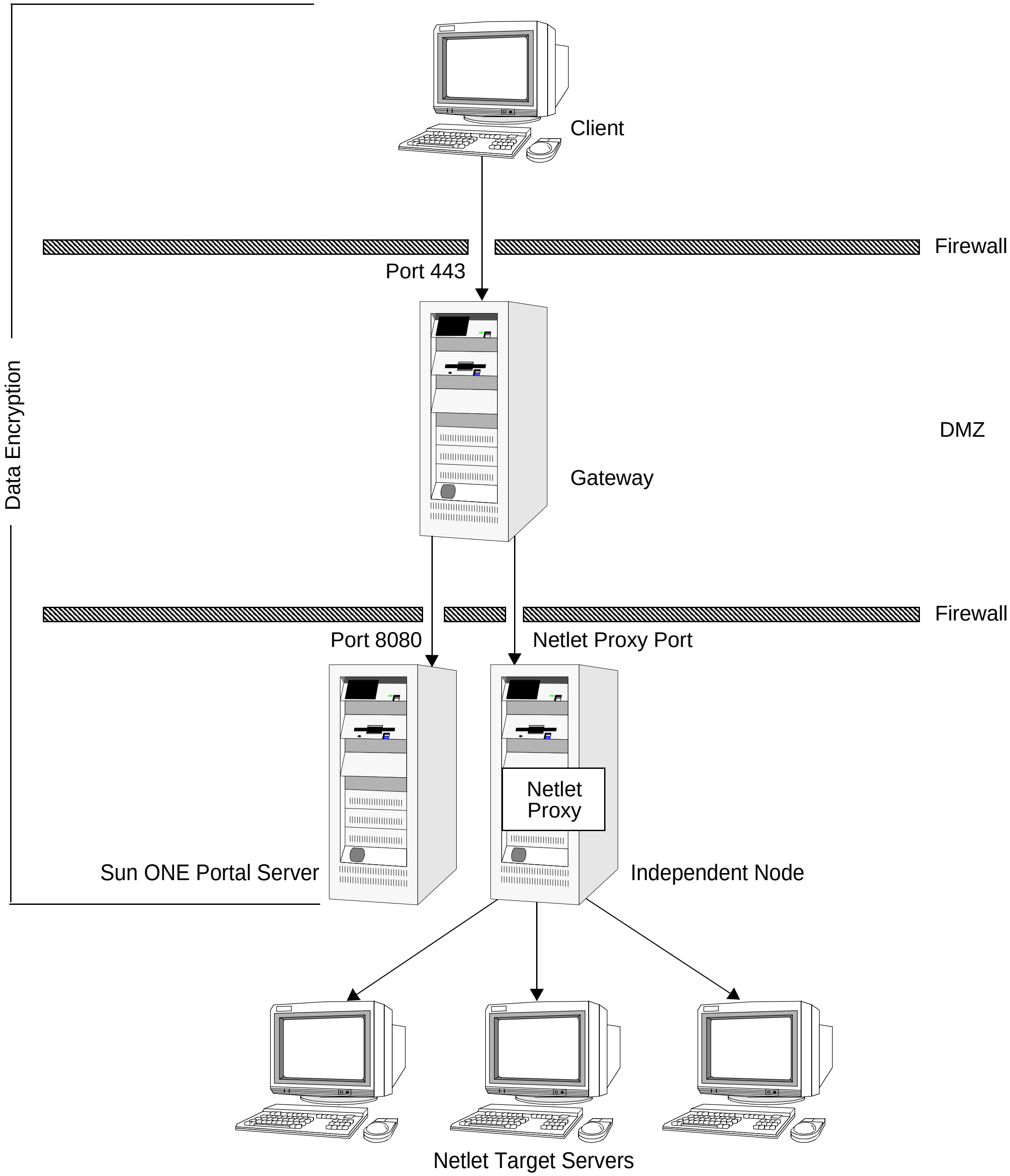


Figure 3-11 Netlet Proxy Installed on a Separate Node



NetFile

NetFile enables remote access and operation of file systems that reside within the corporate intranet in a secure manner, when accessed through the gateway. NetFile includes NetFile Java 1, an AWT-based user interface, and NetFile Java 2, a Swing-based user interface.

NetFile uses standard protocols such as NFS, SMB, 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 *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide* for more information.

NetFile 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 plugins.
- **NetFile servlet(s)**—There are two NetFile servlets present in the web container (of the server of your deployment), 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.

NetFile Initialization

When a user selects a NetFile link in the Portal Desktop, the NetFile servlet checks if the user has a valid Identity Server single sign-on (SSO) token (available only on successful authentication with the Identity Server), and permission to execute NetFile. If the user has a valid SSO token and permission to execute NetFile, 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.

Server and Shares

NetFile uses the underlying Network File System (NFS) and File Transport Protocol (FTP) to access file systems on UNIX platforms, Server Message Block (SMB) protocol to access file systems on Windows, and FTP to access file systems on Novell platforms.

All three platforms provide share-based access to their file systems. A share is a directory tree that you can configure for access from remote systems. On some platforms, you can associate a password with the share. You can have multiple shares on a single server, each with a distinct name associated with it. Users can connect to these systems using a specified protocol, and gain access to files under these shares after validating their credentials.

Using NetFile, users enter the name of the server and share along with their credentials. NetFile stores the credentials and reuses them for authentication, if the user saves the NetFile settings. You can configure the shares (known as common hosts) that users can access at a organization or role level. Users do not need to enter their credentials to access these shares (but a valid SSO token from Identity Server is necessary at all times). You can also deny access to common hosts for a set of users by not inheriting this attribute from an organizational or role level.

If a particular share is part of the common hosts, and also part of the denied list, users are denied access to the share.

NetFile supports access to shares from UNIX (as long as there is NFS support), Windows, Windows 95, Windows 98, Windows 2000, Windows XP, X86 and FTP over NetWare servers.

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 of the Identity Server to grant access to NFS file systems. For file systems that are accessed over FTP and SMB protocols, NetFile uses the methods provided by the protocol itself to validate the credentials.

NetFile 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 Allow or Deny lists to allow or deny access to servers at the organization, role, or user level. The Deny list takes precedence over the Allow list. The Allow and Deny lists can contain the * wildcard to allow or deny access to a set of servers under a single domain or subdomain.

NetFile 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 SMB protocol.These problems are removed by using Jcifs for accessing Windows hosts.

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

The Rewriter is an independent component that translates all URLs (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 *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide* for more information.

In a typical scenario, you install the gateway in a DMZ, with firewalls on either side (extranet and intranet). For the gateway to access machines on the intranet, you need to open extra ports in the firewall between the gateway and the intranet. T o minimize the number of open ports in the firewall, you use the Rewriter Proxy. You install the Rewriter Proxy within the intranet and you configure the gateway to use the Rewriter Proxy. Instead of trying to retrieve the contents directly, the gateway simply forwards all requests to the Rewriter Proxy, which fetches and returns the contents to the gateway.

SRA Authentication

SRA uses the authentication features of Identity Server to authenticate users. Even session validation happens through Identity Server's single sign-on (SSO) API. In fact, the gateway component treats the login pages just like any other content page and passes them to the browser. For Personal Digital Certificate (PDC) authentication, the gateway obtains the client certificate and passes it to the Identity Server host for authentication.

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 user has successfully authenticated, the gateway simply presents (after rewriting) the page returned by Identity Server to the user. In Portal Server, by default, this is the Portal Desktop page.

SRA Configuration Files and Directory Structure

This section describes the SRA directory structure and configuration files used to store configuration and operational data.

SRA Directories

Table 3-2 shows the platform-specific directory structures that are installed for SRA.

Table 3-2 Portal Server, SRA Directories

Description	Location
Default installation directory	*portal-server-install-root*`/`
Default installation directory for Identity Server executables, the web server, and the deployed applications	*portal-server-install-root*`/SUNWam`
Default installation directory for configuration information	`/etc/`*portal-server-install-root*`/SUNWps`
Log files	`/var/`*portal-server-install-root*`/SUNWam/logs`
Debug log files	`/var/`*portal-server-install-root*`/SUNWps/debug`

SRA Configuration Files

All SRA configuration data is stored using the Identity Server Services Management function. Identity Server provides the bootstrap configuration file that is needed to find the 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
```

Chapter 4

Analyzing Your Portal Requirements

This chapter describes how to analyze your organization’s needs and requirements that lead to designing your Sun™ ONE Portal Server software deployment.

This chapter contains the following sections:

- Identifying and Evaluating Your Business and Technical Requirements
- Mapping Portal Server Features to Your Business Needs

Identifying and Evaluating Your Business and Technical Requirements

The first step in planning your deployment is identifying your Portal Server business and technical requirements.

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.

Your technical requirements (often called functional requirements) 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. You need to gather both business and technical requirements before you can address architecture and design issues.

After obtaining both business and technical requirements, carefully evaluate them. Identify how realistic each requirement is. What would be the best design approach to satisfy each requirement and related requirements? Consider all the associated constraints (costs, time to deploy) and decide if any requirements need to be modified before determining the deployable solution. Evaluating your business and technical requirements will help you formulate a design that:

- Meets your business objectives today
- Offers the best available performance
- Provides high availability
- Is highly scalable
- Deploys easily
- Does not contain any single points of failure
- Provides just the right capacity to meet growth for the next several months
- Provides enough capacity to meet above normal peek usage
- Enables upgrade and migration paths
- Is cost-effective
- Can be deployed in a reasonable timeframe (A tight deadline might result in your changing of the architecture.)

Determining Your Business and Technical Requirements

This section provides a series of questions that you use to determine your business and technical requirements. Answering these questions alone does not provide the ultimate answer of what your portal architecture and deployment will look like. Instead, this is the first step in gathering your requirements in such a way as to describe the problems and opportunities facing your organization but without yet proposing a specific solution.

The questions in this section are grouped in the following areas:

- Business Objectives
- Technical Goals
- User Behaviors and Patterns
- Back-End Systems
- Front-End Systems
- Data Centers
- Growth Projections
- Security
- Search Engine
- Performance
- Availability
- Maintainability

Some questions in these areas will not apply to your portal design, and in some cases, you will identify and have to address issues that are not presented here.

NOTE Many organizations often contain their business and technical requirements within a single requirements document.

The Architectural Decision to Use Secure Remote Access Software

There are no other questions in this chapter pertaining directly to Sun™ ONE Portal Server, Secure Remote Access (SRA) software. Deploying SRA is an architectural decision, not an identification of requirements.

Business Objectives

The business goals of your portal affect deployment decisions. 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 (B2B), business-to-consumer (B2C), business-to-enterprise (B2E), 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

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.)

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?

Back-End Systems

Examine your back-end systems to verify that they can support your portal. Scalability, performance, and your data center organization are among the factors you need to assess.

Use these questions to help you understand your back-end systems:

- Does your organization have the technical competence and support organization to deploy and maintain the portal system?
- Are back-end systems scaled to levels that your portal will need?
- Can back-end systems support concurrent users during normal use? During peak use?
- What is the data bandwidth of your intranet?
- Can your existing back-end systems support the number of concurrent users expected for your portal?
- Will response time and other performance metrics of your existing back-end systems be adequate?
- How many geographic locations are involved? How does traffic in these areas vary?

- What response times will back-end systems deliver during normal times? During peak times?
- What type of redundancy do you have?
- How do you manage maintenance and production upgrades?

Front-End Systems

Analyze the front-end systems that will be used for access to your portal. This enables you to identify how your users will connect to your portal and what kinds of browsers they will use. These factors will affect your requirements.

Use these questions to help you understand your front-end systems:

- How will users access your portal?
- What environments do your users have? Will they be using airport kiosks, Internet cafes, homes, or corporate locations?
- What browser features do your users have? Do they have Java™ applications? Is JavaScript™ technology enabled? Is cookie support enabled? Are tables supported?

NOTE Minimum browser requirements will be determined by the types of applications you deploy through the portal.

Data Centers

Your data center structure and requirements often have an affect on your deployment decisions. The number of data centers and their location are factors to define. Accessing data from remote data centers significantly impacts overall portal response times.

Use these questions to help you assess data center requirements for your portal:

- How many data centers are used to host the portal?
- What data center requirements and restrictions exist?
- Where are the data centers located?
- What data bandwidth is available at each data center?
- What are the typical response times between identified data centers?

- How many routers, firewalls, or other devices exist between each data center and are there any bandwidth throttling devices installed?

Growth Projections

In addition to determining what capacity you need today, assess what capacity you will need in the future, within a time frame that you can plan for. Growth expectations and changes in how your portal is used are factors you need to accommodate growth.

Use these questions to help you set growth projections for your portal:

- What is the projected growth for the portal? How fast will the growth occur?
- How will your business objectives change in the next two or three years?
- What plans do you have for future content?

Security

Determine whether security is needed for your portal. If so, you must assess what kind is appropriate.

Use these questions to help you identify security requirements for your portal:

- Do users need to access intranet information through the Internet?
- Is SSL required for authentication to the portal?
- Is SSL required for any other part of the portal?
- What are your security policies?
- What are your single sign-on requirements?
- Are there requirements for running SSL internally between servers, and if so, what is the projected data flow? (This is important to know for deployment.)
- Do users have a universal user ID and password? If not, where is the user ID and password information stored and how will the portal access it if SSO functionality is required?

Search Engine

How you implement Search affects how you size the server you use for the Search Engine.

Use these questions to help you identify Search Engine requirements for your portal site:

- Which organizations will be using Search?
- In each organization how many documents are expected to be indexed in the Search database?
- How many documents or resource descriptions (RDs) will be in the Search database?
- How many concurrent Search users are expected? Will use occur at certain times of the day? Or at certain times in a business process? When are these times? How many concurrent users will occur at each time?
- How many concurrent searches are expected?
- Will you use document level security?
- Will you use a taxonomy for browsing?
- Will you use the discussion feature?
- Will you use the subscription feature?

Performance

The performance that your portal must deliver directly affects your deployment requirements. Scalability, capacity, and high availability are some of the standards you need to consider.

Use these questions to help you evaluate performance requirements for your portal:

- What performance requirements exist?
- What high availability requirements exist?
- What response times are acceptable? How do the response times of your stand-alone systems compare with response time requirements of your portal?

- If you size your portal infrastructure for good response times during regular hours, can you tolerate a possible degradation in performance during peak load times?
- How many concurrent sessions, or connected users, are likely during peak use? (Count only users who are active. Do not include users who are, for example, away on vacation, or on leave.)
- What is the above-normal peak time? How does this information affect your peak concurrent user estimate?
- What sort of user activity occurs during peak periods? Logins or reloads?
- 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?

Availability

How you implement a highly-available system affects the ability of the system to provide agreed system access levels over time.

Use these questions to help you assess the availability requirements for your portal:

- What high availability requirements exist specifically for the portal server?
- What are the requirements for preventing any single point of failure to the system?
- Have data centers and delivery points, network, and back-end systems been designed to be highly available?
- How effective is the production support organization? Does the support staff have the necessary skills, processes, and procedures in place to adequately maintain a production portal system?
- Understand your availability requirements for each component of your deployment. Different components might require different levels of availability, and hence a different design and configuration approach.
- Are you concerned about denial of service attacks?
- Do the size and availability requirements of your portal warrant the use of a clustered LDAP server? If so, does your organization have the expertise to operate and maintain the cluster in the event of problems?

Maintainability

Determine how you want to administer and maintain your portal.

The following type of questions will help you identify maintainability requirements for your portal:

- What are the requirements for backing up and restoring your portal?
- How will user provisioning be handled?
- Will you use delegated administrators for your portal?
- Does your support organization have the processes in place and technical abilities to troubleshoot and fix portal problems?
- Will you train your support organization in maintenance and operations of the portal?

Mapping Portal Server Features to Your Business Needs

The previous sections posed questions to you about the various areas of the Portal Server platform 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 ONE 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 4-1 shows the identity management features and their benefits.

Table 4-1 Identity Management Features and Benefits

Feature	Description	Benefit
Directory service	Sun ONE Portal Server uses Sun™ ONE Identity Server and Sun™ ONE Directory Server.	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. 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 Directory Server products.
User, policy, and provisioning management	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.	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. Consolidates management of users and applications. Personalizes content and service delivery. Simplifies and streamlines information and service access. Reduces costs associated with managing access and delivery. Provides secure policy-based access to applications. Ensures secure access as portal deployments expand beyond employee LAN access.

Table 4-1 Identity Management Features and Benefits *(Continued)*

Feature	Description	Benefit
Web 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.

Personalization

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

Table 4-2 shows the personalization features and their benefits.

Table 4-2 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.

Table 4-2 Personalization Features and Benefits *(Continued)*

Feature	Description	Benefit
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 4-3 shows the aggregation and integration features and their benefits.

Table 4-3 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.

Table 4-3 Aggregation Features and Benefits *(Continued)*

Feature	Description	Benefit
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.

Search Services

Portal Server includes a secure Search Engine, enabling users to search content and receive only those results that they are authorized to receive.

Table 4-4 shows the Search features and their benefits.

Table 4-4 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 4-4 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.

SRA

Adding SRA extends your portal to remote and mobile employees or business partners without the additional cost of administration and maintenance found in a traditional Virtual Private Network (VPN) solution.

Table 4-5 shows the SRA features and their benefits.

Table 4-5 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.	Extends an enterprise's content, applications, files, and services located behind firewalls to authorized suppliers, business partners, and employees. To prevent denial of service attacks, you can use both internal and external DMZ-based gateways.
Remote access	Users achieve remote access through three components: • Gateway • NetFile • Netlet	The gateway presents content securely from internal web servers and application servers through a single interface to a remote user. NetFile, a file manager application, enables remote access and operation of file systems and directories. Netlet facilitates the running of popular or company-specific applications on remote computers in a secure manner. After you implement the Netlet at your site, users can securely run common TCP/IP services, such as Telnet and SMTP, and HTTP-based applications such as pcAnywhere or Lotus Notes.
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

Table 4-5 SRA Features and Benefits *(Continued)*

Feature	Description	Benefit
Netlet Proxy	Provides an optional component that extends the secure tunnel from the client, through the gateway to the Netlet Proxy that resides in the intranet.	Restricts the number of open ports in a firewall between the demilitarized zone (DMZ) and the intranet.
Rewriter Proxy	Redirects HTTP requests to the Rewriter Proxy instead of directly to the destination host. The Rewriter Proxy in turn sends the request to the destination server.	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. • HTTP traffic is now secure between the gateway and the intranet even if the destination server only supports HTTP protocol (no HTTPS).

SHARP Features

SHARP (Scalability, High Availability, Reliability, and Performance) features within Portal Server provide horizontal scalability (for example, adding additional hardware to increase overall system capacity) and vertical scalability (by adding additional portal instances to maximize hardware utilization).

Table 4-6 on page 120 shows the SHARP features and their benefits.

Table 4-6 SHARP Features and Benefits

Feature	Description	Benefit
Scalability	You can configure Portal Server to meet the demands of different deployment scenarios. You can scale a server horizontally by: -adding additional servers to your portal. The overall goal is to provide a system that is both fault tolerant and has no single point of failure from both a software and hardware perspective. You can scale a server vertically by: - adding additional software instances of Portal Server (deployed in a web server container), thus providing fault tolerance on a single server. - adding more system resources, such as CPUs, memory, and disks.	*Scalability* enables a system to increase load or improve overall system performance. *Horizontal scaling* distributes the workload among different systems. Horizontal scaling allows for a building module approach to increasing overall portal system capacity. See "Working with Portal Server Building Modules" on page 162 for more information. *Vertical scaling* enables an organization to increase fault tolerance and maximize the performance of an existing system. Within Portal Server, vertical scaling is achieved by running multiple instances of Portal Server, each with its own JVM™. **Note**: Vertical scalability is only available in web server deployment.
High Availability	Provides redundant services and the ability to redirect requests in the event of a service failure.	High availability is achieved through software replication. You can configure the portal system to run multiple instances of each web application, thereby providing a backup if one of the instances fails. The portal system uses Identity Server services for session management and non-local data access. Therefore, the portal system inherits all the benefits and constraints of Identity Server with respect to high availability. The Identity Server 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. Configuring Sun ONE Directory Server with multiple masters ensures that users can always login and authenticate. If one directory master fails, another is able to take over. Also, Directory Server offers a way to prevent denial of service attacks by setting limits on the resources allocated to a particular bind DN.

Table 4-6 SHARP Features and Benefits *(Continued)*

Feature	Description	Benefit
Reliability	Provides for no single point of failure (NSPOF) when you use portal building modules in your deployment. See Chapter 7, “Creating Your Portal Design” for more information.	A portal 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 reliability. Increased reliability is introduced with load balancing, which is responsible for detecting Portal Server failures and redirecting users’ requests to a backup building module.
Performance	Overall Portal Server performance is a complex equation involving all aspects of the network and the applications it needs for data retrieval. However, if you design and build the portal system for fault tolerance, no single point of failure, and the capacity to exceed projected user loads, overall system performance should meet requirements.	When deployed using the building module configuration (see Chapter 7, “Creating Your Portal Design”), Portal Server shows that performance and capacity increase linearly when additional resources are added within a building module (that is, CPU and memory), and when more building modules are added.

Chapter 5

Sizing Your Portal

This chapter describes how to establish a baseline sizing figure for your Sun™ ONE 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.

This chapter contains the following sections:

- Overview of the Portal Sizing Process
- Portal Sizing Process for SRA
- Portal Sizing Tips

Overview of the Portal Sizing Process

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™ ONE Portal Server, Secure Remote Access (SRA) software are covered in "Portal Sizing Process for SRA" on page 132.

Your Sun ONE 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
- Customization
- Hardware and Applications
- Back-end Servers
- Transaction Time
- Workload Conditions

A discussion of the these performance factors follows.

Peak Numbers

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

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

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

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

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

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

The calculation for this example is:

```
40000 / 10 = 4000
```

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

Average Session Time

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

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

Search 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:

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

The average size adjusts for variations in sizes of RDs. A collection of long, complex RDs with many indexed terms and a list of short RDs with a few indexed terms require different search times, even if 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 ONE 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 occurs 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 "Portal Sizing Process for SRA" on page 132.

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.

Portal Sizing Process for SRA

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 ONE 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 ONE 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 then on the number of users, active users, or user sessions. Any sizing information for the gateway has to be based on a set of assumptions. See "Secure Remote Access Example" on page 152 fore 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 ONE 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 admin 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 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
- Secure Portal Pilot Measured Numbers

NOTE After your Sun ONE 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 more 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, in 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

Here are two examples for SRA:

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 ecommerce applications.

See the *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator’s Guide* for more information on the Sun Crypto Accelerator 1000 board.

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, as the communication between the gateway and Rewriter Proxy is HTTPS and communication between the gateway and Netlet Proxy is equal to SSL (Netlet Proxy uses Netlet protocol, not HTTP). The gateway instead tunnels the Netlet requests to the Netlet Proxy.

About the 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.

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.

Chapter 6

Understanding the Portal Deployment Life Cycle

This chapter provides on overview of the portal deployment life cycle. Use this chapter to help you develop your overall portal deployment project plan.

This chapter contains the following sections:

- Overview of the Portal Deployment Life Cycle
- Creating the Portal Deployment Plan
- Understanding the High-level and Low-level Portal Design
- Implementing and Verifying the Portal
- Moving to a Production Environment

Overview of the Portal Deployment Life Cycle

Deploying a complex product such as Sun™ ONE Portal Server software requires considerable planning effort. To simplify this effort, the following high-level steps break down the portal deployment life cycle into more manageable phases:

1. Creating the deployment plan
2. Creating the high-level portal design
3. Creating the low-level portal design
4. Implementing and verifying the trial portal
5. Rolling out the trial portal to a production deployment
6. Ongoing monitoring and tuning of the portal

The following sections in this chapter discuss these deployment phases in more detail.

Creating the Portal Deployment Plan

Your portal deployment plan is essentially a task list that assigns resources to specific tasks and deliverables. The initial project plan provides important task-level information regarding the steps and order of implementation. Compare your known business and technical requirements against the project plan to look for proper alignment, order of execution, appropriate duration, and any tasks that you might have overlooked.

Though each organization will differ in its own project plans and project management approaches, there are similar elements that should be included in a portal project plan for the deployment to be effective. Ultimately, your project plan will become a functional roadmap for your deployment.

NOTE The portal deployment plan should be read and understood by all system stakeholders with an interest in the detailed workings of the system. Most importantly, this includes those individuals who are building the system and those who need to use it to carry out their business responsibilities.

Your portal project plan answers questions such as:

- Are the tasks in the appropriate order?
- Are the staff resources appropriate?
- Is the method of integrating third-party products to support the overall solution appropriate?
- What is the approach to integrating your existing applications and clients with the portal?
- Have you established reasonable milestones along with signoffs from the appropriate stakeholders?

See Appendix B, “Portal Deployment Worksheets”” for a list of the key portal design tasks. This list, in the form of a worksheet, will assist you with developing your project plan. Though these tasks will vary depending on your organization and the scale of each deployment, the worksheet presented in the appendix represents the most common phases and tasks encountered.

Defining Project Objectives and Scope

As the first step in planning your portal deployment, you define your project objectives (what you want to accomplish) and scope (what you plan to include and exclude).

When you define your project objectives, specify your organization’s business goals and how Portal Server helps to meet those goals. For example, you might have a business goal of having users authenticate once and only once to be able to use web applications and data. Portal Server meets this goal by making use of the SSO API contained within the Sun™ ONE Identity Server product. By being clear in your project’s objectives and scope, can better manage expectations that arise during the project.

In addition to making sure that your project plan clearly states project objectives and defines its scope, provide methods and ways to measure successes and the overall progress. For example, you might use a web site within your organization to track portal deployment status, or send out regular deployment status emails to stakeholders.

When defining a portal project’s scope, keep in mind the following demands and requirements that your organization might have:

- Delivery of a portal solution to meet today’s business objectives
- Best performance
- High availability
- Scalability
- Straight-forward, easy deployment
- No single point of failure
- Delivery of the right capacity to meet future growth
- Delivery of enough capacity to meet above normal peak
- Easy migrations and upgrades to future releases

When you scope your portal solution, you must meet the above requirements while at the same time balancing these requirements with a solution that is deliverable within a reasonable timeframe. The ultimate goal of the project objectives and scope is to provide hardware and software recommendations, an initial man-hour estimate for all work, and a recommendation for the network layout.

TIP To run an efficient project, carefully prioritize the requirements, based on input from all stakeholders, and manage its scope. Too many projects suffer from developers working on features the developer finds interesting and challenging, rather than focusing early on tasks that mitigate risk in the project or stabilize the architecture of the application.

To make sure that you resolve or mitigate risks in a project as early as possible, develop your system incrementally. Carefully choose requirements for each increment that alleviate known risks in the project. To do so, you need to negotiate the scope (of each iteration) with the stakeholders of the project. This typically requires good skills in managing expectations of the output from the project in its different phases. You also need to have control of the sources of requirements, of how the deliverables of the project look, as well as the development process itself.

Understanding the High-level and Low-level Portal Design

The high-level and low-level design steps of the portal deployment life cycle are where you create the actual portal design or architecture. The high- and low-level designs represent the complete set of designs for your deployment.

In designing your portal architecture, start by describing your high-level architecture and implementation, which arise out of your business requirements and sizing needs, as identified in Chapter 4, “Analyzing Your Portal Requirements” and Chapter 5, “Sizing Your Portal”. The high-level architecture is intended to communicate the architecture of the system and provide the basis for developing the detailed design for your portal solution.

The low-level design gets into specifics, including such aspects as your complex of servers, networking considerations, content design and implementation, Identity Server architecture, and application integration.

See Chapter 7, “Creating Your Portal Design” for more information on the high- and low-level design concepts.

Implementing and Verifying the Portal

After you have completed your high- and low-level portal design, you begin the implementation and verification stages of deploying your portal. This stage involves development, testing, validating against your performance criteria, and taking all this into account to implement a redesign of the portal architecture, if necessary.

Content Aggregation

Content aggregation is a portal feature that enables content from various disparate sources to be combined and presented to users in a single interface, known as the Portal Desktop. The Portal Server providers produce the content in the form of channels in the Portal Desktop.

One of the most important aspects of a portal is its ability to integrate 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.

Items to consider when developing your portal content include:

- Container and channel display profile definitions
- Whether to use static or dynamic content
- Content feeds
- Caching
- Content licensing and permissions
- Personalization

The availability and response time of the servers providing the content (and source of the content) will impact the Portal Desktop in the following ways:

- Channels might appear without data.
- The loading, and overall performance, of the Portal Desktop can be affected if the response of content servers is slow.

Content Management

Content management functionality is critical to managing your portal content. Content management covers the full life cycle of publishing information to the portal, ranging from content creation and collaboration to workflow, version control, staging, and publishing of new and updated content to the portal. While content is most often internal data that is useful to portal users, it can also be external news feeds and data sources. Because the portal content might come from many different sources, content management functionality is required to prevent erroneous or unapproved content being placed on the portal.

NOTE Portal Server does not provide tools for creating and adding portal content. You must obtain a third-party product to perform content management. See "Content and Document Management ISVs" on page 33 for more information.

Source Control

Though not a requirement for Portal Server, using source control software is recommended. Source control software provides a systematic way of controlling a complex migration and development process. Using source control provides your organization with a definitive snapshot of the production system for any given code release, and an easy way of reverting to a previous release level if you encounter problems.

Source control software also ensures that a file's source is not changed improperly. This includes making sure that your developers are using the latest version of the file, and that no two developers edit the file at the same time, resulting in overwritten changes. Source control also ensures that the files are not deleted, moved, or otherwise changed in such a way that they cannot be rolled back to the original files.

Add all portal files that you will modify to the source control system, including configuration, HTML, JavaServer Pages™ files, and Java™ source code files. Portal Server does not provide a source control system, so you will need to purchase one, if your organization does not already use one.

TIP When you set up your source control directory structure, use one that is similar to the UNIX® directory structure where the portal software is installed. Also, create "editions" or "tag" your source control tree when deploying content to a production system. This way you have complete control over the system and know exactly what state the system is in.

Testing the Portal

Your overall portal testing strategy involves performing a series of complementary testing activities in a phased approach on all system components. This testing strategy includes the following:

- **Unit testing**—Performed by a developers to ensure that modules they develop are working per the design document.
- **Functional testing**—Performed by business test owners and technical test owners to ensure that combinations of individually unit-tested pieces of code and basic functionality perform as they were designed to, as they are combined into a complete subsystem. This testing phase is performed within the application, interface, and infrastructure component boundaries.
- **Integration testing**—Performed by business test owners and technical test owners to ensure that there is proper interoperability between subsystems. In addition, cross platform business processes are tested to ascertain they work according to specifications. This testing phase is performed across the boundaries of multiple applications, interfaces, and infrastructure components.
- **User acceptance testing**—Performed by business test owners to verify that all requirements are met by the system, prior to sign-off.
- **Performance and stress testing**—Performed by stress testing owners using a load-testing tool. Load balancing and failover are tested at this time.
- **Security testing**—Performed by an organization's security group to verify that users can access only those functions and data that they have permissions for, and to verify that only those users with access to the systems, applications, and data are permitted to access them.

Analyzing Performance Test Results

Perform the following types of analysis on your performance test results:

- **How channel response times degrade under increasing server load**—When the server load is light, response times can be acceptable. However, as the load increases, performance usually drops off. Comparing a light load test with a heavy load test shows if there is a problem in the channel design.
- **How different profile types react under the same load**—These test results show how the addition of specific channels introduces latency. The average page return time between tests with the same concurrent user loads should produce similar numbers. If significant time discrepancies appear between tests using different profiles, this identifies a potential problem area. Expect small increases in the average page return time between profiles because additional information is being retrieved and rendered each time the customized page is displayed.

Conducting the Portal Trial

After you test your portal design in a non-production environment, such as a lab, you will want to conduct a trial portal deployment in your production environment. A trial portal deployment usually involves a limited number of users. Conducting a trial run of your portal deployment minimizes risks that you might encounter in a full-scale portal deployment.

Benefits of conducting a trial include:

- Verification that your design works in the production environment as expected
- Confirmation that your design meets your organization’s business requirements
- Experience in the deployment life cycle with installation, configuration, and customization of the product
- Ability to document and revise production deployment documentation, such as a “run book”

During the trial, users must be able to provide feedback on how the portal works for them. You can use this feedback to fix any problems that might arise and to develop an idea in terms of what kind of support you will need for the full-scale deployment. A portal trial will ultimately lead you to conclude that a full deployment can proceed, or that you need to spend more time resolving issues.

In general, you structure your trial into phases, to further minimize risks during deployment.

Conducting the trial process is iterative, and involves the following general steps:

1. Creating the trial deployment plan
2. Deploying the trial
3. Supporting and monitoring the trial
4. Obtaining feedback
5. Modifying portal design; if necessary, repeating steps 2 - 4
6. Moving to full-scale deployment

A plan for your trial portal addresses the following:

- **Scope and objectives**—Use specific objectives that your trial needs to meet. Also, identify criteria to gauge the trial's success.
- **Who is participating**—Establish the appropriate selection criteria to decide which users you want to participate in the trial. Choose users who are typical of your organization.
- **Training for participants**—Before the trial begins, decide how and when to train your participants. Be sure to identify your training resources.
- **Support plan**—Address who in your organization will provide support for the trial, to what level that support extends, and how users will report problems and seek resolution.
- **How you want to communicate trial status**—Describe how the trial participants will receive information prior to the start of the trial, and how status reports about the trial will be delivered.
- **Rollback plan**—Develop the rollback procedures needed in case the trial runs into problems or fails. Include criteria for when to use the rollback plan, and possibly establish levels of severity for potential problems.

Moving to a Production Environment

The last phase of of the deployment life cycle is moving from a trial environment to a production environment. Moving to a production environment occurs after your have thoroughly tested your portal and operated it as a trial deployment to test and refine your design.

Factors to consider in your move to a production environment include:

- Staging
- Quality assurance

- Using change control
- Managing the site
- Monitoring and tuning
- Documenting the portal

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've defined. Identify which services 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.

See the *Sun ONE Portal Server 6.2 Installation Guide* for more information on the configuration parameters for optimizing the performance and capacity of Portal Server. The `perftune` script (located in the *portal-server-install-root*`/SUNWps/bin` directory), bundled with Portal Server, automates most of the tuning process discussed in this guide. See Chapter 8, "Monitoring and Tuning Your Portal", in this guide, for additional monitoring and tuning information.

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.

Chapter 7

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

At this point in the Sun™ ONE 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™ ONE Portal Server, Secure Remote Access (SRA), high availability, security (including Sun™ ONE Identity Server), and Directory Server architectural components. See "Logical Portal Architecture" on page 153 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 infrastucture, 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™ 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.
- Identity 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.

The low-level design is described in more detail in later portions of this chapter.

Logical Portal Architecture

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

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

The logical architecture 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.

Understanding the Goals of Portal High-Level Design

Table 7-1 provides a series of goals for developing your portal high-level design. Prioritize these goals according to your organization's own requirements.

Table 7-1 Goals of Portal High-Level Design

Goal	Description
Costs	Keep in mind the cost-benefit ratio for designing an architecture that focuses too much on less essential or time-critical information.
Open standards	Avoid proprietary solutions that can limit your options. Strive for maximum interoperability and flexibility.
Simplicity	Keep your solutions straightforward and easy for all to comprehend and implement.
Performance and scalability	Design your portal to be elegant while at the same time capable of handling current loads efficiently. Build solutions that support future growth over time with a minimum of disruption and cost.
Modularity (separation of functionality)	Create a high-level design that shields the implementation as much as possible from end users. Design an architecture that you can modify easily, using replaceable components that use standard, well-defined interfaces.
Accessibility	Guarantee access to applications and data to all those who might need access in the future. For example, plan for future use of internal, group-specific information by other members of your organization or the external world of suppliers, co-suppliers, vendors, and customers. At the same time, provide easy-to-install and easy-to-manage security mechanisms to protect your organization's assets as required.
Availability	Design your solution for maximum robustness and redundancy for mission-critical applications and data.
Manageability	Provide your administrators with the ability to view and control the enterprise from the highest level down to the lowest level of detail as needed.

Designing Portal SHARP Features

A major focus of your portal design concerns SHARP (Scalability, High Availability, Reliability, and Performance) features. SHARP features provide horizontal (for example, the directory's replication mechanisms) and vertical (for example, multiple instance support) scaling.

In the portal design phase, you need to consider high availability, distributed multi-site functionality, high performance, and other requirements that will impact or stress the architecture. Also consider any known technical or business requirements that are deemed a high-risk. Then address these high risks in the design phase, at least in conceptual or strategic terms.

NOTE This deployment guide does not address topics such as high availability configurations for external content providers, legacy system applications, calendar servers, and messaging servers. Refer to the documentation on those specific products for more information on high availability scenarios.

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, "Analyzing Your Portal Requirements" and Chapter 5, "Sizing Your Portal" 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 run the Portal Server software on one machine and the Directory Server software on another. 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 162, discusses an approach to a specific type of configuration that provides optimum performance and horizontal scalability.

Portal Server and High Availability

High Availability ensures 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:

```
Availability = uptime / (uptime + 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, 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 Components

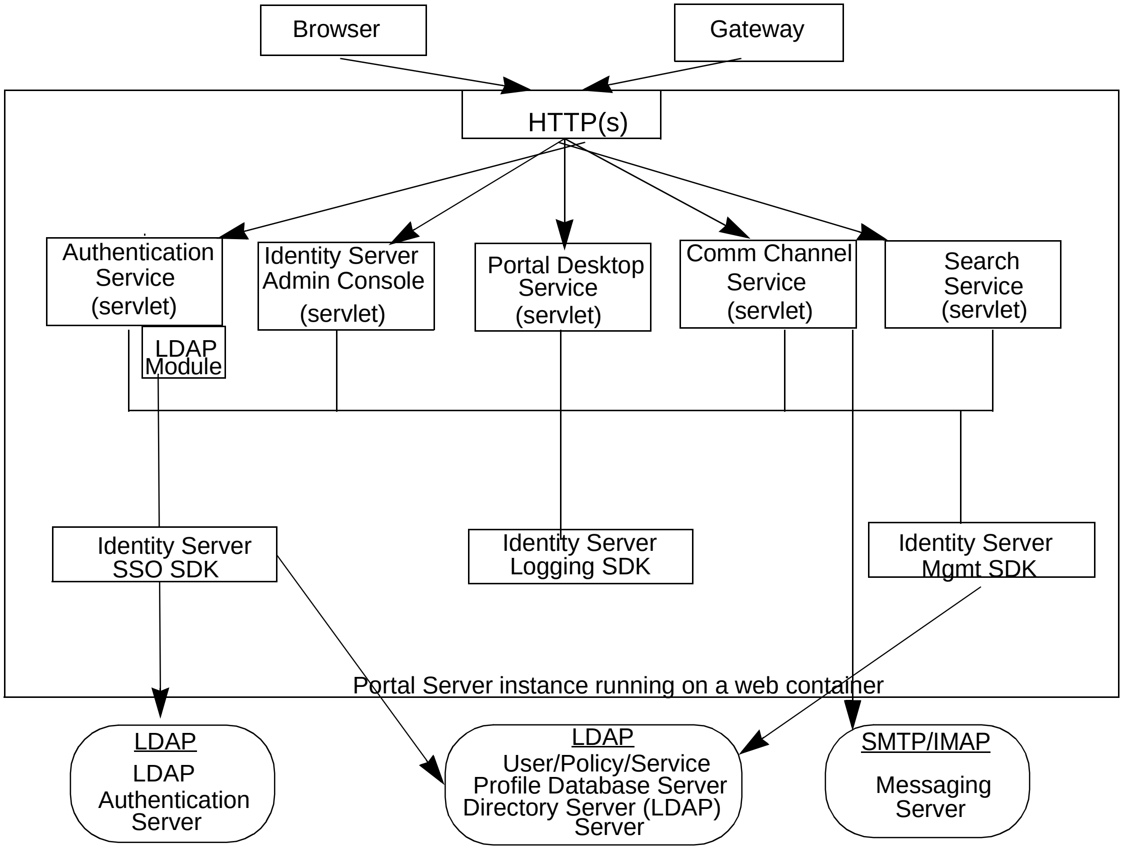
Making Portal Server highly available involves ensuring high availability on each of the following components:

- **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.
- **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™ ONE Web Server product.)
- **Directory Server**—A number of options make the LDAP directory highly available. See "Building Modules and High Availability Scenarios" on page 163 for more information.
- **Netlet Proxy**—In the case of a software crash, a watchdog process automatically restarts the Netlet Proxy. In addition, the gateway performs load balancing for the Netlet Proxy as well. The gateway also supports failure detection failover for Netlet proxies.

Portal Server System Communication Links

Figure 7-1 on page 160 shows the processes and communication links of a Portal Server system that are critical to the availability of the solution.

Figure 7-1 Portal Server Communication Links


Browser
Gateway
HTTP(s)
Authentication Service (servlet)
LDAP Module
Identity Server Admin Console (servlet)
Portal Desktop Service (servlet)
Comm Channel Service (servlet)
Search Service (servlet)
Identity Server SSO SDK
Identity Server Logging SDK
Identity Server Mgmt SDK
Portal Server instance running on a web container
LDAP
LDAP Authentication Server
LDAP
User/Policy/Service Profile Database Server Directory Server (LDAP) Server
SMTP/IMAP
Messaging Server

In this figure, the box encloses the Portal Server instance running on Sun ONE 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 7-1 on page 160 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 (either Sun ONE Web Server or certain application servers). 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™ ONE 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 SRA 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 the Netlet Proxy on a separate node.

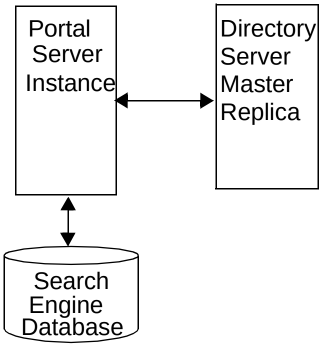
Working with Portal Server Building Modules

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™ ONE 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 7-2 shows the building module architecture. This figure shows the building module components, which include a Portal Server instance, a Directory Server master replica, and a search engine database.

Figure 7-2 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

Sun™ ONE Portal Server 6.2 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 7-2 summarizes these high availability scenarios along with their supporting techniques.

Table 7-2 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

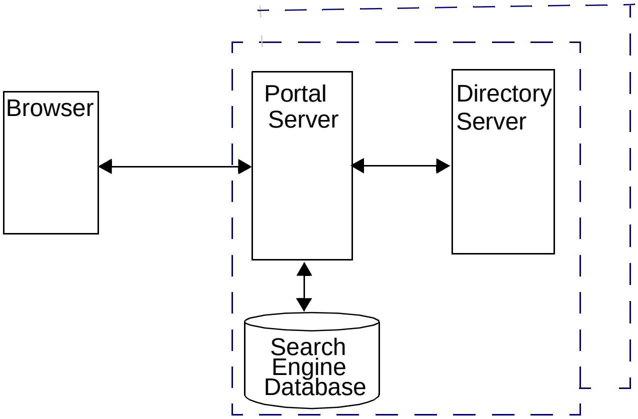
NOTE Load balancing is not provided out-of-the-box with the Sun ONE Web Server web application container.

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 7-3 shows a small, best effort deployment using the building module architecture.

Figure 7-3 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.) For sizing information, see Chapter 5, "Sizing Your Portal".

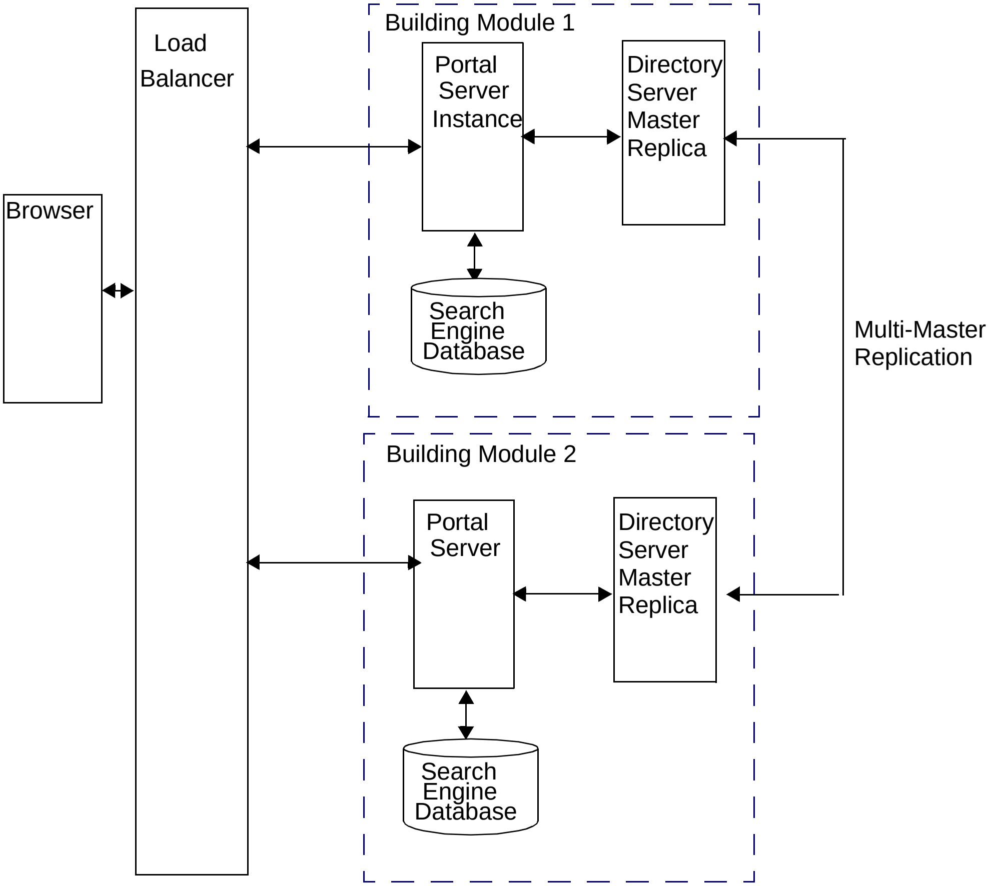
This scenario might suffice for task critical requirements. Its major weakness is that a maintenance action necessitating a system shutdown results in service interruption.

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 7-4 on page 166 shows an NSPOF scenario.

Figure 7-4 No Single Point of Failure Example Scenario



As stated earlier, a building module consists of a 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™ ONE 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 places between the bulding modules. The changes that occur on each directory are replicated to the other, which means that each directory plays both roles of supplier and consumer. For more information on MMR, refer to the *Sun™ ONE Directory Server Deployment Guide.*

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

Redundancy is equally important to the directory master so that profile changes through the administration console or the Portal Desktop, along with consumer replication across building modules, can be always 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.

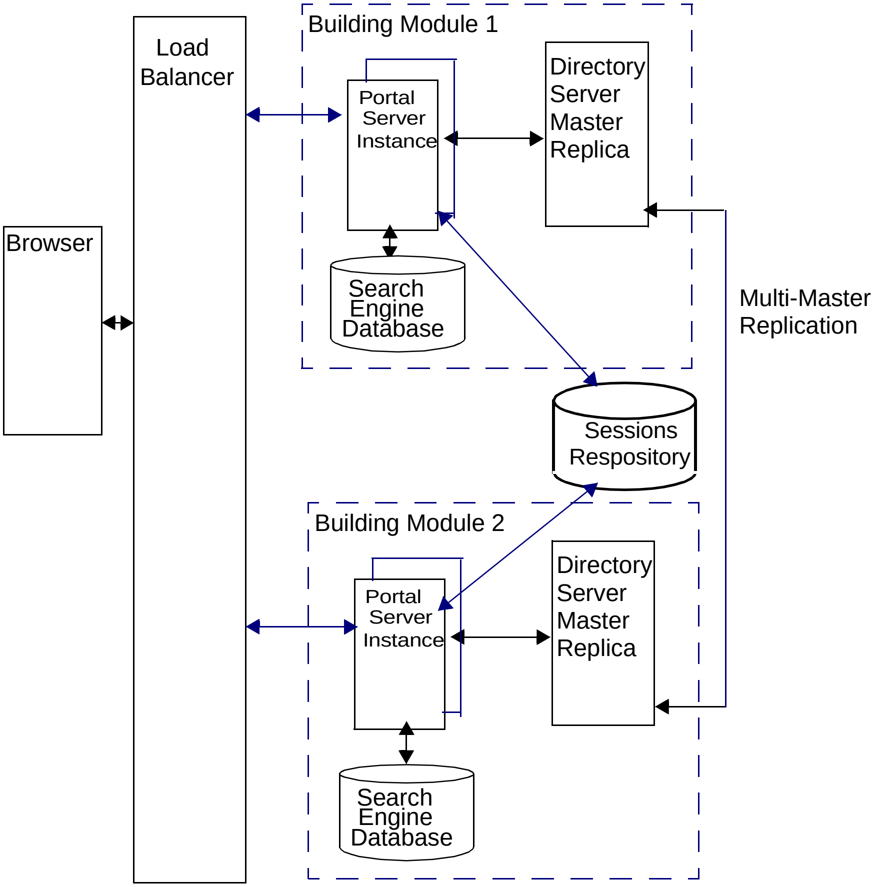
The NSPOF high availability scenario is suitable to business critical deployments. However, some high availability limitations in this scenario might not fulfill the requirements of a mission critical deployment.

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 7-5 on page 169 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 respository.

Figure 7-5 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 ONE Web Server and Sun ONE Application Server do not support HTTPSession 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 171.

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 *Sun ONE 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 7-2 on page 163.)
- 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 *Sun ONE Portal Server 6.2 Administrator's 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.

NOTE The goal of use cases is to describe the "what," not the "how" of the portal. Do not try to design your system by using use cases.

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 7-3 describes a use case for a portal user to authenticate with the portal. This is a two-column table. The first column describes the use case item and the second column provides a description.

Table 7-3 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, the 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 7-3 Use Case: Authenticate Portal User *(Continued)*

Item	Description
Description	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, reduces 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. For more information 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 *Sun ONE Portal Server 6.2 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 *Sun ONE Portal Server 6.2 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. For more information, see:

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

Using a Demilitarized Zone (DMZ)

Central to security is creating a demilitarized zone (DMZ) where the gateway servers typically reside. This is where the outermost firewall enables only SSL traffic from the Internet to the gateways, which then direct traffic to servers on the internal network. For maximum security, the gateway is installed in the DMZ between two firewalls.

Using the Gateway

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 for communication between the user's browser and Portal Server.

Designing SRA Deployment Scenarios

The gateway, Netlet, NetFile, Netlet Proxy, and Rewriter Proxy 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.

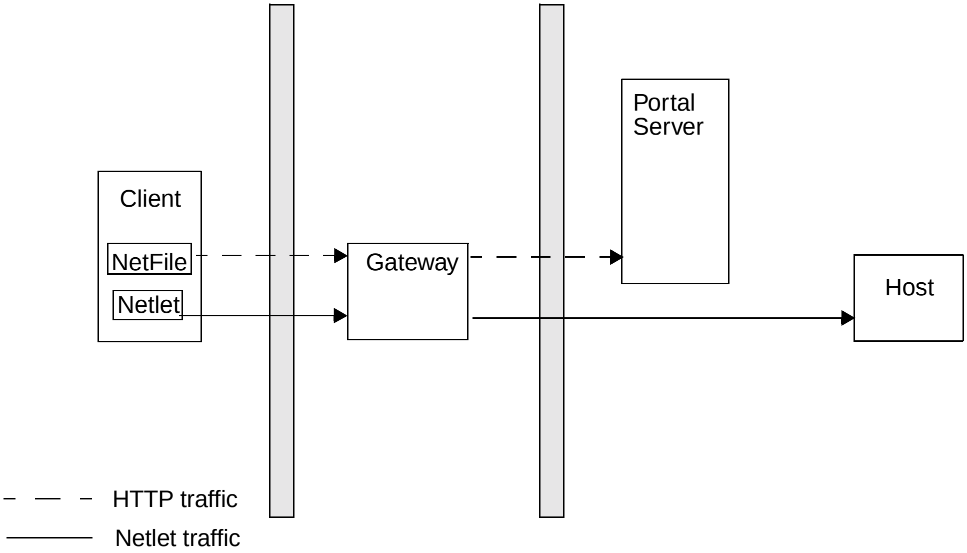
Scenario 1: Basic Configuration

Figure 7-6 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 demilitarized zone (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. This is not recommended if security policies do not permit it. Instead, 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 7-6 Scenario 1: Basic Configuration

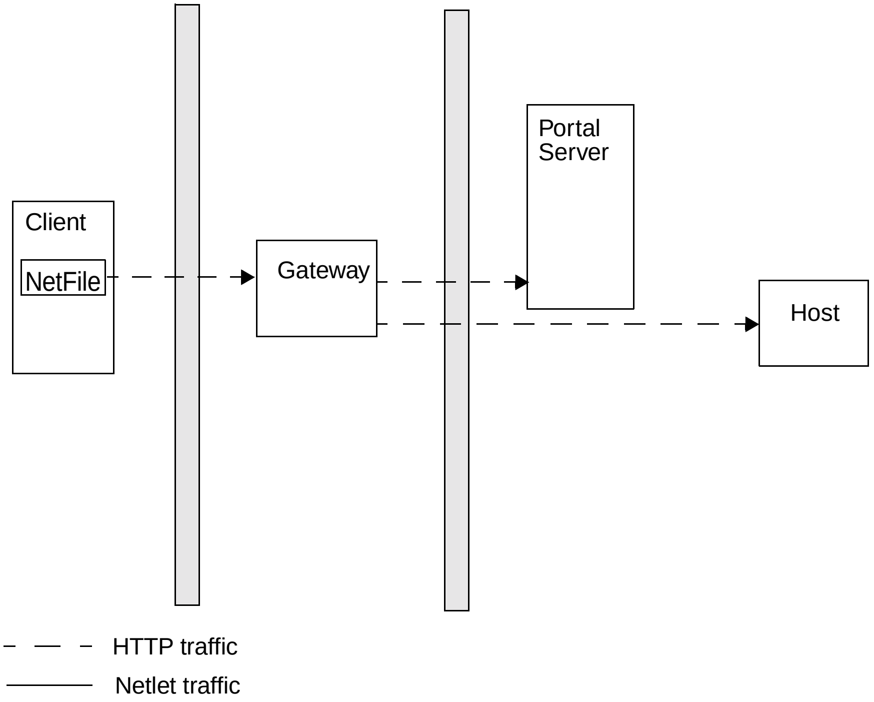


Scenario 2: Disabled Netlet

Figure 7-7 shows a scenario similar to Scenario 1 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 the performance improvement that it provides.

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

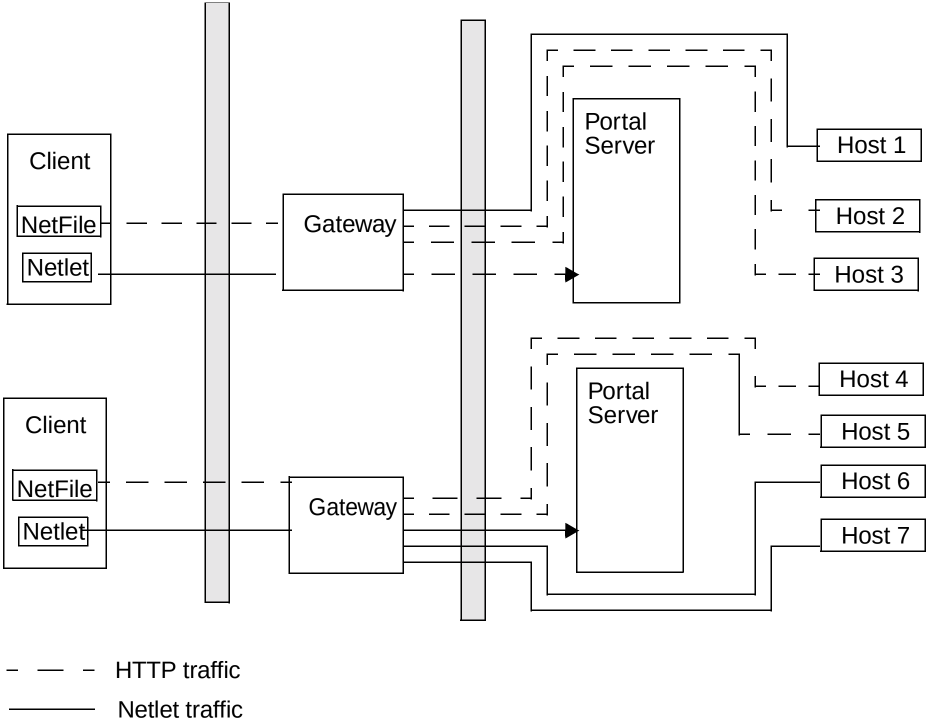
Figure 7-7 Scenario 2: Disabled Netlet



Scenario 3: Multiple Gateway Instances

Figure 7-8 shows an extension of Scenario 1. 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 *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator's Guide* for details.

Figure 7-8 Scenario 3: Multiple Gateway Instances



NOTE Although Figure 7-8 on page 181 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. Scenario 4 overcomes this problem by using the Netlet and Rewriter proxies.

Scenario 4: Netlet and Rewriter Proxies

Figure 7-9 shows a configuration that overcomes the security issues of Scenario 3 by having a Netlet Proxy and a Rewriter Proxy on the intranet.

The gateway need not contact the application hosts directly now, but will forward all Netlet traffic to the Netlet 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.

Also, to provide end-to-end SSL up to the Portal Server node, you can install a Rewriter Proxy on the Portal Server node. This ensures that the traffic between the gateway in the DMZ to the Portal Server node within the intranet is also SSL and hence secure.

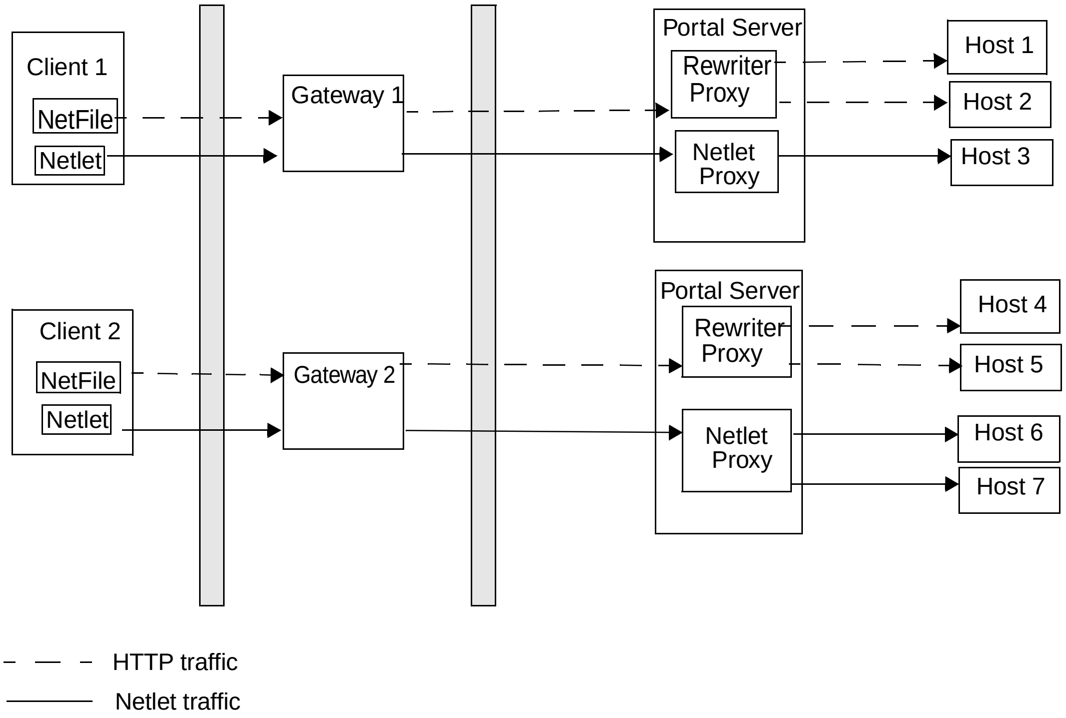
The traffic between the gateway in the DMZ and the Netlet Proxy is encrypted, and gets decrypted only at the Netlet Proxy, thereby enhancing security.

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.

Including the Netlet and Rewriter proxies in the configuration reduces the number of ports opened in the second firewall to 2.

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 in Scenario 5 where the Netlet Proxy is installed on a separate node to reduce load on the Portal Server node.

Figure 7-9 Scenario 4: Netlet and Rewriter Proxies



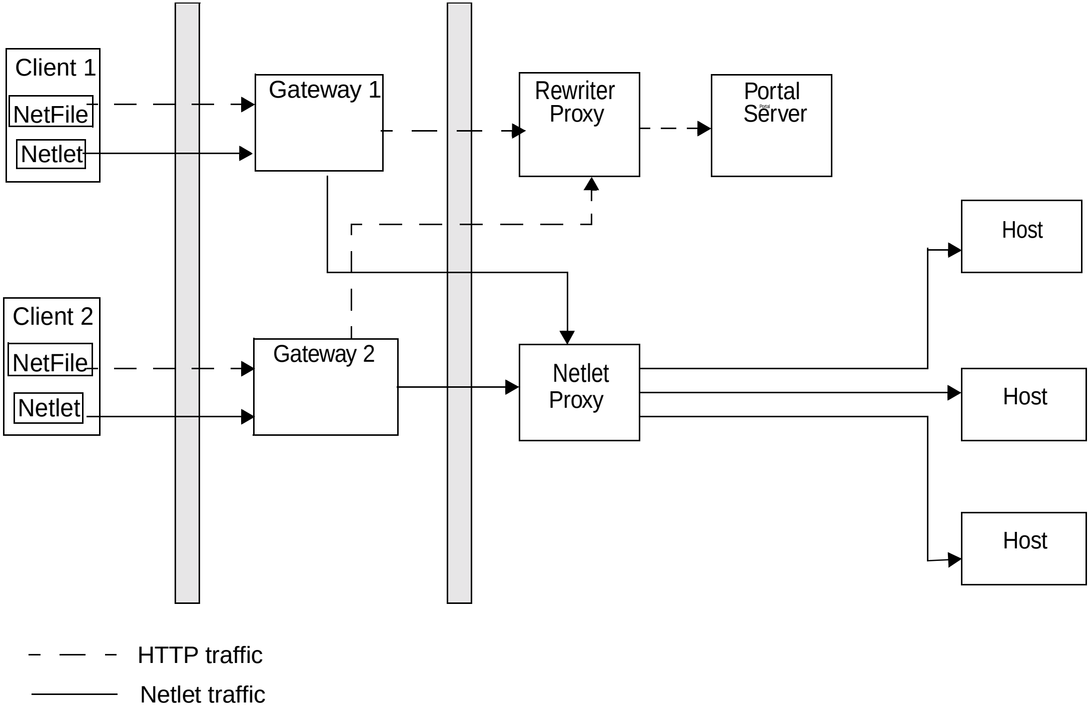
Scenario 5: Netlet Proxy on a Separate Node

To reduce the load on the Portal Server node and still provide the same level of security at increased performance, install the Netlet Proxy on a separate node. 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 Netlet Proxy needs to be directly accessible from the DMZ.

Figure 7-10 shows the Netlet Proxy on a separate node. All the Netlet traffic from the gateway is directed to this separate node, which in turn directs the traffic to the required intranet hosts.

You can have multiple instances or installations of the Netlet Proxy. You can configure each gateway to try to contact various instances of the Netlet Proxy in a round robin manner depending on availability. See Chapter 3, “Configuring the Netlet,” in the *Sun ONE Portal Server, Secure Remote Access 6.2 Administrator’s Guide* for details.

Figure 7-10 Scenario 5: Proxies on Separate Nodes



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 lanuguage 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 *Sun ONE Portal Server 6.2 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 *Sun ONE Portal Server 6.2 Developer's Guide* and *Sun ONE Portal Server 6.2 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.

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 *Sun ONE Portal Server 6.2 Administrator's Guide*, *Sun ONE Directory Server Deployment Guide*, and the *Sun ONE Identity Server Deployment Guide* for more information on planning your Identity Server and Directory Server structure.

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. Readying 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 Sun ONE 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.

See “Independent Software Vendor Integrations with Portal Server” on page 31 for more information on third-party applications that have been integrated to work with Portal Server.

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**—Here, 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.

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 the Java programming language. Using this API, applications can access message stores, and compose and send messages.

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 Aggregration 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:

- `URLScraperProvider`—Typically you use this provider to access dynamic content that is supplied by another web container's web-based system. It uses HTTP and HTTPS calls to retrieve the content. This provider puts high requirements on the back-end system, as the back-end system has to be highly scalable and available. Performance needs to be in 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.

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

NOTE The URL scraper provider that is part of Sun™ ONE Portal Server 6.2 can also function as a file scraper provider.

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

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

This is the best performing provider, in terms of how fast it 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 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.

- `XMLProvider`—Transforms an XML document into HTML using an XSLT (XML Style Sheet Language) file. You must create the appropriate XSLT file to match the XML document type. `XMLProvider` is an extension of `URLScraperProvider`. This provider uses the JAXP 1.2 JAR files that come with Sun™ ONE Web Server 6.2 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 `URLScraperProvider`.
- Database provider—This type of provider utilizes some back-end database for its content. It requires that you build database connection polling and that you use small queries (either single queries, or no more than a couple). You might also have to perform extra work in the way of 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 *Sun ONE Portal Server 6.2 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.

The Sun™ ONE Portal Server, Mobile Access 6.2 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 browers.

Mobile Access software supports nearly all of the 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 Sun ONE Portal Server, Mobile Access could support any number of channels, including automobiles, set-top boxes, PDAs, cellular phones, and voice.

Chapter 8

Monitoring and Tuning Your Portal

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

This chapter contains the following sections:

- Monitoring Sun ONE™ Portal Server
- Tuning Portal Server

Monitoring Sun ONE™ 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™ ONE Identity Server
- Portal Desktop
- Sun™ ONE Directory Server
- Java™ Virtual Machine

While there are emerging technologies that will enable you to perform detailed monitoring of Portal Server components, 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. As described in Chapter 7, "Creating Your Portal Design", 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 sort out 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 describes rules of thumb 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 not allocated 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 UltraSPARCIII™, a full GC on a 2 GB heap can take over ten seconds. During that period of time, the system is unavailable to respond to web requests. During a reliability test, full GCs are clearly visible as spikes in the response time 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` JVM™ parameter.

CPU Utilization

When deployed using the building module concept (as described in Chapter 7, "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 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 Identity Server schema.

Tuning Portal Server

The `perftune` script (in the *portal-server-install-root*`/SUNWps/bin` directory), bundled with Portal Server, automates most of the portal tuning process. The `perftune` script does the following:

- Tunes the Solaris™ Operating System Kernel and TCP settings
- Modifies configuration files for:
    - Sun™ ONE Web Server 6.0
    - Sun™ ONE Application Server 7.0
    - Sun™ ONE Directory Server
    - Sun™ ONE Identity Server
    - Sun™ ONE Portal Server Desktop
    - Sun™ ONE Identity Server authentication service

See the *Sun ONE Portal Server 6.2 Installation Guide* for complete information.

Appendix A

Troubleshooting Your Portal Deployment

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

This appendix contains the following sections:

- Troubleshooting Sun ONE™ Portal Server
- Troubleshooting SRA

Troubleshooting Sun ONE™ Portal Server

This sections contains troubleshooting information for 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.

Directory Server:

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

Identity Server:

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

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
```

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.

Stopping and Starting Portal Server

Use the following commands to stop and start the portal and its associated processes. You do not need to stop the server to restart it. If you start a server that is already running, the server is stopped and restarted.

➤ To Stop Portal Server

Type:

```
/etc/init.d/amserver stop
```

➤ To Start Portal Server

Type:

```
/etc/init.d/amserver start
```

To start multiple instances of Portal Server, type:

```
/etc/init.d/amserver startall
```

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™ ONE 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 Web Server.

Configuring a Sun ONE 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 ONE support personnel need to troubleshoot problems in your deployment.

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™ ONE 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 Portal Server, Portal Server with Secure Remote Access support, 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™ ONE Identity Server, Sun™ ONE Directory Server, and Sun™ ONE 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 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*

Appendix B

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 B-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 B-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 Sun ™ ONE Portal Server to your partners? Suppliers? Customers? Employees?

8. Who is the target community for your portal?

Table B-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 B-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 B-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 B-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 B-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 B-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 B-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 B-8 lists the major portal deployment phases and key design tasks. Use this task list in conjunction with Chapter 6, "Understanding the Portal Deployment Life Cycle", 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 B-8 Key Design Task List *(1 of 7)*

Major Phases and Tasks	Subtasks
1. Project Start and Coordination	
Project Planning	• Perform general project management

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

Major Phases and Tasks	Subtasks
Project Plan Review	• Review pre-implementation • Review business requirements • Review technical requirements • Review architectural documents • Review hardware and infrastructure
Coordinate Resources	• Identify skills required • Identify resources • Schedule resources • Assemble project team members • Review work plan with project team members
Define Requirements	• Collect business requirements • Summarize requirements • Confirm functional requirements • Collect technical requirements • Summarize technical requirements • Confirm technical requirements • Prepare combined requirements document • Deliver requirements
2. Design	
Develop Solution Architecture	• Design software architecture • Design server topology • Document architecture
Develop Portal Integration	• Understand system integration approach • Define container and channel layout • Define content aggregation • Define SSO approach • Develop custom Netlet and authentication modules
User Interface Design	• Prepare or modify user interface design • Develop or update screen specifications • Review and approve user interface model

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

Major Phases and Tasks	Subtasks
Directory Design	• 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
3. Develop and Integrate	
Install Software for Testing and Development Environments	• Install Sun™ ONE Portal Server software and optionally Sun™ ONE 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	• Install Portal Server and optionally Sun™ ONE Portal Server, Secure Remote Access (install appropriate supporting software) • Install application server, if needed • Install other software • Test server software components • Document test findings
Software Configuration	• Apply specific software configuration requirements • Create product configuration matrix

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

Major Phases and Tasks	**Subtasks**
Portal Server, Application Server, and Other Software Modifications	• Review your organization's requirements and expectations • Establish modifications for software • Establish methods for software modifications • Create software modification plan • Design software modifications • Establish software modification teams • Create modifications • Test modifications • Obtain appropriate stakeholder and organizational review and approval of modifications
LDAP Directory Setup	• Confer with stakeholders to establish proper schema • Establish modifications for software • Establish methods for software modifications • Create software modification plan • Design software modifications • Establish software modification teams • Create schema • Set up LDAP • Receive and verify data • Modify mapping as required for LDAP • Establish data update methods • Test directory • Create client user documentation for update methods
Legacy Software Integration (such as PeopleSoft, SAP)	• Perform integration • Prepare package integration test plan • Perform integration test • Produce package integration test results

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

Major Phases and Tasks	**Subtasks**
Reporting	• Establish reporting requirements for organization • Create reporting plan • Establish reporting team • Design reports • Create reports • Test reports • Review reports with customer • Provide information and training on report tool
Test	• Establish test plan
Plan User Acceptance Test	• Identify user acceptance test manager • Develop user acceptance test strategy and procedures • Review strategy and procedures with customer • Obtain approval for strategy and procedures • Develop user acceptance test roles and responsibilities • Obtain integration test scenarios • Review test conditions and acceptance criteria and revise • Develop user acceptance test schedule • Prepare acceptance test log and update with scenario test assignments
Conduct User Acceptance Test	• Execute user acceptance test • Identify and document user acceptance test discrepancies • Resolve user acceptance test discrepancies • Re-execute user acceptance tests and track user acceptance test progress • Catalog and prioritize known limitations and process improvement opportunities identified during testing • Review test results with quality assurance advisors, summarize and communicate results to stakeholders • Obtain acceptance test approval from stakeholders

Table B-8 Key Design Task List *(6 of 7)*

Major Phases and Tasks	Subtasks
Conduct Integration and System Test	• Ensure establishment of integration test environment • Identify test team and assign test scenario ownership • Train team on integration test procedures, roles, and responsibilities • Review and revise integration test execution schedule, as required • Execute integration test • Identify and document integration test discrepancies • Resolve integration test discrepancies and document • Identify required modifications (such as configuration enhancements, interfaces, reports) • Re-execute integration tests • Update as required • Track test progress • Obtain test approval • Summarize and communicate results to stakeholders
4. Deployment Production	
Confirm Approach	• Review with stakeholders and establish implementation locations and configurations • Develop implementation approach • Repeat appropriate tasks from development hardware and software installation
Review and Update Deployment	• Review existing documentation of results of tests • Validate scope, objectives, and critical success factors • Update deployment approach • Review and approve deployment
Implement Deployment	• Review and reconcile system operations • Review organization and system procedures • Promote to production • Update current operations • Revise system release and deployment materials • Provide transition support

Table B-8 Key Design Task List *(7 of 7)*

Major Phases and Tasks	Subtasks
Training	• Confirm organization commitment and expectations • Establish training requirements for all personnel • Establish training schedules • Establish training staff • Prepare materials for training • Train administrators • Train maintenance providers • Capture training feedback • Incorporate feedback for training improvement
Document Portal	• Create “run book” for system administrators

Appendix C

Portal Server and Application Servers

This appendix provides an overview of the Sun™ ONE Portal Server 6.2 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™ ONE Portal Server product provides support for the following application servers to be used as the web application container, in addition to the Sun™ ONE Web Server 6.1 software:

- Sun™ ONE Application Server 7.0, Update 1
- BEA WebLogic Server™ Server 6.1 (SP5)
- IBM WebSphere® Application Server 4.0.5

NOTE Portal Server runs in the context of a web application container, which can be either a web server (Sun™ ONE 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™ ONE Application Server)

Portal Server on an Application Server Cluster

This section describes how Sun™ ONE 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 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 ONE Application Server

Sun ONE 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 ONE 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 *Sun ONE Portal Server 6.2 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.

Index