Sun Java System Reference Configuration Series: Portal Service on Application Server Cluster

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

Part No: 820–2195
May, 2008

080605@20209

Contents

Figures

Tables

Preface

This guide describes a portal service that is built from Sun Java™ Enterprise System (Java ES) components.

The guide covers the development and deployment life cycle of the Portal Service on Application Server Cluster reference configuration, including the following topics:

- the portal service business and technical requirements that are the basis for the reference configuration design
- the solution architecture
- step-by-step instructions for implementing the architecture

The purpose of the reference configuration, and this reference configuration guide, is to make it easy for Sun customers to implement a well-designed, tested solution for their portal services.

This preface contains the following sections:

Note – this reference architecture supports Solaris™ systems that use the SPARC® and x86 families of processor architectures: UltraSPARC®, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`. This document cites any implementation differences between the platform types.

Who Should Use This Reference Configuration Guide

This guide is intended for system architects who are developing portal services for their organizations. It is also intended for system administrators who will install and configure the portal service architecture that is described in this guide.

This guide assumes familiarity with the following:

- The UNIX® operating system
- Internet protocol (IP) computer networks
- Enterprise-level software products
- Java ES and its components

Before You Read This Reference Configuration Guide

Familiarize yourself with the basics of Java ES before reading this guide. See "Java ES Documentation" on page 15 for more information about documentation resources.

How This Reference Configuration Guide is Organized

This reference configuration guide provides information about Sun's development of the reference configuration architecture and step-by-step instructions for implementing the reference configuration architecture on your network. This reference configuration guide is organized into the following chapters:

- Chapter 1, "Introduction to the Portal Service Reference Configuration," describes the overall goals of the reference configuration and the business and technical requirements for portal services for medium-sized organizations.
- Chapter 2, "Reference Configuration Architecture," describes the architecture that is needed to meet the requirements. Use this chapter to understand the architecture before you attempt to implement it.
- Chapter 3, "Deployment Specifications," describes how to prepare the detailed technical specifications that you will need to deploy the reference configuration architecture in your environment. The chapter includes a set of sample specifications that you can adapt for your own use.
- Chapter 4, "Implementation Module 1: Directory Server With Multimaster Replication," contains detailed procedures for installing and configuring the directory service module in your environment.
- Chapter 5, "Implementation Module 2: Access Manager With Session Failover on Application Server," contains detailed procedures for installing and configuring the Access Manager service module in your environment.

- Chapter 6, "Implementation Module 3: Portal Server With Portlet Session Failover on Application Server Cluster," contains detailed procedures for installing and configuring the portal service module in your environment.
- Chapter 7, "Implementation Module 4: Secure Remote Access Gateway," contains detailed procedures for installing and configuring the Secure Remote Access (SRA) Gateway module in your environment.

Java ES Documentation

For more information about the Java ES components that are used in the reference configuration, see the resources in the Java ES documentation set. The documentation set provides extensive information about Java ES, its components, and its implementation. For the latest documentation, go to `http://docs.sun.com/prod/entsys.5`. A wealth of Java ES resources is also available at `http://www.sun.com/bigadmin/hubs/javaes/overview`.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click **Send comments**. In the online form, provide the full document title and part number. The part number is the 7- or 9-digit number on the book's title page or in the document's URL. For example, the part number for this document is 819-2095.

◆ ◆ ◆ CHAPTER 1

Introduction to the Portal Service Reference Configuration

To meet the needs of portal service customers, Sun is developing a series of portal service reference configurations that use Sun Java™ Enterprise System (Java ES) components. Each reference configuration is a tested, documented, and performance-tuned portal service solution that consists of a specific deployment architecture that uses a specific network topology and a set of recommended hardware.

This guide documents the Portal Service on Application Server Cluster reference configuration on the Solaris™ Operating System (Solaris OS). This reference configuration is designed to meet specific business and technical requirements, and covers the following aspects of the software solution:

- Identification of the business and technical requirements
- Design of the logical and deployment architectures
- Implementation of the deployment architecture

This chapter introduces the Portal Service on Application Server Cluster reference configuration and includes the following sections:

Objectives of the Reference Configuration

The purpose of a reference configurations is to enable customers to deploy proven technology solutions with a minimum of design time and implementation difficulty. A reference configuration starts with a business problem and finishes with a deployed solution that meets specified business and technical requirements.

The Portal Service on Application Server Cluster reference configuration responds to the business needs of medium-sized organizations for portal services. The reference configuration provides a basic portal service architecture that organizations can customize with their own

portal content and integrate with their existing applications and content management systems. This reference configuration guide explains the basic solution architecture and describes how to implement it. Once you have successfully installed, configured, and verified a reference configuration, you can customize it to meet your specific needs.

How to Use This Guide

This guide is the first of a number of portal service reference configuration guides. Each guide documents a different reference configuration architecture. The following table summarizes the steps you take, using this guide, to adopt and implement the Portal Service on Application Server Cluster reference configuration.

TABLE 1–1 Development and Deployment Cycle of the Reference Configuration

Step	Covered In
1. Determine if the reference configuration (or one of its deployment options) meets your organization's needs.	"Business and Technical Requirements" on page 19 discusses the requirements that the Portal Service on Application Server Cluster reference configuration is designed to meet."Deployment Options for the Reference Configuration" on page 42 presents deployment options that might better meet your needs.
2. Understand the reference configuration deployment architecture.	Chapter 2, The Reference Configuration Deployment Architecture describes the Portal Service on Application Server Cluster architecture.
3. Modify the detailed deployment specifications. Deployment specifications contain the low-level, detailed information needed to deploy the reference configuration in your environment. You need to modify the deployment specifications that appear in this guide by substituting the values you will be using in your own environment.	Chapter 3, "Deployment Specifications," describes how to develop the actual values that you use in the implementation procedures.
4. Implement the deployment architecture.	Four chapters in this guide describe the installation and configuration procedures for the four implementation modules of the Portal Service on Application Server Cluster architecture.

The intent of this guide is to describe the portal service on Application Server Cluster reference configuration in such a way that you can re-create the architecture in your own environment. With very little adaptation, the information presented in this guide can be used to help you deploy your own portal service solution.

Business and Technical Requirements

This section describes the business and technical requirements that are the basis for the Portal Service on Application Server Cluster reference configuration. You can use the information in this chapter in the following ways:

- Compare the requirements in this chapter to your organization's requirements for a portal service to determine whether the reference configuration (or one of its options) will meet your organization's needs.
- Use the requirements in this chapter to help you develop your own organization's requirements for a portal service.

This section divides the business and technical requirements into two categories:

- "Basic Functional Requirements" on page 19
- "Quality-of-Service Requirements" on page 20

Basic Functional Requirements

This section describes the basic features that the reference configuration provides. These features will meet the portal service needs of most medium-sized organizations. The basic feature requirements are the following:

- The portal service must serve content to users over HTTP, if accessed from a trusted network, or over HTTPS, if accessed from an unsecured network, such as the public Internet.
- The portal service must support the ability to add content to the portal and to customize the appearance of the content.

Note – The reference configuration described in this guide provides a portal service that displays the default desktop sample. You can use the default desktop to test your basic platform configuration. After verifying that the basic portal platform is working, you can customize content and develop a customized portal desktop.

- The portal service must be able to authenticate users who are authorized to access the portal and prevent unauthorized access.
- The portal service must be able to provide different kinds of content to different users, depending on each user's privileges.
- The portal service must be able to integrate content from existing applications that the organization is running, such as messaging, calendar, and content management systems.
- The portal service must support single sign-on that works with the applications that the portal owner integrates into the portal service.

- The portal service must let users customize the appearance and content of their portal desktops within the limits that are imposed by each user's privileges.

Quality-of-Service Requirements

In designing a successful software solution, you must establish the relevant quality-of-service requirements for your business needs. Five important service qualities are used to specify such requirements, as summarized in the following table.

TABLE 1–2 Reference Configuration Service Qualities

Service Quality	Description
Performance	A measure of response time and latency with respect to user load conditions.
Availability	A measure of how often a system's resources and services are accessible to end users (the *uptime* of a system).
Security	A complex combination of factors that describe the integrity of a system and its users. Security includes the physical security of computer systems, network security, application and data security (authentication and authorization of users), as well as the secure transport of information.
Scalability	The ability to add capacity to a deployed system over time. Scalability typically involves adding resources to the system but should not require changes to the deployment architecture.
Serviceability	The ease by which a deployed system can be maintained, including monitoring the system, repairing problems that arise, and upgrading hardware and software components.

The requirements regarding these service qualities have a big impact on how application and infrastructure components are deployed in a physical environment.

The following sections describe the quality-of-service requirements upon which the Portal Service on Application Server Cluster reference configuration is based:

- "Performance Requirements" on page 21
- "Availability Requirements" on page 21
- "Security Requirements" on page 22
- "Scalability Requirements" on page 23
- "Serviceability Requirements" on page 23

Performance Requirements

A portal service is an end-user service, and a fairly high level of performance (an acceptably short response time) is expected. The performance of Sun Java System Portal Server is generally measured by the response time of the standard channels that are available in the default desktop sample. The reference configuration is designed to provide a response time under two seconds for these channels at peak load levels. In a typical deployment, however, response time is dependent upon not only the portal service, but also the back-end applications that it aggregates.

Availability Requirements

Availability is a crucial requirement for a portal service. In many organizations, the portal is the employee's (or the customer's) gateway to critical information that is aggregated and displayed by the portal service. If the portal services fails, the employee or customer has no other way to access the information needed to conduct business.

Portal services can be classified according to the following levels of availability requirements:

- *Low availability*. This level has no real availability requirements. If the system goes down, it is acceptable to take days to repair it. This level of availability is suitable for software development, unit testing, or demonstration systems.
- *Service Availability*. With this level, the portal service must always be accessible to users, where failures will affect the work of employees or customers.
- *Session State Availability*. In addition to service availability, this level requires that session state is not lost when a user is redirected to another service instance (service failover) when failure occurs. The following are two types of session state availability requirements for portal services:
  - *User Session State Availability*. A user is not required to log in again when service failover occurs. In other words, that user session state is preserved in case of a service failure.
  - *Application Session State Availability*. A user is not required to restart a business operation when service failover occurs. In other words, the session state of applications that are providing the service is preserved in case of a service failure, and the user will not notice the failure.

The Portal Service on Application Server Cluster reference configuration is designed to provide service availability with both user session state and application session state availability. However, if application session state availability and/or user session state availability are not requirements of your organization, you can choose deployment architecture options that do not include them.

The reference configuration is not designed to sustain the complete failure of a data center. To overcome such failures, the portal service needs to be distributed across multiple locations. This kind of implementation is out of scope for the reference configuration.

The availability of a system should be measured from the user's perspective. Users care about how often a system fails and how long it takes to recover. There is no difference between a system being unavailable due to a systems failure or because of a scheduled maintenance window. Consequently, when measuring availability and when designing a highly available system, both planned and unplanned downtime needs to be considered. The reference configuration is designed to have no single points of failure. If implemented in conjunction with appropriate operational procedures and staffing, the reference configuration should result in less than one hour of unplanned downtime per year (99.99 percent availability).

Security Requirements

Portal services deliver varied content to varied users, often over the public Internet. In many cases, the content is confidential and should only be viewed by authorized users. Hence, the following security features are included in the basic feature requirements for portal services:

- Authentication of all users, including remote and wireless client access and mobile access
- Role-based access control

In addition, a more general set of security requirements is needed to provide secure access to confidential data. These requirements are shown in the following table.

TABLE 1–3 Security Requirements for Portal Services

Security Category	Requirement
Physical	■ Housed within a secure data center to which only authorized personnel have access
Network	■ Internet firewall protection ■ Subnet design that secures vital services ■ Secure transfer and storage of data
Privacy	■ All data stored in a manner that follows applicable regulations, corporate security policies, and corporate privacy policies
Transport	■ Authentication must be secure ■ Compatible with Secure Socket Layer (SSL)-enabled browsers and Transport Layer Security (TLS) ■ Strong encryption

The Portal Service on Application Server Cluster reference configuration is designed to support these security requirements.

Scalability Requirements

Most organizations anticipate growth in their user populations and want to know that their portal service can grow along with the size of their user populations.

As a result, no computer system should be more than 80 percent utilized under daily peak load. Also the deployed system should accommodate long-term growth of 10 percent per year.

While there are upper limits to how much any system can scale, due to the increased interactions among its components and the limitations of the network infrastructure, the Portal Service on Application Server Cluster reference configuration is designed to be easily scalable up to these limits.

Serviceability Requirements

Because a portal service is normally critical to conducting business, it must be maintained with minimal disruption and downtime.

Common servicing operations include database backups, replacement of applications and system software, upgrades, and other maintenance. Analyzing the solution's requirements for such servicing, and techniques to facilitate the servicing, should be a priority when you design the system.

For example, on an intranet-oriented portal, users are generally most active during the 8:00 a.m. to 5:00 p.m. working hours. This means all system servicing operations can be done after hours. However, if your organization is geographically distributed over multiple time zones, or users need 24–by-7 access to the system, there is no servicing window. Instead, the system needs to be designed so that all maintenance operations can be done with the system in operation or having little impact on the system's availability. In addition to an appropriate deployment architecture, it is necessary to have well-defined and tested operational procedures that ensure minimum downtime.

The Portal Service on Application Server Cluster reference configuration is designed to maximize serviceability, both with respect to scaling the portal service and upgrading software components in the configuration.

◆ ◆ ◆ CHAPTER 2

Reference Configuration Architecture

This chapter describes the design of the Portal Service on Application Server Cluster reference configuration, based on the functional and quality-of-service requirements that are specified in "Business and Technical Requirements" on page 19.

Read this chapter to understand the design rationale of the reference configuration before attempting to implement the deployment architecture in your own hardware environment.

The design of the reference configuration consists of a two-step process, first developing the logical architecture and then developing the deployment architecture, as described in the following sections:

Note – This reference configuration architecture uses a web container provided by Sun Java™ System Application Server. While the architecture would not change substantially if Sun Java System Web Server were used to provide the web container, the implementation procedures would be substantially different.

Logical Architecture of the Reference Configuration

A *logical architecture* shows the software components (and the interactions between them) that are needed to provide a specific set of services to end users.

An analysis of the reference configuration's functional requirements and quality-of-service requirements (which specify the required performance, availability, scalability, security, and serviceability) is the basis for determining the main Java ES software components that are needed to meet these requirements. In most cases, these components interact with or are dependent upon other, secondary software components. For information about Java ES

components, the services they provide, and interdependencies between those components, see the *Sun Java Enterprise System 5 Update 1 Technical Overview*.

The following sections describe the Java ES components that are used in the portal service reference configuration, their roles within the reference configuration, and the interactions between them:

Logical Architecture Diagram

The various components that are needed to meet the reference configuration requirements depend on their functions as distributed infrastructure services or their roles within a tiered application framework. In other words, the various components represent two views or dimensions that define a logical architecture: the logical tier dimension and the distributed infrastructure services dimension. These dimensions are described in the *Sun Java Enterprise System 5 Update 1 Technical Overview*.

The positioning of reference configuration components in such a two-dimensional framework is shown in the following logical architecture diagram. Components are placed within a horizontal dimension that represents standard logical tiers and within a vertical dimension that represents infrastructure service dependency levels. The positioning of a component in this matrix helps describe the role that the component plays in the logical architecture.

For example, Access Manager is a component that is used by presentation and business service tier components to provide security and policy infrastructure services. However, Application Server is a component that is used by presentation and business service tier components to provide distributed runtime services.

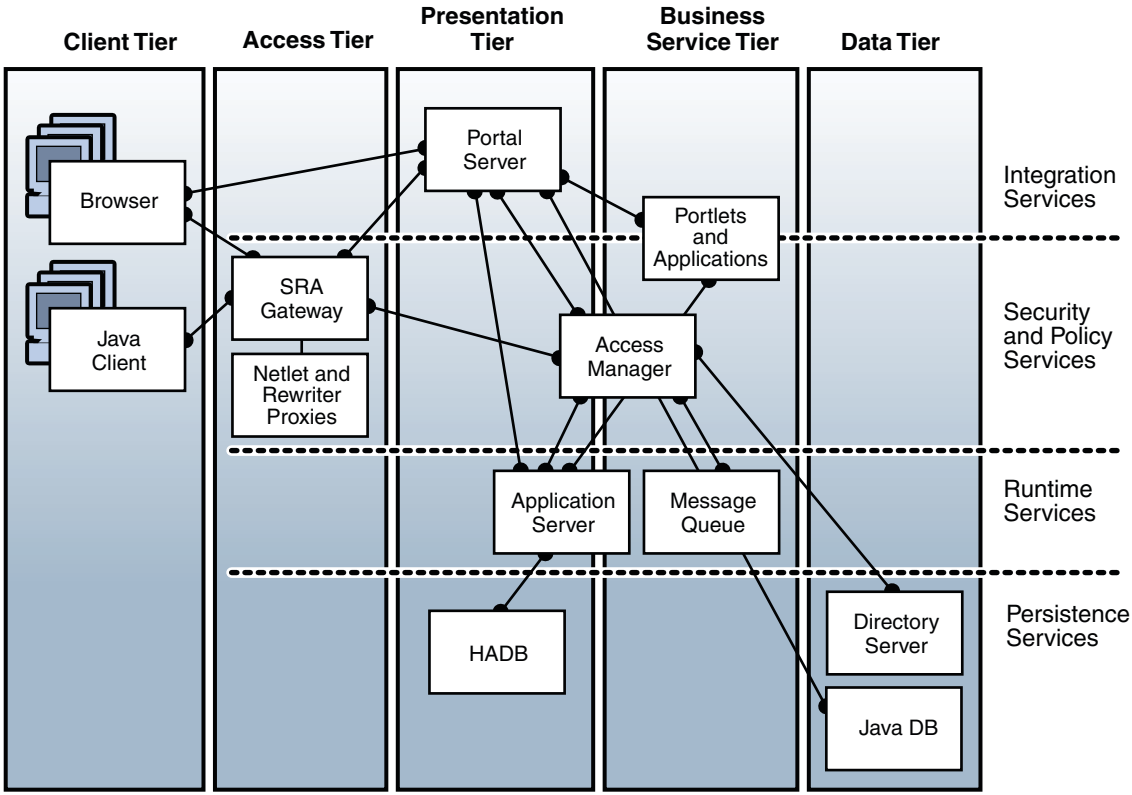

FIGURE 2–1 Logical Architecture of the Reference Configuration

A description of the tiers shown in Figure 2–1 is provided in the following table.

TABLE 2–1 Logical Tiers in the Architecture Diagram

Tier	Description
Client	In the Client tier are applications that are used by users to access portal services. In this reference configuration, the only client applications that are used are a browser and a stand-alone Java client.
Access	This tier enables remote users to securely access their organization's network and its services over the Internet. The Access tier acts as a communication relay between the Client tier and the Presentation tier, and includes the Portal Server Secure Remote Access components needed to securely access portal services from the Internet.
Presentation	This tier provides aggregation and presentation capabilities that enable users to access relevant information and personalize their desktop to best meet their needs. In addition, this tier provides community, collaboration, content, and knowledge management capabilities. This tier is implemented using Portal Server software.

TABLE 2–1 Logical Tiers in the Architecture Diagram *(Continued)*

Tier	Description
Business Service	This tier contains the back-end services that are aggregated and presented to users by services in the Presentation tier. Examples of applications that might reside in this tier include: email systems, calendar servers, and Enterprise Resource Planning (ERP) applications (SAP, PeopleSoft, Siebel, and so forth.). Also, this tier contains portlets and application components that are deployed in a web container or application server.
Data	This tier provides a permanent repository that business services can use to store persistent information. This tier includes Directory Server (used by Access Manager and Portal Server to store user profiles) and Java DB (used to store application data). High Availability Session Store (HADB), which is used to store portlet session state, is placed in the Presentation tier to indicate its functional relationship to Portal Server.

Software Components in the Logical Architecture

While Figure 2–1 is indicative of the role of the different components within the reference configuration's logical architecture, the following table describes more precisely the purpose of each component.

TABLE 2–2 Software Components in the Logical Architecture

Component	Component's Role in the Architecture
Web Browser client	While not formally a component of the reference configuration, the browser client is included in the architecture diagram to show how users will access portal services. There are two access scenarios: ■ Access from a trusted network: browser clients (for example, an organization's employees) connect to portal services over the local network (or intranet) or from the Internet by using a virtual private network (VPN) or a similar solution. ■ Access from an unsecured network: Web browser clients (of a business-to-business or business-to-consumer portal) connect to portal services over the public Internet. This access scenario is supported by the Secure Remote Access (SRA) Gateway.

TABLE 2–2 Software Components in the Logical Architecture *(Continued)*

Component	Component's Role in the Architecture
Remote client (optional)	In addition to browsers, users can use applets that are included with Portal Server SRA software: ■ *Netlet*. The Netlet applet runs on the browser and sets up an encrypted TCP/IP tunnel between the remote client and intranet applications in the Business Service tier. Netlet listens to and accepts connections on preconfigured ports, and routes both incoming and outgoing traffic between the client and the destination server. In this way, Netlet enables client applications to securely access intranet business service components. ■ *Netfile*. NetFile is a file manager application that allows remote access and operation of file systems. ■ *Proxylet*. Proxylet is a dynamic proxy server that runs on the browser and redirects a URL to the SRA Gateway. It does so by reading and modifying the proxy settings of the browser on the client so that the settings point to the local proxy server or Proxylet. Proxylet is used to reduce the number of ports that must be opened in a firewall through which the SRA Gateway(see next item) connects to Internet hosts. It is also used to minimize or eliminate the dependency on the Rewriter Proxy (see next item) and Rewriter rulesets.
Sun Java System Portal Server Secure Remote Access (Portal Server SRA)	Portal Server SRA provides a gateway service that allows secure connections over the public Internet to applications and content on an internal intranet, but only to authorized users. In addition to the SRA Gateway, SRA includes the following two optional components, depending on your requirements: ■ *Netlet Proxy*. The Netlet proxy is an stand-alone Java process that enhances the security between the SRA Gateway and the intranet by extending the secure tunnel from the client through the Gateway to the Netlet proxy that resides in the intranet. Netlet packets are decrypted by the proxy and then sent to their destinations. This mechanism helps to reduce the number of ports that must be opened in a firewall. ■ *Rewriter Proxy*. The Rewriter proxy is a stand-alone Java process that is installed on the intranet. The SRA Gateway forwards all requests to the Rewriter proxy, which fetches and returns the content of the request to the Gateway. This mechanism helps to reduce the number of ports that must be opened in a firewall.
Sun Java System Portal Server (Portal Server)	Portal Server provides key portal services, such as content aggregation and personalization, to browser-based clients that are accessing business applications or services in the Business Service tier.

TABLE 2–2 Software Components in the Logical Architecture *(Continued)*

Component	Component's Role in the Architecture
Sun Java System Access Manager (Access Manager)	Access Manager provides access management services such as authentication and role-based authorization for user access to applications and services. In cases where Access Manager is remote from a local component, Access Manager SDK provides an interface to the remote Access Manager services.
Sun Java System Application Server (Application Server)	Application Server provides the Java Platform, Enterprise Edition (Java EE) web container that is needed to support web components, such as Portal Server, Access Manager, portlet applications, and so forth. While a web container can also be provided by Sun Java System Web Server, the Portal Service on Application Server Cluster reference configuration uses Application Server.
Applications	Various kinds of applications provide the content for Portal Server channels that are accessed by end users. These applications can include email systems, calendar servers, ERP applications, custom or third-party portlet applications deployed on a web container, and so forth.
Sun Java System Directory Server (Directory Server)	Directory Server provides an LDAP repository for storing information about portal users, such as identity profiles, user credentials, access privileges, application resource information, and so forth. This information is used by Access Manager for authentication and authorization and by Portal Server to build users' portal desktops.
Sun Java System Message Queue (Message Queue)	Message Queue is a reliable asynchronous messaging service that is used by Access Manager to write user session state into a replicated session database and to retrieve such state information when necessary.
High Availability Session Store (HADB)	HADB provides a data store that makes application data, especially session state data, available even in the case of failure.
Java DB	Java DB is the default relational database used by Portal Server to support community features and selected portal applications.

Interactions Between Reference Configuration Components

To design a logical architecture, you must understand the software dependencies and interactions between the various components that are listed in Table 2–2. These interactions can be somewhat complicated and difficult to illustrate in a single diagram such as Figure 2–1. The main interactions between components in the reference configuration are therefore described briefly in the table below, in the context of typical portal service operations.

Two access scenarios are incorporated into the following table:

- Direct access of portal services from a trusted network
- Indirect access of portal services from an unsecured network (such as the public Internet) by way of SRA Gateway

TABLE 2–3 Interactions Between Reference Configuration Components

Step	What Happens
1	The user starts a browser and opens the portal service or SRA Gateway service URL, depending on the access scenario being used.
2	If portal services are accessed directly, Portal Server returns the anonymous desktop, which includes the login channel. If portal services are accessed through the SRA Gateway, the Gateway redirects the user request to Access Manager. Access Manager returns the login page (by way of the Gateway).
3	The user logs in by typing a user ID and password in the appropriate form and clicking Login.
4	Access Manager interacts with Directory Server to retrieve the user's profile, which contains authentication, authorization, and application-specific information.
5	Access Manager authenticates the user's ID and password against the LDAP directory information and creates a session object.
6	When the user has been authenticated, Access Manager returns a session cookie to the user's browser and redirects the browser to Portal Server. Portal Server uses the session cookie to interact with Access Manager to access information in the user's profile (cached by Access Manager). Portal Server uses the information to build the user's personalized portal desktop. Portal Server returns the desktop to the user's browser (by way of the Gateway).
7	The user reviews his or her portal desktop, and clicks a portal channel.
8	Portal Server interacts with Access Manager to validate the status of the user session. Access Manager authorizes the channel content that is being requested by the user.
9	When appropriate, Portal Server creates a portlet session and returns channel content to the user's browser.
10	The user logs out or the session times out.
11	Portal Server closes the portlet session, if any, and Access Manager deletes the user's session object.

The understanding of component interactions represented in the logical architecture can be used later in the design process when you estimate the load on different components for sizing purposes and when you create a network connectivity specification.

Deployment Architecture of the Reference Configuration

A *deployment architecture*is a mapping of software components to a hardware environment. More specifically, for the Portal Service on Application Server Cluster reference configuration, it represents how to map the components in the logical architecture to networked computers in a way that achieves the specified quality-of-service requirements.

The logical architecture in Figure 2–1 identifies the components that are needed to meet the functional requirements of the reference configuration. The deployment architecture, however, shows how to do so with the specified quality of service.

The following sections present the deployment architecture diagram and discuss how the deployment architecture addresses the quality-of-service requirements:

Summary of Quality-of-Service Requirements

The quality-of-service requirements for the portal service on Application Server Cluster reference configuration are summarized in the following table.

TABLE 2–4 Quality-of-Service Requirements for the Reference Configuration

Service Quality	Requirements
Performance	Response time under two seconds for default portal channels at peak load levels. See "Performance Requirements" on page 21.
Availability	Service availability with both user session availability and application session state availability. No single points of failure. See "Availability Requirements" on page 21.

TABLE 2–4 Quality-of-Service Requirements for the Reference Configuration *(Continued)*

Service Quality	Requirements
Security	Protected services in separate network subnets.
	Firewall protection for Internet access and for portal service subnet zone.
	Encrypted Internet transport over SSL.
	See "Security Requirements" on page 22.
Scalability	Easily scalable so that no computer system is more than 80% utilized under daily peak load. Also the deployed system should accommodate long-term growth of 10% per year.
	See "Scalability Requirements" on page 23.
Serviceability	Minimize planned downtime needed to scale the portal service or to upgrade software components in the configuration.
	See "Serviceability Requirements" on page 23.

Deployment Architecture Diagram

Figure 2–2 is a graphical representation of the deployment architecture for the Portal Service on Application Server Cluster reference configuration. It shows the following features of the deployment architecture:

- The computers that are used to support the reference configuration and the components that are installed on each computer
- The redundancy strategies that are used to achieve scalability and availability
- The grouping of computers, components, and load balancers into service modules

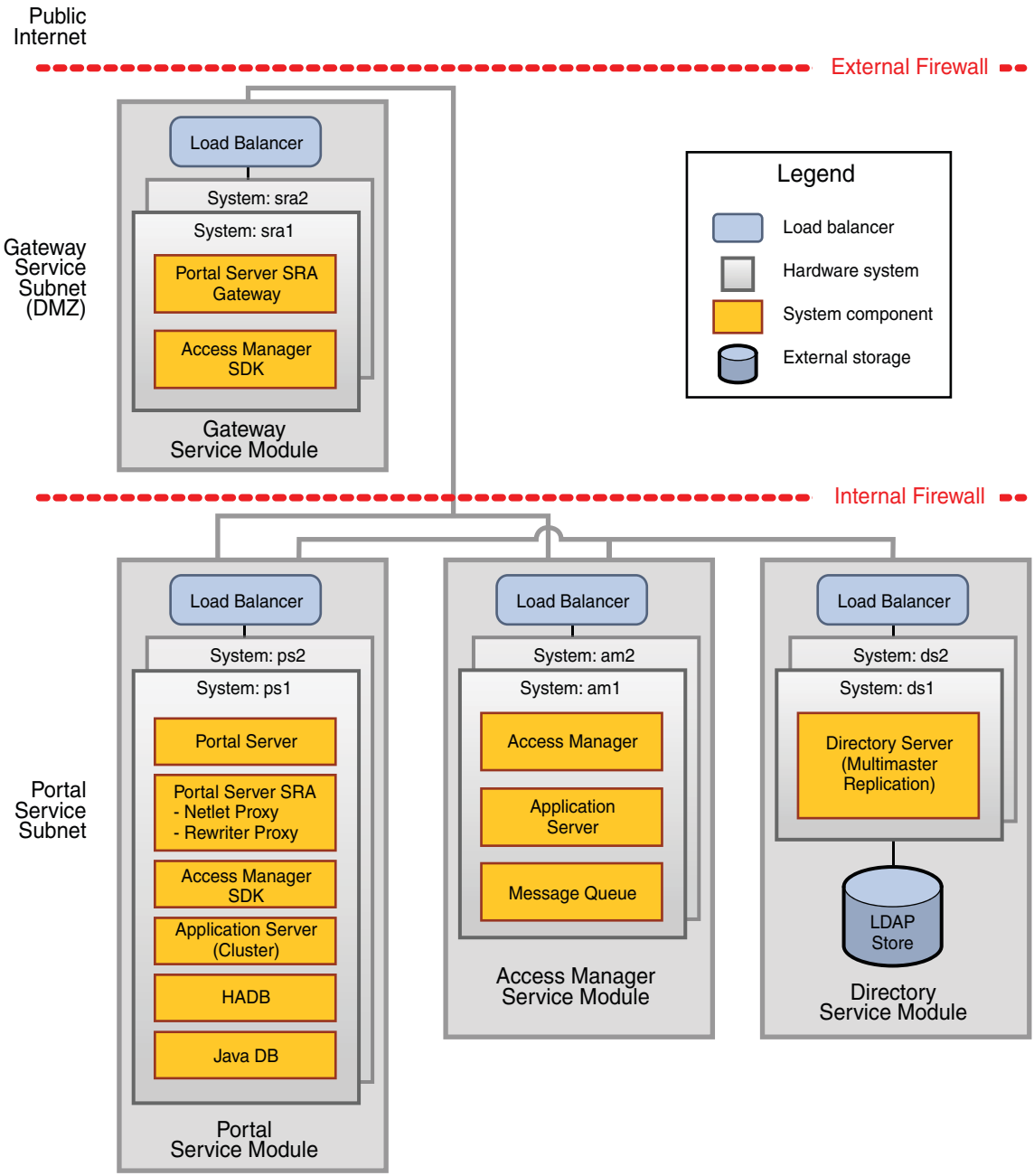

FIGURE 2–2 Deployment Architecture of the Reference Configuration

Modularity in the Deployment Architecture

The reference configuration deployment architecture is based on a Service Delivery Network Architecture (SDNA) approach, in which individual services within a solution are modularized (see https://wikis.sun.com/display/BluePrints/The+Service+Delivery+Network-+A+Case+Study). The result is a deployment architecture consisting of four independent service modules: SRA Gateway, portal, Access Manager, and directory.

In accordance with SDNA principles, each service module in the reference configuration independently implements its own level of availability, security, scalability, and serviceability. The overall solution can therefore be easily deployed, secured, maintained, and upgraded. An explanation of how the reference configuration's modular architecture facilitates quality-of-service objectives is provided in subsequent sections of this chapter.

The service modules that make up the reference configuration, shown in Figure 2–2, have the following common SDNA characteristics:

- Each module consists of two or more instances of the service configured to meet quality-of-service requirements. Each module includes the components that are needed to provide a single service to the overall reference configuration.

  For example, the two Directory Server instances in Figure 2–2 can be considered a unit that provides directory services for the other components in the deployment.
- Each service module is accessed through a load balancer. The load balancer is configured to establish a virtual IP address, or virtual service address, for the module. Other components in the reference configuration are configured to send requests to the virtual service address rather than to the individual component instances.

  For example, when Access Manager needs information from the directory service, it addresses its request to the virtual directory service address that is provided by the load balancer, rather than to a specific instance of Directory Server.

  Each load balancer is responsible for routing incoming requests to a specific service instance, and, where desired, to route successive requests from the same user to the same service instance.
- The component instances in a module can be reconfigured without changing the virtual service address of the module, making changes within the module transparent to other components. Depending on the kinds of usage patterns you experience, you can independently assign system resources and scale each module to meet load requirements, using scaling techniques that are best suited to the module.

While the modular architecture depicted in Figure 2–2 has many advantages, as described in subsequent sections of this chapter, alternative approaches in common practice do exist. The drawbacks of two such alternatives, which are not supported by this reference configuration, are discussed below.

Not Supported: Portal Server and Access Manager Combined

In some situations, the modular architecture of Figure 2–2 might result in lower resource utilization than could be achieved by combining components on the *same* computer and running them in the same web container. In fact, many deployment architects have traditionally deployed Portal Server and Access Manager in the same web container in an effort to maximize resource utilization and reduce network traffic in updating Access Manager session information. However, such designs cannot realize the availability, security, scalability, and serviceability benefits of SDNA modularity, which generally outweigh the drawbacks.

Not Supported: Access Manager Internal Configuration for Multiple Directory Server Instances

Access Manager supports, by way of post-installation configuration, multiple LDAP directories for each Access Manger service. In this way, Access Manager can detect failure of a primary Directory Server instance and fail over to an standby instance. This built-in mechanism has several drawbacks:

- Configuration of these multiple Directory Server instances needs to be done for each of the Access Manger services: user profiles, policies, LDAP authentication, Membership authentication, and so forth.
- Access Manager does not load balance directory requests: only the primary DS instance is used, while the other(s) are inactive.
- Upon a failure of a primary instance, Access Manager switches over to the standby instance, but if the primary instance comes back online, there is no mechanism to revert back to the original configuration.

By contrast, the modular architecture of Fig 2-2 has the following advantages:

- The only required Access Manager configuration is the load balancer's virtual service address, specified at installation time.
- The directory services load balancer In the reference configuration routes requests to all Directory Server instances, monitors the health of these instances, automatically performs the failover and restoration of a failed instance.
- The modular architecture allows you to configure, manage, scale and monitor the Directory Server instances independent of the Access Manager instances.

Note – In the multimaster replication approach of Figure 2-2, write operations are synchromized between directory instances. In environments with many write operations, the overhead of the multimaster replication process can slow down Directory Server processing of client requests. In these situations, the best approach is to direct all write operations to a single master by placing a Directory Proxy Server instance in front of each Directory Server instance. Such situations are not common in portal service deployments, so the reference configuration does not include Directory Proxy Server.

Availability in the Deployment Architecture

The deployment architecture that is represented in Figure 2–2 uses several strategies to meet the availability requirements of the reference configuration. Availability requirements fall into the two categories that are discussed in the following sections:

- "Service Availability" on page 37
- "Session State Availability" on page 38

Service Availability

Service availability means that a service is available, even when a service provider fails. Service availability is generally achieved using multiple identically configured service instances (redundancy). Redundancy eliminates single points of failure (assuming that simultaneous failure of all instances is extremely unlikely). If one instance providing a service fails, another instance is available to take over. This mechanism is known as *service failover*.

Service failover is supported in the reference configuration through two mechanisms:

- *Load balancing*. Load balancing uses redundant hardware and software components to distribute requests for a service among multiple component instances that provide the service. This redundancy provides greater capacity than would be possible with a single instance. This redundancy also means that if any one instance of a components fails, other instances are available to assume a heavier load. Depending on the latent capacity that is built into the deployment, a failure might not result in significant degradation of performance. Load balancing is used in all of the service modules in the reference configuration.
- *Directory Server multimaster replication*. The preferred solution for Directory Server, this mechanism provides data that is crucial to the operation of the entire deployment. Multimaster replication is specifically designed to synchronize data between the two (or more) Directory Server instances shown in the deployment architecture. Multimaster replication is the simplest directory service failover implementation and is suitable for all but the highest-end deployments that need to support millions of users.

Session State Availability

Session state availability means that data associated with a user session is not lost during a service failover. When a service failover occurs, the session state data that is stored by the failed instance is made available to the failover instance. This mechanism is known as *session failover*. The result is that the service failover is transparent to the user: the user will not be required to log in again or to restart a business operation.

Session failover is supported in the reference configuration through two mechanisms:

- *Access Manager session failover*. Access Manager session information is created when a user is authenticated and stored in a replicated database. This database is shared by Access Manager instances and accessed through Message Queue. If an Access Manager instance fails, the load balancer routes all user requests to a failover instance (service failover). The failover instance retrieves session information from the shared database and maintains the session.
- *Portlet session failover*. The JSR 168 portlet specification requires portlets to map state information to an HTTP session. If a web container supports highly available HTTP sessions, and if a Portal Server instance fails, the HTTP session state can be recovered by the failover instance. In the reference configuration, Portal Server is deployed in an Application Server cluster, in which High Availability Session Store (HADB) is used to store and replicate portlet session state. The failover instance retrieves session information from HADB and maintains the session.

  Portlet session failover requires availability of Access Manager session state. An Access Manager failure could therefore interfere with portlet session failover, unless Access Manager session failover is also implemented.

Note – When a user is successfully authenticated with Access Manager, the browser is redirected to a Portal Server instance. A portal desktop session is created on this instance and is mapped to the user's Access Manager session. This portal desktop session is used to track Portal Server specific information such as the user's merged display profile and provider properties. If a Portal Server instance fails, the desktop session is automatically re-created by using the user's display profile and attributes that are stored in the Access Manager's user session. However, provider properties that are stored in local memory are lost.

Security in the Deployment Architecture

The security requirements of the portal service reference configuration (see "Security Requirements" on page 22) are met through several mechanisms, each of which are discussed in the following sections:

■ "Secure Remote Access" on page 40

Authentication and Authorization

Each user's access must be limited to the portal services and data channels that he or she is authorized to view.

The reference configuration uses Access Manager and Directory Server to control user access to portal content. The directory service maintains each user's portal desktop profile. This profile includes any desktop customization that is performed by the user, as well as mechanisms for determining what content the user is authorized to view.

Separate Administration

The modularized architecture makes it easy for different organizations to administer different service modules so that each organization has the level of administrative security it needs. In most enterprises, for example, directory services and Access Manager services are administered by security-oriented organizations, while portal services are administered by end-user applications organizations.

Network Segmentation

The portal service must be secured against unauthorized and unauthenticated access.

The deployment architecture uses a secure network topology for the portal service, which includes the use of firewalls, controlled access through load balancers with virtual service addresses, and private subnets behind the firewall.

Figure 2–2 shows a portal services zone in which the portal service, Access Manager service, and directory service modules are deployed behind the Internal Firewall. Within this zone, the deployment architecture protects the service modules in the following ways:

- A load balancer provides a single point of contact for the portal service, even though the service consists of two Portal Server instances that are running on two computers. This means that there is only one opening in the firewall for the portal service, and all of the traffic for the portal service is routed through the load balancer. Note that employees connected to the main corporate network also access the portal through this load balancer.
- Local access to the portal service is only from trusted computers on the corporate network, by users who have authenticated themselves to the corporate network.
- Not shown in Figure 2–2, but implied in the deployment architecture, is a network topology that creates separate subnets for accessing each service module. The IP addresses that are used in the subnets are private IP addresses, making the subnets invisible to the outside world. These subnets are connected only through the load balancers, further impeding the ability of intruders to access the actual computers behind the public URL. For more information on the network topology, see "Network Connectivity Specification" on page 51.

Not shown in Figure 2–2 is that the individual computers hosting service instances are hardened and that the operating system installations are minimized. Minimizing the number of installed Solaris OS packages means fewer security holes. Because the majority of system penetrations are through exploitation of operating system vulnerabilities, minimizing the number of installed operating system packages will reduce the number of vulnerabilities. Minimizing the operating system is covered in detail in "Computer Hardware and Operating System Specification" on page 48.

Secure Remote Access

The secure remote access option provides secure access to portal services, applications, and other content on an internal intranet to employees or customers on the public Internet. This option prevents such access to unauthorized people.

The requirement for secure remote access is met in the Portal Service on Application Server Cluster reference configuration through Portal Server SRA components, specifically the SRA Gateway service, and by network access zones, demarcated by firewalls, that take maximum advantage of the SRA Gateway service. The access zones and the firewalls are represented in Figure 2–2.

The outermost zone in Figure 2–2 is the so-called *demilitarized zone*, or *DMZ*, which contains the SRA Gateway service. The Gateway service can only be accessed through the External Firewall at one specific URL. Employees or customers who connect to the portal service with remote browser clients or mobile clients do so by accessing the Gateway service at the specified URL. The External Firewall blocks all other ports and addresses.

Because remote access to the portal service from the public Internet is through the Gateway service, the portal service itself can reside behind an additional firewall (the Internal Firewall) and an additional layer of hardware load balancing.

In addition to deploying the Gateway service behind an Internet-facing firewall, the deployment architecture secures the Gateway service in the following ways:

- The Gateway service requires the authentication of all users. Users who access the URL for the Gateway service in their browsers are presented with a login page and must type a user ID and password to gain access to any content.
- The Gateway service instances are behind a hardware load balancer. The load balancer provides a single point of contact for the Gateway service, even though multiple component instances are running on multiple computers. As a result, only one port in the firewall is needed for the Gateway service, and all requests are routed through the load balancer.
- The communication between the browser and the Gateway service load balancer is encrypted through using the SSL protocol. This protocol is required because this traffic will circulate through an unsecured network (the Internet). The SSL protocol also requires the use of server certificates to ensure that service providers have not been tampered with. Optionally, client certificates can be used to better authenticate access to the Gateway service.

Scalability in the Deployment Architecture

The modular nature of the reference configuration's deployment architecture means that you can scale each module independently, depending on the kind of traffic that your portal service receives.

Each service module in the deployment architecture is composed of two or more service instances running on separate computers behind a load balancer. This architecture allows you to scale any of the modules vertically (by adding CPUs or memory to the host computers) or horizontally (by adding additional service instances). Some modules are better suited to vertical scaling, and some modules are better suited to horizontal scaling.

The recommended techniques for scaling each module in the reference configuration are as follows:

- Scaling the directory service module:

  Directory Server scales almost linearly up to 12 CPUs, so vertical scaling is an effective technique for this module.

  A limitation on the performance of Directory Server is the complexity of the LDAP directory tree. Access Manager creates access control instructions (ACIs) for each Access Manager organization. Creating multiple organizations increases the load on Directory Server, as it must process more requests from Access Manager. At some point (at about 1000 organizations), vertical scaling is no longer effective.

  In that case it becomes more effective to scale horizontally, keeping the multiple Directory Server instances synchronized by using the Directory Server's multimaster replication feature. Other approaches include trimming down the number of ACIs created for each organization and running Access Manager in realm mode instead of legacy mode. Having thousands of organizations is not a common requirement, so the reference configuration does not explore the architectural implications of large numbers of Access Manager organizations.

- Scaling the portal and Access Manager service modules:

  These modules can be scaled effectively by adding computers running additional component instances to the module. This approach is cost-effective and also helps maintain availability because spreading the load over additional computers ensures that only a relatively smaller amount of capacity will be lost if a single hardware system fails.

  Both Portal Server and Access Manager run in web containers. When they run in a 32–bit web container, as described in this reference configuration, the maximum process size is 4 Gbytes of memory, limiting the number of user session objects that can be stored. If increased memory is needed or increased throughput is desired, these modules should be scaled horizontally.

It might seem that to better utilize memory (the computers used in the reference configuration have 16 Gbytes of memory), it would be possible to run multiple instances on the same hardware. However, this kind of vertical scaling breaks the modularity of the architecture and does not substantially increase throughput (the number of pages that are rendered per second).

- Scaling the gateway service module:

  The Gateway service can be scaled effectively by adding computers running additional component instances to the module. This approach is cost effective and also helps maintain availability because spreading the load over additional computers ensures that only a relatively smaller amount of capacity will be lost if a single hardware system fails.

Serviceability in the Deployment Architecture

The reference configuration architecture builds the portal service out of several subservices, such as the Access Manager service and directory service. Because each subservice is implemented in a separate module, it is possible to maintain each module independently.

In addition, the reference configuration architecture creates each subservice as a virtual service, which means that interoperability among the subservices is not dependent on specific hardware connections, and the individual subservices are maintained, upgraded, replaced, and scaled without affecting each other. For example, if it is necessary to add an Access Manager instance to the architecture, the Portal Server instances that depend on Access Manager do not need to be modified or affected in any way.

Deployment Options for the Reference Configuration

Depending on your quality-of-service requirements, certain parts of the reference configuration can be changed or omitted. This section briefly discusses these options which include the following:

- "Omitting Portlet Session Failover" on page 42
- "Omitting Access Manager Session Failover" on page 43
- "Omitting Secure Remote Access " on page 43
- "Using a Different Web Container" on page 44

Omitting Portlet Session Failover

The reference configuration deployment architecture supports portlet session failover, as described in "Session State Availability" on page 38. It does this by deploying Portal Server in an Application Server cluster that uses High Availability Session Store (HADB) to store and replicate portlet session state.

If your business solution does not involve portlets that store session state, then portlet session failover might not be a requirement for your portal service deployment. If that is the case, you do not need to deploy Portal Server in an Application Server cluster. However, if you have other reasons beside portlet session failover for deploying Portal Server in an Application Server cluster, you can use this guide, but omit the section on implementing portlet session failover.

Portal Server can be deployed in a web container provided by nonclustered Application Server instances. This approach would substantially change the implementation of the portal service module described in Chapter 6, "Implementation Module 3: Portal Server With Portlet Session Failover on Application Server Cluster."

At the present time, however, an alternative implementation for Portal Server on Application Server (without portlet session failover) has not yet been documented.

Omitting Access Manager Session Failover

The reference configuration deployment architecture supports Access Manager session failover, as described in "Session State Availability" on page 38. It does so by configuring Access Manager to use Message Queue and a highly available database to store and replicate Access Manager session state.

If your business solution permits users to log in again to reestablish a session after a service failover, then Access Manager session failover is not a requirement for your portal service deployment. If that is the case, you do not need to configure Access Manager for Access Manager session failover, and Message Queue would not be included as a component in the Access Manager service module. This approach would change the implementation of the Access Manager service module by not requiring the procedures in "Implementing Session Failover for Access Manager " on page 104.

Omitting Secure Remote Access

The reference configuration deployment architecture supports secure access to portal services, applications, and other content on an internal intranet to users on the public Internet. This feature is supported by the SRA Gateway module, as described in "Secure Remote Access" on page 40.

If your business solution does not require secure access to portal services, applications, and other content over the public Internet, then secure remote access is not a security requirement for your portal service deployment. For example, you might be using one of the following alternate scenarios to access the portal service:

- An Internet-accessible portal service that communicates over SSL and is deployed in a DMZ
- A portal service that is located behind an organization's firewalls and accessed only locally or through VPN connections

- An internal portal service that is only accessed on a corporate network

In these scenarios, you can omit Chapter 7, "Implementation Module 4: Secure Remote Access Gateway," from the reference configuration architecture. However, depending on the scenario, you might need to modify the network topology of the reference configuration accordingly.

Using a Different Web Container

Two of the components in the reference configuration, Portal Server and Access Manager, run in web containers. The Java ES component set gives you the choice of using either Sun Java System Web Server or Sun Java System Application Server for a web container.

You need to consider both technical and non-technical factors when you choose a web container.

The following technical factors address the abilities of the different containers to run different types of portal content:

- Portlets and providers are Portal Server mechanisms for building presentation channels that can aggregate content from other applications. If your plans include developing portlets or providers that use Java EE APIs that are not supported by Web Server, such as the Enterprise JavaBeans (EJB) or Java Connector Architecture (JCA) interfaces, then you must use Application Server as your web container.
- Web Server 7.0 supports a lightweight mechanism for HTTP session failover. This mechanism can be eventually used to enable portlet session failover in the same way that HADB and Application Server clusters enable such failover. However, this new feature of Web Server and its impact on the reliability, security, and performance has not yet been fully analyzed.
- A reference configuration guide that documents a portal service deployment on Web Server does not yet exist.

If none of the technical factors are decisive for your organization, the following non-technical considerations could prove decisive:

- Does your organization have existing standards for a web container? If so, you are likely to use that web container to implement the portal service reference configuration.
- What does a price-to-performance comparison of the web containers reveal? Your organization might choose a web container based on the cost of the licenses that are needed to support the organization's user base. Your organization might have a volume discount agreement with a vendor that affects this decision.
- Your organization might have support agreements with a web container vendor.
- You might want to choose the same web container for all elements of your portal service even if you are not colocating Portal Server instances and portal channel applications. For example, if you have portal channels that are running in Application Server, you might want to deploy Portal Server in Application Server for the sake of consistency.

- If there is no compelling reason to use Application Server in your portal, Web Server can be easier to administer.

◆ ◆ ◆ CHAPTER 3

Deployment Specifications

The deployment architecture is a high-level design of the portal service reference configuration. Before you can actually deploy the reference configuration in your environment, you need to specify additional information required during the installation and configuration process. The deployment specifications are meant to help you gather and organize this additional information.

This chapter describes the deployment specifications that are needed for the portal service reference configuration. It consists of the following sections:

Software Component Specification

The reference configuration that is described in this book uses specific versions of the software components in the deployment architecture shown in Figure 2–2. In particular, the deployment architecture is implemented using Sun Java Enterprise System 5, Update 1, which includes the following versions of the components that are used in the reference configuration:

- Sun Java System Portal Sever 7.1 Update 2
- Sun Java System Access Manager 7.1
- Sun Java System Application Server Enterprise Edition 8.2 patch 2
- Sun Java System Message Queue 3.7 UR2
- Sun Java System High Availability Session Store (HADB) 4.4.3
- Java DB 10.2.2.1
- Sun Java System Directory Server Enterprise Edition 6.2

Computer Hardware and Operating System Specification

A computer hardware and operating system specification describes the hardware and operating system configuration for the computers in your deployment. You want to size your hardware to the level of performance you require.

Table 3–1 lists the computer hardware that has been chosen for the Portal Service on Application Server Cluster reference configuration. This specification is meant to satisfy the requirements in Chapter 1, "Performance Requirements" on page 21.

In general, a hardware specification is based upon a sizing analysis that takes into account the size of the user base, the resource needs of each component, and the relative number of interactions (or hits) that are made on each component (see "Interactions Between Reference Configuration Components" on page 30). For the reference configuration, however, the approach has been to select the *same* hardware for each computer in the deployment architecture, and then use performance tests to determine the utilization of each computer under load conditions.

Using this approach, the absolute and relative sizing of the different computers in the deployment architecture can be determined and documented. For this purpose, the Sun Fire™ T2000 server was selected as a basic, low-end, high-performance computer.

Note – The T2000 server has performance limitations for deployments in which write-intensive Directory Server operations are required. Write operations are serialized and the T2000 cannot perform them in parallel. As a result, CPU utilization can be lower then 50 percent. This reference configuration does not involve write-intensive operations. However, if your solution has such requirements, consider using computers with a faster clock rate than the T2000 for the directory service module.

If your performance requirements are significantly different than the requirements of the reference configuration, you can specify hardware with more or less CPUs, more or less memory, and so on.

TABLE 3–1 Computer Hardware and Operating System Specification

Computer(s)	Service Module	Components Installed	Hardware Model	Operating System
ds1, *ds2*	Directory Service	Directory Server	Sun Fire T2000 server, 8 core 1.2 GHz UltraSPARC® T1 processor, 16 Gbyte DDR2 memory	Solaris 10 8/07 OS with the Solaris Zones facility

TABLE 3–1 Computer Hardware and Operating System Specification *(Continued)*

Computer(s)	Service Module	Components Installed	Hardware Model	Operating System
am1, *am2*	Access Manager Service	Access Manager Message Queue Application Server	Sun Fire T2000 server, 8 core 1.2 GHz UltraSPARC T1 processor, 16 Gbyte DDR2 memory	Solaris 10 8/07 OS with the Solaris Zones facility
ps1, *ps2*	Portal Service	Portal Server Application Server Access Manager SDK Java DB HADB	Sun Fire T2000 server, 8 core 1.2 GHz UltraSPARC T1 processor, 16 Gbyte DDR2 memory	Solaris 10 8/07 OS with the Solaris Zones facility
sra1, *sra2*	SRA Gateway Service	Portal Server SRA Access Manager SDK	Sun Fire T2000 server, 8 core 1.2 GHz UltraSPARC T1 processor, 16 Gbyte DDR2 memory	Solaris 10 8/07 OS with the Solaris Zones facility

Solaris OS Minimization and Hardening

The Solaris OS version that is used to build the Portal Service on Application Server Cluster reference configuration is Solaris 10 8/07. However, the architecture and implementation is expected to be supported by later versions of the Solaris 10 operating system.

For maximum security of your portal service, use a minimized version of the Solaris 10 OS. Most implementations of the reference configuration portal service will be exposed to the Internet or some other public or untrusted network, which makes minimization especially important. If your portal service will be exposed to these conditions, you must reduce the Solaris OS installation to the minimum number of packages that are required to support the portal service components. This minimization of services, libraries, and component software increases security by reducing the number of subsystems that must be disabled, patched, and maintained.

Minimization increases the security of the computer systems, but it also limits the software that you can run on the computer systems. Therefore, you need to use the appropriate minimal configuration for your environment. Minimizing the operating system you use for a portal service involves the following:

- Minimizing the Solaris OS on the computers that will be running the portal service components.
- Hardening the Solaris OS on the computers that will be running the portal service components. Sun provides the Solaris Security Toolkit (`http://www.sun.com/software/security/jass/`), which hardens a system by changing the system parameters, disabling any unused services, and providing a quick way to audit the system. The Toolkit is based on the field experience of security experts.

The operating systems that were used in testing the reference configuration described in this guide were installed with the minimal number of Solaris packages required to run the Java Enterprise System components, as described in the "Platform Requirements and Issues" in *Sun Java Enterprise System 5 Release Notes for UNIX*. Most of the required packages are included in the "Core System Solaris Software Group (SUNWCreq)." The additional packages needed are:

- `SUNWadmc`
- `SUNWadmfr`
- `SUNWadmfw`
- `SUNWpl5u`
- `SUNWxcu4`
- `SUNWxcu6`
- `SUNWctpls`
- `SUNWmfrun`
- `SUNWxwfnt`
- `SUNWxwise`
- `SUNWxwplr`
- `SUNWxwplt`
- `SUNWxwrtl`

Solaris Zones

The Solaris 10 OS provides the Solaris Zones facility, which allows application components to be isolated from one another, even though the zones share a single instance of the operating system. From an application perspective, a zone is a fully functional Solaris OS environment. Multiple zones can be created on a single computer system, each zone serving its own set of applications. Detailed information about the use and features that are provided by Solaris zones can be found in the Solaris OS documentation.

It is possible to replace each of the computers in the portal service reference configuration's deployment architecture with a dedicated zone. The installation and configuration steps in this document would apply to a deployment in Solaris non-global zones. The installation of Java ES components in Solaris zones (whole root or sparse) is supported with certain restrictions as described in the Java Enterprise 5 Update 1 documentation.Appendix A, "Java ES and Solaris 10 Zones," in *Sun Java Enterprise System 5 Installation Planning Guide*

One reason to use Solaris zones is for improved security. A non-global zone can be used to run applications (for example, Directory Server, Access Manager, Portal Server, and so forth), while the administration and monitoring can be done from the global zone. A non-global zone cannot access resources in the global zone. So the management and monitoring applications installed in the global zone will not be visible and will not interfere with the applications installed in the non-global zones.

Another reason to use Solaris zones is for better resource utilization. The portal service reference configuration uses a modularized deployment architecture that is based on a number

of dedicated computers. This approach improves the manageability, scalability, and availability of the reference configuration. Using zones, it is possible to install multiple modules on the same computer and still achieve the reference configuration quality-of-service goals. For example, it is possible to install directory, Access Manager, and portal service modules on a single computer, with each using a dedicated Solaris zone. You need to size the individual systems properly, so the memory, disk, and processing power of each component is considered in sizing the whole computer. Solaris Resource Management can be used in conjunction with Solaris zones. The benefit of this approach is that resources (memory, CPU cycles) can be dynamically allocated for each zone, providing a better overall resource utilization.

Beyond this general explanation, this guide does not provide procedures for implementing the reference configuration in Solaris zones. The procedures are very similar, except that the zones need to be configured and networked before you install any of the Java ES components.

Network Connectivity Specification

Before you can install and configure Java ES components, the computers that you are using must be assigned IP addresses and attached to your network. The network topology for the portal service reference configuration uses several subnets with different ranges of IP addresses for each subnet. A network connectivity specification shows the network connections and the IP addresses that are needed to implement the reference configuration.

A network connectivity specification is typically a graphical representation of the required network configuration. The following figure shows the specification for the Portal Service on Application Server Cluster reference configuration. In the specification, all computers are shown in a *pstest.com* domain and are assigned the IP addresses that are used to establish the required network topology.

Note – The procedures in this guide use the host names, domain name, and IP addresses shown in Figure 3–1. However, you must map these host names, domain name, and IP addresses to equivalent names and addresses in your environment. For this reason, the procedures in this guide show host names, domain name, and IP addresses as variables.

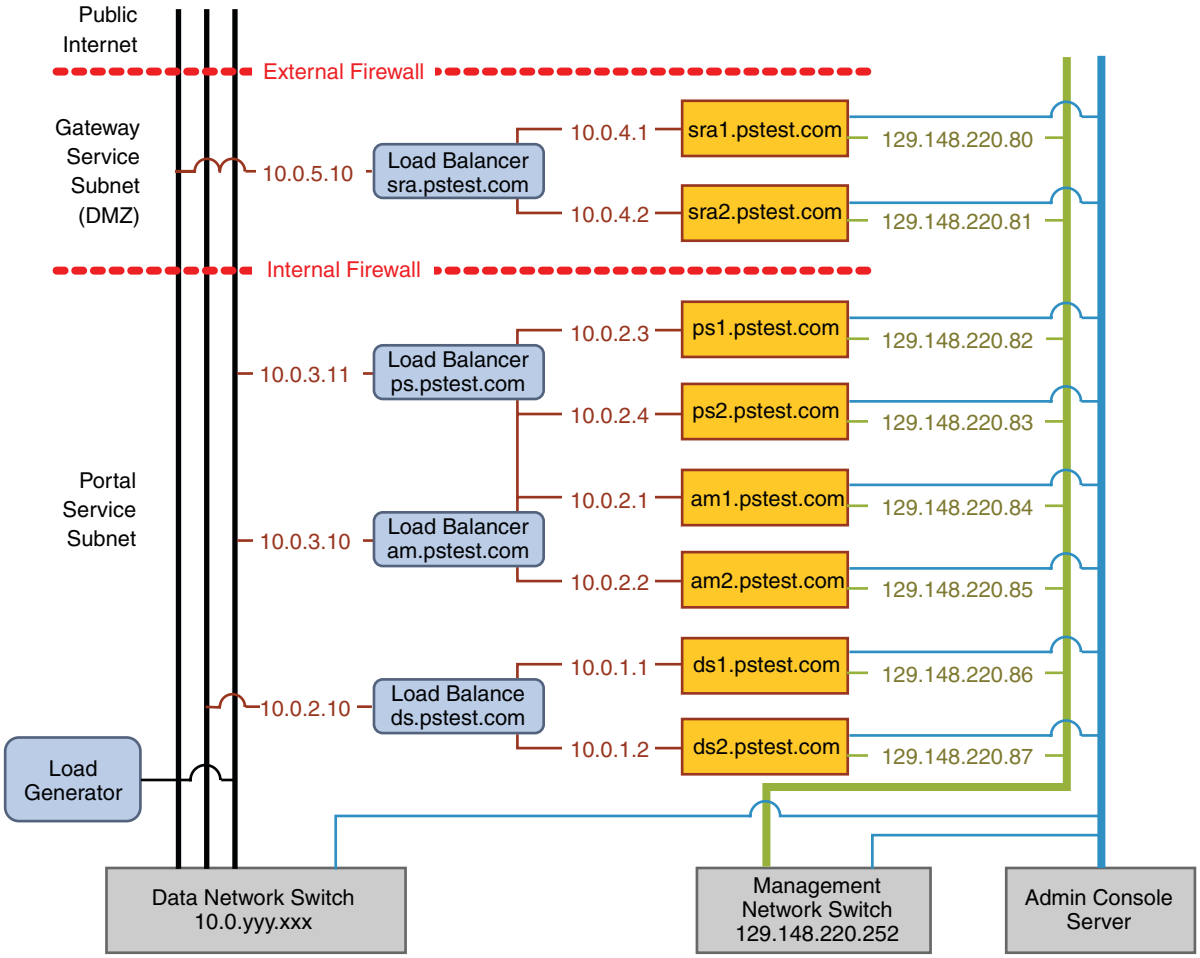

FIGURE 3–1 Network Connectivity Specification for the Reference Configuration Deployment

This figure illustrates how the different modules in the architecture are connected. Each module consists of two component instances, as well as a load balancer that provides a single entry point for the module. Each load balancer is configured to provide a virtual service address that accepts all requests for its respective service. The load balancer is configured to route such requests among the component instances in the module.

Portal Service Subnet

In Figure 3–1, the directory, Access Manager, and portal service modules reside in a network zone that is isolated from the main corporate network. Within this zone are separate subnets that are used to help secure each service.

Each service is accessed only through its respective load balancer. Clients of the service address their requests to the virtual IP address that is configured into the load balancer. Behind the load balancer, the computers that are running the component instances are isolated on their own subnets with private IP addresses. In Figure 3–1, the following five subnets are used:

- Directory service subnet: *10.0.1.0/24*
- Access Manager/portal service subnet: *10.0.2.0/2*4
- Access Manager/portal service load balancer subnet: *10.0.3.0/24*
- Gateway service subnet: *10.0.4.0/24*
- Gateway service load balancer subnet: *10.0.5.0/24*

The directory service load balancer is on the same subnet as the Access Manager and Portal Server instances because the latter directly access directory services.

These subnets are bridged by the load balancers, and all communications between the subnets is routed through routers. Therefore, if one subnet is compromised, there is no direct route to other services.

Gateway Service Subnet

The Gateway service runs in a separate subnet (the DMZ) that is isolated from the portal service subnet by an Internal Firewall and from the public Internet by an External Firewall, as shown in Figure 3–1.

In the DMZ, only the Gateway service load balancer (at *sra.pstest.com*) is exposed to traffic from the public Internet, and only through the External Firewall. Other hardware in the DMZ is assigned a private IP address, in keeping with the philosophy of minimizing the surface of attack. In Figure 3–1, the DMZ subnet is created with private IP addresses in the *10.0.4.0/24* range. These private addresses are not recognized by the Internet and are not routed outside the network.

Note – In Figure 3–1, the gateway service load balancer is shown with the IP address *10.0.5.10*. When you deploy your reference configuration, you must configure this load balancer with a real, publicly accessible IP address that is appropriate for your site.

The firewall rules that are used to establish the Gateway service subnet are shown in the following tables.

TABLE 3–2 Internal Firewall Rules

Rule Number	Source	Destination	Type/Port	Action
1	*sra1.pstest.com* *sra2.pstest.com*	*am.pstest.com*	TCP/80	ALLOW
2	*sra1.pstest.com* *sra2.pstest.com*	*ps.pstest.com* (Portal Server)	TCP/80	ALLOW
3	*sra1.pstest.com* *sra2.pstest.com*	*ps.pstest.com* (Rewriter Proxy)	TCP/10433	ALLOW
4	*sra1.pstest.com* *sra2.pstest.com*	*ps.pstest.com* (Netlet Proxy)	TCP/10555	ALLOW
5	*am1.pstest.com* *am2.pstest.com*	*sra1.pstest.com* *sra2.pstest.com*	TCP/443	ALLOW
6	*	*	*	DENY

The first two rules in the previous table allow the Gateway instances to reach the virtual service IP addresses (the load balancers) for the Access Manager and portal services. Rule 3 allows the session notifications that are generated by the Access Manager instances to reach the Gateway instances. The firewall automatically adds rules to allow the response traffic.

TABLE 3–3 External Firewall Rules

Rule Number	Source	Destination	Type/Port	Action
1	*	*sra.pstest.com*	TCP/443	ALLOW
2	*	*	*	DENY

The rules in the previous table allow only the Gateway service load balancer to be accessed from the Internet.

DNS Considerations

In implementing a network connectivity specification, you must coordinate the setting of virtual service IP addresses with the configuration of your DNS servers (or whatever naming service your network is using). Doing so ensures that the correct service names and IP addresses are routed publicly. In Figure 3–1, the externally accessible DNS server maps the URL *www.pstest.com* to the virtual service IP address for the Gateway service load balancer. For example, the internal DNS server maps the host name *sra.pstest.com* to the same virtual service address.

Other Networks

Figure 3–1 also shows two additional networks that are often used to implement a deployment architecture:

- The management network is used for performing software component installations and for monitoring component instance behaviors. The management network shown in Figure 3–1 uses a subnet with IP addresses *129.148.220.0/24*.
- The Console Server network is used to install the operating systems and manage startup of all the computers. The Console Server in Figure 3–1 accesses all computers through a serial port.

Load Balancer Configuration Specification

In the reference configuration's modular architecture, each module has a load balancer that routes traffic among the component instances in the module. For each module, the load balancer is configured with a virtual IP address for the service that the module provides. All of the requests for the service are delivered to the load balancer. The load balancer then routes this traffic among the component instances in the module.

For example, in Figure 3–1, the directory service module consists of two computers that are running instances of Directory Server (*ds1.pstest.com* and *ds2.pstest.com*) and a load balancer (*ds.pstest.com*) that is placed in front of the two computers. Requests for directory services are addressed to the load balancer at *ds.pstest.com*, and the load balancer is configured to distribute these requests between the Directory Server instances running on *ds1.pstest.com* and *ds2.pstest.com*.

In configuring a load balancer, three categories of configurable parameters need to be specified, as described in the following sections:

- "IP Address Configuration" on page 55
- "Configuration of Routing Characteristics" on page 56
- "Health-check Configuration" on page 56

IP Address Configuration

The virtual IP addresses and real IP addresses that are used to configure each load balancer are shown in Figure 3–1. In configuring your load balancers, substitute the service names, host names, and IP addresses that you will be using on your network. Details of setting up each load balancer are provided in the implementation procedure for the respective module.

Configuration of Routing Characteristics

The following table specifies characteristics that are required for each load balancer in the reference configuration to properly route requests. For example, the bottom row of the table below describes how each load balancer needs to be configured to maintain session persistence (stickiness).

TABLE 3–4 Specification for Load Balancer Routing

Parameter	Directory Service	Access Manager Service	Portal Service	Gateway Service
Virtual Service Name	*ds.pstest.com*	*am.pstest.com*	*ps.pstest.com*	*sra.pstest.com*
Protocol	LDAP	HTTP	HTTP	HTTPS or HTTP, depending on whether SSL is terminated or not
Port	389	80	80	443
Virtual Service Type	Layer-4 (TCP)	Layer-7 (HTTP)	Layer-7 (HTTP)	Layer-7 (HTTP) or SSL, depending on whether SSL is terminated
Scheduling	Least Connections or Round Robin	Least Connections or Round Robin	Least Connections or Round Robin	Least Connections or Round Robin
Session Persistence (Stickiness)	Long Persistent TCP Connections	Based on server-side cookie `amlbcookie`	Load balancer-managed cookie	SSL Session ID or Load balancer-managed cookie

Health-check Configuration

Load balancers use a health-check mechanism to establish if a service instance is properly working and if it can process requests from clients. If the health-checks succeed, the load balancer includes the service instance in the pool of available instances, and requests are routed to the instance based on the existing scheduling rules. However, if the health-checks fail, the instance is removed from the load balancer's scheduling list.

A health-check is considered failed if the response is different than the one expected, or if no response is received after a specified timeout value. The timeout mast be properly tuned because if it is too short, a sporadically overloaded service that is slow to respond can be considered down. If the timeout is too long, the load balancer will take too much time to detect failures, and users will notice the lack of response.

The simplest health-check is to try to open a TCP connection to the service instance. However, this health-check only proves that the application is listening on the assigned port. It does not show that the instance can process requests. To better establish that the instance is properly working, the health-check must actually exercise the service instance.

The load balancer performs health-checks at a specified interval. The interval needs to be as short as possible so that the load balancer will quickly detect failures. However, too many health-check requests can cause performance degradation. In the worst case, frequent health-checks can overload the service instances.

To determine if a server instance is down, the load balancer monitors the number of consecutive failed health-checks. If this number reaches a specified threshold, an instance is considered down. The time it takes to make this determination equals the number of consecutive failed health-checks, multiplied by the health-check interval. During this time, the load balancer considers a failed instance to be operating correctly, and users will notice a lack of response.

The health-check parameters need to be tuned separately for each service module. The following table specifies health-check parameter values that can be used as a starting point for the reference configuration.

TABLE 3–5 Specification for Load Balancer Health-Checks

Parameter	Directory Service	Access Manager Service	Portal Service	Gateway Service
Health-check Type	LDAP (simple, anonymous bind)	HTTP	HTTP	HTTP
Query	DN: <None> Base:dc=pstest,dc=com Scope: Base Query: (objectclass=*)	GET `/amserver/isAlive.jsp`	GET `/portal`	GET
Expected Result	Any LDAP success code	HTTP 200	HTTP 302	HTTP 302
Health-check Timeout	20 seconds	10 seconds	5 seconds	5 seconds
Interval Between Checks	60 seconds	30 seconds	30 seconds	30 seconds
Consecutive Failed Health-check Threshold	3	3	3	3

Note – In the reference configuration, Gateway SSL sessions are terminated at the load balancer, and the Gateway instances run plain HTTP. If the SSL sessions are terminated at the Gateway instances instead of at the Gateway load balancer, then the Health-check needs to be configured to use the SSL protocol.

Administrator Account Specification

When deploying the portal service reference configuration, you install and configure a number of components with administrative interfaces, as well as administrator accounts for accessing these interfaces. Some of these administrator accounts are used by multiple components.

In many environments, different administrator accounts are used to manage different services. However, if there are no specific reasons to use different passwords for the different administrator accounts, you can streamline the installation, configuration, and maintenance of your deployment by using the same password for all such accounts.

Note – It is important to determine, *in advance*, the administrative account IDs and passwords that you will use when deploying the reference configuration.

The following table shows the administrator account IDs that are needed to deploy the reference configuration, the variables that are used in this guide to represent the corresponding passwords, and the interfaces that are managed by each of the administrator accounts.

TABLE 3–6 Administrator Accounts in Reference Configuration

Account ID	Password Variable	Interfaces
`admin`	*directory-admin-password*	Directory Server `dsconf` command
		Directory Service Control Center (DSCC)
`cn=Directory Manager`	*directory-manager-password*	Accessing directory data
		`ldapmodify` and `ldapsearch` commands

TABLE 3–6 Administrator Accounts in Reference Configuration *(Continued)*

Account ID	Password Variable	Interfaces
`amadmin`	*access-manager-admin-password*	Access Manager `amadmin` command
		Portal Server `psadmin` command
		Access Manager Console
		Portal Server Console
`amldapuser`	*access-manager-LDAP-password*	Access Manager's Directory Server account
`admin`	*app-server-admin-password*	Application Server `asadmin` command
		Application Server Admin Console
	app-server-master-password	Application Server cluster features
`admin`	*MQ-admin-password*	Message Queue `imqcmd` command

When you use command-line interfaces in the implementation procedures in this guide, you can provide the administrator account password in any of the following ways:

- Type a password on the command line by using the option that is provided by the command (usually `-p` *password*).

  This approach is not very secure, and is not supported by the Portal Server `psadmin` command.

- Create a password file that contains the password, and reference the password file on the command line by using the option provided by the command (usually `-f` *password-file*)

  To create a password file, type the following command:

  `# echo` *password*`>`*password-file*

  This approach is more secure than typing the password on the command line. However, the contents of the password file can be stolen.

- Type only the administrator user ID on the command line, and type the password only when prompted.

  This approach is quite secure but makes scripting of procedures more difficult.

When implementing the reference configuration, you are free to choose whichever approach you wish. For consistency, however, the last approach is used in all the implementation procedures in this guide.

User Management Specification

The process of deploying the portal service reference configuration establishes an LDAP directory schema and the basic tree structure of the LDAP directory. Before beginning the installation and configuration process, analyze your directory requirements and design a schema and a directory tree structure that supports your application system needs. Preparing a user management specification, in advance, ensures that you have the directory you need after having completed deployment.

LDAP Schema

Installing and configuring the reference configuration components creates a basic LDAP schema, as follows:

- When you install Directory Server, the basic schema is created.
- When you install Access Manager, the basic schema is extended to support Access Manager. (This is sometimes referred to as schema 2 in Access Manager's legacy mode.)
- After you deploy and test your reference configuration, and you begin to add custom content and service channels to your portal, you normally need to extend the LDAP schema further. Depending on the content and services your portal service will provide, you will probably need to add object classes and attributes to the schema. For more information about managing schema to support custom content, see Chapter 11, "Directory Server Schema," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

Directory Tree

Access Manager introduced a new data structure configuration with Access Manager 7.0. The new *realm mode* separates configuration data and user data into different repositories, thus supporting different data formats for user data and corresponding interfaces for accessing that data. In contrast to the previous *legacy mode*, in which both configuration data and user data are stored in a single LDAP directory tree, realm mode enables Access Manager to plug in multiple user repositories, while storing service configuration data in a single realm repository.

The Portal Service on Application Server Cluster reference configuration is based on legacy mode configuration of Access Manager. Legacy mode fully supports Portal Server access to data. In this mode, the Access Manager service and policy configuration data are merged with user data in the same LDAP directory.

However, realm mode can also support Portal Server as long as Access Manager is configured to use the Access Manager SDK datasource plugin that Portal Server uses to access service data in Directory Server. Using Access Manager in realm mode for the reference configuration requires additional configuration to map elements in the realm repository to elements in the user repository. Nevertheless, this realm mode configuration is outside the scope of this reference configuration guide.

Installing and configuring the reference configuration in legacy mode creates a basic LDAP directory tree. Input supplied during the installation and configuration process determines the directory tree root suffix, as follows:

- When you install Directory Server, you specify the directory tree's base suffix.
- When you install Access Manager, you configure it to look for user data under the directory's root suffix.

The procedures in this guide for installing Directory Server create the directory tree structure shown in the following figure.

The root suffix in the figure is shown as `dc=pstest,dc=com`.

Note – The procedures in this guide use the root suffix shown in Figure 3–2. However, you must specify a root suffix different from `dc=pstest,dc=com` that is suitable for your organization. For this reason, the procedures in this guide show `dc=pstest,dc=com` as a variable.

Additional user management specifications are needed to support custom content and service channels in your portal.

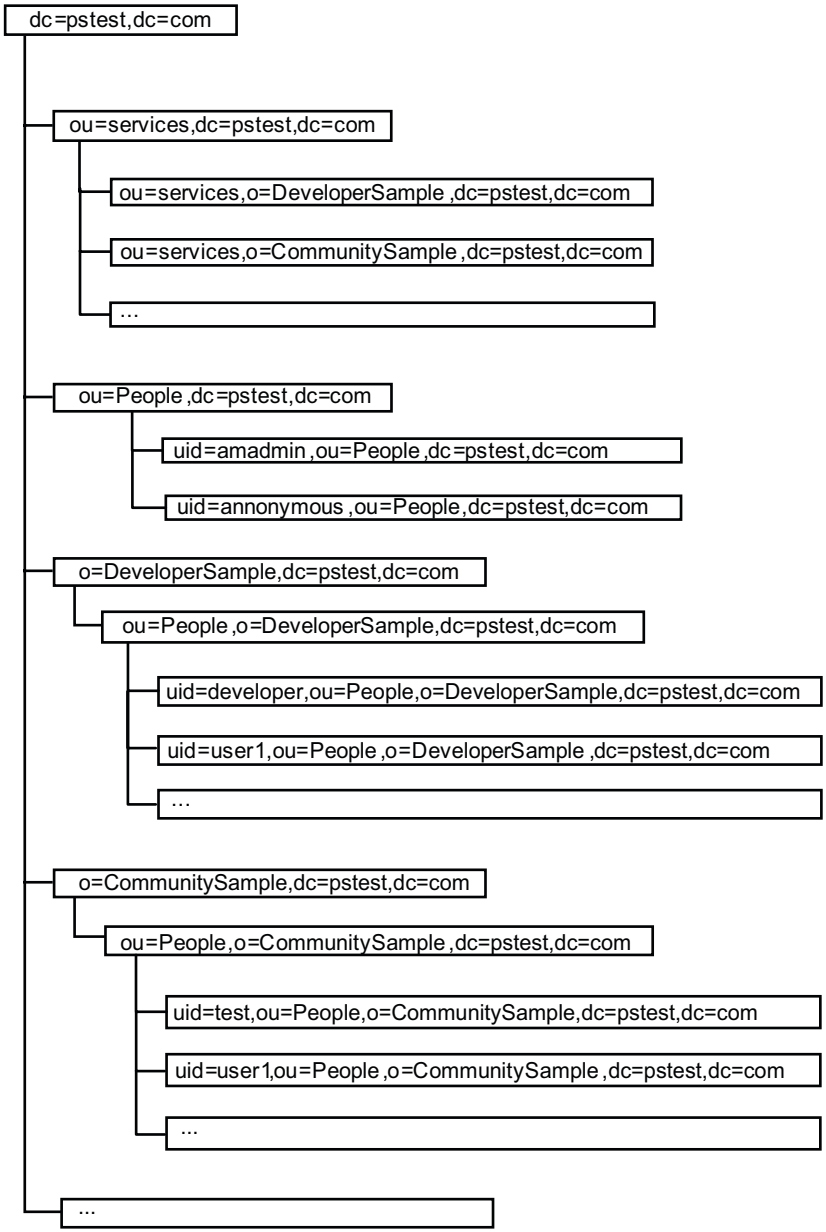

FIGURE 3–2 Basic LDAP Directory Tree for the Reference Configuration

◆ ◆ ◆ CHAPTER 4

Implementation Module 1: Directory Server With Multimaster Replication

This chapter provides an overview of the directory service module in Figure 2–2 and documents the tasks that are required to implement it. The chapter includes the following sections:

Overview of the Directory Service Module

The directory service module of the reference configuration's deployment architecture illustrated in Figure 2–2 consists of two instances of Sun Java System Directory Server running on two different computers. The module makes use of a hardware load balancer that is configured to provide service failover capability between the two Directory Server instances. All requests for directory services are addressed to the virtual service name and IP address of the load balancer. The load balancer then directs each request to one of the two Directory Server instances.

In this module, the two Directory Server instances use multimaster replication to synchronize their data and to provide for a highly available directory service.

The architecture of the directory service module is shown in the following illustration.

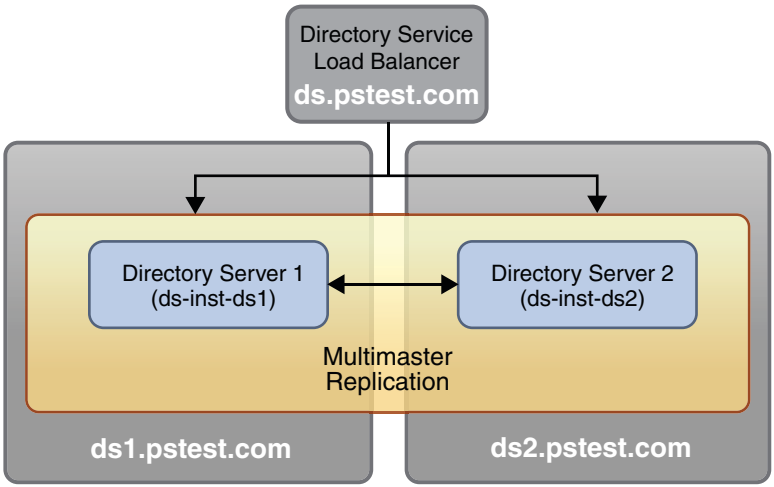

FIGURE 4–1 Directory Service Module

When implementing the directory service module, you set up the Directory Server instances on *ds1* and *ds2*. However, you do not implement multimaster replication until *after* you have installed and configured all of the other components in the reference configuration. The process of installing and configuring the other components writes configuration data to the directory, and in some cases will extend the schema. It is is easier to ensure that the directory is updated correctly if the configuration data and schema extensions are written to a single Directory Server instance. The data is replicated on the other Directory Server instance only after such configuration is complete.

Note – The procedures in this chapter use the host names, domain name, and IP addresses shown in Figure 3–1 and Figure 4–1. However, you must map these host names, domain name, and IP addresses to equivalent names and addresses in your environment. For this reason, the procedures in this chapter show host names, domain name, and IP addresses as variables.

Setting Up Directory Server on *ds1*

This task consists of the following procedures:

- Install Directory Server on *ds1*
- Start and Verify Directory Server on *ds1*

▼ To Install Directory Server on *ds1*

This procedure assumes that you are installing Directory Server on Solaris 10 8/07 OS or later version. Hence, no operating system patches need to be installed. The Java ES installer evaluates the state of the operating system and indicates if you need to install a patch. If you are using versions of the operating system older than Solaris 10 8/07 OS, it is better to install any required patches before you begin the actual Directory Server installation procedure.

The following procedure runs the Java ES installer without saving a state file. You can choose to run the installer and capture your input in a state file (`-saveState` *state-filename*). You could then use the state file to re-create the installation if, for example, you needed to reinstall Directory Server.

1 Download the Java ES software distribution to *ds1*.

The procedure is documented in "To Download the Software Distribution" on page 185.

2 Log in as root or become superuser.

```
# su -
```

3 Start the Java ES installer.

```
# cd /portdist_71u2/Solaris_sparc

# ./installer
```

This procedure uses the GUI installer. The installer can also be run in text mode by using the `-nodisplay` option.

The Welcome panel opens.

4 In the Welcome panel, click Next.

The Software License Agreement panel opens.

5 In the Software License Agreement panel, review the license terms and click Yes, Accept License.

The Choose Software Components panel opens.

6 In the Choose Software Components panel, select the following components:

- Directory Server Enterprise Edition 6.2
  - Directory Server 6 Core Server
  - Directory Service Control Center
  - Directory Server Command-line Utility
- Directory Preparation Tool 6.4 (selected automatically)
- Install Multilingual Package(s) for all selected components (selected automatically, but optional if using English)

7 Click Next.

The Java SE Software Development Kit Upgrade Required panel opens.

8 In the Java SE Software Development Kit Upgrade Required panel, select Automatic Upgrade to the Version Included with the Installer and click Next.

The installer evaluates the Java ES shared components on the computer and determines if any upgrades are required. On a fresh copy of the Solaris 10 8/07 OS, shared component upgrades are needed, and the Shared Components Upgrades Required panel opens.

9 In the Shared Components Upgrades Required panel, click Next.

The installer upgrades the shared components. The Specify Installation Directories panel opens.

10 In the Specify Installation Directories panel, type the following values and click Next.

Input Field	Value
Directory Preparation Tool	`/opt/SUNWcomds`
Directory Server	`/opt/SUNWdsee`

The installer checks the system, and the System Check panel opens.

11 In the System Check panel, evaluate the results of the system check.

If the system check is favorable, click Next.

The Choose a Configuration Type panel opens.

12 In the Choose a Configuration Type panel, select Configure Now and click Next.

The Common Server Settings panel opens.

13 In the Common Server Settings panel, type the following values and click Next.

Be sure to first read "Administrator Account Specification" on page 58.

Input Field	Value
Host Name	*ds1*
DNS Domain Name	*pstest.com*
Host IP Address	*10.0.1.1*
Administrator User ID	`admin`
Administrator Password	*directory-admin-password*

Input Field	Value
System User	`root`
System Group	`root`

The Directory Server: Create Directory Instance panel opens.

14 In the Directory Server: Create Directory Instance panel, type the following value and click Next.

Input Field	Value
Create a Directory Server Instance	`Yes`

The Directory Server: Specify Instance Creation Information panel opens.

15 In the Directory Server: Specify Instance Creation Information panel, type the following values and click Next.

Input Field	Value
Instance Directory	`/var/opt/SUNWdsee/ds-inst-`*ds1*
Directory Instance Port	`389`
Directory Instance SSL Port	`636`
Directory Manager DN	`cn=Directory Manager`
System User	`root`
System Group	`root`
Directory Manager Password	*directory-manager-password*
Suffix	*dc=pstest,dc=com*

The Ready to Install panel opens.

16 In the Ready to Install panel, indicate whether you want to open the software registration window during installation.

This panel enables you to register the components that you have selected for installation with Sun Connection. Sun Connection is a Sun-hosted service that helps you track, organize, and maintain Sun hardware and software. For example, Sun Connection can inform you of the latest available security fixes, recommended updates, and feature enhancements.

If you choose to register, information about the installation is sent to the Sun Connection database. You can also register at a later date, after installation has been completed.

17 Click Install.

The installer copies files to the computer, creates a Directory Server instance and a database using the information in Step 15.

18 During the installation process, click N to decline an Internet connection.

Directory Server is being installed behind two firewalls, so it is unlikely that an Internet connection can be established for updates.

19 When the installation is complete, review the installation in the Summary field.

20 Click Exit to exit the installer.

21 Check the installation log files for any installation errors.

```
# cd /var/sadm/install/logs
```

```
# egrep -i 'fall|error' Java*
```

▼ To Start and Verify Directory Server on *ds1*

The following procedure confirms that Directory Server has been installed by starting and connecting to a Directory Server instance on *ds1*.

1 Start the Directory Server instance on *ds1*.

```
# /opt/SUNWdsee/ds6/bin/dsadm start /var/opt/SUNWdsee/ds-inst-ds1
```

2 Check that the Directory Server instance is running.

```
# /opt/SUNWdsee/ds6/bin/dsadm info /var/opt/SUNWdsee/ds-inst-ds1
```

The `State:` line in the output should indicate that the instance is running.

3 Check that you can connect to the Directory Server instance and perform a basic operation.

```
# ldapsearch -b "dc=pstest,dc=com" -h ds1 -p 389 -D "cn=Directory Manager"
"objectClass=*"
```

When prompted, type the *directory-manager-password*.

The response should resemble the following:

```
version: 1
dn: dc=pstest,dc=com
dc: pstest
objectClass: top
objectClass: domain
```

Setting Up Directory Server on *ds2*

This task consists of the following procedures:

- Install Directory Server on *ds2*
- Start and Verify Directory Server on *ds2*

▼ To Install Directory Server on *ds2*

- **Repeat the procedure that appears in "To Install Directory Server on *ds1*" on page 65, except for the following:**
  - Replace all occurrences of *ds1* with *ds2*.
  - When you are prompted for the host IP address, type *10.0.1.2* instead of *10.0.1.1*.

▼ To Start and Verify Directory Server on *ds2*

- **Repeat the procedure that appears in "To Start and Verify Directory Server on *ds1*" on page 68, except for the following:**
  - Replace all occurrences of *ds1* with *ds2*.
  - Replace the instance name of `ds-inst-`*ds1* with `ds-inst-`*ds2*.

Configuring the Directory Server Control Center

The Directory Server Control Center (DSCC) is a tool for managing Directory Server instances. DSCC is accessed through Sun Java Web Console™ (Web Console), a web application that provides a single user interface framework for Sun system management applications.

This task consists of the following procedures:

- Create an Instance of the Directory Server Control Center
- Register Your DSCC Instance With Sun Java Web Console
- Register Your Directory Server Instances with DSCC
- Verify Configuration of the DSCC

▼ To Create an Instance of the Directory Server Control Center

1 Assess the current status of the control center.

On *ds1*, run the following commands:

```
# cd /opt/SUNWdsee/dscc6/bin
```

```
# ./dsccsetup status
```

The response should resemble the following:

```
***
DSCC Application is registered in Sun Java (TM) Web Console
***
DSCC Agent is registered in Cacao
***
DSCC Registry has not been created yet
***
```

This response indicates that the installer has installed the DSCC packages but did not create a DSCC instance.

2 Start the DSCC configurator.

```
# ./dsccsetup install
```

The response should resemble the following:

```
### 'install' subcommand is obsolete.
### Use 'ads-create' subcommand instead.
Choose password for Directory Server Manager:
```

3 When prompted, type the *directory-admin-password*.

The response should resemble the following:

```
Confirm password for Directory Service Manager: Creating DSCC registry...
DSCC Registry has been created successfully.
```

4 Confirm that your new DSCC instance is running.

```
# ps -ef | grep dscc6
```

The response should resemble the following:

```
/opt/SUNWdsee/ds6/lib/64/ns-slapd -D /var/opt/SUNWdsee/dscc6/dcc/ads
-i /var/opt
```

5 If the DSCC instance is not running, start it.

```
# /opt/SUNWdsee/ds6/bin/dsadm start /var/opt/SUNWdsee/dscc6/dcc/ads
```

▼ To Register Your DSCC Instance With the Web Console

If the `dsccsetup status` command in Step 1 of "To Create an Instance of the Directory Server Control Center" on page 70 does not indicate that the DSCC application is registered in the Web Console, then perform the following steps.

1. **Check the status of Web Console.**

   ```
   # cd /usr/share/webconsole/bin
   # ./smcwebserver status
   ```

   The output should resemble the following:

   ```
   Sun Java(TM) Web Console is stopped
   ```

2. **If the Web Console is not running, start the Web Console.**

   ```
   # ./smcwebserver start
   ```

3. **Register your DSCC instance.**

   a. **Run the following command:**

      ```
      # /opt/SUNWdsee/dscc6/bin/dsccsetup smreg
      ```

      The response prompts you to automatically restart the Web Console.

   b. **Type `Y` and press `Return`.**

▼ To Register Your Directory Server Instances With DSCC

To manage your Directory Server instances, you must register your instances with the DSCC. Doing so modifies the Directory Server instance's `cn=config` tree.

To complete this task, you work in both the command-line and the DSCC Web Console interfaces.

1. **Start a Browser.**

2. **Go to the Web Console login page.**

   `https://`*ds1.pstest.com*`:6789`

   The Web Console login page opens.

3. **Log in to the Web Console by typing the following values and clicking Login.**

Input Field	Value
User ID	`root` (Any authorized user can log in to the Web Console, but you must log in as root to register the DSCC.
password	*root-password*

The DSCC main page in Web Console opens.

4 In the DSCC main page, locate the list of services and click the link for the Directory Server Control Center.

The Directory Server Control Center page opens.

5 Type the following values and click Login.

Input Field	Value
User ID	`admin`
Password	*directory-admin-password*

The Directory Service Control Center Common Tasks panel appears.

6 Interrupt the registration procedure to Enable DSCC audit logging.

The audit logs will show the DSCC entries to be added in the registration steps that follow.

a. Run the following command on *ds1*:

```
# /opt/SUNWdsee/ds6/bin/dsconf set-log-prop -p 389 audit enabled:on
```

You are prompted to accept a certificate.

b. Type `Y` to accept the certificate and press Return.

c. When prompted, type the *directory-manager-password* and press Return.

The response should resemble the following:

```
time: 20080220175511
dn: cn=config
changetype: modify
replace: nsslapd-auditlog-logging-enabled
nsslapd-auditlog-logging-enabled: on
```

7 Returning to the Web Console, click the Directory Servers tab.

The Directory Servers tab is displayed, and the Enter Host Info panel opens.

8 Register the Directory Server instance on *ds1*.

a. In the Directory Servers tab, locate the More Server Actions drop-down menu and select Register Existing Server.

The Register Existing Directory Server wizard opens, displaying the Step 1. Enter Host and Server Information panel.

b. In the Enter Host and Server Information panel, type the following values and click Next.

Otherwise, keep the default values.

Input Field	Value
Instance Path	`/var/opt/SUNWdsee/ds-inst-`*`ds1`*
Description	`ds-inst-`*`ds1`*

The Review Server Certificate panel opens.

c. Click Next to accept the certificate.

The Provide Authentication Information panel opens. Keep the default values.

d. Type the *directory-manager-password* and click Next.

The Summary panel opens stating that a restart is required

e. Click Finish.

Your Directory Server instance (`ds-inst-`*`ds1`*) restarts and registers with the DSCC.

f. When the registration process is complete, click Close.

The Register Existing Directory Server wizard closes.

9 Register the Directory Server instance on *ds2*.

Repeat Step 8, except replace all occurrences of *ds1* with *ds2* (for example, in the instance name, `ds-inst-`*`ds2`*).

You now see your Directory Server instances (`ds-inst-`*`ds1`* and `ds-inst-`*`ds2`*) in the DSCC's list of registered servers.

10 Check the audit logs for both Directory Server instances.

```
# tail -100 /var/opt/SUNWdsee/ds-inst-ds1/logs/audit
```

```
# tail -100 /var/opt/SUNWdsee/ds-inst-ds2/logs/audit
```

The audit logs should resemble the following:

```
time: 20080421170848
dn: cn=pass through authentication,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginarg0
nsslapd-pluginarg0: ldap://localhost:3998/cn=dscc
- replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
- replace: modifiersname
modifiersname: cn=directory manager
- replace: modifytimestamp
modifytimestamp: 20080421160847Z
-
time: 20080421170848
dn:
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0; acl "Enable full access for Directory Services Managers";
 allow (all)(userdn = "ldap:///cn=*,cn=Administrators,cn=dscc");)
aci: (targetattr = "aci") (targetscope = "base") (version 3.0; acl "Enable root ACI modification
 by Directory Services Managers"; allow (all)(userdn = "ldap:///cn=*,cn=Administrators,cn=dscc");)
```

11 Check the audit logs for the DSCC registry instance.

```
# tail -100 /var/opt/SUNWdsee/dscc6/dcc/ads/logs/audit
```

▼ To Verify Configuration of the DSCC

1 List the Directory Server instances that are registered with DSCC.

```
# /opt/SUNWdsee/dscc6/bin/dsccreg list-servers
```

When prompted, type the *directory-admin-password*.

2 Press Enter.

The response should resemble the following:

```
Hostname Port sPort Type Owner iPath Description
ds1 389 636 DS root /var/opt/SUNWdsee/ds-inst-ds1 ds-inst-ds1 on ds1
ds2 389 636 DS root /var/opt/SUNWdsee/ds-inst-ds2 ds-inst-ds2 on ds2
```

Implementing Load Balancing for the Directory Service

This task consists of the following procedures:

- Configure the Directory Service Load Balancer
- Configure Directory Server Instances for Load Balancing
- Verify Directory Service Load Balancing

▼ To Configure the Directory Service Load Balancer

This procedure describes how to configure the directory service load balancer (*ds.pstest.com* at IP address *10.0.2.10*). The steps are relatively generic; the details depend on the load balancer you are using.

1 Populate the load balancer's Hosts Table.

Add the IP address for *ds1.pstest.com* and *ds2.pstest.com* to the load balancer's hosts table.

2 Populate the load balancer's Real Service Table.

Add the real services for *ds1.pstest.com* and *ds2.pstest.com*. A real service is identified by its IP address and port. Add *10.0.1.1*:389 and *10.0.1.2*:389.

3 Populate the load balancer's Service Group Table

Add the service group for directory services. The service groups are sets of the real services that you defined in Step 2. The real services in the group must be capable of fulfilling the same type of request. The load balancer will distribute requests among the real services in the service group. When you define the service group for *ds.pstest.com*, you add the real services that specify the Directory Server instances, *10.0.1.1*:389 and *10.0.1.2*:389.

4 Populate the load balancer's Virtual IP Table.

A virtual service definition includes the outward facing IP address and the port at which the load balancer accepts requests for a service, as well as the service group that you specified in Step 3, which actually handles the requests. The load balancer will accept requests at the virtual service address and distribute them among the service group. The virtual service definition for the directory service should be *ds.pstest.com*, with the virtual IP address of *10.0.2.10*:389, and with the service group consisting of the computers *ds1.pstest.com* and *ds2.pstest.com*.

5 Configure the load balancer to use Layer-4 (TCP layer) load balancing.

If you are using a load balancer that supports long, persistent TCP connections, this is the best option. There is no need for stickiness at the directory service load balancer because the TCP connections remain open.

6 **Configure the load balancer with a scheduling type of either least connections or round robin.**

Both scheduling types initially distribute the connections evenly between the Directory Server instances. Both scheduling types keep connections evenly distributed if the connections are restarted.

7 **Configure the health-check settings for the load balancer.**

The recommended settings are specified in Table 3–5.

▼ To Configure Directory Server Instances for Load Balancing

Timeout problems can arise when a load balancer (or firewall) is placed between Access Manager and Directory Server, as explained in "Setting Connection Timeouts for Access Manager " on page 102. To prevent such problems, set the idle timeout for the Directory Server connections that are used by Access Manager to a value less than the idle timeout value of the load balancer (or firewall).

To perform this procedure, an `amldapuser` entry must exist in the directory. However, this entry is not created until you deploy Access Manager, as documented in Implementation Module 2: Access Manager Running on Application Server. You therefore must set the Directory Server idle timeout value *after* you install and configure Access Manager.

For this reason, the procedure for setting the idle timeout for Directory Server connections used by Access Manager is documented in Module 2.

- **See the procedure in "To Configure the Connection Timeout of the Directory Service " on page 102.**

▼ To Verify Directory Service Load Balancing

This procedure assumes that `ds-inst-`*ds1* on *ds1* and `ds-inst-`*ds2* on *ds2* are running.

1 **Shut down the Directory Server instance on *ds1*.**

```
# /opt/SUNWdsee/ds6/bin/dsadm stop /var/opt/SUNWdsee/ds-inst-ds1
```

2 **Verify that you can access the Directory Server instance on *ds2* through the load balancer.**

Run the following command from a computer that can access *ds.pstest.com*.

```
# ldapsearch -b "dc=pstest,dc=com" -h ds.pstest.com -p 389 -D "cn=Directory Manager"
"objectClass=*"
```

When prompted, type the *directory-manager-password*.

A list of object classes currently in the directory is displayed.

3 **Start the Directory Server instance on *ds1*.**

```
# /opt/SUNWdsee/ds6/bin/dsadm start /var/opt/SUNWdsee/ds-inst-ds1
```

4 **Shut down the Directory Server instance on *ds2*.**

```
# /opt/SUNWdsee/ds6/bin/dsadm stop /var/opt/SUNWdsee/ds-inst-ds2
```

5 **Verify that you can access the Directory Server instance on *ds1* through the load balancer.**

Run the following command from a computer that can access *ds.pstest.com*.

```
# ldapsearch -b "dc=pstest,dc=com" -h ds.pstest.com -p 389 –D "cn=Directory Manager"
"objectClass=*"
```

When prompted, type the *directory-manager-password*.

A list of object classes currently in the directory is displayed.

Confirming That the Directory Server Instance on *ds2* Is Stopped.

In the previous procedure ("To Verify Directory Service Load Balancing" on page 76) the Directory Server instance on *ds2* was stopped in order to verify proper operation of the directory service load balancer. This instance must remain shut down while the remaining modules in the reference configuration are implemented. Once these modules have been implemented, and configuration data has been written to the Directory Server instance on *ds1*, then the Directory Server instance on *ds2* can be restarted and multimaster replication can be implemented.

▼ To Confirm That the Directory Server Instance on *ds2* Is Stopped

- **Run the following command on *ds2*:**

```
# /opt/SUNWdsee/ds6/bin/dsadm info /var/opt/SUNWdsee/ds-inst-ds2
```

The `State:` line in the output should indicate that the instance is stopped. If it does not, then shut down the Directory Server instance on *ds2*, and perform the above step to confirm that it is stopped.

Implementing Multimaster Replication

Note – Do not implement multimaster replication until you have installed and configured all of the other components in the reference configuration deployment. Otherwise required configuration data will be missing from the Directory Server instance on *ds1*.

Multimaster replication ensures that both Directory Server instances are synchronized as data is changed for either instance. During the implementation of the Access Manager, Portal Server, and Portal Server Secure Remote Access modules, configuration data is written to the Directory Server instance on *ds1*. When implementing multimaster replication, the Directory Server instance on *ds2* is restarted and synchronized with the Directory Server instance on *ds1*. The only complication is that the directory indexes that support other components must be re-created manually on the Directory Server instance on *ds2*.

This task consists of the following procedures:

- Restart the Directory Server Instance on *ds2*
- Enable Multimaster Replication
- Create Replication Agreements
- Replicate Directory Data
- Verify Multimaster Replication
- Update the Directory Indexes

▼ To Restart the Directory Server Instance on *ds2*

1 **Start the Directory Server instance on *ds2*.**

```
# /opt/SUNWdsee/ds6/bin/dsadm start /var/opt/SUNWdsee/ds-inst-ds2
```

2 **Check that the Directory Server instance is running.**

```
# /opt/SUNWdsee/ds6/bin/dsadm info /var/opt/SUNWdsee/ds-inst-ds2
```

The `State:` line in the output should indicate that the instance is running.

▼ To Enable Multimaster Replication

The steps for enabling replication on both Directory Server instances are performed only on *ds1*.

1 **Enable multimaster replication for the *dc=pstest,dc=com* suffix on *ds1*.**

```
# cd /opt/SUNWdsee/ds6/bin
```

```
# ./dsconf enable-repl -h ds1 -p 389 -i -d 1 master dc=pstest,dc=com
```

When prompted, type the *directory-admin-password*.

2 **Enable multimaster replication for the** `dc=pstest,dc=com` **suffix on** *ds2*.

```
# ./dsconf enable-repl -h ds2 -p 389 -i –d 2 master dc=pstest,dc=com
```

When prompted, type the *directory-admin-password*.

▼ To Create Replication Agreements

The steps for creating replication agreements between the Directory Server instances are performed only on *ds1*.

1 **Create a replication agreement between** *ds1* **and** *ds2* **for the** *dc=pstest,dc=com* **suffix.**

```
# ./dsconf create-repl-agmt -i -h ds1 -p 389 dc=pstest,dc=com ds2:389
```

When prompted, type the *directory-admin-password*.

2 **Create a replication agreement between** *ds2* **and** *ds1* **for the** *dc=pstest,dc=com* **suffix.**

```
# ./dsconf create-repl-agmt -i -h ds2 -p 389 dc=pstest,dc=com ds1:389
```

When prompted, type the *directory-admin-password*.

▼ To Replicate Directory Data

- **Copy data from** *ds1* **to** *ds2* **by running the following command on** *ds1*:

```
# ./dsconf init-repl-dest -i -h ds1 -p 389 dc=pstest,dc=com ds2:389
```

When prompted, type the *directory-admin-password*.

The response should resemble the following:

```
Started initialization of “ds2:389”; MMM DD YYYY HH:MM:SS
Sent 644 entries...
Sent 1229 entries...
Sent 2170 entries...
Sent 2965 entries...
Sent 3463 entries...
Sent 3902 entries...
Sent 4361 entries...
Sent 4362 entries...
Completed initialization of “ds2:389; MMM DD YYYY HH:MM:SS
```

▼ To Verify Multimaster Replication

1 Check that your Directory Server instances are synchronized.

Run the following command on *ds1*:

```
# /opt/SUNWdsee/ds6/bin/insync -D "cn=Directory Manager" -s ds2:389
```

When prompted, type the *directory-manager-password*.

The response should resemble the following:

```
Replica DN Consumer Supplier Delay
dc=pstest,dc=com ds1:389 -1 *CSN has not been intialized. No changes received.
```

2 Add an entry to the Directory Server instance on *ds1*.

```
# ldapmodify -h ds1 -p 389 -D "cn=Directory Manager" <<EOF
dn: o=id1,dc=pstest,dc=com
changetype: add
objectClass: top
objectClass: organization
description: ds1
EOF
```

When prompted, type the *directory-manager-password*.

The response should resemble the following:

```
adding new entry o=id1,dc=pstest,dc=com
```

3 Verify that the entry is replicated on *ds2*.

```
# ldapsearch -h ds2 -p 389 -D "cn=Directory Manager" -b o=id1,dc=pstest,dc=com
objectClass=*
```

When prompted, type the *directory-manager-password*.

The response should resemble the following:

```
dn: o=id1,dc=pstest,dc=com
objectClass: top
objectClass: organization
description: ds1
o: id1
version: 1
```

4 Add an entry to the Directory Server instance on *ds2*.

```
# ldapmodify -h ds2 -p 389 -D "cn=Directory Manager" <<EOF
```

```
dn: o=id2,dc=pstest,dc=com

changetype: add

objectClass: top

objectClass: organization

description: ds2

EOF
```

When prompted, type the *directory-manager-password*.

The response should resemble the following:

```
adding new entry o=id2,dc=pstest,dc=com
```

5 **Verify that the entry is replicated on *ds1*.**

```
# ldapsearch -h ds1 -p 389 -D "cn=Directory Manager" -b o=id2,dc=pstest,dc=com
"objectClass=*"
```

When prompted, type the *directory-manager-password*.

The response should resemble the following:

```
dn: o=id2,dc=pstest,dc=com
objectClass: top
objectClass: organization
description: ds2
o: id2
version: 1
```

▼ To Update the Directory Indexes

This procedure uses the index file on *am1* to update the Directory Server instance on *ds2* with indexes that support Access Manager.

1 **Copy the following file from *am1* to `/tmp` on *ds2*:**

```
/etc/opt/SUNWam/config/ldif/index.ldif
```

2 **Add the indexes to the Directory Server instance on *ds2*.**

```
# ldapmodify -D "cn=Directory Manager" -c -a -h ds2 -p 389 -f /tmp/index.ldif
```

When prompted, type the *directory-manager-password*.

The response should resemble the following:

```
adding new entry cn=nsroledn,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config ldap_add: Already exists
adding new entry cn=memberof,cn=index,cn=pstest,
```

```
    cn=ldbm database, cn=plugins,cn=config
adding new entry cn=iplanet-am-static-group-dn,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=iplanet-am-static-group-dn,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=iplanet-am-modifiable-by,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=iplanet-am-user-federation-info-key,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=sunxmlkeyvalue,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=o,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=ou,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=sunPreferredDomain,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=associatedDomain,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
adding new entry cn=sunOrganizationAlias,cn=index,cn=pstest,
    cn=ldbm database,cn=plugins,cn=config
```

3 Using the Web Console, re-index the *dc=pstest, dc=com* suffix.

a. Start a Browser.

b. Go to the Web Console login page.

`https://`*ds1.pstest.com*`:6789`

The Web Console login page opens.

c. Log in to the Web Console by typing the following values and clicking Login.

Input Field	Value
User ID	`root`
password	*root-password*

The DSCC main page in Web Console opens.

d. In the DSCC main page, click the Servers tab.

e. Locate the link for *ds2*`:389` and click the link.

f. Click the Suffix tab.

g. Locate the link for *dc=pstest,dc=com* and click the link.

h. Click the Indexes tab.

i. Locate the list of Suffix Actions and select Regenerate Indexes.
The Index Regeneration panel opens.

j. In the Index Regeneration panel, click Add All.
All the listed attributes move from the Available list to the Selected list.

k. Click OK.
The indexes are regenerated.

l. Wait for the regeneration process to complete and click Close.

Taking a Snapshot of the Module

When you have completed deploying the directory service module of the reference configuration, and before you move on to the next module, it is good practice to take a snapshot of the data in the Directory Server instance. By exporting `ds-inst-`*ds1*, you preserve the current state of your deployment in case you subsequently need to roll back directory information to this point in the reference configuration deployment process. The directory serves as the repository for service and user configuration information and therefore changes as each reference configuration module is deployed.

▼ To take a snapshot of the directory on *ds1*

In this procedure you use the `db2ldif` command to export the directory to an `ldif` file. If you want to subsequently restore the directory, use an equivalent procedure with the `ldif2db` command.

1 On *ds1* change directory as follows:

```
# cd /var/opt/SunWdsee/ds-inst-ds1
```

2 Stop the Directory Server instance.

```
# ./stop-slapd
```

3 Export the current state of the *pstest* directory to an `ldif` file.

```
# ./db2ldif -n pstest
```

The output should resemble the following:

```
ldiffile: /var/opt/SunWdsee/ds-inst-ds1/ldif/2008_05_20_140750.ldif
[20/May/2008:14:07:56 +0100] - export pstest: Precessed 1000 entries (26%)
...
[20/May/2008:14:08:02 +0100] - export pstest: Precessed 4165 entries (100%)
```

4 Rename the `ldif` file to something meaningful.

```
# mv /var/opt/SunWdsee/ds-inst-ds1/ldif/2008_05_20_140750.ldif
/var/opt/SunWdsee/ds-inst-ds1/ldif/ds_module_complete.ldif
```

5 Restart the Directory Server instance.

```
# ./start-slapd
```

◆ ◆ ◆ CHAPTER 5

Implementation Module 2: Access Manager With Session Failover on Application Server

This chapter provides an overview of the Access Manager service module in Figure 2–2 and documents the tasks required to implement it. The chapter includes the following sections:

Overview of the Access Manager Service Module

The Access Manager service module of the reference configuration's deployment architecture illustrated in Figure 2–2. The module consists of two instances of Sun Java System Access Manager running on two different computers. The module makes use of a hardware load balancer that is configured to provide service failover capability between the two Access Manager instances. All requests for Access Manager services are addressed to the virtual service name and IP address of the load balancer. The load balancer directs each request to one of the two Access Manager instances.

This module implements Access Manager session failover. When a user logs in, the load balancer routes the login request to one of the Access Manager instances, which authenticates the user and creates a session object. Subsequent requests from the user are directed to the same Access Manager instance.

If an Access Manager instance fails, the system recovers as follows:

- *Service Failover*. Subsequent requests are routed by the load balancer to the other Access Manager instance.

- *Access Manager Session Failover*. The new Access Manager instance retrieves session information from an Access Manager session database, thus making the service failover transparent to the user. The session failover mechanism is designed to revert back to the original Access Manager instance, if that instance subsequently comes back on line.

Access Manager's session failover mechanism is designed to be web container independent. It uses Message Queue and a session database to provide session failover between the two Access Manager instances.

The architecture of the Access Manager service module is shown in the following illustration.

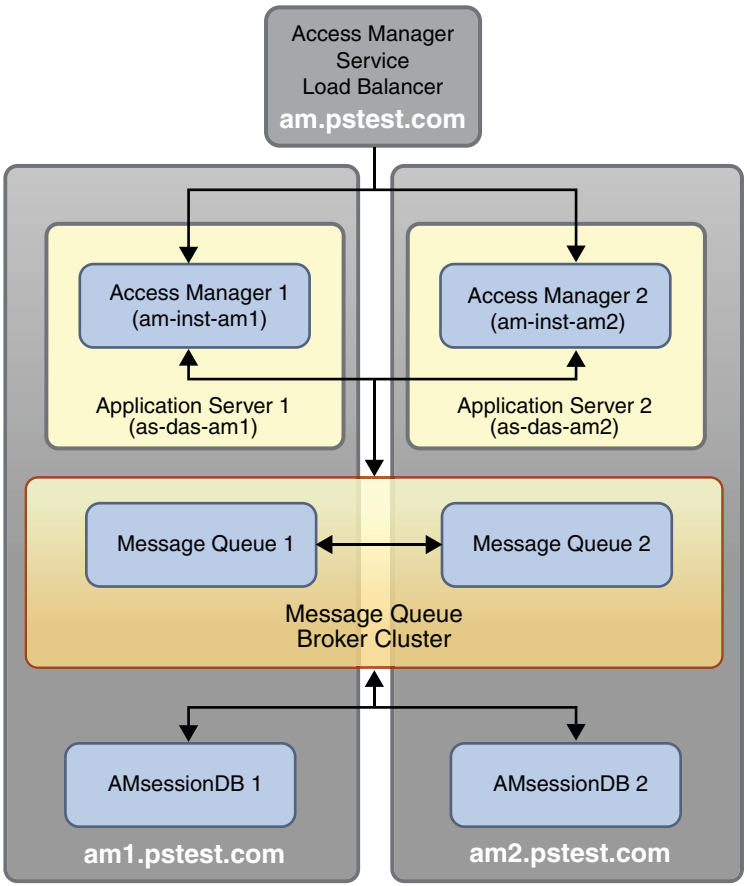

FIGURE 5–1 Access Manager Service Module

The Access Manager instances run in a web container that is provided by Sun Java System Application Server. Each Access Manager instance runs in the Domain Administration Server (DAS) instance of its respective computer. A Message Queue broker cluster, consisting of one Message Queue broker on each computer, is used by Access Manager to write session information to (and retrieve session information from) an Access Manager session database, which is replicated on each computer. The broker cluster and replicated session database are meant to avoid a single point of failure.

The Message Queue brokers and session database instances can reside on different computers from the Access Manager instances. However, it is simpler to set up the failover mechanism locally.

The general approach to implementing this module is to first set up Access Manager on each computer. In doing so, the Java ES installer is run in Configure Now mode to install and configure Application Server, Message Queue, and Access Manager. Following these procedures, load balancing is implemented to provide Access Manager service failover and then Access Manager session failover is set up.

This module can be scaled horizontally by adding an additional computer like *am2* and its respective components, and following the instructions in this chapter that apply to *am2*. However, the procedures for implementing Access Manager session failover might require some adjustment.

Note – The procedures in this chapter use the host names, domain name, and IP addresses shown in Figure 3–1 and Figure 5–1. However, you must map these host names, domain name, and IP addresses to equivalent names and addresses in your environment. For this reason, the procedures in this chapter show host names, domain name, and IP addresses as variables.

Setting Up Access Manager on *am1*

This task consists of the following procedures:

- Install Access Manager on *am1*
- Verify Access Manager on *am1*

▼ To Install Access Manager on *am1*

This procedure assumes that you are installing Access Manager on Solaris 10 8/07 OS or later version. Hence, no operating system patches need to be installed. The Java ES installer evaluates the state of the operating system and indicates if you need to install a patch. If you are using versions of the operating system older than Solaris 10 8/07 OS, it is better to install any required patches before you begin the actual Access Manager installation procedure.

The following procedure runs the Java ES installer without saving a state file. You can choose to run the installer and capture your input in a state file (`-saveState` *state-filename*). You could then use the state file to re-create the installation if, for example, you needed to reinstall Access Manager.

1 Download the Java ES software distribution to *am1*.

The procedure is documented in "To Download the Software Distribution" on page 185.

2 Log in as root or become superuser.

```
# su -
```

3 Start the Java ES installer.

```
# cd /portdist_71u2/Solaris_sparc
```

```
# ./installer
```

This procedure uses the GUI installer. The installer can also be run in text mode by using the `-nodisplay` option.

The Welcome panel opens.

4 In the Welcome panel, click Next.

The Software License Agreement panel opens.

5 In the Software License Agreement Panel, review the license terms and click Yes, Accept License.

The Choose Software Components panel opens.

6 In the Choose Software Components panel, select the following components:

- Application Server Enterprise Edition 8.2
  - Domain Administration Server
  - Application Server Node Agent (Not needed for this module, but can be installed for possible future use)
  - Command-Line Administration Tool
  - Sample Application (Not needed for this module, but can be installed for possible future use)
- High Availability Session Store 4.4 (Not needed for this module, but is a dependency of Application Server that is automatically selected and cannot be unselected)
- Java DB 10.1 (Not needed for this module, but is a dependency of Application Server that is automatically selected and cannot be unselected)
  - Java DB Client
  - Java DB Server

- Access Manager 7.1
  - Access Manager Core Services
  - Access Manager Administration Console
  - Common Domain Services for Federation (Not needed for this module, but can be installed for possible future use)
  - Access Manager SDK
  - Access Manager Session Failover Client
- Message Queue 3.7 UR1
- Install Multilingual Package(s) for all selected components (selected automatically, but optional if using English)

Also, unselect Directory Server Enterprise Edition 6.2 if it is automatically selected.

7 Click Next.

The Dependency Warning panel opens.

8 In the Dependency Warning panel, choose Use Directory Server Installed on a Remote Machine and click OK.

The installer evaluates the Java SE Software Development Kit on the computer and determines if an upgrade is required. On a fresh copy of Solaris 10 8/07 OS, an upgrade is needed, and the Java SE Software Development Kit Upgrade Required panel opens.

9 In the Java SE Software Development Kit Upgrade Required panel, select Automatic Upgrade to the Version Included with the Installer and click Next.

The installer evaluates the Java ES shared components on the computer and determines if any upgrades are required. On a fresh copy of the Solaris 10 8/07 OS, shared component upgrades are needed, and the Shared Components Upgrades Required panel opens.

10 In the Shared Components Upgrades Required panel, click Next.

The installer upgrades the shared components. The Specify Installation Directories panel opens.

11 In the Specify Installation Directories panel, type the following values and click Next.

Input Field	Value
Access Manager	`/opt`
Application Server	`/opt/SUNWappserver`
Application Server Data and Configuration	`/var/opt/SUNWappserver`

The installer checks the system, and the System Check panel opens.

12 In the System Check panel, evaluate the results of the system check.

If the system check is favorable, click Next.

The Choose a Configuration Type panel opens.

13 In the Choose a Configuration Type panel, select Configure Now and click Next.

The Custom Configuration Panel opens.

14 In the Custom Configuration Panel, note the following message and click Next.

```
The following component products cannot be configured during installation:
Java DB
Click Next to configure the other components.
```

The Specify Administrator Account Preferences panel opens.

15 In the Specify Administrator Account Preferences panel, type the following values and click Next.

If you are using different administrator accounts for different services (Java ES products), select the corresponding checkbox in the panel (see "Administrator Account Specification" on page 58).

Input Field	Value
Administrator User ID	`admin`
Administrator Password	*app-server-admin-password*

The Common Server Settings panel opens.

16 In the Common Server Settings panel, type the following values and click Next.

Input Field	Value
HostName	*am1*
DNS Domain Name	*pstest.com*
Host IP Address	*10.0.2.1*
System User	root
System Group	root

The Application Server: High Availability Session Store (HADB) panel opens.

17 In the Application Server: High Availability Session Store (HADB) panel, type the following values, uncheck the Automatically Start HADB checkbox, and click Next.

Input Field	Value
HADB Management Port	1862
HADB Resource Directory	`/var/opt`
HADB Administrator Group	root

The Application Server: Domain Administration Server panel opens.

18 In the Application Server: Domain Administration Server panel type the following values and click Next.

Input Field	Value
Admin Port	4849
JMX Port	8686
HTTP Port	80
HTTPS Port	8181
Master Password	*app-server-master-password*

The Access Manager: Specify Configuration Information panel opens.

19 In the Access Manager: Specify Configuration Information panel, type the following values and click Next.

Also, record the values that you specify in this panel. They will be needed when installing other components in the reference configuration.

Input Field	Value
Install Type	`Legacy Mode`
Administrator User ID	`amadmin` (This administrator account is different from the `admin` account in step 15, Administrator Account Preferences panel.)
Administrator Password	*access-manager-admin-password*
LDAP User ID	`amldapuser`
LDAP Password	*access-manager-LDAP-password*
Password Encryption Key	*password-enc-key* (This password must be at least 12 characters. A value is proposed by the installer.)

The Access Manager: Choose Deployment Container panel opens.

20 In the Access Manager: Choose Deployment Container panel, type the following values and click Next.

Input Field	Value
Sun Java System Application Server	Yes

The Access Manager: Specify Sun Java System Application Server panel opens.

21 In the Access Manager: Specify Sun Java System Application Server panel type the following values and click Next.

Input Field	Value
Secure Server Instance Port	`No`
Secure Administration Server Port	`Yes`

The Access Manager: Specify Web Container for Running Access Manager Services panel opens.

22 In the Access Manager: Specify Web Container for Running Access Manager Services panel, type the following values and click Next.

Input Field	Value
Host Name	*am1.pstest.com*
Services Deployment URI	`amserver`
Common Domain Deployment URI	`amcommon`
Cookie Domain	*pstest.com*
Password Deployment URI	`ampassword`
Console Protocol	`HTTP`

The Access Manager: Choose Access Manager Console panel opens.

23 In the Access Manager: Choose Access Manager Console panel, type the following values and click Next.

Input Field	Value
Administration Console	`Deploy New Console`

Input Field	Value
Console Deployment URI	`amconsole`
Console Host Name	*am1.pstest.com*
Console Port	`80`

The Access Manager: Specify Directory Server Information panel opens.

24 In the Access Manager: Specify Directory Server Information panel, type the following values and click Next.

Input Field	Value
Directory Server Host	*ds.pstest.com* (The logical service name was created when the directory service module's load balancer was configured.)
Directory Server Port	`389`
Access Manager Directory Root Suffix	*dc=pstest,dc=com* (The suffix was defined when Directory Server was installed.)
Directory ManagerDN	`cn=Directory Manager`
Directory Manager Password	*directory-manager-password* (This password was defined when Directory Server was installed.)

The Access Manager: Specify Directory Server Data panel opens.

25 In the Access Manager: Specify Directory Server Data panel, type the following values and click Next.

Input Field	Value
Is Directory Server Provisioned With User Data?	`No`

The Ready to Install panel opens.

26 In the Ready to Install panel, indicate whether you want to open the software registration window during installation.

This panel enables you to register the components that you have selected for installation with Sun Connection. Sun Connection is a Sun-hosted service that helps you track, organize, and maintain Sun hardware and software. For example, Sun Connection can inform you of the latest available security fixes, recommended updates, and feature enhancements.

If you choose to register, information about the installation is sent to the Sun Connection database. You can also register at a later date, after installation has been completed.

27 Click Install.

The installer copies files to the computer, modifies configuration files based on the values provided, and deploys Access Manager to the Application Server's Domain Administration Server (DAS) instance.

28 When the installation is complete, review the installation in the Summary field.

29 Click Exit to exit the installer.

30 Check the installation log files for any installation errors.

```
# cd /var/sadm/install/logs
```

```
# egrep -i 'fail|error' Java*
```

▼ To Start and Verify Access Manager on *am1*

The following procedure confirms that Access Manager has been installed by starting the Access Manager Console login page.

1 Check that Application Server is running.

```
# netstat -an | grep 80
```

2 If Application Server is not running, start it.

```
# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

3 Verify the operation of Access Manager on *am1*.

a. Start a browser.

b. Open the Access Manager Console login page:

`http://`*am1.pstest.com*`/amconsole`

The login page opens.

c. Log in to the Access Manager Console by typing the following values and clicking Login.

Input Field	Value
User ID	`amadmin`

Input Field	Value
Password	*access-manager-admin-password*

The Access Manager Console opens, which confirms that Access Manager is deployed and running in the web container.

Setting Up Access Manager on *am2*

This task consists of the following procedures:

- Install Access Manager on *am2*
- Verify Access Manager on *am2*

▼ To Install Access Manager on *am2*

- **Repeat the procedure that appears in "To Install Access Manager on *am1*" on page 87, using the same parameter values, except for the following:**
  - Replace all occurrences of *am1* with *am2*.
  - For the IP address, replace *10.0.2.1.* with *10.0.2.2*.
  - When you see the prompts in the following table, type the values in the table. These values were not requested in "To Install Access Manager on *am1*" on page 87 because no data was in the directory at that time.

Input Field	Value for *am2* Installation
Is Directory Server Provisioned With User Data	`Yes`
Organization Marker Object Class	`sunISManagedOrganization`
Organization Naming Attribute	`o`
User Marker Object Class	`inetorgperson`
User Naming Attribute	`uid`

▼ To Start and Verify Access Manager on *am2*

- **Repeat the procedure in "To Start and Verify Access Manager on *am1*" on page 94, except for the following:**

  When you are prompted for input values, type the values that apply to *am2*. For example, when you open the Access Manager Console, type the URL with *am2* instead of *am1*.

Implementing Load Balancing for the Access Manager Service

This task consists of the following procedures:

- Configure the Load Balancer for the Access Manger Service
- Configure Access Manager as a Load-balanced Server Site
- Configure Access Manager Instances for Load Balancing
- Verify Load Balancing for the Access Manager Service

▼ To Configure the Load Balancer for the Access Manger Service

This procedure describes how to configure the Access Manager service load balancer (*am.pstest.com* at IP address *10.0.3.10*). The steps are relatively generic; the details depend on the load balancer you are using.

1 Populate the load balancer's Hosts Table.

Add the IP address for *am1.pstest.com* and *am2.pstest.com* to the load balancer's hosts table.

2 Populate the load balancer's Real Service Table.

Add the real services for *am1.pstest.com* and *am2.pstest.com*. A real service is identified by its IP address and port. Add *10.0.2.1*`:80` and *10.0.2.2*`:80`

3 Populate the load balancer's Service Group Table.

Add the service group for Access Manager services. The service groups are sets of the real services that you defined in Step 2. The real services in the group must be capable of fulfilling the same type of request. The load balancer will distribute requests among the real services in the service group. When you define the service group for *am.pstest.com*, you add the real services that specify the Access Manager instances, *10.0.2.1*`:80` and *10.0.2.2*`:80`.

4 Populate the load balancer's Virtual IP Table.

A virtual service definition includes the outward facing IP address and the port at which the load balancer accepts requests for a service, as well as the service group that you specified in Step 3, which actually handles the requests. The load balancer will accept requests at the virtual service address and distribute them among the service group. The virtual service definition for the Access Manager service should be *am.pstest.com*, with the virtual IP address of *10.0.3.10*`:80`, and with the service group consisting of the computers *am1.pstest.com* and *am2.pstest.com*.

5 Configure the load balancer to use Layer-7 (HTTP layer) load balancing.

6 **Configure the load balancer with a scheduling type of either least connections or round robin.**

Both scheduling types initially distribute the connections evenly between the Access Manager instances. Both scheduling types keep the connections evenly distributed if the connections are restarted.

7 **Configure the load balancer for sticky routing based on a server-side cookie.**

In the case of Access Manager services, a session token (or cookie) is provided when a user is first authenticated. The load balancer must be configured to identify this session token (`amlbcookie`) in each request and to route all requests within the same user session to the same Access Manager instance.

You typically accomplish the configuration of session persistence by establishing forwarding rules based on the HTTP Request and Response header predicate `COOKIE`, as in the following example:

```
{COOKIE has amlbcookie eq 01}
```

where `01` is the Access Manager instance ID.

If the load balancer is not configured to stick sessions to the instance that creates them, but instead routes them randomly, the instances that receive subsequent requests will maintain shadow sessions. The instances in the module will communicate among themselves about session changes. Maintaining the shadow sessions requires more memory and decreases system performance. It also generates more network traffic among the Access Manager instances in order to keep the session caches synchronized.

8 **Configure the health-check settings for the load balancer.**

The recommended settings are specified in Table 3–5.

▼ To Configure Access Manager as a Load-Balanced Server Site

To implement Access Manager session persistence, you define what is called a load-balanced Access Manager *site*. Once a site is defined and configured, Access Manager automatically sets the value of a cookie, with the default name of `amlbcookie`, to be equal to the instance ID of the Access Manager instance that first authenticates an Access Manager client request.

The instance ID is defined at the moment the instance is created. For example, the Access Manager instance created on *am1.pstest.com*will have an instance ID equal to `01` because this is the first instance created.

1 **Start a browser.**

2 Go to the Access Manager Console login page.

`http://`*`am1.pstest.com`*`/amconsole`

The Access Manager Console login page opens.

3 Log in to the Access Manager Console by typing the following values and clicking Login.

Field	Input Value
User ID	`amadmin`
Password	*access-manager-admin-password*

The Access Manager Console opens.

4 Modify the organization properties as follows:

a. In the Identity Management tab, select the Organizations view.

The Organizations section is displayed in the right pane.

b. In the Organizations section, locate the General Properties and the list of Organization Aliases.

c. In the list of Organization Aliases, add the following value:

am.pstest.com

d. Click Save.

The following message is displayed: "The organization properties have been saved."

5 In the Access Manager Console, navigate to the load balancer setup information.

a. Click the Service Configuration tab.

The Service Configuration tab displays a list of services that you can configure.

b. In the Service Configuration tab, locate the Service Name list in the left pane.

c. In the Service Name list, locate Platform. Click the arrow to the right of Platform.

The configuration options for the Platform service are displayed in the right pane.

d. In the list of configuration options, locate the section for Global options.

e. Locate the entry field for the Site List and type the following value:

`http://`*`am.pstest.com`*`:80|10`

where `10` is an arbitrary site number.

f. **Click Add.**
The load balancer's name is added to the Site List.

g. **Click Save.**
The following message is displayed: "The service properties have been saved."

6 **Under the Site List, locate the Server List, and in the server list locate the Instance Name.**
Do the following:

a. **Add** *am2.pstest.com*`:80|02|10`

b. **Add** *am1.pstest.com*`:80|01|10`

c. **Remove** *am1.pstest.com*`:80|01`

d. **Remove** *am2.pstest.com*`:80|02`

e. **Click Save.**
Your changes are saved.

▼ To Configure Access Manager Instances for Load Balancing

The `AMconfig.properties` file for each Access Manager instance must be configured to recognize the load balancer as the virtual Access Manager service login host.

1 **Modify the login URL property for the Access Manager instance on** *am1*.

a. **On** *am1*, **open the** `AMconfig.properties` **file in a text editor.**
The file is located at:

```
/etc/opt/SUNWam/config/AMConfig.properties
```

b. **Modify the login URL property:**

```
com.sun.identity.loginurl=http://am.pstest.com:80/amserver/UI/Login
```

c. **Restart the Access Manager instance.**

```
# /opt/SUNWappserver/appserver/bin/asadmin stop-domain
# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

2 **Modify the login URL property for the Access Manager instance on *am2*.**

Use the same procedure as in Step 1, except on *am2*.

▼ To Verify Load Balancing for the Access Manager Service

This procedure verifies the following:

- that you can interact with Access Manager instances through the load balancer
- that the load balancer provides service failover when an Access Manager instance fails

1 **Start the Access Manager instances on *am1* and *am2*, if they are not already running.**

2 **Start a browser.**

3 **Go to the Access Manager Console login page by using the load balancer URL**

`http://`*am.pstest.com*`/amconsole`

The Access Manager Console login page opens.

4 **Log in to the Access Manager Console by typing the following values and clicking Login.**

Input Field	Value
User ID	`amadmin`
Password	*access-manager-admin-password*

The Access Manager Console opens, which confirms that the load balancer has routed the login request to one of the Access Manager instances.

5 **Determine which Access Manager instance handled the login request in Step 4.**

a. **Click on the Current Sessions tab.**

The left panel shows both Access Manager instances: `http://`*am1.pstest.com*`:80` and `http://`*am2.pstest.com*`:80`

b. **Check for an `amadmin` session on each instance.**

You can display the sessions existing on each instance by clicking on the small triangle adjacent to each.

c. Note the instance that owns the `amadmin` session.

6 Simulate a failure of the Access Manager instance that was noted in Step 5.

Failure of an Access Manager instance can result from a computer failure, a software failure, or a network failure. The method employed for simulating a failure in this service failover verification procedure is to shut down the Access Manager instance (by shutting down the Application Server instance in which it runs). Additionally, you could also simulate failure by unplugging the network cable or disabling the interface.

Run the following command on the computer (*am1* or *am2*) hosting the instance identified in Step 5.

```
# /opt/SUNWappserver/appserver/bin/asadmin stop-domain
```

7 In the Access Manager Console, click on the Identity Management tab.

If service failover has succeeded, the login page should once again be displayed, indicating that content is now being served from the remaining Access Manager instance.

8 Log in once again, as directed in Step 4.

9 Confirm the service failover.

a. Click on the Current Sessions tab.

The left panel shows both Access Manager instances: `http://`*am1.pstest.com*`:80` and `http://`*am2.pstest.com*`:80`

b. Check for an `amadmin` session on the failed instance.

You can display the existing sessions by clicking on the small triangle adjacent to the instance. In this case, an "Failed to get the valid sessions..." error message should be displayed (the instance is shut down).

c. Check for an `amadmin` session on the remaining instance.

In this case, an `amadmin` session should be displayed.

10 Recover the simulated failure of your original Access Manager instance.

Run the following command on the computer (*am1* or *am2*) on which the Application Server instance was shut down in Step 6.

```
# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

Setting Connection Timeouts for Access Manager

Access Manager connections to directory services can inadvertently time out with negative consequences if idle timeout values are not set correctly with respect to Directory Server (or the directory service load balancer).

This task consists of the following procedures:

- Configure the Connection Timeout of the Directory Service
- Configure the Persistent Search Timeout for Access Manager

▼ To Configure the Connection Timeout of the Directory Service

Access Manager uses a pool of open connections to access the directory service. If these connections remain idle for longer than the Directory Server's idle timeout period, the connections will be closed on the Directory Server end, and Access Manager will restart them.

However, if a load balancer (or firewall) is located between Access Manager and Directory Server, the idle timeout of the load balancer (or firewall) might close the connection before Directory Server does. Some load balancers (or firewalls) do not close the connection cleanly, and Access Manager is not notified of the closure. In this case, connections in the pool can be exhausted, requiring a restart of Access Manager. In addition, when a connection is not closed cleanly by a load balancer (or firewall), the Directory Server might not close the socket, causing the open sockets to accumulate.

To avoid this set of circumstances, the Directory Server's idle timeout for Access Manager connections must be less than the idle timeout interval of the directory service load balancer (or firewall).

- **Set the value of the Directory Server idle timeout to less than that of the directory service load balancer.**

  Run the following command on *ds1*:

```
# ldapmodify -h ds1.pstest.com -p 389 -D "cn=Directory Manager" <<EOF

dn: cn=amldapuser,ou=DSAME Users, dc=example,dc=com

changetype: modify

add: nsIdleTimeout

nsIdleTimeout: timeout-value

EOF
```

  where *timeout-value* is a value in seconds less then the load balancer's idle timeout.

When prompted, type the *directory-manager-password*.

▼ To Configure the Persistent Search Timeout for Access Manager

Access Manager uses Directory Server persistent searches to obtain asynchronous notifications of changes on the Directory Server. The persistent search mechanism provides an active channel through which information about changes that occur can be communicated back to Access Manager.

Each active, persistent search requires that an open TCP connection be maintained between Access Manager and Directory Server. If the persistent search connections are made through a load balancer (or firewall), then these connections are subject to being closed by the load balancer (or firewall). For some load balancers (and firewalls), the connection is not closed cleanly. As a result, the persistent searches are not automatically restarted, and change notifications are interrupted until a persistent search connection is re-established.

This interruption in persistent searches can be prevented by configuring the Access Manager idle timeout for persistent search to be shorter than the TCP idle timeout of the directory service load balancer (or firewall). Hence, persistent searches are restarted before the load balancer (or firewall) can time out.

1 **On *am1*, open the `AMconfig.properties` file in a text editor.**

The file is located at:

```
/etc/opt/SUNWam/config/AMconfig.properties
```

2 **Locate the persistent search timeout property:**

```
com.sun.am.event.connection.idle.timeout
```

This property specifies the timeout value in minutes after which persistent searches will be restarted. A value of "`0`" (the default) indicates that the connection does not time out, so that searches will not be restarted.

3 **Set the persistent search timeout value as follows and save the change:**

```
com.sun.am.event.connection.idle.timeout=timeout-value
```

where *timeout-value* is a value in minutes less than the load balancer's idle timeout value.

4 **Restart the Access Manager instance, `am-inst-`*am1* on *am1*.**

```
# /opt/SUNWappserver/appserver/bin/asadmin stop-domain

# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

5 **Repeat Steps 1–4 on *am2*.**

Replace all occurrences of *am1* with *am2* in these steps.

Implementing Session Failover for Access Manager

The implementation of session failover involves establishing a persistence layer that uses Message Queue to write session information to a replicated Access Manager session database (see Figure 5–1). If the Access Manager instance that owns a session fails, the session information is retrieved and passed to another Access Manager instance.

You implement session failover by first creating what is called a *secondary configuration instance*. The secondary configuration instance specifies values that are needed to store and recover persistent session information. You then use Access Manager utilities to set up the session persistence database.

Note – Access Manager provides the `amsfoconfig` script, which performs some of the procedures that are needed to implement service failover and session failover. However, the `amsfoconfig` script is not used in this reference configuration because service failover and session failover are independent functions that are best implemented and verified separately.

- Create a Secondary Configuration Instance
- Configure Session Failover on *am1*.
- Configure Session Failover on *am2*
- Verify Session Failover

▼ To Create a Secondary Configuration Instance

1 **Log in to the Access Manager Console.**

a. **Confirm that Application Server on *am1* is running.**

```
# /opt/SUNWappserver/appserver/bin/asadmin list-domains --user admin
--terse=true
```

When prompted, type the *app-server-admin-password*.

If the Application Server is not running, then start it.

```
# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

b. **Start a browser.**

c. Go to the Access Manager login page:

`http://`*am.pstest.com*`/amconsole`

The Access Manager Console login page opens.

d. Log in to the Access Manager Console by typing the following values and clicking Login.

Input Field	Value
User ID	`amadmin`
Password	*access-manager-admin-password*

The Access Manager Console opens.

2 In the Access Manager Console main page, click the Service Configuration tab.

3 In the Service Configuration tab, navigate to the New Secondary Configuration dialog box.

a. Locate the Service Name pane on the left side of the tab.

b. In the Service Name pane, scroll down and click the arrow to the right of Session.

The Session detail is displayed in the right pane.

c. In the Session detail pane, locate the Secondary Configuration Instances and click New.

The New Secondary Configuration dialog box opens.

4 In the New Secondary Configuration dialog box, type the following values and click Add.

Input Field	Value
Instance Name	`http://`*am.pstest.com*`:80` (this is the URL for the load balancer)
Session Store User	*am-svr-usr* (the user name you establish in the following procedure for connecting to the Message Queue Server)
Session Store Password	*am-svr-usr-password* (the password you set in the following procedure)
Maximum Wait Time	`5000`
Database URL	*am1.pstest.com*`:7777,`*am2.pstest.com*`:7777` (the list of broker addresses in the Message Queue broker cluster)

The Secondary Configuration dialog box closes.

5 **Click Save to save your changes.**

▼ To Configure Session Failover on *am1*

1 **Shut down the Access Manager instance on *am1*.**

You shut down the Access Manager instance by shutting down the Application Server instance in which it runs.

```
# /opt/SUNWappserver/appserver/bin/asadmin stop-domain
```

The response should resemble the following:

```
Domain domain1 stopped.
```

2 **Add the required Java Archive (JAR) files to the web container classpath.**

a. **Start a browser.**

b. **Go to the following URL:**

`https://`*am1*`.pstest.com:4849`

The Application Server login page opens.

c. **Log in to the Application Server Admin Console by typing the following values and clicking Login.**

Input Field	Value
User ID	`admin`
Password	*app-server-admin-password*

The Application Server Admin Console opens.

d. **Click on the small triangle next to Configurations on the Common Tasks panel.**

The configurations are expanded.

e. **Click on the small triangle next to `server-config`.**

The `pscluster` configuration is expanded.

f. **Click on JVM Settings.**

The frame on the right shows the configuration options.

g. In the right frame, select the Path Settings tab.

The JVM Classpath Settings panel opens.

h. Add `/usr/share/lib/imq.jar` and `/usr/share/lib/jms.jar` to the `Classpath Suffix` list.

i. Click Save.

3 Create a Message Queue user for Access Manager session failover.

This user will be used internally to send and retrieve session information. To use the `imqusermgr` utility in the following steps, you must first create a default user repository, which is done automatically the first time you start the Message Queue broker.

a. Start the Message Queue broker to be used for session failover.

```
# bash
```

This opens the bash shell, which supports background processes.

```
# /usr/bin/imqbrokerd -name aminstance -port 7777 &
```

Note – Before using port 7777, check that it is not being used by some other process.

The output should resemble the following:

```
[25/Oct/2007:16:17:00 MEST]
================================================================================
Sun Java(tm) System Message Queue 3.7
Sun Microsystems, Inc.
Version:  3.7 UR2  (Build 3-b)
Compile:  Mon May  7 22:37:30 PDT 2008
Copyright (c) 2007 Sun Microsystems, Inc.  All rights reserved.
SUN PROPRIETARY/CONFIDENTIAL.  Use is subject to license terms.
This product includes code licensed from RSA Data Security.
================================================================================
Java Runtime: 1.5.0_12 Sun Microsystems Inc. /usr/jdk/instances/jdk1.5.0/jre
[25/Oct/2007:16:17:00 MEST]    IMQ_HOME=/
[25/Oct/2007:16:17:00 MEST] IMQ_VARHOME=/var/imq
[25/Oct/2007:16:17:00 MEST] SunOS 5.10 sparc am1(24 cpu) root
[25/Oct/2007:16:17:00 MEST] Max file descriptors: 65536 (65536)
[25/Oct/2007:16:17:00 MEST] Java Heap Size: max=174784k, current=35328k
[25/Oct/2007:16:17:00 MEST] Arguments:
[25/Oct/2007:16:17:00 MEST] [B1060]: Loading persistent data...
[25/Oct/2007:16:17:00 MEST] Using built-in file-based persistent store:
   /var/imq/instances/aminstance/
[25/Oct/2007:16:17:01 MEST] [B1039]: Broker "aminstance@am1:7777" ready.
```

b. Change the default Message Queue administrative user password.

```
# /usr/bin/imqusermgr update -i aminstance -u admin -p MQ-admin-psssword
```

The response should resemble the following:

```
User repository for broker instance: aminstance
Are you sure you want to update user admin? (y/n) y
User admin successfully updated.
```

c. Add a new Message Queue user to be used for Access Manager session failover..

```
# /usr/bin/imqusermgr add -i aminstance -u am-svr-usr -p am-svr-usr-password
```

The response should resemble the following:

```
User repository for broker instance: aminstance
User amSvrUsr successfully added.
```

d. Delete the default guest user.

```
# /usr/bin/imqusermgr update -i aminstance -u guest -a false
```

The response should resemble the following:

```
User repository for broker instance: aminstance
Are you sure you want to update user guest? (y/n) y
User guest successfully updated.
```

e. Shut down the Message Queue broker.

```
# imqcmd shutdown bkr -b am1:7777 -u admin
```

When prompted, type the *MQ-admin-password*.

4 Check the installation directories in the `amsessiondb` file.

a. In a text editor, open the following file:

```
/opt/SUNWam/bin/amsessiondb
```

b. If you have installed Access Manager, JDK, or Message Queue in non-default directories, you must make the appropriate changes to the `amsessiondb` file.

5 Generate an encrypted password file.

```
# /opt/SUNWam/bin/amsfopassword -f /opt/SUNWam/.password -e am-svr-usr-password
os.name=SunOS
SUCCESSFUL
```

6 Edit the `amsfo.conf` file.

a. Open the `amsfo.conf` file in a text editor.

The file, which is used to configure Access Manager session failover, is located at:

`/opt/SUNWam/lib/amsfo.conf`

b. Type the following values:

Parameter	Value
AM_HOME_DIR	`/opt/SUNWam`
AM_SFO_RESTART	`true`
CLUSTER_LIST	*am1.pstest.com*`:7777,`*am2.pstest.com*`:7777`
DATABASE_DIR	`/tmp/amsession/sessiondb`
LOG_DIR	`/tmp/amsession/logs`
START_BROKER	`true`
BROKER_INSTANCE_NAME	`aminstance`
BROKER_PORT	`7777`
USER_NAME	*am-svr-usr*
lbServerPort	`80`
lbServerProtocol	`http`
lbServerHost	*am.pstest.com*`:80`
SiteID	`10`
JAVA_HOME	`/usr/jdk/entsys-j2se`

7 Run the `amsfo` script:

```
# /opt/SUNWam/bin/amsfo start
```

The script starts the Message Queue broker on *am1*, the Access Manager session database on *am1*, and initializes the Message Queue and Access Manager session database clients needed to implement session persistence.

8 Verify that the Message Queue connections are working.

Open the following log file:

`/tmp/amsession/logs/amsessiondb.log`

Check for errors in the file.

9 Restart the Access Manager instance on *am1*.

You start the Access Manager instance by starting the Application Server instance in which it runs.

```
# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

▼ To Configure Session Failover on *am2*

- **Repeat the procedure that appears in "To Configure Session Failover on *am1*" on page 106, except for the following:**

  Replace all occurrences of *am1* with *am2*.

▼ To Verify Session Failover

In this procedure, a user logs in to the Access Manager Console, and you determine the Access Manager instance that is handling the Access Manager Console request. You then simulate a failure of that instance, have the user make another Access Manager Console request, and note which Access Manager instance is handling the second request. If session failover is working properly, the Access Manger service fails over to the failover Access Manager instance without the user having to log in a second time.

1 Log in to the Access Manager Console, if you are not already logged in.

a. Start a browser.

b. Go to the Access Manager Console login page by using the load balancer URL

`http://`*am.pstest.com*`/amconsole`

The Access Manager Console login page opens.

c. Log in to the Access Manager Console by typing the following values and clicking Login.

Input Field	Value
User ID	`amadmin`
Password	*access-manager-admin-password*

The Access Manager Console opens.

2 Determine which Access Manager instance handled the login request in Step 1.

a. Click on the Current Sessions tab.

The left panel shows both Access Manager instances: `http://`*am1.pstest.com*`:80` and `http://`*am2.pstest.com*`:80`

b. Check for an `amadmin` session on each instance.

You can display the sessions existing on each instance by clicking on the small triangle adjacent to each.

c. Note the instance that owns the `amadmin` session.

3 Simulate a failure of the Access Manager instance that was noted in Step 2.

Failure of an Access Manager instance can result from a computer failure, a software failure, or a network failure. The method employed for simulating a failure in this session failover verification procedure is to shut down the Access Manager instance (by shutting down the Application Server instance in which it runs). Additionally, you could also simulate failure by unplugging the network cable or disabling the interface.

Run the following command on the computer (*am1* or *am2*) hosting the instance identified in Step 2.

```
# /opt/SUNWappserver/appserver/bin/asadmin stop-domain
```

4 Perform another Console request.

For example, click the Identity Management tab, then click the *pstest* link.

If session failover is working correctly, the Console session will fail over to the other Access Manager instance and display a list of Organization Aliases in the right pane. The fact that you did not have to log in again confirms that session failover is working.

5 Confirm that your Access Manager Console session is now owned by the other Access Manager instance.

You can do this step by repeating Step 2 or by checking the access logs on your web containers.

6 Recover the simulated failure of your original Access Manager instance.

Run the following command on the computer (*am1* or *am2*) on which the Application Server instance was shut down in Step 3.

```
# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

Tuning Access Manager Instances

Access Manager performs best when the web container in which it runs is tuned to optimize Access Manager performance. So, you should tune the Application Server instances that host Access Manager. In this module, Access Manager runs in the Application Server's DAS instance.

In the portal service reference configuration, no other component or application besides Access Manager runs in the Application Server's DAS instance. However, if you run other components or applications in the same Application Server instance, then be aware that optimizing the Application Server instance for Access Manager might negatively impact the performance of other components.

▼ To Tune Application Server Instances for Access Manager

Tuning the Application Server instances that host Access Manager is performed by using the `amtune` utility. For additional information about `amtune`, see Part I, "Basic Performance Tuning," in *Sun Java System Access Manager 7.1 Performance Tuning and Troubleshooting Guide*.

Perform the following procedure on both *am1* and *am2*..

1 Change to the following directory:

```
# cd /opt/SUNWam/bin/amtune
```

2 Open the `amtune-env` file in a text editor.

3 Confirm, or if necessary, modify the following values and save the changes.

```
WEB_CONTAINER=AS8
ASADMIN_PORT=80
DOMAIN_NAME=pstest.com
AMTUNE_PCT_MEMORY_TO_USE=100
AMTUNEAMTU_MIN_PERM_SIZE_AS8=128
AMTUNE_MODE=REVIEW
```

Note – In a Solaris zones deployment, also set `AMTUNE_TUNE_OS=false`.

---

4 Run the `amtune` utility.

```
# ./amtune directory-manager-password app-server-admin-password
```

The utility proposes changes to optimize Access Manager performance.

5 **Review any suggested changes.**

If no undesired changes are proposed, continue to the next step.

6 **Modify the `amtune-env` file with the following value and save the change.**

```
AMTUNE_MODE=CHANGE
```

7 **Repeat Step 4.**

The utility runs a second time, but makes all the changes proposed in Step 4.

8 **Restart the computer.**

Restarting the computer will affect changes to the operating system.

9 **Start the Access Manager instance.**

```
# /opt/SUNWappserver/appserver/bin/asadmin start-domain --user admin domain1
```

When prompted, type the *app-server-admin-password*.

Taking a Snapshot of the Module

When you have completed deploying the Access Manager module of the reference configuration, and before you move on to the next module, it is good practice to take a snapshot of the data in the Directory Server instance. By exporting `ds-inst-`*ds1*, you preserve the current state of your deployment in case you subsequently need to roll back directory information to this point in the reference configuration deployment process. The directory serves as the repository for service and user configuration information and therefore changes as each reference configuration module is deployed.

▼ To take a snapshot of the directory on *ds1*

In this procedure you use the `db2ldif` command to export the directory to an `ldif` file. If you want to subsequently restore the directory, use an equivalent procedure with the `ldif2db` command.

1 **On *ds1* change directory as follows:**

```
# cd /var/opt/SunWdsee/ds-inst-ds1
```

2 **Stop the Directory Server instance.**

```
# ./stop-slapd
```

3 **Export the current state of the *pstest* directory to an `ldif` file.**

```
# ./db2ldif -n pstest
```

The output should resemble the following:

```
ldiffile: /var/opt/SunWdsee/ds-inst-ds1/ldif/2008_05_20_140750.ldif
[20/May/2008:14:07:56 +0100] - export pstest: Precessed 1000 entries (26%)
...
[20/May/2008:14:08:02 +0100] - export pstest: Precessed 4165 entries (100%)
```

4 Rename the `ldif` file to something meaningful.

```
# mv /var/opt/SunWdsee/ds-inst-ds1/ldif/2008_05_20_140750.ldif
/var/opt/SunWdsee/ds-inst-ds1/ldif/am_module_complete.ldif
```

5 Restart the Directory Server instance.

```
# ./start-slapd
```

◆ ◆ ◆ CHAPTER 6

Implementation Module 3: Portal Server With Portlet Session Failover on Application Server Cluster

This chapter provides an overview of the portal service module in Figure 2–2 and documents the tasks that are required to implement it. The chapter includes the following sections:

Overview of the Portal Service Module

The portal service module of the reference configuration's deployment architecture illustrated in Figure 2–2consists of two instances of Sun Java System Portal Server running on two different computers. The module makes use of a hardware load balancer that is configured to provide service failover capability between the two Portal Server instances. All requests for portal services are addressed to the virtual service name and IP address of the load balancer. The load balancer directs each request to one of the two Portal Server instances.

This module implements portlet session failover. When a user logs in and is authenticated by the Access Manager service, the load balancer routes the portal request to one of the Portal Server instances, which builds and returns a Portal Desktop to the user. As the user accesses various portal channels and portlet applications from the desktop, the requests from the user are routed to the same Portal Server instance. Portlet session state is maintained by the web container supporting the Portal Server instance.

If the Portal Server instance fails, the system recovers as follows:

- *Service Failover*. Subsequent requests are routed by the load balancer to the other Portal Server instance.
- *Portlet Session Failover*. The new Portal Server instance rebuilds the Portal Desktop by using the Access Manager session state and retrieves portlet session state from a high availability database, thus making the service failover transparent to the user.

Portlet session failover depends upon high availability mechanisms provided by the web container. Hence, portlet session failover for the portal service module is achieved by using an Application Server cluster and its High Availability Session Store (HADB).

The architecture of the portal service module is shown in the following illustration.

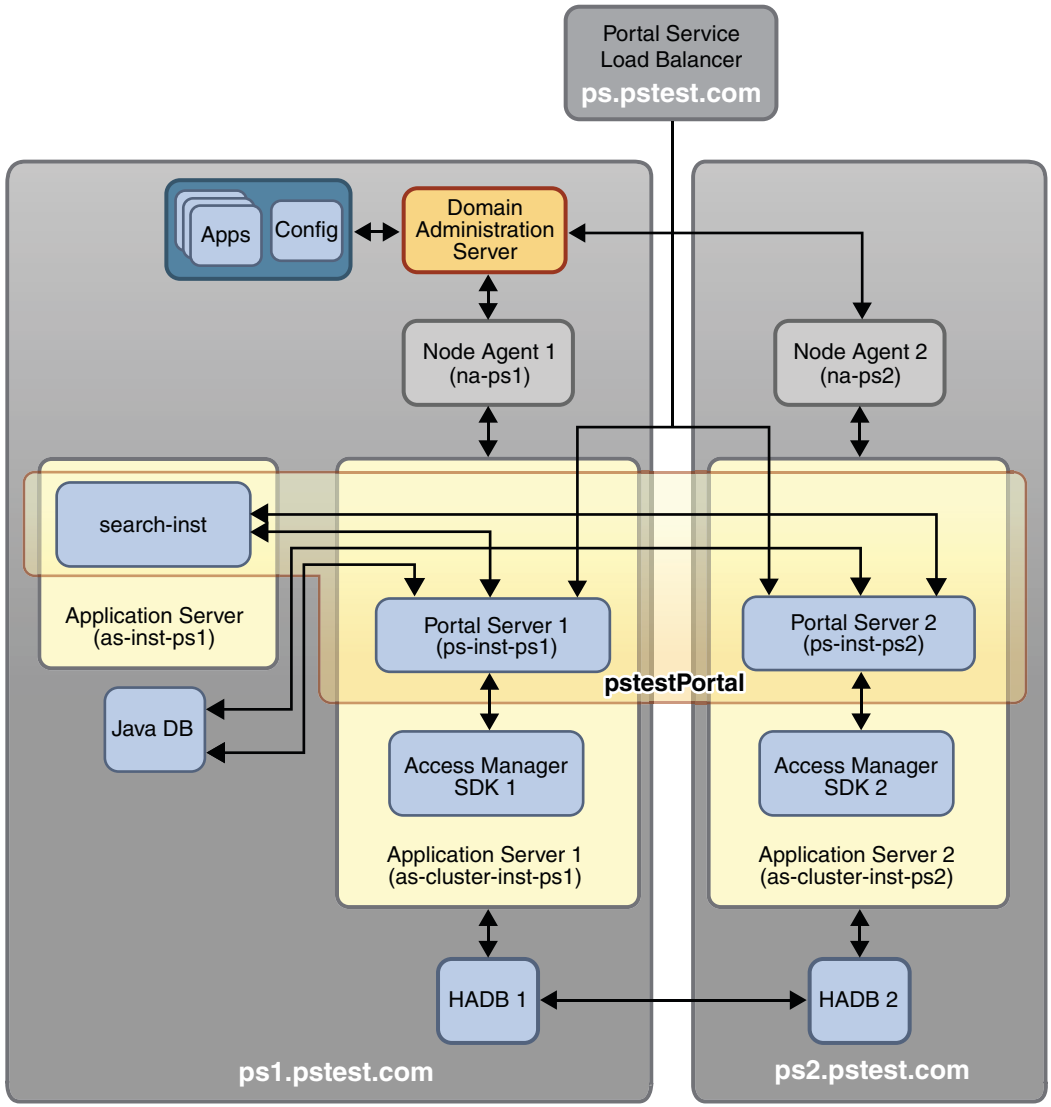

FIGURE 6–1 Portal Service Module

The Portal Server instances run in a web container that is provided by Application Server. Each Portal Server instance runs in an Application Server instance on its respective computer. The instances are managed by Application Server node agents that are acting on behalf of the Domain Administration Server (DAS).

As shown in Figure 6–1, the Portal Server instances together constitute a single portal service, `pstestPortal`.

Also indicated in Figure 6–1 is that the Application Server instances are configured to operate within the context of an Application Server cluster. The cluster provides for high availability through an HADB cluster consisting of an HADB instance on *ps1* and *ps2*.

The module also includes two components that are not replicated and therefore represent single points of failure:

- *Java DB*. Java DB is the default relational database that is used to support Portal Server features such as the search engine, community membership, and portlet sample applications (wiki, survey, and filesharing). However, Java DB cannot be configured for high availability.

  If support for communities is required in your deployment, you can switch to another relational database that can be configured for high availability. For example, instructions for configuring Portal Server to use Oracle RDBMS can be found at the Sun Developer Network web site (`http://developers.sun.com/portalserver/reference/techart/databases.html`).
- *Portal Server's Search engine*. In most architectures, a single search instance is deployed on a dedicated node and is accessed by all Portal Server instances. However, for simplicity, the search instance is deployed in this module in a separate, non-clustered Application Server instance on *ps1*. It is not possible to have multiple instances of the search engine using the same database (not shown in Figure 6–1) and at the same time be accessible through a load balancer. As a result, if the search instance fails, the search channel and the subscriptions and collaboration services, which depend upon Java DB, will become inoperative.

The general approach to implementing this module is to first set up the Application Server cluster (plus an additional Application Server instance in which to deploy the search instance), and then to install, deploy, and configure the `pstestPortal` portal within the cluster. In setting up the cluster, the Java ES installer is run in Configure Now mode to install and configure Application Server, HADB, and Access Manager SDK. In setting up the `pstestPortal` portal, the Java ES installer is run in Configure Later mode to install Portal Server, Java DB, and Service Repository libraries. Following these procedures, load balancing is implemented to provide portal service failover. Then portlet session failover is set up.

This module can be scaled horizontally by adding an additional computer like *ps2* and its respective components, and following the instructions in this chapter that apply to *ps2*.

Note – The procedures in this chapter use the host names, domain name, and IP addresses shown in Figure 3–1 and Figure 6–1. However, you must map these host names, domain name, and IP addresses to equivalent names and addresses in your environment. For this reason, the procedures in this chapter show host names, domain name, and IP addresses as variables.

Setting Up an Application Server Cluster Node on *ps1*

This task creates the infrastructure for an Application Server cluster and sets up the first node of that cluster. It consists of the following procedures:

- Install Application Server on *ps1*
- Start the Application Server Domain on *ps1*
- Start a Node Agent on *ps1*
- Create a Cluster Configuration File
- Create and Start an Application Server Instance on *ps1*

▼ To Install Application Server on *ps1*

This procedure assumes that you are installing Java ES components on Solaris 10 8/07 OS or later version. Hence, no operating system patches need to be installed. The Java ES installer evaluates the state of the operating system and indicates if you need to install a patch. If you are using versions of the operating system older than Solaris 10 8/07 OS, it is better to install any required patches before you begin the actual Java ES installation procedure.

The following procedure installs Application Server, HADB, Java DB, and Access Manager SDK, all of which are needed to support Portal Server on this computer, and all of which can be configured using the Java ES installer's Configure Now capability. Installation of Portal Server requires custom configuration and is performed in a subsequent procedure (see "Setting Up Portal Server on *ps1*" on page 132).

The procedure runs the Java ES installer without saving a state file. You can choose to run the installer and capture your input in a state file (`-saveState` *state-fileName*). You could then use the state file to re-create the installation, if, for example you needed to reinstall these components.

1 Download the Java ES software distribution to *ps1*.

The procedure is documented in "To Download the Software Distribution" on page 185.

2 Log in as root or become superuser.

```
# su -
```

3 Start the Java ES installer.

```
# cd /portdist_71u2/Solaris_sparc
```

```
# ./installer
```

This procedure uses the GUI installer. The installer can also be run in text mode by using the `-nodisplay` option.

The Welcome panel opens.

4. **In the Welcome panel, click Next.**

   The Software License Agreement panel opens.

5. **In the Software License Agreement Panel, review the license terms and click Yes, Accept License.**

   The Choose Software Components panel opens.

6. **In the Choose Software Components panel, select the following components:**

   - Application Server Enterprise Edition 8.2 patch 2
     - Domain Administration Server
     - Application Server Node Agent
     - Command Line Administration Tool
   - High Availability Session Store 4.4.3 (automatically selected)
   - Java DB 10.2.2.1 (automatically selected)
     - Java DB Client
     - Java DB Server
   - Access Manager 7.1
     - Access Manager SDK
   - Install Multilingual Package(s) for all selected components (this is selected automatically but optional if using English)

7. **Click Next.**

   The Dependency Warning panel opens.

8. **In the Dependency Warning panel, choose Use Access Manager 7.1 Installed on a Remote Machine and click OK.**

   The installer evaluates the Java SE Software Development Kit on the computer and determines if an upgrades is required. On a fresh copy of Solaris 10 8/07 OS, an upgrade is needed, and the Java SE Software Development Kit Upgrade Required panel opens.

9. **In the Java SE Software Development Kit Upgrade Required panel select Automatic Upgrade to the Version Included with the Installer and click Next.**

   The installer evaluates the Java ES shared components on the computer and determines if any upgrades are required. On a fresh copy of the Solaris 10 8/07 OS, shared component upgrades are needed, and the Shared Components Upgrades Required panel opens.

10. **In the Shared Components Upgrades Required panel, click Next.**

    The installer upgrades the shared components. The Specify Installation Directories panel opens.

11. **In the Specify Installation Directories panel, type the following values and click Next.**

Input Field	Value
Application Server	`/opt/SUNWappserver`
Application Server Data and Configuration	`/var/opt/SUNWappserver`
Access Manager	`/opt`

The installer checks the system, and the System Check panel opens.

12 In the System Check panel, evaluate the results of the system check.

If the system check is favorable, click Next.

The Choose a Configuration Type panel opens.

13 In the Choose a Configuration Type panel, select Configure Now and click Next.

The Custom Configuration Panel opens.

14 In the Custom Configuration Panel, note the following message and click Next.

```
The following component products cannot be configured during installation:
Java DB
Click Next to configure the other components.
```

The Specify Administrator Account Preferences panel opens.

15 In the Specify Administrator Account Preferences Panel, type the following values and click Next.

Input Field	Value
Administrator User ID	`admin`
Administrator Password	*app-server-admin-password*

The Common Server Settings Panel opens.

16 In the Common Server Settings panel, type the following values and click Next.

Input Field	Value
Host Name	*ps1*
DNS Domain Name	*pstest.com*
Host IP Address	*10.0.2.3*
System User	`root`

Input Field	Value
System Group	`root`

The High Availability Session Store (HADB) panel opens.

17 In the Application Server:High Availability Session Store (HADB) panel, type the following values and click Next.

Input Field	Value
HADB Management Port	`1862`
HADB Resource Directory	`/var/opt`
HADB Administrator Group	`root`

The Application Server: Domain Administration Server panel opens.

18 In the Application Server: Domain Administration Server panel, type the following values and click Next.

Input Field	Value
Admin Port	`4849`
JMX Port	`8686`
HTTP Port	`8080`
HTTPS Port	`8181`
Master Password	*app-server-master-password*

The Application Server: Node Agent panel opens.

19 In the Application Server: Node Agent panel, type the following values and click Next.

Input Field	Value
Admin Host Name	*ps1.pstest.com*
Master Password	*app-server-master-password*
Admin Port	`4849`
Node Agent Name	`na-`*ps1*

The Access Manager: Specify Configuration Information panel opens.

20 In the Access Manager: Specify Configuration Information panel, type the following values and click Next.

The following values must match the values that were used when Access Manager was installed on *am1* and *am2*).

Input Field	Value
Install Type	`Legacy Mode`
Administrator User ID	`amAdmin`
Administrator Password	*access-manager-admin-password*
LDAP User ID	`amldapuser`
LDAP Password	*access-manager-LDAP-password*
Password Encryption Key	*password-enc-key*

The Access Manager: Specify Directory Server panel opens.

21 In the Access Manager: Specify Directory Server Information panel, type the following values and click Next.

The following values must match the values that were used when Directory Server was installed on *ds1* and *ds2*, except for the host and port values. Those values must match the directory service load balancer values.

Input Field	Value
Directory Server Host	*ds.pstest.com*
Directory Server Port	`389`
Access Manager Directory Root Suffix	*dc=pstest,dc=com*
Directory Manager DN	`cn=Directory Manager`
Directory Manager Password	*directory-manager-password*

The Access Manager: Specify Directory Server Data panel opens.

22 In the Access Manager: Specify Directory Server Data panel, type the following values and click Next.

Input Field	Value
Is Directory Server Provisioned With User Data?	`Yes`

Input Field	Value
Organization Marker Object Class	`sunISManagedOrganization`
Organization Naming Attribute	`o`
User Marker Object Class	`inetorgperson`
User Naming Attribute	`uid`

The Access Manager: Specify Web Container for Running Access Manager Services panel opens.

23 In the Access Manager: Specify Web Container for Running Access Manager Services panel, type the following values and click Next.

Input Field	Value
Remote Host	*am.pstest.com*
Services Deployment URI	`amserver`
Cookie Domain	*.pstest.com*
Remote Services Port	`80`
Remote Services Protocol	`HTTP`

The Ready to Install panel opens.

24 In the Ready to Install panel, indicate whether you want to open the software registration window during installation.

This panel enables you to register the components that you have selected for installation with Sun Connection. Sun Connection is a Sun-hosted service that helps you track, organize, and maintain Sun hardware and software. For example, Sun Connection can inform you of the latest available security fixes, recommended updates, and feature enhancements.

If you choose to register, information about the installation is sent to the Sun Connection database. You can also register at a later date, after installation has been completed.

25 Click Install.

The installer copies files to the computer. The installer also configures the Access Manager SDK to interoperate with the Access Manager service and the Directory Server service. The installer also creates an instance of the Application Server Domain Administration Server (DAS) for the default Application Server domain, which is `domain1`.

26 When the installation is complete, review the installation in the Summary field.

27 Click Exit to exit the installer.

28 Check the installation log files for any installation errors.

```
# cd /var/sadm/install/logs
# egrep -i 'fail|error' Java*
```

▼ To Start the Application Server Domain

This procedure starts a Domain Administration Server (DAS) for the default domain (`domain1`) and the Application Server instance on *ps1* in which it runs.

- **Run the `start–domain` command:**

  ```
  # /opt/SUNWappserver/sbin/asadmin start-domain --user admin domain1
  ```

  When prompted, type the *app-server-admin-password*.

  The response should indicate that you successfully started the domain:

  ```
  Domain domain1 started
  ```

▼ To Start a Node Agent on *ps1*

This procedure starts the node agent (`na-`*ps1*) on *ps1* that was specified during Portal Server installation.

- **Run the `start–node–agent` command:**

  ```
  # /opt/SUNWappserver/sbin/asadmin start-node-agent --user admin na-ps1
  ```

  When prompted, type the *app-server-admin-password*.

  The response should indicate that you successfully started a node agent:

  ```
  Command start-node-agent executed successfully.
  ```

▼ To Create a Cluster Configuration

This procedure creates an Application Server cluster (`pscluster`) in the default domain (`domain1`). New Application Server instances will be created as members of this cluster on *ps1* and *ps2*. The cluster configuration file must be created before you create the Application Server instances that constitute the cluster.

- **On *ps1*, run the `create–cluster` command:**

  ```
  # /opt/SUNWappserver/sbin/asadmin create-cluster --user admin pscluster
  ```

  When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully created the cluster and its cluster configuration file, `pscluster-config`:

```
Command create-cluster executed successfully.
```

▼ To Create and Start an Application Server Instance on *ps1*

This procedure creates and starts a new Application Server instance (`as-cluster-inst-`*ps1*) on *ps1*, which belongs to `pscluster`.

1 Create an Application Server instance.

```
# /opt/SUNWappserver/sbin/asadmin create-instance --user admin --cluster
pscluster --nodeagent na-ps1 --systemproperties HTTP_LISTENER_PORT=80
as-cluster-inst-ps1
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully created the instance:

```
Command create-instance executed successfully.
```

2 Start the instance created in Step 1.

```
# /opt/SUNWappserver/sbin/asadmin start-instance --user admin
as-cluster-inst-ps1
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully started the instance:

```
Command start-instance executed successfully.
```

Setting Up an Application Server Cluster Node on *ps2*

This task creates the second node of an Application Server cluster. It consists of the following procedures:

- Install Application Server on *ps2*
- Start a Node Agent on *ps2*
- Create and Start an Application Server Instance on *ps2*

▼ To Install Application Server on *ps2*

This procedure is very similar to the procedure for installing Application Server on *ps1*, except that you do not install the Domain Administration Server (DAS) component on *ps2* because only one DAS instance is required in an Application Server cluster.

The procedure assumes that you are installing Java ES components on Solaris 10 8/07 OS or later version. Hence, no operating system patches need to be installed. The Java ES installer evaluates the state of the operating system and indicates if you need to install a patch. If you are using versions of the operating system older than Solaris 10 8/07 OS, it is better to install any required patches before you begin the actual Java ES installation procedure.

The following procedure installs Application Server, HADB, Java DB, and Access Manager SDK, all of which are needed to support Portal Server on this computer, and all of which can be configured using the Java ES installer's Configure Now capability. Installation of Portal Server requires custom configuration and is performed in a subsequent procedure (see "To Install Portal Server on *ps2*" on page 138).

The procedure runs the Java ES installer without saving a state file. You can choose to run the installer and capture your input in a state file (`-saveState` *state-fileName*). You could then use the state file to re-create the installation, if, for example you needed to reinstall these components.

1. **Download the Java ES software distribution to *ps2*.**

   The procedure is documented in "To Download the Software Distribution" on page 185.

2. **Log in as root or become superuser.**

   ```
   # su -
   ```

3. **Start the Java ES installer.**

   ```
   # cd /portdist_71u2/Solaris_sparc

   # ./installer
   ```

   This procedure uses the GUI installer. The installer can also be run in text mode by using the `-nodisplay` option.

   The Welcome panel opens.

4. **In the Welcome panel, click Next.**

   The Software License Agreement panel opens.

5. **In the Software License Agreement Panel, review the license terms and click Yes, Accept License.**

   The Choose Software Components panel opens.

6. **In the Choose Software Components panel, select the following components:**

- Application Server Enterprise Edition 8.2 patch 2
  - Application Server Node Agent
  - Command Line Administration Tool
- High Availability Session Store 4.4.3 (automatically selected)
- Java DB 10.2.2.1 (automatically selected)
  - Java DB Client
  - Java DB Server
- Install Multilingual Package(s) for all selected components (this is selected automatically but optional if using English)

7 Click Next.

The Dependency Warning panel opens.

8 In the Dependency Warning panel, choose Use Access Manager 7.1 Installed on a Remote Machine and click OK.

The installer evaluates the Java SE Software Development Kit on the computer and determines if an upgrades is required. On a fresh copy of Solaris 10 8/07 OS, an upgrade is needed, and the Java SE Software Development Kit Upgrade Required panel opens.

9 In the Java SE Software Development Kit Upgrade Required panel select Automatic Upgrade to the Version Included with the Installer and click Next.

The installer evaluates the Java ES shared components on the computer and determines if any upgrades are required. On a fresh copy of the Solaris 10 8/07 OS, shared component upgrades are needed, and the Shared Components Upgrades Required panel opens.

10 In the Shared Components Upgrades Required panel, click Next.

The installer upgrades the shared components. The Specify Installation Directories panel opens.

11 In the Specify Installation Directories panel, type the following values and click Next.

Input Field	Value
Application Server	`/opt/SUNWappserver`
Application Server Data and Configuration	`/var/opt/SUNWappserver`

The installer checks the system, and the System Check panel opens.

12 In the System Check panel, evaluate the results of the system check.

If the system check is favorable, click Next.

The Choose a Configuration Type panel opens.

13 In the Choose a Configuration Type panel, select Configure Now and click Next.

The Custom Configuration Panel opens.

14 In the Custom Configuration Panel, note the following message and click Next.

```
The following component products cannot be configured during installation:
Java DB
Click Next to configure the other components.
```

The Specify Administrator Account Preferences panel opens.

15 In the Specify Administrator Account Preferences Panel, type the following values and click Next.

Input Field	Value
Administrator User ID	admin
Administrator Password	*app-server-admin-password*

The Common Server Settings Panel opens.

16 In the Common Server Settings panel, type the following values and click Next.

Input Field	Value
Host Name	*ps2*
DNS Domain Name	*pstest.com*
Host IP Address	*10.0.2.4*
System User	`root`
System Group	`root`

The High Availability Session Store (HADB) panel opens.

17 In the Application Server:High Availability Session Store (HADB) panel, type the following values and click Next.

Input Field	Value
HADB Management Port	`1862`
HADB Resource Directory	`/var/opt`
HADB Administrator Group	`root`

The Application Server: Domain Administration Server panel opens.

18 In the Application Server: Node Agent panel, type the following values and click Next.

Input Field	Value
Admin Host Name	*ps1.pstest.com*
Master Password	*app-server-master-password*
Admin Port	4849
Node Agent Name	na-*ps2*

The Ready to Install panel opens.

19 In the Ready to Install panel, indicate whether you want to open the software registration window during installation.

This panel enables you to register the components that you have selected for installation with Sun Connection. Sun Connection is a Sun-hosted service that helps you track, organize, and maintain Sun hardware and software. For example, Sun Connection can inform you of the latest available security fixes, recommended updates, and feature enhancements.

If you choose to register, information about the installation is sent to the Sun Connection database. You can also register at a later date, after installation has been completed.

20 Click Install.

The installer copies files to the computer. The installer also configures the Access Manager SDK to interoperate with the Access Manager service and the Directory Server service. The installer also creates an instance of the Application Server Domain Administration Server (DAS) for the default Application Server domain, which is `domain1`.

21 When the installation is complete, review the installation in the Summary field.

22 Click Exit to exit the installer.

23 Check the installation log files for any installation errors.

```
# cd /var/sadm/install/logs
# egrep -i 'fail|error' Java*
```

▼ To Start a Node Agent on *ps2*

This procedure starts the node agent (`na-`*ps2*) on *ps2* that was specified during Portal Server installation.

- **Run the `start–node–agent` command:**

  ```
  # /opt/SUNWappserver/sbin/asadmin start-node-agent --user admin na-ps2
  ```

  When prompted, type the *app-server-admin-password*.

  When prompted, type the *app-server-master-password*.

  The response should indicate that you successfully started a node agent:

  ```
  Command start-node-agent executed successfully.
  ```

▼ To Create and Start an Application Server Instance on *ps2*

This procedure creates and starts a new Application Server instance (`as-cluster-inst-`*ps2*) on *ps2*, which belongs to `pscluster`.

1. **Run the `create–instance` command:**

   ```
   # /opt/SUNWappserver/sbin/asadmin create-instance --user admin --host
   ps1.pstest.com --cluster pscluster --nodeagent na-ps2 --systemproperties
   HTTP_LISTENER_PORT=80 as-cluster-inst-ps2
   ```

   When prompted, type the *app-server-admin-password*.

   The response should indicate that you successfully created the instance:

   ```
   Command create-instance executed successfully.
   ```

2. **Run the `start–instance` command:**

   ```
   # /opt/SUNWappserver/sbin/asadmin start-instance --user admin --host
   ps1.pstest.com as-cluster-inst-ps2
   ```

   When prompted, type the *app-server-admin-password*.

   The response should indicate that you successfully started the instance:

   ```
   Command start-instance executed successfully.
   ```

Setting Up a Non-Cluster Application Server Instance on *ps1*

This task creates an Application Instance on *ps1* in which to deploy a Portal Server search instance. The Application Server instance is not part of the Application Server cluster (`pscluster`) in which the Portal Server instances are to be deployed. The task consists of the following procedure:

▼ To Create and Start a Non-Cluster Application Server Instance on *ps1*

This procedure creates and starts a new non-cluster Application Server instance (`as-inst-`*ps1*) on *ps1*, in which a search instance is to be deployed.

1 Create an Application Server instance.

```
# /opt/SUNWappserver/sbin/asadmin create-instance --user admin --nodeagent
na-ps1 --systemproperties HTTP_LISTENER_PORT=5050 as-inst-ps1
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully created the instance:

```
Command create-instance executed successfully.
```

2 Start the instance created in Step 1.

```
# /opt/SUNWappserver/sbin/asadmin start-instance --user admin as-inst-ps1
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully started the instance:

```
Command start-instance executed successfully.
```

Setting Up Portal Server on *ps1*

This task consists of the following procedures:

- Install Portal Server on *ps1*
- Create a Portal Server Instance on *ps1*
- Modify the Configuration of the Portal Server Instance on *ps1*
- Start and Verify Portal Server on *ps1*

▼ To Install Portal Server on *ps1*

This procedure assumes that you are installing Portal Server on Solaris 10 8/07 OS or later version. Hence, no operating system patches need to be installed. The Java ES installer evaluates the state of the operating system and indicates if you need to install a patch. If you are using versions of the operating system older than Solaris 10 8/07 OS, it is better to install any required patches before you begin the actual Portal Server installation procedure.

This procedure runs the Java ES installer in Configure Later mode. After installation is complete, you manually configure a Portal Server instance to run in the Application Server cluster instance (`as-cluster-inst-`*ps1*).

The following procedure runs the Java ES installer without saving a state file. You can choose to run the installer and capture your input in a state file (`-saveState` *state-filename*). You could then use the state file to re-create the installation if, for example, you needed to reinstall Portal Server.

1 On *ps1*, navigate to the directory with the unzipped installer.

```
# cd /portdist_71u2/Solaris_sparc
```

2 Start the Java ES installer.

```
# ./installer
```

This procedure uses the GUI installer. The installer can also be run in text mode by using the `-nodisplay` option.

The Welcome panel opens.

3 In the Welcome panel, click Next.

The Software License Agreement panel opens.

4 In the Software License Agreement Panel, review the license terms and click Yes, Accept License.

The Choose Software Components panel opens.

5 In the Choose Software Components panel, select the following components:

- Portal Server 7.1
  - Netlet Proxy
  - Rewriter Proxy
- Portal Server Secure Remote Access 7.1
- Service Registry 3.1
  - Service Registry Client Support

Unselect Access Manager and Directory Server, which are automatically selected by the installer.

Note – Service Registry is used by Portal Server to support the Web Services for Remote Portals (WSRP) standard, in particular the WSRP producer implementation, which enables the publishing of portlets for use by remote WSRP consumers. Because the dependency on Service Registry is for such a specialized capability, installation of Service Registry is optional. For that reason, Service Registry was not included in the logical and deployment architectures of the reference configuration.

---

6 Click Next.

The Dependency Warning panel opens.

7 In the Dependency Warning panel, choose Use Access Manager 7.1 Installed on a Remote Machine and click OK.

The Specify Installation Directories panel opens.

8 In the Specify Installation Directories panel, type the following values and click Next.

Input Field	Value
Portal Server	`/opt`
Service Registry	`/opt`

The installer checks the system, and the System Check panel opens.

9 In the System Check panel, evaluate the results of the system check.

If the system check is favorable, click Next.

The Choose a Configuration Type panel opens.

10 In the Choose a Configuration Type panel, select Configure Later and click Next.

The Ready to Install Panel opens.

Note – When you select the Configure Later option, Portal Server and Service Registry files will be copied to the computer, but no configuration takes place. You must configure these components after installation is complete.

For example, the Configure Now option would have automatically deployed Portal Server in the default Domain Administration Server instance. However, you want instead to deploy Portal Server in a clustered Application Server instance. Therefore, you install in Configure Later mode, and subsequently create the Portal Server instance and deploy it to the Application Server cluster instance, `as-cluster-inst-`*ps1*.

---

11 In the Ready to Install panel, indicate whether you want to open the software registration window during installation.

This panel enables you to register the components that you have selected for installation with Sun Connection. Sun Connection is a Sun-hosted service that helps you track, organize, and maintain Sun hardware and software. For example, Sun Connection can inform you of the latest available security fixes, recommended updates, and feature enhancements.

If you choose to register, information about the installation is sent to the Sun Connection database. You can also register at a later date, after installation has been completed.

12 Click Install.

The installer copies files to the computer.

13 When the installation is complete, review the installation in the Summary field.

14 Click Exit to exit the installer.

15 Check the installation log files for any installation errors.

```
# cd /var/sadm/install/logs
# egrep -i 'fail|error' Java*
```

▼ To Create a Portal Server Instance on *ps1*

This procedure uses the `psconfig` command and a configuration file to create a portal (`pstestPortal`) as well as a Portal Server instance (`ps-inst-`*ps1*) and a search instance (`search-inst-`*ps1*) that are both associated with the portal. The procedure deploys the search instance into the non-cluster Application Server instance on *ps1* (`as-inst-`*ps1*) and the portal Server instance into all instances in Application Server cluster (`pscluster`).

You begin with an appropriate example configuration file as a template and edit the file to specify parameter values that are needed for the reference configuration.

1 Create a `config-`*ps1* configuration file.

Use the `example14.xml` file as a template.

```
# cd /opt/SUNWportal/samples/psconfig
# cp example14.xml config-ps1.xml
```

2 Open the `config-`*ps1*`.xml` file in a text editor.

3 Modify `config-`*ps1*`.xml` to use the non-default values in the following table:

Parameter	Value
AdministratorUserPassword (@ADMIN.PASSWORD@)	*access-manager-admin-password*
LDAPUserIdPassword (@AMLDAP.PASSWORD@)	*access-manager-LDAP-password*
DirectoryManagerPassword (@DIRMGR.PASSWORD@)	*directory-manager-password*
SearchServerID	`search-inst-`*ps1*
Host (@HOST.DOMAIN@)	*ps1.pstest.com*
Port (Search instance)	`5050`

Parameter	Value
WebContainerInstanceName (Search instance)	`as-inst-`*ps1*
WebContainerInstanceDir (Search instance)	`/var/opt/SUNWappserver/nodeagents/na-`*ps1*`/as-inst-`*ps1*
WebContainerDocRoot (Search instance)	`/var/opt/SUNWappserver/nodeagents/na-`*ps1*`/as-inst-`*ps1*`/docroot`
WebContainerAdminPort (@ADMIN.PORT@)	`4849`
WebContainerAdminPassword (@PASSWORD@)	*app-server-admin-password*
WebContainerMasterPassword (@MASTER.PASSWORD@)	*app-server-master-password*
PortalAccessURL	`http://`*ps.pstest.com*`:80/portal`
PrimaryPortalHost	*ps1.pstest.com*
PortalID	`pstestPortal`
InstanceID (Portal Server instance)	`ps-inst-`*ps1*
Port (Portal Server instance)	`80`
WebContainerInstanceName (Portal Server instance)	`pscluster`
WebContainerInstanceDir (Portal Server instance)	`/var/opt/SUNWappserver/nodeagents/na-`*ps1*`/as-cluster-inst-`*ps1*
WebContainerDocRoot (Portal Server instance)	`/var/opt/SUNWappserver/nodeagents/na-`*ps1*`/as-cluster-inst-`*ps1*`/docroot`

4 Save the modified `config-`*ps1*`.xml` file.

The modified file is reproduced in "Example Configuration File: Portal Server Instance on *ps1*" on page 187.

5 Run the `psconfig` command.

```
# /opt/SUNWportal/bin/psconfig --config
/opt/SUNWportal/samples/psconfig/config-ps1.xml
```

The response should resemble the following:

```
Successfully created PSConfig.properties file
Copying config templates from: /opt/SUNWportal/template/config
Successfully created PortalDomainConfig.properties file
Validating the Input Config XML File
Configuring Cacao Agent for Portal Software
Configuring Derby Server Instance
Connecting to Cacao MBean Server
Creating Portals
```

```
Domain domain1 started
Successfully created Portal: pstestPortal
Configuring Samples
```

▼ To Modify the Configuration of the Portal Server Instance on *ps1*

The previous procedure deployed Portal Server by using `pscluster` as the target Application Server instance. The `instance.id` entry, however, needs to be targeted to `as-cluster-inst-`*ps1*, rather than to the cluster. The following procedure removes this entry from the `pscluster` configuration and adds it to the `as-cluster-inst-`*ps1* configuration.

1 **Remove the `instance.id` entry from the cluster configuration.**

```
# /opt/SUNWappserver/sbin/asadmin delete-jvm-options --user admin
--target pscluster "-Dcom.sun.portal.instance.id=ps-inst-ps1"
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully removed the instance from the cluster configuration:

```
Command delete-jvm-options executed successfully.
```

2 **Add the `instance.id` entry to the `as-cluster-inst-`*ps1* configuration.**

```
# /opt/SUNWappserver/sbin/asadmin create-system-properties --user admin
--target as-cluster-inst-ps1 com.sun.portal.instance.id=ps-inst-ps1
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully added Portal Server instance information to the Application Server instance on *ps1*:

```
Command create-system-properties executed successfully.
```

3 **Restart the `pscluster` cluster.**

The cluster needs to be restarted for changes in configuration to take effect.

```
# /opt/SUNWappserver/sbin/asadmin stop-cluster --user admin pscluster
```

```
# /opt/SUNWappserver/sbin/asadmin start-cluster --user admin pscluster
```

▼ To Start and Verify Portal Server on *ps1*

You start the Portal Server instance by restarting the Application Server instance (`as-cluster-inst-`*ps1*) in which it is deployed. You then verify that the instance is running by accessing the portal Welcome page in a browser.

1. **Stop the Application Server instance.**

   ```
   # /opt/SUNWappserver/sbin/asadmin stop-instance --user admin
   as-cluster-inst-ps1
   ```

   When prompted, type the *app-server-admin-password*.

   The response should indicate that you successfully stopped the instance:

   ```
   Command stop-instance executed successfully.
   ```

2. **Restart the Application Server instance.**

   ```
   # /opt/SUNWappserver/sbin/asadmin start-instance --user admin
   as-cluster-inst-ps1
   ```

   When prompted, type the *app-server-admin-password*.

   The response should indicate that you successfully started the instance:

   ```
   Command start-instance executed successfully.
   ```

3. **Verify that the Portal Server instance is running.**

   a. **Start a browser.**

   b. **Go to the following URL:**

      ```
      http://ps1.pstest.com:80/portal/welcome
      ```

      The portal Welcome page opens, confirming that your Portal Server instance is running.

Setting Up Portal Server on *ps2*

This task consists of the following procedures:

- Install Portal Server on *ps2*
- Configure Access Manager SDK on *ps2*
- Create a Portal Server Instance on *ps2*
- Modify the Configuration of the Portal Server Instance on *ps2*
- Start and Verify Portal Server on *ps2*

▼ To Install Portal Server on *ps2*

To properly set up Portal Server on *ps2*, Access Manager SDK must be configured manually. As a result, Access Manager SDK is installed with Portal Server on *ps2* using the Configure Later option of the installer rather than being installed with Application Server using the Configure Now option (as it was on *ps1*).

- **Repeat the procedure that appears in "To Install Portal Server on *ps1*" on page 132, except for the following:**

  In the Choose Software Components panel (Step 5), select the following components:

  - Portal Server 7.1
    - Netlet Proxy
    - Rewriter Proxy
  - Portal Server Secure Remote Access 7.1
  - Access Manager 7.1
    - Access Manager SDK
  - Service Registry 3.1
    - Service Registry Client Support

  Unselect Directory Server, which is automatically selected by the installer.

▼ To Configure Access Manager SDK on *ps2*

Because Access Manager SDK was installed using the Configure Later option, you need to configure Access Manager SDK by modifying Access Manager configuration files. The standard approach for making these modifications is to run the `amconfig` command with an input file.

1. **Change to the directory that contains the `amconfig` input file template, `amsamplesilent`.**

   ```
   # cd /opt/SUNWam/bin
   ```

2. **Copy the template to a new file.**

   ```
   # cp amsamplesilent amconfigps2
   ```

3. **In a text editor, edit the `amconfigps2` file to set the Access Manager SDK configuration parameters.**

   Locate the configuration parameters that are listed in the following table, and change their values to the values shown in the table.

Parameter	Value
DEPLOY_LEVEL	`4`
SERVER_Name	*am*
SERVER_HOST	*am.pstest.com*
SERVER_PORT	`80`

Parameter	Value
ADMIN_PORT	`4849`
DS_HOST	*ds.pstest.com*
DS_DIRMGRPASSWD	*directory-manager-password*
ROOT_SUFFIX	`"dc=pstest,dc=com"`
ADMINPASSWD	*access-manager-admin-password*
AMLDAPUSERPASSWD	*access-manager-LDAP-password*
COOKIE_DOMAIN	*pstest.com*
AM_ENC_PWD	*password-enc-key*
NEW_OWNER	`root`
NEW_GROUP	`other`
PAM_SERVICE_NAME	`other`
AS81_INSTANCE	`pscluster`
AS81_INSTANCE_DIR	`/var/opt/SUNWappserver/nodeagents/na-`*ps2*`/as-cluster-inst-`*ps2*
AS81DOCS_DIR	`/var/opt/SUNWappserver/nodeagents/na-`*ps2*`/as-cluster-inst-ps2/docroot`

4 Run the `amconfig` command with the input file you modified in Step 3.

```
# /opt/SUNWam/bin/amconfig -s amconfigps2
```

The output should show failures after checking if Application Server is already configured with Access Manager. These errors are expected because the Access Manager configuration was already added to the Application Server cluster configuration when Access Manager SDK was installed and configured on *ps1*.

5 Verify that the Access Manager SDK is properly configured.

```
# /opt/SUNWam/bin/amadmin –u amadmin –m http://am.pstest.com:80
```

When prompted, type the *access-manager-admin-password*.

The output should show current session information.

▼ To Create a Portal Server Instance on *ps2*

When you created a Portal Server instance (`ps-inst-`*ps1*) on *ps1* (see "To Create a Portal Server Instance on *ps1*" on page 135), you deployed Portal Server to `pscluster` and created additional portal and instance configuration information on *ps1*.

To create a Portal Server instance (`ps-inst-`*ps2*) on *ps2*, you need to set up the portal's administrative infrastructure that is needed to copy the portal and instance configuration information from *ps1* to *ps2*.

In addition, before you can use the portal's administrative interface to create `ps-inst-`*ps2*, you must remove `ps_util.jar` from the `pscluster` classpath, as described in the following procedure. If you try to run the `psadmin -create-instance` command without first removing `ps_util.jar`, `psadmin` will detect the presence of `ps_util.jar`and exit without creating a new instance.

The following procedure sets up the portal's administrative infrastructure, removes `ps_util.jar` from the `pscluster` classpath, creates a custom configuration file for creating `ps-inst-`*ps2*, and then uses the `psadmin -create-instance` command to create the new instance.

You begin with an appropriate example configuration file as a template and edit the file to specify parameter values that are needed for the reference configuration.

1 Set up the portal's administrative infrastructure.

a. Create a `config-`*ps2* configuration file.

Use the `example2.xml` file as a template.

```
# cd /opt/SUNWportal/samples/psconfig
```

```
# cp example2.xml config-ps2.xml
```

b. Open the `config-`*ps2*`.xml` file in a text editor.

c. Modify the `config-`*ps2*`.xml` file to use parameter values that are appropriate for the reference configuration.

The modified `config-`*ps2*`.xml` file is reproduced in "Example Configuration File: Portal Server Instance on *ps2*" on page 189.

d. Run the `psconfig` command:

```
# /opt/SUNWportal/bin/psconfig --config
/opt/SUNWportal/samples/psconfig/config-ps2.xml
```

The response should resemble the following:

```
Logs redirected to /var/opt/SUNWportal/logs/config/portal.fabric.0.0.log
Creating directory: /etc/opt/SUNWportal
Successfully created PSConfig.properties file
Copying config templates from: /opt/SUNWportal/template/config
Successfully created PortalDomainConfig.properties file
Validating the Input Config XML File
Configuring Cacao Agent for Portal Software
Configuring Derby Server Instance
Closing MBean Server connection ...
Resetting log level ...
Configuration Successful
```

2 Remove `ps_util.jar` from the `pscluster` classpath.

a. Start a browser.

b. Go to the following URL:

`https://`*`ps1`*`.pstest.com:4849`

The Application Server login page opens.

c. Log in to the Application Server Admin Console by typing the following values and clicking Login.

Input Field	Value
User ID	`admin`
Password	*app-server-admin-password*

The Application Server Admin Console opens.

d. Click on the small triangle next to Configurations on the Common Tasks panel.

The configurations are expanded.

e. Click on the small triangle next to `pscluster-config`.

The `pscluster` configuration is expanded.

f. Click on JVM Settings.

The frame on the right shows the configuration options.

g. In the right frame, select the Path Settings tab.

The JVM Classpath Settings panel opens.

h. Remove `/opt/SUNWportal/lib/ps_util.jar` from the `Classpath Suffix` list.

i. Click Save.

j. Restart the `pscluster` cluster.

The cluster needs to be restarted for the change in configuration to take effect. The following commands are run on *ps2*, but can also be run on *ps1* without using the `--host` option.

```
# /opt/SUNWappserver/sbin/asadmin stop-cluster --user admin --host
ps1.pstest.com pscluster
```

```
# /opt/SUNWappserver/sbin/asadmin start-cluster --user admin --host
ps1.pstest.com pscluster
```

3 Create a custom configuration file to use when creating a new Portal Server instance.

a. Copy the configuration template to a new file.

```
# cd /opt/SUNWportal/template
```

```
# cp Webcontainer.properties.SJSAS81 Webcontainer.properties.ps-inst-ps2
```

b. Open `Webcontainer.properties.ps-inst-`*ps2* in a text editor.

c. Set the parameters that are listed in the following table.

Parameter	Value
Host	*ps2.pstest.com*
Port	`80`
Scheme	`http`
WebContainerInstallDir	`/opt/SUNWappserver/appserver`
WebContainerInstanceName	`pscluster`
WebContainerDomainName	`domain1`
WebContainerInstanceDir	`/var/opt/SUNWappserver/nodeagents/na-`*ps2*`/as-cluster-inst-`*ps2*
WebContainerDocRoot	`/var/opt/SUNWappserver/nodeagents/na-`*ps2*`/as-cluster-inst-`*ps2*`/docroot`
WebContainerAdminHost	*ps1.pstest.com*
WebContainerAdminPort	`4849`

Parameter	Value
WebContainerAdminScheme	`https`
WebContainerAdminPassword	*app-server-admin-password*
WebContainerMasterPassword	*app-server-master-password*

d. Save your changes and close the file.

4 Create a new Portal Server instance.

Execute the following command on *ps2*:

```
# /opt/SUNWportal/bin/psadmin create-instance -u amadmin -p pstestPortal
--instance ps-inst-ps2 --webconfig Webcontainer.properties.ps-inst-ps2
```

▼ To Modify the Configuration of the Portal Server Instance on *ps2*

The previous procedure deployed Portal Server by using `pscluster` as the target Application Server instance. The `instance.id` entry, however, needs to be targeted to `as-cluster-inst-`*ps2*, rather than to the cluster. The following procedure removes this entry from the `pscluster` configuration and adds it to the `as-cluster-inst-`*ps2* configuration.

The following commands are run on *ps2*, but can also be run on *ps1* without using the `--host` option.

1 Remove the `instance.id` entry from the cluster configuration.

```
# /opt/SUNWappserver/sbin/asadmin delete-jvm-options --user admin
--target pscluster "-Dcom.sun.portal.instance.id=ps-inst-ps2" --host
ps1.pstest.com
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully removed the instance from the cluster configuration:

```
Command delete-jvm-options executed successfully.
```

2 Add the `instance.id` entry to the `as-cluster-inst-`*ps2* configuration.

```
# /opt/SUNWappserver/sbin/asadmin create-system-properties --user admin
--target as-cluster-inst-ps2 com.sun.portal.instance.id=ps-inst-ps2 --host
ps1.pstest.com
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully added Portal Server instance information to the Application Server instance on *ps2*:

```
Command create-system-properties executed successfully.
```

3 Restart the `pscluster` cluster.

The cluster needs to be restarted for changes in configuration to take effect.

```
# /opt/SUNWappserver/sbin/asadmin stop-cluster --user admin pscluster --host
ps1.pstest.com
```

```
# /opt/SUNWappserver/sbin/asadmin start-cluster --user admin pscluster --host
ps1.pstest.com
```

▼ To Start and Verify Portal Server on *ps2*

You start the Portal Server instance by restarting the Application Server instance (`as-cluster-inst-`*ps2*) in which it is deployed. You then verify that the instance is running by accessing the portal Welcome page in a browser.

1 Stop the Application Server instance.

```
# /opt/SUNWappserver/sbin/asadmin stop-instance --user admin
as-cluster-inst-ps2
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully stopped the instance:

```
Command stop-instance executed successfully.
```

2 Restart the Application Server instance.

```
# /opt/SUNWappserver/sbin/asadmin start-instance --user admin
as-cluster-inst-ps2
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully started the instance:

```
Command start-instance executed successfully.
```

3 Verify that the Portal Server instance is running.

a. Start a browser.

b. Go to the following URL:

```
http://ps2.pstest.com:80/portal/Welcome
```

The portal Welcome page opens, confirming that your Portal Server instance is running.

Implementing Load Balancing for the Portal Service

This task consists of the following procedures:

- Configure the Load Balancer for the Portal Service
- Verify Load Balancing for the Portal Service

▼ To Configure the Load Balancer for the Portal Service

This procedure describes how to configure the portal service load balancer (*ps.pstest.com* at IP address *10.0.3.11*). The steps are relatively generic; the details depend on the load balancer you are using.

1 Populate the load balancer's Hosts Table.

Add the IP address for *ps1.pstest.com* and *ps2.pstest.com* to the load balancer's hosts table.

2 Populate the load balancer's Real Service Table.

Add the real services for *ps1.pstest.com* and *ps2.pstest.com*. A real service is identified by its IP address and port. Add *10.0.2.3*`:80` and *10.0.2.4*`:80`.

3 Populate the load balancer's Service Group Table.

Add the service group for portal services. The service groups are sets of the real services that you defined in Step 2. The real services in the group must be capable of fulfilling the same type of request. The load balancer will distribute requests among the real services in the service group. When you define the service group for the *ps.pstest.co*m, you add the real services that specify the Portal Server instances, *10.0.2.3*`:80` and *10.0.2.4*`.80`.

4 Populate the load balancer's Virtual IP Table

A virtual service definition includes the outward-facing IP address and the port at which the load balancer accepts requests for a service, as well as the service group that you specified in Step 3, which actually handles the requests. The load balancer will accept requests at the virtual service address and distribute them among the service group. The virtual service definition for the Portal Server service should be *ps.pstest.com*, with the virtual IP address of *10.0.3.11*`:80`, and with the service group consisting of the computers *ps1.pstest.com* and *ps2.pstest.com*.

5 Configure the load balancer to use Layer-7 (HTTP layer) load balancing.

6 **Configure the load balancer with a scheduling type of either least connections or round robin.**

Both scheduling types initially distribute the connections evenly between the Portal Server instances. Both scheduling types keep the connections evenly distributed if the connections are restarted.

7 **Configure the load balancer for sticky routing.**

The portal service load balancer must maintain session persistence; it must route all user requests subsequent to the first request, to the same Portal Server instance (except in the case of failure).

There are two options for sticking the user's portal session to the same Portal Server instance:

- Load balancer passive cookies (also known as *managed cookies*). If your load balancer has this feature, it is the preferred solution.
- Portal Server provides a mechanism analogous to the Access Manager's `amlbcookie`. You can specify the name of a cookie (for example, `pslbcookie`) in the `lb.cookie.name` property of the following file on both *ps1* and *ps2*:
  `/var/opt/SUNWportal/portals/pstestPortal/config/desktopconfig.properties`

  Each Portal Server instance assigns a value to this cookie at runtime. The value, which identifies the Portal Server instance, has the following syntax:

  `<portalID>.<instanceID>`

  For example, in the reference configuration, the value of the cookie for *ps1.pstest.com* will be `pstestPortal.ps-inst-`*ps1* and for *ps2.pstest.com* the value will be `pstestPortal.ps-inst-`*ps2*.

  The load balancer is configured for session persistence using this cookie.

8 **Configure the health-check settings for the load balancer.**

The recommended settings are specified in Table 3–5.

▼ To Verify Load Balancing for the Portal Service

This procedure verifies that you can interact with Portal Server instances through the load balancer and that the load balancer provides service failover when a Portal Server instance fails.

1 **Configure the Portal Server instances on *ps1* and *ps2* to support session cookies.**

Portal Server provides a mechanism that is analogous to the Access Manager's `amlbcookie` in which you can specify the name of a session cookie, as follows:

a. **In a text editor, open the following configuration file:**

`/var/opt/SUNWportal/portals/pstestPortal/config/desktopconfig.properties`

b. Specify the name of a session cookie, as follows:

```
lb.cookie.name=pslbcookie
```

Each Portal Server instance assigns a value to this cookie at runtime. The value, which identifies the Portal Server instance, has the following syntax:

```
<portal ID>.<instance ID>
```

2 Restart the Portal Server instances on *ps1* and *ps2* by restarting the Application Server instances in which they are deployed.

See "To Start and Verify Portal Server on *ps1*" on page 137 and "To Start and Verify Portal Server on *ps2*" on page 145.

3 Log in to the DeveloperSample desktop.

a. Open the portal service in a browser.

Use the load balancer URL:

`http://`*ps.pstest.com*`/portal`

The portal Welcome page opens.

b. On the Samples box, click on the DeveloperSample.

The anonymous desktop for the DeveloperSample should be displayed.

c. Log in by typing the following values and clicking Login.

Input Field	Value
User ID	developer
Password	developer

The DeveloperSample desktop is displayed.

4 Determine which Portal Server instance handled the login request in Step 3.

You can determine the instance by opening the `pslbcookie` cookie in your browser and checking it's instance ID value. For example in the Firefox browser, do the following:

a. Choose Edit→Preferences→Privacy→Show Cookies.

b. Select the *pstest.com* portal.

c. Select the `pslbcookie` cookie.

d. Note the instance ID in the Content field.

For example, the value of the cookie for *ps1.pstest.com* will be `pstestPortal.ps-inst-`*ps1*, and for *ps2.pstest.com* the value will be `pstestPortal.ps-inst-`*ps2*.

5 Simulate a failure of the Portal Server instance that was noted in Step 4.

Failure of an Portal Server instance can result from a computer failure, a software failure, or a network failure. The method employed for simulating a failure in this service failover verification procedure is to shut down the Portal Server instance (by shutting down the Application Server instance in which it runs). Additionally, you could also simulate failure by unplugging the network cable or disabling the interface.

Run the following command on the computer (*ps1* or *ps2*) hosting the instance identified in Step 2.

```
# /opt/SUNWappserver/sbin/asadmin stop-instance --user admin
as-cluster-inst-ps1|ps2
```

When prompted, type the *app-server-admin-password*.

6 Repeat Step 3, above (log in to the DeveloperSample desktop).

If service failover is working correctly, the portal Welcome page opens, confirming that the load balancer has routed the login request to the remaining online Portal Server instance.

7 Recover the simulated failure of your original Portal Server instance.

Restart the Application Server instance that was shut down in Step 5.

```
# /opt/SUNWappserver/sbin/asadmin ststt-instance --user admin
as-cluster-inst-ps1|ps2
```

When prompted, type the *app-server-admin-password*.

Implementing Portlet Session Failover

The following procedures use of an example portlet(`PortletSessionCounter`) to demonstrate how to set up and test portlet session failover. `PortletSessionCounter` is a portlet that stores session state information. In particular, it counts the number of times a user interacts with the portlet by accessing the developer desktop.

To implement portlet session failover, you first enable high availability for the Application Server cluster in which your Portal Server instances are running. You then deploy and set up high availability for the portlet. As a result, session state information that is created by the portlet will be saved in the Application Server's High Availability Session Store (HADB). If a Portal Server instance fails, the session state information is made available to a failover Portal Server instance.

This section consists of the following procedures.

- Configure the Availability Service for `pscluster`.
- Set Up Session Failover for a Portlet
- Verify Portlet Session Failover

To Configure the Availability Service for `pscluster`

To implement portlet session failover, you must first create an HADB cluster and configure Application Server availability service settings for `pscluster`, as described in this procedure.

1 Create an HADB cluster and add it to your Application Server cluster.

Run the following command on either *ps1* or *ps2*:

```
# /opt/SUNWappserver/sbin/asadmin configure-ha-cluster --user admin --hosts
ps1.pstest.com,ps2.pstest.com pscluster
```

When prompted, type the *app-server-admin-password*.

2 Start the Application Server Admin Console.

a. Start a browser.

b. Go to the following URL:

`https://`*ps1.pstest.com*`/4849`

The Application Server Admin Console login page opens.

c. Log in to the Application Server Admin Console by typing the following values and clicking Login.

Input Field	Value
User ID	`admin`
Password	*app-server-admin-password?*

The Application Server Admin Console opens.

3 Modify Availability Service settings.

a. Click on the small triangle next to Configurations in the left pane under Common Tasks.

The configurations are expanded.

b. Click on the small triangle next to `pscluster-config`.

The `pscluster` configuration is expanded.

c. Click on the Availability Service.

The Availability Service settings are displayed in the right pane.

d. Type the Availability Service settings shown in the following table.

Input Field	Value
Availability Service	`Enabled`
Store Pool Name	`pscluster-hadb-pool`
Ha-agent-hosts	*10.0.2.3*, *10.0.2.4*
Ha-agent-ports	`1862`
Ha-db-name	`pscluster`

e. Click Save.

4 Close the Console.

5 Verify that HADB is working and that *ps1* and *ps2* are part of the HADB cluster.

a. Check the HADB status.

```
# /opt/SUNWhadb/4.4.3-5/bin/hadbm status pscluster
```

When prompted, type the *app-server-admin-password*.

The status of the `pscluster` high availability database should be `FaultTolerant`.

```
Database  Status
pscluster FaultTolerant
```

b. Check that *ps1* and *ps2* are part of the HADB cluster.

```
# /opt/SUNWhadb/4.4.3-5/bin/hadbm status --nodes pscluster
```

When prompted, type the *app-server-admin-password*.

When prompted, type the *app-server-master-password*.

All Application Server nodes should be in the running state.

```
 NodeNo HostName       Port  NodeRole NodeState MirrorNode
 0      ps1.pstest.com 15200 active   running   1
 1      ps2.pstest.com 15220 active   running   0
```

▼ To Set Up Session Failover for a Portlet

This procedure uses an example `Session Counter` portlet to demonstrate how to set up portlet session failover. Using this example, the procedure describes how to deploy a portlet into the `pstestPortlet` portal, enable high availability for the portlet, and create a desktop channel for the portlet.

1 **Download the `Session Counter` portlet from the Open Portal Portlet Repository site:**

`https://portlet-repository.dev.java.net/public/Download.html`

The `sessioncounter.war` file is downloaded.

2 **Deploy the `Session Counter` portlet to `DeveloperSample` in the `pstestPortal` portal.**

```
# /opt/SUNWportal/bin/psadmin deploy-portlet -u amadmin
-d o=DeveloperSample,dc=pstest,dc=com -p pstestPortal sessioncounter.war
```

When prompted, type the *access-manager-admin-password*.

3 **Enable high availability for the `Session Counter` portlet.**

```
# /opt/SUNWappserver/sbin/asadmin set --user admin
"domain.applications.web-module.sessioncounter.availability-enabled=true"
```

When prompted, type the *app-server-admin-password*.

4 **Enable high availability for the Portal web application.**

```
# /opt/SUNWappserver/sbin/asadmin set --user admin
"domain.applications.web-module.portal.availability-enabled=true"
```

When prompted, type the *app-server-admin-password*.

5 **Create a channel for the `Session Counter` portlet.**

a. **Create a display profile for the `Session Counter` portlet.**

Save the display profile document in "Example Display Profile: `Session Counter` Portlet" on page 191 to `/tmp/developer-dp.xml`.

b. **Add the display profile to the `developer` user.**

```
# /opt/SUNWportal/bin/psadmin add-display-profile -u amadmin
-d uid=developer,ou=People,o=DeveloperSample,dc=pstest,dc=com
-p pstestPortal /tmp/developer-dp.xml
```

When prompted, type the *access-manager-admin-password*.

▼ To Verify Portlet Session Failover

In this procedure, you exercise the `Session Counter` portlet by successively reloading the developer desktop page in a browser. You first exercise the portlet while the Portal Server instance on *ps2* is shut down, noting the session counter value. You then restart the Portal Server instance on *ps2* and simulate a failure of the Portal Server instance on *ps1*so that the portal service fails over to the instance on *ps2*.

You then continue to refresh the browser and note the session counter value. If portlet session failover is working properly, the session counter value should continue to increase from its value before the failover occurred. If the session counter value is reset to `1`, then session failover is not working.

1 Make sure that `ps-inst-`*ps1* on *ps1* is running and that `ps-inst-`*ps2* on *ps2* is shut down.

a. Start `ps-inst-`*ps1* by starting the Application Server instance on *ps1* in which it is deployed.

```
# /opt/SUNWappserver/sbin/asadmin start-instance --user admin
as-cluster-inst-ps1
```

When prompted, type the *app-server-admin-password*.

b. Shut down `ps-inst-`*ps2* by shutting down the Application Server instance on `ps2` in which it is deployed.

```
# /opt/SUNWappserver/sbin/asadmin stop-instance --user admin
as-cluster-inst-ps2
```

When prompted, type the *app-server-admin-password*.

2 Create a portlet session on *ps1*.

You create a session by accessing the `Session Counter` portlet on the Developer Sample desktop, as follows:

a. Open the portal service in a browser:

`http://`*ps.pstest.com*`/portal`

b. On the Samples box, click on the DeveloperSample.

The anonymous desktop for the DeveloperSample should be displayed.

c. Log in by typing the following values and clicking Login.

Input Field	Value
User ID	developer

Input Field	Value
Password	developer

The DeveloperSample desktop is displayed.

d. Click the Portlet Samples tab.

e. Look at the `Session Counter` channel.

The channel should show (in addition to other information):

```
Counter Value in Session is 1.
```

3 Click Reload in your browser.

Now the `Session Counter` channel should show:

```
Counter Value in Session is 2.
```

Every time you reload the desktop page, the counter value will increase.

4 Start `ps-inst-`*ps2*.

```
# /opt/SUNWappserver/sbin/asadmin start-instance --user admin
as-cluster-inst-ps2
```

5 Simulate a failure of `ps-inst-`*ps1*.

In the configuration interface for your load balancer (*ps.pstest.com*), remove the real server instance, `ps-inst-ps1`, from the service group.

6 Click Reload in your browser.

Now the `Session Counter` channel should show:

```
Counter Value in Session is N.
```

Where the value of *N* is one integer greater than the last value in Step 3.

If session failover is not working, then the channel would show:

```
Counter Value in Session is 1.
```

7 Recover the simulated failure of the `ps-inst-`*ps1* instance.

Return to the configuration interface for your load balancer, and re-enable the instance.

Tuning Portal Server Instances

Consider the following issues when tuning Portal Server instances to maximize performance:

Impact of Java DB on Performance

The Portal Service on Application Server Cluster reference configuration includes an instance of Java DB running on *ps1*. Java DB is the default database used to store configuration and membership data for Portal Server's community feature. Java DB is also used by portlet applications such as wiki, fileshare, and survey.

However, the use of Java DB has an impact on the overall performance of Portal Server. If you do not need the community feature and are not using a portlet application that requires Java DB, you can improve portal service performance by disabling the use of Java DB by the portal desktop and then shutting down the database.

Caution – Be careful not to shut down Java DB without disabling Java DB for the desktop, as this action seriously degrades performance.

This section consists of the following procedures:

- Disable the Use of Java DB by the Desktop
- Shut Down Java DB

▼ To Disable the Use of Java DB by the Desktop

1. **On *ps1*, change to the following directory:**

   ```
   # cd /var/opt/SUNWportal/portals/portal-ID/config
   ```

2. **Edit the `communitymc.properties` file.**

   Remove the `jdo` entry from the `manager.contributors` list.

3. **Restart the Portal Server instance by restarting the Application Server instance in which it is deployed.**

   a. **Stop the Application Server instance.**

      ```
      # /opt/SUNWappserver/sbin/asadmin stop-instance --user admin
      as-cluster-inst-ps1
      ```

      When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully stopped the instance:

```
Command stop-instance executed successfully.
```

b. Start the Application Server instance.

```
# /opt/SUNWappserver/sbin/asadmin start-instance --user admin
as-cluster-inst-ps1
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully started the instance:

```
Command start-instance executed successfully.
```

4 Repeat Steps 1 through 3 on *ps2*, except for the following:

Replace all occurrences of *ps1* with *ps2*, including the instance ID (replacing `ps-inst-`*ps1* with `ps-inst-`*ps2*).

▼ To Shut Down Java DB

This procedure is performed only on *ps1* where the Java DB instance is running (see Figure 6–1).

- **Run the following command on *ps1*:**

```
# java -cp /opt/SUNWjavadb/lib/derbynet.jar
org.apache.derby.drda.NetworkServerControl shutdown
```

Tuning Application Server Instances

Portal Server performs best when the web container in which it runs is tuned to optimize Portal Server performance. So, you must tune the Application Server instances that host Portal Server. For this module, Portal Server runs in the Application Server instances in the `pstest` cluster.

In the portal service reference configuration, no other component or application besides Portal Server runs in the Application Server instance. However, if you run other components or applications in the same Application Server instance, then be aware that optimizing the Application Server instance for Portal Server might negatively impact the performance of other components.

This section consists of the following procedures:

- To Tune the Application Server Instance for Portal Server on *ps1*
- Tune the Application Server Instance for Portal Server on *ps2*

▼ To Tune the Application Server Instance for Portal Server on *ps1*

Tuning the Application Server instances that host Portal Server is performed by using the `perftune` utility. This utility runs the `amtune` utility and also performs some tuning of Portal Server thread usage. For additional information about `amtune`, see Part I, "Basic Performance Tuning," in *Sun Java System Access Manager 7.1 Performance Tuning and Troubleshooting Guide*.

1 On *ps1*, change to the following directory:

```
# cd /opt/SUNWam/bin/amtune
```

2 Open the `amtune-env` file in a text editor.

3 Confirm, or if necessary, modify the following values and save the changes.

```
WEB_CONTAINER=AS8
ASADMIN_PORT=80
DOMAIN_NAME=pstest.com
AMTUNE_PCT_MEMORY_TO_USE=100
AMTUNEAMTU_MIN_PERM_SIZE_AS8=128
AMTUNE_TUNE_WEB_CONTAINER=false
AMTUNE_MODE=REVIEW
```

Also, change all variables that are named `CONTAINER_*` to point to the `as-cluster-inst-`*ps1* instance:

`CONTAINER_*=/var/opt/SUNWappserver/nodeagents/`*ps1.pstest.com*`/as-cluster-inst-`*ps1*

Note – In a Solaris zones deployment, also set `AMTUNE_TUNE_OS=false`.

---

4 Run the `perftune` utility.

`# /opt/SUNWportal/bin/perftune` *directory-manager-password app-server-admin-password*

The utility proposes suggested changes.

5 Review the suggested changes.

If no undesired changes are proposed, continue to the next step.

6 Modify the `amtune-env` file with the following value and save the changes.

```
AMTUNE_MODE=CHANGE
```

7 Repeat Step 4.

The utility will make all the changes proposed in Step 4.

8 Restart the computer.

Restarting the computer will affect changes to the operating system.

9 **Start the Portal Server instance.**

Starting Portal Server will affect changes made to the web container. You start Portal Server by starting the Application Server instance in which it is deployed.

```
# /opt/SUNWappserver/sbin/asadmin start-instance --user admin
as-cluster-inst-ps1
```

When prompted, type the *app-server-admin-password*.

The response should indicate that you successfully started the instance:

```
Command start-instance executed successfully.
```

▼ To Tune the Application Server Instance for Portal Server on *ps2*

- **Repeat the procedure in "To Tune the Application Server Instance for Portal Server on *ps2*" on page 158, except for the following:**

  Replace all occurrences of *ps1* with *ps2*.

Taking a Snapshot of the Module

When you have completed deploying the portal service module of the reference configuration, and before you move on to the next module, it is good practice to take a snapshot of the data in the Directory Server instance. By exporting `ds-inst-`*ds1*, you preserve the current state of your deployment in case you subsequently need to roll back directory information to this point in the reference configuration deployment process. The directory serves as the repository for service and user configuration information and therefore changes as each reference configuration module is deployed.

▼ To take a snapshot of the directory on *ds1*

In this procedure you use the `db2ldif` command to export the directory to an `ldif` file. If you want to subsequently restore the directory, use an equivalent procedure with the `ldif2db` command.

1 **On *ds1* change directory as follows:**

```
# cd /var/opt/SunWdsee/ds-inst-ds1
```

2 **Stop the Directory Server instance.**

```
# ./stop-slapd
```

3 **Export the current state of the *pstest* directory to an `ldif` file.**

```
# ./db2ldif -n pstest
```

The output should resemble the following:

```
ldiffile: /var/opt/SunWdsee/ds-inst-ds1/ldif/2008_05_20_140750.ldif
[20/May/2008:14:07:56 +0100] - export pstest: Precessed 1000 entries (26%)
...
[20/May/2008:14:08:02 +0100] - export pstest: Precessed 4165 entries (100%)
```

4 **Rename the `ldif` file to something meaningful.**

```
# mv /var/opt/SunWdsee/ds-inst-ds1/ldif/2008_05_20_140750.ldif
/var/opt/SunWdsee/ds-inst-ds1/ldif/ps_module_complete.ldif
```

5 **Restart the Directory Server instance.**

```
# ./start-slapd
```

◆ ◆ ◆ CHAPTER 7

Implementation Module 4: Secure Remote Access Gateway

This chapter provides an overview of the Secure Remote Access (SRA) Gateway module in Figure 2–2 and documents the tasks required to implement it. The chapter includes the following sections:

Overview of the SRA Gateway Module

The SRA Gateway module of the reference configuration deployment architecture illustrated in Figure 2–2consists of two Sun Java System Portal Server Secure Remote Access (SRA) Gateway instances running on two different computers, with additional, optional Rewriter Proxy and Netlet Proxy components residing on the computers hosting Portal Server instances. The module makes use of a hardware load balancer that is configured to provide SRA Gateway service failover capability between the two Gateway instances. All external Internet requests for portal services are addressed to the virtual service name and IP address of the Gateway service load balancer. The load balancer directs each request to one of the Gateway instances.

The Access Manager SDK library is required for each Gateway instance because the Gateway service and Gateway profile are stored as Access Manager services in Directory Server. The Netlet Proxy and Rewriter Proxy instances are accessed directly by the Gateway instances by using a round-robin scheduling algorithm.

The architecture of the SRA Gateway module is shown in the following illustration.

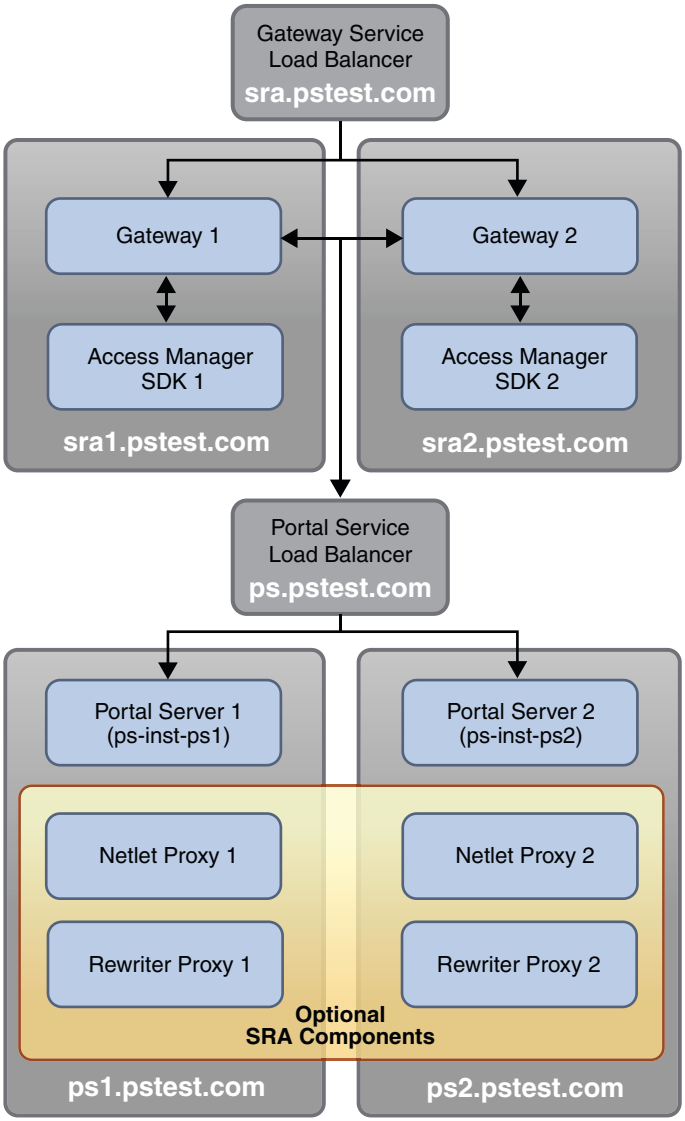

FIGURE 7–1 SRA Gateway Module

The general approach to implementing this module is to first set up a Gateway profile for the SRA layer. Each Portal Server instance is then configured for SRA operation, after which the Gateway instances themselves are set up. Following these procedures, load balancing is implemented to provide Gateway service failover.

This module can be scaled horizontally by adding an additional computer like *sra2* and its respective components, and following the instructions in this chapter that apply to *sra2*.

When you install and configure the SRA Gateway module, you configure it to interoperate with the other modules in the reference configuration. This chapter describes the procedures for implementing the SRA Gateway module in the following sections.

Note – The procedures in this chapter use the host names, domain name, and IP addresses shown in Figure 3–1 and Figure 7–1. However, you must map these host names, domain name, and IP addresses to equivalent names and addresses in your environment. For this reason, the procedures in this chapter show host names, domain name, and IP addresses as variables.

Setting Up a Gateway Profile

This task consist of the following procedures:

- Verify a Default Gateway Profile
- Enable the portal service for SRA
- Provision the Gateway Profile
- Verify the Updated Gateway Profile

▼ To Verify the Default Gateway Profile

A default Gateway profile is created at installation time. To verify that this profile exists, use the following procedure on *ps1*.

- **List all instances of Gateway.**

  ```
  # /opt/SUNWportal/bin/psadmin list-sra-instances -u amadmin -t gateway
  ```

  When prompted, type the *access-manager-admin-password*.

  The output shows the profile name, but no instances are yet listed:

  ```
  default:
  ```

▼ To Enable the portal service for SRA

The following procedure is performed on *ps1*, though it affects the portal service provided by both Portal Server instances..

1. **Check whether the portal service is enabled for secure remote access.**

   ```
   # /opt/SUNWportal/bin/psadmin get-sra-status -u amadmin
   ```

   When prompted, type the *access-manager-admin-password*.

   The following output shows that secure remote access is not enabled:

```
off
```

2 **Enable the portal service for secure remote access.**

```
# /opt/SUNWportal/bin/psadmin switch-sra-status -u amadmin on
```

When prompted, type the *access-manager-admin-password*.

This command toggles the SRA status of the portal service between disabled and enabled mode.

▼ To Provision the Gateway Profile

This procedure updates the Gateway profile with information such as the non-authenticated URL list.

- **Run the following command:**

```
/opt/SUNWportal/bin/psadmin provision-sra -u amadmin -p pstestPortal --console
--console-url http://ps.pstest.com/psconsole --gateway-profile default --enable
```

When prompted, type the *access-manager-admin-password*.

▼ To Verify the Updated Gateway Profile

This procedure is performed using the Portal Server Console (`psconsole`).

1 **Go to the following URL in a browser window:**

```
http://ps.pstest.com/psconsole
```

2 **Log in to the Portal Server Console by typing the following values and clicking Login.**

Input Field	Value
User ID	`amadmin`
Password	*access-manager-admin-password*

The Portal Server Console opens.

3 **Verify the URL path to portal in the Gateway profile.**

a. **Click the Secure Remote Access Tab.**

b. **Under the Profile section, click** `default`.

c. Under Basic Options, check that the Portal Server's URL path is set to the following:

`http://`*ps.pstest.com*`:80/portal`

4 Edit the list of URLs to Which User Session Cookie Is Forwarded.

The list should contain the following values:

`http://`*ps.pstest.com*
`http://`*ps1.pstest.com*
`http://`*ps2.pstest.com*
`http://`*am.pstest.com*
`http://`*am1.pstest.com*
`http://`*am2.pstest.com*

5 Check that Enable Cookie Management is on.

6 If you made changes, click Save at the bottom of the screen.

7 Click the Security tab.

8 Edit the list of URL's in the Non-authenticated URL box:

The list should contain the following values:

`http://`*am.pstest.com*`:80/amserver/UI/Login`
`http://`*am.pstest.com*`:80/amserver/UI/Logout`
`http://`*am.pstest.com*`:80/amconsole/console/css`
`http://`*am.pstest.com*`:80/amconsole/console/images`
`http://`*am.pstest.com*`:80/amconsole/console/js`
`http://`*am.pstest.com*`:80/amserver/css`
`http://`*am.pstest.com*`:80/amserver/images`
`http://`*am.pstest.com*`:80/amserver/js`
`http://`*am.pstest.com*`:80/amserver/login_images`
`http://`*ps.pstest.com*`:80/portal/images`
`http://`*ps.pstest.com*`:80/portal/desktop/images`
`http://`*ps.pstest.com*`:80/portal/desktop/tabs/images`
`http://`*ps.pstest.com*`:80/portal/desktop/css`
`http://`*ps.pstest.com*`:80/portal/console/images`
`http://`*ps.pstest.com*`:80/portal/netlet/jnlpclient.jar`
`http://`*ps.pstest.com*`:80/portal/netlet/netletjsse.jar`
`http://`*ps.pstest.com*`:80/portal/proxylet/jnlpclient.jar`
`http://`*ps.pstest.com*`:80/portal/proxylet/regx-win32-native.jar`

Configuring *ps1* for SRA Operation

This task involves enabling the Portal Server instance for SRA operations.

It also involves setting up the optional Netlet Proxy and Rewriter Proxy instances, which enable you to make full use of the functionality of the Gateway service. These components were installed on the computers running Portal Server when you implemented the Portal service module (see "To Install Portal Server on *ps1*" on page 132). You now need only to create instances of these components and configure Portal Server to interoperate with them.

The task consists of the following procedures:

- Set Up a Netlet Proxy Instance on *ps1*
- Set Up a Rewriter Proxy Instance on *ps1*
- Configure Portal Server to Interoperate With the Netlet Proxy and Rewriter Proxy Instances on *ps1*

▼ To Set Up a Netlet Proxy Instance on *ps1*

In this procedure you create and start a Netlet Proxy Instance on *ps1*.

1 Create a working copy of the Netlet Proxy configuration file.

```
# cp /opt/SUNWportal/samples/psconfig/example11.xml /tmp/nlp-ps1.xml
```

2 Edit your working copy of the configuration file in a text editor.

Locate the configuration parameters that are listed in the following table, and change their values to the values shown in the table.

Parameter	Value
@HOST.DOMAIN@	*ps1.pstest.com*
@LBHOST.DOMAIN@	*ps.pstest.com*
@PSHOST.DOMAIN@	*pstest.com*
@PORT@	`80`
@AMADMIN.PASSWORD@	*access-manager-admin-password*
@AMLDAPUSER.PASSWORD@	*access-manager-LDAP-password*
@DIRMGR.PASSWORD@	*directory-manager-password*
@NETLET.PROXY.PORT@	`10555`

Parameter	Value
@IPADDRESS@	*10.0.2.3*
@SRA.LOGUSER.PASSWORD@	*loguser-password*
Organization	*your-organization*
Division	*your-division*
StateProvince	*your-state*
CountryCode	*your-country*
CertificateDatabasePassword	*cert-DB-password*
@SRA.CERTDB.PASSWORD@	*cert-DB-password*

3 Create a Netlet Proxy instance.

```
# /opt/SUNWportal/bin/psconfig --config ./tmp/nlp-ps1.xml
```

4 Start the Netlet Proxy Instance.

```
# /opt/SUNWpoartal/bin/psadmin start-sra-instance -u amadmin -N default
-t nlproxy
```

When prompted, type the *access-manager-admin-password*.

▼ To Set Up a Rewriter Proxy Instance on *ps1*

1 Create a working copy of the Rewriter Proxy configuration file.

```
# cp /opt/SUNWportal/samples/psconfig/example12.xml /tmp/rwp-ps1.xml
```

2 Edit your working copy of the configuration file in a text editor.

Locate the configuration parameters that are listed in the following table, and change their values to the values shown in the table.

Parameter	Value
@HOST.DOMAIN@	*ps1.pstest.com*
@LBHOST.DOMAIN@	*ps.pstest.com*
@PSHOST.DOMAIN@	*pstest.com*
@PORT@	`80`
@AMADMIN.PASSWORD@	*access-manager-admin-password*

Parameter	Value
@AMLDAPUSER.PASSWORD@	*access-manager-LDAP-password*
@DIRMGR.PASSWORD@	*directory-manager-password*
@REWRITER.PROXY.PORT@	`10443`
@IPADDRESS@	*10.0.2.3*
@SRA.LOGUSER.PASSWORD@	*loguser-password*
Organization	*your-organization*
Division	*your-division*
StateProvince	*your-state*
CountryCode	*your-country*
@SRA.CERTDB.PASSWORD@	*cert-DB-password*

3 Create a Rewriter Proxy instance.

```
# /opt/SUNWportal/bin/psconfig --config ./tmp/rwp-ps1.xml
```

4 Start the Rewriter Proxy instance.

```
# /opt/SUNWpoartal/bin/psadmin start-sra-instance -u amadmin -N default
-t rwproxy
```

When prompted, type the *access-manager-admin-password*.

▼ To Configure Gateway Instances to Interoperate With the Netlet Proxy and Rewriter Proxy Instances on *ps1*

This procedure changes the Gateway profile to use the Netlet and Rewriter proxies on *ps1*.

1 Start a browser.

2 Go to the following URL:

```
http://ps.pstest.com/psconsole
```

The Portal Server Console (`psconsole`) opens.

3 Log in to the Portal Server Console by typing the following values and click Log in.

Input Field	Value
User ID	`amadmin`
Password	*access-manager-admin-password*

The Portal Server Console opens.

4 **Click the Secure Remote Access tab.**

5 **Modify the Gateway profile.**

In the Secure Remote Access tab, do the following:

a. **In the Profile section, click** `default`**.**

b. **Click the Deployment tab.**

c. **Locate the section for Rewriter Proxy and Netlet Proxy.**

d. **Click the checkbox that enables Rewriter Proxy.**

e. **Locate the Rewriter Proxy list.**

f. **Add** `https://`*ps1.pstest.com*`:10443` **to the list.**

g. **Click the checkbox that enables Netlet Proxy.**

h. **Locate the Netlet Proxy List.**

i. **Add** *ps1.pstest.com*`:10555` **to the list.**

j. **Click Save.**

Configuring *ps2* for SRA Operation

This task is the same as "Configuring *ps1* for SRA Operation" on page 166. It consists of the following procedures:

- Set Up a Netlet Proxy Instance on *ps2*
- Set Up a Rewriter Proxy Instance on *ps2*
- Configure Portal Server to Inter0perate With the Netlet Proxy and Rewriter Proxy Instances on *ps2*

▼ To Set Up a Netlet Proxy Instance on *ps2*

- **Repeat the procedure that appears in To Set Up a Netlet Proxy Instance on ps2, except for the following:**
  - Replace all occurrences of *ps1* with *ps2*.
  - Replace *10.0.2.3* with *10.0.2.4*.

▼ To Set Up a Rewriter Proxy Instance on *ps2*

- **Repeat the procedure that appears in "To Set Up a Rewriter Proxy Instance on *ps1*" on page 167, except for the following:**
  - Replace all occurrences of *ps1* with *ps2*.
  - Replace *10.0.2.3* with *10.0.2.4*.

▼ To Configure Gateway Instances to Interoperate With the Netlet Proxy and Rewriter Proxy Instances on *ps2*

- **Repeat the procedure that appears in "To Configure Gateway Instances to Interoperate With the Netlet Proxy and Rewriter Proxy Instances on *ps1*" on page 168, except for the following:**

  Replace all occurrences of *ps1* with *ps2*. In particular, replace *ps1.pstest.com* with *ps2.pstest.com*.

Setting Up the Gateway Service on *sra1*

This task consists of the following procedures:

- Install SRA Gateway on *sra1*
- Configure Access Manager SDK on *sra1*
- Create a Gateway Instance on *sra1*
- Start and Verify the Gateway Service on *sra1*

▼ To Install SRA Gateway on *sra1*

This procedure assumes that you are installing Portal Server SRA Gateway on Solaris 10 8/07 OS or later version. Hence, no operating system patches need to be installed. The Java ES installer evaluates the state of the operating system and indicates if you need to install a patch. If you are using versions of the operating system older than Solaris 10 8/07 OS, it is better to install any required patches before you begin the actual SRA Gateway installation procedure.

This procedure runs the installer in Configure Later mode. After installation is complete, you manually configure a Gateway instance.

The following procedure runs the Java ES installer without saving a state file. You can choose to run the installer and capture your input in a state file (`-saveState` *state-filename*). You could then use the state file to re-create the installation if, for example, you needed to reinstall SRA Gateway.

1 Download the Java ES software distribution to *sra1*.

The procedure is documented in "To Download the Software Distribution" on page 185.

2 Log in as root or become superuser.

```
# su -
```

3 Start the Java ES installer.

```
# cd /portdist_71u2/Solaris_sparc
```

```
# ./installer
```

This procedure uses the GUI installer. The installer can also be run in text mode by using the `-nodisplay` option.

The Welcome panel opens.

4 In the Welcome panel, click Next.

The Software License Agreement panel opens.

5 In the Software License Agreement Panel, review the license terms and click Yes, Accept License.

The Choose Software Components panel opens.

6 In the Choose Software Components panel, select the following components:

- Portal Server Secure Remote Access 7.1
  - Gateway
- Access Manager 7.1
  - Access Manager SDK

7 Click Next.

The Dependency Warning panel opens.

8 In the Dependency Warning panel, choose Use Directory Server Installed on a Remote Machine and click OK.

The installer evaluates the Java SE Software Development Kit on the computer and determines if an upgrade is required. On a fresh copy of Solaris 10 8/07 OS, an upgrade is needed, and the Java SE Software Development Kit Upgrade Required panel opens.

9 In the Java SE Software Development Kit Upgrade Required panel, select Automatic Upgrade to the Version Included with the Installer and click Next.

The installer evaluates the Java ES shared components on the computer and determines if any upgrades are required. On a fresh copy of the Solaris 10 8/07 OS, shared component upgrades are needed, and the Shared Components Upgrades Required panel opens.

10 In the Shared Components Upgrades Required panel, click Next.

The installer upgrades the shared components. The Specify Installation Directories panel opens.

11 In the Specify Installation Directories panel, type the following values and click Next.

Input Field	Value
Portal Server Secure Remote Access	`/opt`
Access Manager	`/opt`

The System Check panel opens.

12 In the System Check panel, evaluate the results of the system check.

If the system check is favorable, click Next.

The Choose a Configuration Type panel opens.

13 In the Choose a Configuration Type panel, select Configure Later and click Next.

The Ready to Install panel opens.

14 In the Ready to Install panel, indicate whether you want to open the software registration window during installation.

This panel enables you to register the components that you have selected for installation with Sun Connection. Sun Connection is a Sun-hosted service that helps you track, organize, and maintain Sun hardware and software. For example, Sun Connection can inform you of the latest available security fixes, recommended updates, and feature enhancements.

If you choose to register, information about the installation is sent to the Sun Connection database. You can also register at a later date, after installation has been completed.

15 Click Install.

The installer copies files to the computer.

16 When the installation is complete, review the installation in the Summary field.

17 Click Exit to exit the installer.

18 Check the installation log files for any installation errors.

```
# cd /var/sadm/install/logs
```

```
# egrep -i 'fail|error' Java*
```

19 Apply the patch to Portal Server 7.1 Update 2.

The following patch to Portal Server 7.1 Update 2 is needed for the Gateway service to interact with Portal Server through a firewall:

- Solaris SPARC: 124301–10
- Solaris x86: 124302–10
- Linux: 124303–10

The patch revision number (10) is the minimum required for this upgrade. If newer revisions become available, use the newer revisions instead of the preceding patch revisions.

a. Access the SunSolve℠ web site:

`http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

b. Search for the patch ID.

c. Download the patch to /*working-directory*.

d. Apply the patch.

`# patchadd /`*`working-directory/patch-ID`*

The `patchadd` command will instruct you to run `psupdate -a`, but you can safely skip this step.

e. Confirm that the patch upgrade was successful.

`# showrev –p | grep` *`patch-ID`*

The output should return the version of the patch ID that was applied in Step 18d.

▼ To Configure Access Manager SDK on *sra1*

Because Access Manager SDK was installed using the Configure Later option, you need to configure Access Manager SDK by modifying Access Manager configuration files. The standard approach for making these modifications is to run the `amconfig` command with an input file.

1 Change to the directory that contains the `amconfig` input file template, `amsamplesilent`.

```
# cd /opt/SUNWam/bin
```

2 Copy the template to a new file.

```
# cp amsamplesilent amconfigsra
```

3 In a text editor, edit the `amconfigsra` file to set the Access Manager SDK configuration parameters.

Locate the configuration parameters that are listed in the following table, and change their values to the values shown in the table.

Parameter	Value
DEPLOY_LEVEL	`3`
SERVER_HOST	*am.pstest.com*
SERVER_PORT	`80`
DS_HOST	*ds.pstest.com*
DS_DIRMGRPASSWD	*directory-manager-password*
ROOT_SUFFIX	`"dc=pstest,dc=com"`
SM_CONFIG_BASEDN	`$ROOT_SUFFIX`
ADMINPASSWD	*access-manager-admin-password*
AMLDAPUSERPASSWD	*access-manager-LDAP-password*
COOKIE_DOMAIN	*pstest.com*
AM_ENC_PWD	*password-enc-key*

4 Run the `amconfig` command with the input file you modified in Step 3.

```
# /opt/SUNWam/bin/amconfig -s amconfigsra
```

5 Verify that the Access Manager SDK is properly configured.

```
# /opt/SUNWam/bin/amadmin –u amadmin –m http://am.pstest.com:80
```

The output should show current session information.

▼ To Create a Gateway Instance on *sra1*

This procedure uses the `psconfig` command and a configuration file to create a Gateway instance. You begin with the appropriate configuration file as a template and edit the file to specify parameter values that are needed for the reference configuration.

1 Create a `config-`*sra1* configuration file.

Use the `example10.xml` file as a template.

```
# cd /opt/SUNWportal/samples/psconfig
# cp example10.xml config-sra1.xml
```

2 Open the `config-`*sra1*`.xml` file in a text editor.

3 Modify `config-`*sra1*`.xml` to use the values in the following table.

Parameter	Value
ConfigurationHostName	*sra1.pstest.com*
AdministratorUID	`amadmin`
AdministratorUserPassword	*access-manager-admin-password*
LDAPUserId	`amldapuser`
LDAPUserIdPassword	*access-manager-LDAP-password*
DirectoryManagerDn	`cn=Directory Manager`
directory-manager-password	*directory-manager-password*
PortalAccessURL	`http://`*ps.pstest.com*`:80/portal`
PrimaryPortalHost	*ps1.pstest.com*
Protocol	`https`
Host	*sra1.pstest.com*
Port	`443`
IPAddress	*10.0.4.1*
LogUserPassword	*log-user-password*
RestrictiveMode	`true`
Organization	*your-organization*
Division	*your-division*

Parameter	Value
CityOrLocality	*your-city*
StateProvince	*your-state*
CountryCode	*your-country*
CertificateDatabasePassword	*cert-DB-password*

The modified `config-sra1.xml` file is reproduced in "Example Configuration File: Gateway Instance on *sra1*" on page 192.

4 Run the `psconfig` command with the configuration file input.

```
# cd /opt/SUNWportal/bin
```

```
# ./psconfig --config /opt/SUNWportal/samples/psconfig/config-sra1.xml
```

The output should resemble the following:

```
Creating directory: /etc/opt/SUNWportal
Copying config templates from: /opt/SUNWportal/template/config
Successfully created PortalDomainConfig.properties file
Validating the Input Config XML File
Configuring Cacao Agent for Portal Software
Connecting to Cacao MBean Server

...
Closing MBean Server connection
Resetting log level
Configuration successful
```

▼ To Start and Verify the Gateway Service on *sra1*

1 Start the Gateway instance.

```
# /opt/SUNWportal/bin/psadmin start-sra-instance -u amadmin -N default -t
gateway --restrictive
```

When prompted, type the *access-manager-admin-password*.

2 Start a browser.

3 Go to the following URL:

`https://sra1.pstest.com`

You are prompted to accept the Gateway's self-signed certificate.

4 **Accept the certificate.**

The Access Manager login page opens.

5 **Log in to the Portal desktop by typing the following values and clicking Login.**

Input Field	Value
User ID	`developer`
Password	`developer`

If you successfully login, the Gateway is operating correctly.

Setting Up the Gateway Service on *sra2*

This task consists of the following procedures:

- Install SRA Gateway on *sra2*
- Configure Access Manager SDK on*sra2*
- Create a Gateway Instance on *sra2*
- Start and Verify the Gateway Service on *sra2*

▼ To Install Portal Server SRA Gateway on *sra2*

- **Repeat the procedure that appears in "To Install SRA Gateway on *sra1*" on page 170.**

  The installer's Configure Later option does not prompt you for configuration information.

▼ To Configure Access Manager SDK on *sra2*

- **Repeat the procedure that appears in "To Configure Access Manager SDK on *sra1*" on page 174.**

▼ To Create a Gateway Instance on *sra2*

- **Repeat the procedure in "To Create a Gateway Instance on *sra1*" on page 175, except for the following:**
  - Replace all occurrences of *sra1* with *sra2*.
  - Replace *10.0.4.1* with *10.0.4.2*.

▼ To Start and Verify the Gateway Service on *sra2*

- **Repeat the procedure in "To Start and Verify the Gateway Service on *sra1*" on page 176, except for the following:**

  Replace all occurrences of *sra1* with *sra2*.

Implementing Load Balancing for the Gateway Service

This task consists of the following procedures:

- Configure the Load Balancer for the Gateway Service
- Configure the Gateway Service on *ps1* for Load Balancing
- Configure the Gateway Service on *ps2* for Load Balancing
- Verify Load Balancing for the Gateway Service

▼ To Configure the Load Balancer for the Gateway Service

This procedure describes how to configure the Gateway service load balancer (*sra.pstest.com* at IP address *10.0.5.10*). The steps are relatively generic; the details depend on the load balancer you are using.

1. **Populate the load balancer's Hosts Table.**

   Add the IP address for *sra1.pstest.com* and *sra2.pstest.com* to the load balancer's hosts table.

2. **Populate the load balancer's Real Service Table.**

   Add the real services for *sra1.pstest.com* and *sra2.pstest.com*. A real service is identified by its IP address and port. Add *10.0.4.1*:443 and *10.0.4.2*:443.

3. **Populate the load balancer's Service Group Table.**

   Add the service group for Gateway services. The service groups are sets of the real services that you defined in Step 2. The real services in the group must be capable of fulfilling the same type of request. The load balancer will distribute requests among the real services in the service group. When you define the service group for the *sra.pstest.com*, you add the real services that specify the Gateway instances, *10.0.4.1*:443 and *10.0.4.2*.443.

4. **Set the load balancer to perform certificate authentication.**

   a. **Generate an SSL key and certificate request.**

      Use the certificate and key manager (CKM) on the load balancer.

b. Obtain a valid X.509 certificate.

Submit the certificate signing request (CSR) to an authorized certificate authority (CA). Alternatively, the load balancer might have a utility for generating a test certificate.

c. Install the X.509 certificate.

The method for installing the certificate depends on the load balancer.

5 Populate the load balancer's Virtual IP Table.

A virtual service definition includes the outward-facing IP address and the port at which the load balancer accepts requests for a service, as well as the service group that you specified in Step 3, which actually handles the requests. The load balancer will accept requests at the virtual service address and distribute them among the service group. The virtual service definition for the Gateway service should be *sra.pstest.com*, with the virtual IP address of *10.0.5.10*:443, and with the service group consisting of the computers *sra1.pstest.com* and *sra2.pstest.com*.

6 Configure the load balancer to use Layer-7 (HTTP layer) load balancing.

7 Configure the load balancer with a scheduling type of either least connections or round robin.

Both scheduling types initially distribute the connections evenly between the Portal Server instances. Both scheduling types keep the connections evenly distributed if the connections are restarted.

8 Configure the load balancer for sticky routing.

Although not mandatory, it is good practice to maintain a binding between the user's session and the Gateway instance that processed the user's initial request. Gateway instances keep, in cache, the Access Manager sessions that are associated with user requests. If there is no session persistence on the Gateway instances, user requests are routed randomly to the Gateway instances, and every instance caches every user session. This duplication creates additional network traffic whenever a Gateway instance needs to refresh a session. It also leads to unnecessary memory use by the Gateway instances.

Configuring the Gateway service load balancer to maintain session persistence with the Gateway instances will prevent these problems.

In the reference configuration, Internet users access the portal service over HTTPS connections to the Gateway service. When users connect over HTTPS, the requests, including any session persistence cookies that help a load balancer route the traffic to the correct instance, are encrypted. The information in the cookies is not available to the load balancer for routing purposes. The following are two ways of handling this situation in the Gateway service load balancer:

- SSL termination

If the load balancer supports SSL termination, then the load balancer can perform all the encryption and decryption work that is needed to terminate SSL at the load balancer. Terminating SSL at the load balancer reduces the load on the Gateway instances and improves performance. When the load balancer decrypts an HTTPS request, the session cookie is available to the load balancer. If the load balancer supports passive cookies, it can be configured to maintain session persistence. This approach is the preferred way to configure a load balancer for sticky routing.

If the load balancer does not support passive cookies, session persistence can be maintained by using the client IP address. However, if multiple users are using a web proxy to reach the Gateway service load balancer, the IP address that the load balancer will see is the IP address of the web proxy. In this case, all users who are using the same web proxy will be routed to the same Gateway instance, possibly resulting in an uneven load on the Gateway instances.

- SSL session ID

  If the load balancer does not support SSL termination, then it cannot read the session cookie. In this case, you can configure the load balancer to read the SSL session ID and make routing decisions based on the value of the session ID.

9 Configure the health-check settings for the load balancer.

The recommended settings are specified in Table 3–5.

▼ To Configure the Gateway Service on *sra1* for Load Balancing

If SSL sessions are terminated at the Gateway service load balancer, the traffic between the load balancer and the Gateway instances are plain HTTP. In that case, it is necessary to configure the Gateway instances to use the load balancer's virtual name (*sra.pstest.com*) and protocol (HTTPS) in all content that is rewritten.

You do so by configuring the attributes on the `platform.conf` file that is associated with the profile that the Gateway instance is using.

1 Open the `platform.conf` file on *sra1* in a text editor.

The file is located at:

```
/etc/opt/SUNWportal/platform.conf.default
```

2 Modify the following properties as follows:

```
gateway.enable.customurl=true
gateway.enable.accelerator=true
gateway.httpurl=https://sra.pstest.com:443
gateway.httpsurl=https://sra.pstest.com:443
gateway.virtualhost=sra.pstest.com  10.0.5.10
```

3 **Restart the Gateway instance on *sra1*.**

a. **Stop the Gateway instance on *sra1*.**

```
# /opt/SUNWportal/bin/psadmin stop-sra-instance -u amadmin -N default
-t gateway
```

When prompted, type the *access-manager-admin-password*.

b. **Start the Gateway instance on *sra1*.**

```
# /opt/SUNWportal/bin/psadmin start-sra-instance -u amadmin -N default
-t gateway --restrictive
```

When prompted, type the *access-manager-admin-password*.

4 **Verify that the Gateway instance is running in non-SSL mode.**

```
# telnet 10.0.4.1 443

GET / HTTP/1.1 <carriage return>

HOST:sra.pstest.com <carriage return>

Connection:Close <carriage return>

<carriage return>
```

The response should resemble the following:

```
HTTP/1.0 302 Moved Temporarily
Date: Fri. 08 Feb 2008 21:27:00 GMT
Server: Redirector
Location: https://sra.pstest.com/http://am.pstest.com/amserver/UI/Login?qw=&...
```

▼ To Configure the Gateway Service on *sra2* for Load Balancing

- **Repeat the procedure in except for the following:**
  - Replace all occurrences of *sra1* with *sra2*.
  - Replace *10.0.4.1* with *10.0.4.2*.

▼ To Verify Load Balancing for the Gateway Service

This procedure verifies that you can interact with Gateway instances through the load balancer and that the load balancer provides service failover when a Gateway instance fails.

1. **Start the Gateway instances on *sra1* and *sra2*, if they are not already running.**

   ```
   # /opt/SUNWportal/bin/psadmin start-sra-instance -u amadmin -N default
   -t gateway --restrictive
   ```

   When prompted, type the *access-manager-admin-password*.

2. **Start a browser.**

3. **Go to the Access Manager login page by using the load balancer URL**

   `http://sra.pstest.com`

   The Access Manager login page opens.

4. **Log in to the portal by typing the following values and clicking Login.**

Input Field	Value
User ID	developer
Password	developer

   The Developer Sample desktop opens, which confirms that the load balancer has routed the login request to one of the Gateway instances.

5. **Determine the Gateway instance handling the login request in Step 4.**

   a. **Open the log file on *sra1*.**

   ```
   # cd /var/opt/SUNWportal/logs/sra/default
   # tail –f portal.gateway.0.0.log
   ```

   b. **Open the log file on *sra2*.**

   ```
   # cd /var/opt/SUNWportal/logs/sra/default
   # tail –f portal.gateway.0.0.log
   ```

   c. **Note which log file displays more output.**

   Whichever Gateway instance is servicing the request will cause more output to be generated.

6. **Simulate a failure of the Gateway instance that was noted in Step 5.**

   In the configuration interface for your load balancer (*sra.pstest.com*), disable the Gateway instance that you identified in Step 5 (or otherwise remove it from the service group).

7 **Refresh the browser page.**

If service failover is working correctly, the Access Manager login page opens, confirming that the load balancer has routed the request to the remaining online Gateway instance.

8 **Recover the simulated failure of your original Portal Server instance.**

Return to the configuration interface for your load balancer, and replace the real server instance that you removed in Step 6 to the load balancer service group.

◆ ◆ ◆ APPENDIX A

Downloading the Software Distribution

The Portal Service on Application Server Cluster reference configuration uses features that are available in Portal Server 7.1, Update 2. This appendix provides instructions for downloading the Java Enterprise System (Java ES) 5 Update 1 software distribution, which contains Portal Server 7.1, Update 2 and all the other Java ES components that are used in the reference configuration.

Download Procedure

The following procedure describes how to download Java ES 5 Update 1 to a working directory, from which specific Java ES components can be installed.

▼ To Download the Software Distribution

The following steps can be performed on any computer on which you want to install Java ES components.

1. **Create a directory for the distribution.**

   ```
   # mkdir /portdist_71u2
   ```

2. **Navigate to the new directory.**

   ```
   # cd /portdist_71u2
   ```

3. **In a browser running on the local host go to the following URL:**

   `http://www.sun.com/software/javaenterprisesystem/getit.jsp`

4. **Click Get the Product Now.**

   The Sun's Software Portfolio page opens.

5 Select Sun Java Enterprise System and click Get Downloads & Media.

6 Log in to the Download Center.

Type your user ID and password, and click OK.

If necessary, create a new Download Center account and then log in.

The Download Center opens.

7 Review the license agreement and click Accept License Agreement.

The page acknowledges that you accepted the license agreement.

8 Locate and then click the link for the Solaris OS distribution.

- SPARC: If you are using SPARC hardware, as specified in the reference configuration, use the following link:

  `Java ES 5 Update 1 Integrated Installer, Multi-language [java_es-5u1-ga-solaris-sparc.zip].`

- x86: If you are using x86 hardware, use the following link:

  `Java ES 5 Update 1 Integrated Installer, Multi-language [java_es-5u1-ga-solaris-x86.zip]`

The download begins, and might take a few minutes to complete..

9 When the download completes, unzip the downloaded file.

```
# unzip /portdist_71u2/java_es-5u1-ga-solaris-arch.zip
```

where *arch* is either `sparc` or `x86`.

You are ready to begin the installation procedure.

◆ ◆ ◆ APPENDIX B

Configuration Files

This appendix provides the full content of configuration files that are used in implementing the Portal Service on Application Server Cluster reference configuration:

Example Configuration File: Portal Server Instance on *ps1*

The `config-`*ps1*`.xml` file that is used to configure the Portal Server Instance on *ps1* as part of implementing the portal service module follows, with parameter values as specified in "To Create a Portal Server Instance on *ps1*" on page 135.

```
<PortalServerConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="file:///opt/SUNWportal/lib/psconfig.xsd"
        SchemaVersion="1.0">
    <Configure ConfigurationHostName="ps1.pstest.com" SystemUser="root"
        SystemGroup="other" Validate="true">
        <SharedComponents
              JavaHome="/usr/jdk/entsys-j2se"
              CacaoProdDir="/usr/lib/cacao"
              CacaoConfigDir="/etc/cacao/instances/default"
              SharedLibDir="/usr/share/lib"
              PrivateLibDir="/usr/share/lib"
              JDMKLibDir="/opt/SUNWjdmk/5.1/lib"
              NSSLibDir="/usr/lib/mps/secv1"
              JSSJarDir="/usr/share/lib/mps/secv1"
              WebNFSLibDir="/opt/SUNWebnfs"
              DerbyLibDir="/opt/SUNWjavadb/lib"
              AntLibDir="/usr/sfw/lib/ant"
```

```
      AntHomeDir="/usr/sfw"
      RegistryLibDir="/opt/SUNWsrvc-registry/lib"
      MFWKLibDir="/opt/SUNWmfwk/lib"
      JAXLibDir="/opt/SUNWjax/share/lib"
/>
<AccessManager>
     <InstallationDirectory
           ProdDir="/opt/SUNWam"
           DataDir="/var/opt/SUNWam"
           ConfigDir="/etc/opt/SUNWam/config"
           ConfigFile="AMConfig.properties"
     />
     <UserCredentials
           AdministratorUID="amadmin"
           AdministratorUserPassword="access-manager-admin-password"
           LDAPUserId="amldapuser"
           LDAPUserIdPassword="access-manager-LDAP-password"
           DirectoryManagerDn="cn=Directory Manager"
           directory-manager-password="directory-manager-password"/>
</AccessManager>
<PortalConfiguration>
    <InstallationDirectory
           ProdDir="/opt/SUNWportal"
           DataDir="/var/opt/SUNWportal"
           ConfigDir="/etc/opt/SUNWportal"/>
    <ComponentsToConfigure>
           <component>portalserver</component>
    </ComponentsToConfigure>
    <SearchServer SearchServerID="search-inst-ps1">
       <WebContainerProperties
           Host="ps1.pstest.com"
           Port="5050"
           Scheme="http"
           WebContainerInstallDir="/opt/SUNWappserver/appserver"
           WebContainerInstanceName="as-inst-ps1"
           WebContainerDomainName="domain1"
           WebContainerInstanceDir="/var/opt/SUNWappserver/nodeagents/
               na-ps1/ns-inst-ps1"
           WebContainerDocRoot="/var/opt/SUNWappserver/nodeagents/
               na-ps1/as-inst-ps1/docroot"
           WebContainerAdminHost="ps1.pstest.com"
           WebContainerAdminPort="4849"
           WebContainerAdminScheme="https"
           WebContainerAdminUid="admin"
           WebContainerAdminPassword="app-server-admin-password"
           WebContainerMasterPassword="app-server-master-password"
           WebContainerType="SJSAS81"
       />
```

```
                </SearchServer>
                <PortalServer PortalAccessURL="http://ps.pstest.com:80/portal"
                            PrimaryPortalHost="ps1.pstest.com"
                            PortalID="pstestPortal"
                            PortalWebappURI="/portal"
                            SearchServerID="search-inst-ps1">
                   <SamplePortal<>
                       <Sample Name="DeveloperPortal"/>
                       <Sample Name="EnterprisePortal"/>
                       <Sample Name="CommunityPortal"/>
                   </SamplePortal>
                   <Instance InstanceID="ps-inst-ps1">
                       <WebContainerProperties
                           Host="ps1.pstest.com"
                           Port="80"
                           Scheme="http"
                           WebContainerInstallDir="/opt/SUNWappserver/appserver"
                           WebContainerInstanceName="pscluster"
                           WebContainerDomainName="domain1"
                           WebContainerInstanceDir="/var/opt/SUNWappserver/
                               nodeagents/na-ps1/as-cluster-inst-ps1"
                           WebContainerDocRoot="/var/opt/SUNWappserver/
                               nodeagents/na-ps1/as-cluster-inst-ps1/docroot"
                           WebContainerAdminHost="ps1.pstest.com"
                           WebContainerAdminPort="4849"
                           WebContainerAdminScheme="https"
                           WebContainerAdminUid="admin"
                           WebContainerAdminPassword="app-server-admin-password"
                           WebContainerMasterPassword="app-server-master-password"
                           WebContainerType="SJSAS81"
                        />
                    </Instance>
                </PortalServer>
            </PortalConfiguration>
        </Configure>
</PortalServerConfiguration>
```

Example Configuration File: Portal Server Instance on *ps2*

The `config-`*ps2*`.xml` file that is used to configure the Portal Server Instance on *ps2*, as part of implementing the portal service module, follows, with parameter values as specified in "To Create a Portal Server Instance on *ps2*" on page 141.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<PortalServerConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:noNamespaceSchemaLocation="file:///opt/SUNWportal/lib/psconfig.xsd"
```

```
            SchemaVersion="1.0">
        <Configure ConfigurationHostName="ps2.pstest.com" SystemUser="root"
            SystemGroup="other" Validate="true">
            <SharedComponents
                JavaHome="/usr/jdk/entsys-j2se"
                CacaoProdDir="/usr/lib/cacao"
                CacaoConfigDir="/etc/cacao/instances/default"
                SharedLibDir="/usr/share/lib"
                PrivateLibDir="/usr/share/lib"
                JDMKLibDir="/opt/SUNWjdmk/5.1/lib"
                NSSLibDir="/usr/lib/mps/secv1"
                JSSJarDir="/usr/share/lib/mps/secv1"
                WebNFSLibDir="/opt/SUNWebnfs"
                DerbyLibDir="/opt/SUNWjavadb/lib"
                AntLibDir="/usr/sfw/lib/ant"
                AntHomeDir="/usr/sfw"
                RegistryLibDir="/opt/SUNWsrvc-registry/lib"
                MFWKLibDir="/opt/SUNWmfwk/lib"
                MFWKBinDir="/opt/SUNWmfwk/bin"
                JAXLibDir="/opt/SUNWjax/share/lib"
            />
            <AccessManager>
                <InstallationDirectory
                     ProdDir="/opt/SUNWam"
                     DataDir="/var/opt/SUNWam"
                     ConfigDir="/etc/opt/SUNWam/config"
                     ConfigFile="AMConfig.properties"
                />
                <UserCredentials
                     AdministratorUID="amadmin"
                     AdministratorUserPassword="adminadm"
                     LDAPUserId="amldapuser"
                     LDAPUserIdPassword="adminadmin"
                     DirectoryManagerDn="cn=Directory Manager"
                     directory-manager-password="adminadm"/>
            </AccessManager>
            <PortalConfiguration>
                <InstallationDirectory
                     ProdDir="/opt/SUNWportal"
                     DataDir="/var/opt/SUNWportal"
                     ConfigDir="/etc/opt/SUNWportal"/>
            </PortalConfiguration>
        </Configure>
    </PortalServerConfiguration>
```

Example Display Profile: Session Counter Portlet

The following display profile is used to implement portlet session failover as documented in "To Set Up Session Failover for a Portlet" on page 152.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">

<DisplayProfile version="1.0" priority="user">
  <Properties/>
    <Channels>
      <Container name="JSPTabContainer" provider="JSPTabContainerProvider">
        <Properties/>
        <Available/>
        <Selected/>
        <Channels>
          <Container name="PortletSamplesTabPanelContainer">
            <Properties/>
            <Available>
               <Reference value="JSPTabContainer/
                PortletSamplesTabPanelContainer/CounterSession"/>
            </Available>
            <Selected>
               <Reference value="JSPTabContainer/
                PortletSamplesTabPanelContainer/CounterSession"/>
            </Selected>
            <Channels>
              <Channel name="CounterSession"
                 provider="__Portlet__sessioncounter.PortletSessionCounter"
                 merge="replace">
                   <Properties/>
              </Channel>
            </Channels>
          </Container>
        </Channels>
      </Container>
    </Channels>
    <Providers/>
</DisplayProfile>
```

Example Configuration File: Gateway Instance on *sra1*

The `config-sra1.xml` file that is used to configure the Gateway Instance on *sra1*, as part of implementing the Gateway service module, follows, with parameter values as specified in "To Create a Gateway Instance on *sra1*" on page 175.

```
<PortalServerConfiguration
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="file:///opt/SUNWportal/lib/psconfig.xsd"
        SchemaVersion="1.0">
    <Configure ConfigurationHostName="sra1.pstest.com" SystemUser="root"
            SystemGroup="other" Validate="true">
        <SharedComponents
            JavaHome="/usr/jdk/entsys-j2se"
            CacaoProdDir="/usr/lib/cacao"
            CacaoConfigDir="/etc/cacao/instances/default"
            SharedLibDir="/usr/share/lib"
            PrivateLibDir="/usr/share/lib"
            JDMKLibDir="/opt/SUNWjdmk/5.1/lib"
            NSSLibDir="/usr/lib/mps/secv1"
            JSSJarDir="/usr/share/lib/mps/secv1"
            WebNFSLibDir="/opt/SUNWportal/lib"
            DerbyLibDir="/opt/SUNWjavadb/lib"
            AntLibDir="/usr/sfw/lib/ant"
            AntHomeDir="/usr/sfw"
            MFWKLibDir="/opt/SUNWmfwk/lib"
            MFWKBinDir="/opt/SUNWmfwk/bin"
            JAXLibDir="/opt/SUNWjax/share/lib"
        />
        <AccessManager>
            <InstallationDirectory
                 ProdDir="/opt/SUNWam"
                 DataDir="/var/opt/SUNWam"
                 ConfigDir="/etc/opt/SUNWam/config"
                 ConfigFile="AMConfig.properties"
            />
            <UserCredentials
                 AdministratorUID="amadmin"
                 AdministratorUserPassword="access-manager-admin-password"
                 LDAPUserId="amldapuser"
                 LDAPUserIdPassword="access-manager-LDAP-password"
                 DirectoryManagerDn="cn=Directory Manager"
                 directory-manager-password="directory-manager-password"/>
        </AccessManager>
        <PortalConfiguration>
            <InstallationDirectory
                 ProdDir="/opt/SUNWportal"
                 DataDir="/var/opt/SUNWportal"
```

```
                ConfigDir="/etc/opt/SUNWportal"/>
            <ComponentsToConfigure>
                <component>gateway</component>
            </ComponentsToConfigure>
            <PortalServer PortalAccessURL="http://ps.pstest.com:80/portal"
                PrimaryPortalHost="ps1.pstest.com">
            </PortalServer>
            <Gateway Profile="default">
                <SRAInstance
                    Protocol="https"
                    Host="sra1.pstest.com"
                    Port="443"
                    IPAddress="10.0.4.1"
                    LogUserPassword="password"
                    RestrictiveMode="true"
                    StartInstance="false"/>
            </Gateway>
                <CertificateInformation
                    Organization="Sun Microsystems"
                    Division="Software"
                    CityOrLocality="Santa Clara"
                    StateProvince="CA"
                    CountryCode="US"
                    CertificateDatabasePassword="password"/>
        </PortalConfiguration>
    </Configure>
</PortalServerConfiguration>
```

◆ ◆ ◆ APPENDIX C

Provisioning Users for Portal Services

This appendix provides information about how to populate Directory Server with user entries that support the reference configuration. In particular, the procedures described in this appendix provision users consistent with the "User Management Specification" on page 60.

Attributes of Portal Service Users

By deploying he reference configuration, in particular the Access Manager module, in accordance with the procedures in this guide, you create an LDAP schema with some basic user attributes. In particular, new user accounts will be provisioned with the following attributes:

```
sn: username
cn: username
userPassword: *********
inetUserStatus: Active
uid: username
objectClass: iplanetpreferences
objectClass iplanet-am-managed-person
objectClass: top
objectClass: iplanet-am-user-service
objectClass: organizationalperson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: person
objectClass: sunamauthaccountlockout
objectClass: inetuser
iplanet-am-user-auth-config: (empty)
```

With these attributes alone, however, user accounts are not able to access the portal desktop or other portal services, such as the SRA Gateway, Netlet, or Proxylet services. To be authorized for portal services, a user entry must include the object classes that are shown in the following table.

TABLE C–1 Object Classes and Corresponding Services

objectClass	Corresponding Service
sunportalportal1desktopperson	portal1 Desktop
sunportalportalpksubscriptionperson	Access List
iplanet-amauth-configuration-service	Authentication Configuration
sunmobileappabperson	Mobile Address Book
sunmobileappcalendarperson	Mobile Calendar
sunmobileappmailperson	Mobile Mail
sunportalnetfileservice	NetFile
sunportalgatewayaccessservice	Gateway
sunportalnetletservice	Netlet
sunportalproxyletservice	Proxylet
sunssoadapterperson	SSO Adapter
sunportalportal1pksubscriptionsperson	portal1 Subscriptions

In provisioning users for portal services, objectClasses in the above table need to be added to all user entries, depending on the portal services desired by the user.

Provisioning Tool Choices

Several tools are available to perform the provisioning of users for portal services. These tools are described briefly, from the highest level, most general tools to the lowest level, most specific tools:

- Identity Manager

  Identity Manager is a set of tools that enable you to automate the provisioning and management of users in multiple user repositories. It can be used to provision an LDAP repository such as Directory Server using, for example, a data feed from a corporate human resources database. It provides users with a central password administrative service, and allows user credentials to be added to or removed from all applications when a new user joins or leaves a company. This tool is most likely to be used to provision users for portal services if Identity Manager is already being used company-wide as user provisioning and management framework.
- Delegated Administrator

Delegated Administrator provisions users for Messaging Server and Calendar Server. Because Delegated Administrator is Access Manager-based, it offers the ability to provision portal service users as well. Delegated Administrator is most likely be used to provision users for portal services at sites that have a combination of portal, messaging, and calendar services, such as Telcos.

- Access Manager tools

  Access Manager provides both GUI and command-line tools to provision users for Access Manager services, such as portal and related services. These tools are the simplest way of provisioning users if only Portal Server and Access Manager are used in a deployed solution. These tools are described in more detail in "Access Manager Provisioning Tools" on page 197.

- Directory-level commands

  Directory-level commands such as `ldapmodify` can be used to add user entries to an LDAP directory. At this level, the directory is not aware of Access Manager or Portal Server. All object and attribute creation must be performed manually.

Access Manager Provisioning Tools

Access Manager Console

The Access Manager Console is the simplest tool to use to provision individual users for portal services.

▼ To Provision a Single Portal Service User

The following procedure provisions a Developer Sample user, `dsuser1`, using the Access Manager Console.

1 Log in to the Access Manager Console if you are not already logged in.

a. Start a browser.

b. Go to the Access Manager Console login page using the load balancer URL:

`http://`*`am.pstest.com`*`/amconsole`

The Access Manager Console login page opens.

c. Log in to the Access Manager Console by typing the following values and clicking Login..

Input Field	Value
User ID	amadmin
Password	*access-manager-admin-password*

The Access Manager Console opens.

2 Click on the DeveloperSample link.

The link is found in the left pane under Organizations.

The DeveloperSample organization opens in the right pane.

3 View DelveloperSample users.

Select Users in the View pull-down menu in the left pane.

4 Define a new user.

a. Click New

The New User wizard opens in the right pane.

b. Enter the user name and password.

c. Select the services desired.

For example, if you select `portal1Desktop`, the new user will be able to log in and view the portal desktop.

d. Click Finish.

The New User wizard closes and the new user entry is saved.

`amadmin` Command

The `amadmin` command is the best tool to use to provision large numbers of users for portal services. Using this command-line option, you can write a script or create an input file that provisions any number of users.

▼ To Provision Multiple Portal Service Users

The following procedure provisions a Developer Sample user, `dsuser1`, using an XML input file to first create a user entry and then another input file to specify portal services for the user entry. Multiple users can be created by using this same procedure.

1 Create a new user entry for `dsuser1`.

a. Create an XML file that specifies the basic user attributes.

An example `CreateUserRequest.xml` file follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
    Copyright (c) 2005 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms.
-->
 <!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun Java System Access Manager 2005Q4 Admin CLI DTD//EN"
           "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>

<!--  CREATE REQUESTS -->
<Requests>
<PeopleContainerRequests DN="ou=People,o=DeveloperSample,dc=pstest,dc=com">
     <CreateUser createDN="dsuser1">
           <AttributeValuePair>
              <Attribute name="cn"/>
              <Value>dsuser1</Value>
           </AttributeValuePair>
           <AttributeValuePair>
              <Attribute name="sn"/>
              <Value>dsuser1</Value>
           </AttributeValuePair>
           <AttributeValuePair>
              <Attribute name="userPassword"/>
              <Value>dsuser1</Value>
           </AttributeValuePair>
     </CreateUser>
</PeopleContainerRequests>
</Requests>
```

b. Run the `amadmin` command with `CreateUserRequest.xml` as an input file.

```
# /opt/SUNWam/bin/amadmin -u amadmin -w password -t CreateUserRequest.xml
```

The output should resemble the following:

```
PeopleContainer: ou=People,o=DeveloperSample,dc=pstest,dc=com
Create Users:
uid=dsuser1,ou=People,o=DeveloperSample,dc=pstest,dc=com
Success 0: Successfully completed.
```

2 Add portal services to the `dsuser1` entry.

a. Create an XML file that specifies the portal services to add.

An example `AddUserServeice.xml` file follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
    Copyright (c) 2005 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms.
-->
<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun Java System Access Manager 2005Q4 Admin CLI DTD//EN"
           "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>

<!--  USER REQUESTS -->
<Requests>
   <UserRequests DN="uid=dsuser1,ou=People,o=DeveloperSample,dc=pstest,dc=com">
       <RegisterServices>
           <Service_Name>sunportalnetletservice </Service_Name>
           <Service_Name>sunportalproxyletservice </Service_Name>
           <Service_Name>sunportalgatewayaccessservice </Service_Name>
           <Service_Name>sunportalportal1desktopservice </Service_Name>
           <Service_Name>iplanet-am-auth-configuration-service </Service_Name>
       </RegisterServices>
   </UserRequests>
</Requests>
```

This input file adds the following portal services:

- Proxylet
- Access List
- portal Desktop
- Authentication Configuration
- Netlet

b. Run the `amadmin` command with `AddeUserServices.xml` as an input file.

```
# /opt/SUNWam/bin/amadmin -u amadmin -w password -t AddUserServices.xml
```

The output should resemble the following:

```
User: uid=dsuser1,ou=People,o=DeveloperSample,dc=pstest,dc=com
Registered services:
  sunportalproxyletservice
  ...
   sunportalnetletservice
Success 0: Successfully completed.
```

Index