Administrator's Guide

Sun™ ONE Portal Server

Version 6.2

816-6748-10
November 2003

Contents

About This Guide

This guide explains how to administer Sun™ ONE Portal Server 6.2. Sun ONE Portal Server provides a platform to create portals for your organization's integrated data, knowledge management, and applications. The Sun ONE Portal Server platform offers a complete infrastructure solution for building and deploying all types of portals, including business-to-business, business-to-employee, and business-to-consumer.

This preface includes the following sections:

- Who Should Read This Book
- What You Need to Know
- How This Book is Organized
- Document Conventions Used in This Guide
- Where to Find Related Information
- Where to Find This Guide Online

Who Should Read This Book

You should read this book if you are responsible for installing, administering, and configuring Sun ONE Portal Server at your site.

What You Need to Know

Before you administer Sun ONE Portal Server, you must be familiar with the following concepts:

- Basic Solaris™ administrative procedures
- LDAP
- Sun™ONE Directory Server
- iPlanet™ Directory Server Access Management Edition
- Sun™ONE Web Server

NOTE The Sun™ ONE family of products was previously branded under the iPlanet name. This product as well as others in the product family were branded and renamed shortly before the launch of this product. The late rebranding and renaming of products has resulted in a situation where some new product names have not been fully integrated into the shipping product. In particular, you will see the Sun™ ONE Identity Server referred to as the iPlanet Directory Server Access Management Edition within the GUI and within the product documentation. For this release, please consider Sun ONE Identity Server and iPlanet Directory Server Access Management Edition as interchangeable names for the same product.

How This Book is Organized

This book contains the following chapters and appendices:

- About This Guide (this chapter)
- Chapter 1, “Introduction to Administering the Sun™ ONE Portal Server”

  This chapter describes the Sun ONE Portal Server 6.2 architecture, protocols, and interfaces, and provides an overview of administering and customizing the product.

- Chapter 2, “Administering Authentication, Users, and Services”

  This chapter describes how to use Sun ONE Identity Server to administer authentication, users, and services.

- Chapter 3, “Configuring Delegated Administration”

  This chapter describes how to configure delegated administration for Sun ONE Portal Server.

- Chapter 4, “Administering the Portal Desktop Service”

This chapter describes how to administer the Sun ONE Portal Server Desktop service.

- Chapter 5, “Administering the Display Profile”

  This chapter describes how to administer the Sun ONE Portal Server display profile component.

- Chapter 6, “Administering the NetMail Service”

  This chapter describes how to administer the NetMail service.

- Chapter 7, “Administering the Rewriter Service”

  This chapter describes how to administer the Rewriter service.

- Chapter 8, “Administering the Search Engine Service”

  This chapter describes how to configure and administer the Search Engine service.

- Chapter 9, “Administering the Search Engine Robot”

  This chapter describes the Search Engine robot and its corresponding configuration files.

- Chapter 10, “The Pre-defined Robot Application Functions”

  This chapter describes the pre-defined robot application functions. You can use these functions to create and modify filter definitions.

- Chapter 11, “Administering the Subscriptions Service”

  This chapter describes how to configure and administer the Subscriptions service.

- Chapter 12, “Configuring the Communication Channels”

  This chapter provides information on the communication channels for Sun™ ONE Portal Server.

- Chapter 13, “Managing the Sun ONE Portal Server System”

  This chapter describes the various administrative tasks associated with maintaining the Sun ONE Portal Server system.

- Chapter 14, “Command-Line Utilities”

  This chapter describes the set of command-line utilities used with Sun ONE Portal Server.

- Appendix A, “Configuration Files”

This appendix provides a reference for the Sun ONE Portal Server configuration files.

- Appendix B, “XML Reference”

  This appendix provides the underlying XML that makes up the display profile and Rewriter Document Type Definitions (DTDs), and also the Desktop service definition.

- Appendix C, “Portal Desktop Attributes”

  This appendix provides a reference for the Desktop Service attributes.

- Appendix D, “NetMail Attributes”

  This appendix provides a reference for the NetMail Service attributes

- Appendix E, “Rewriter Attributes”

  This appendix provides a reference for the Rewriter Service attributes.

- Appendix F, “Search Attributes”

  This appendix provides a reference for the Search Engine Service attributes.

- Appendix G, “Subscriptions Attributes”

  This appendix provides a reference for the Subscriptions Service attributes.

- Appendix H, “SSO Adapter Templates and Configurations”

  This appendix provides a reference for the communication channels for Sun™ ONE Portal Server.

- Appendix I, “Schema Reference”

  This appendix provides a reference for the Sun ONE Portal Server schema definitions.

Document Conventions Used in This Guide

Monospaced Font

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

Bold Monospaced Font

Also, all paths specified in this manual are in Unix format. If you are using a Windows NT-based Sun ONE Portal Server, you should assume the Windows NT equivalent file paths whenever Unix file paths are shown in this book.

`Bold monospaced font` is used to represent text within a code example that you should type. For example, you might see something like this:

```
./pssetup
************************************************************************

Sun(TM) ONE Portal Server (6.0 release)

************************************************************************

Installation log at /var/sadm/install/logs/pssetup.13343/install.log

This product will run without a license. However, you must either
purchase a Binary Code License from, or accept the terms of a Binary
Software Evaluation license with, Sun Microsystems, to legally use
this product.

Do you accept? yes/[no] Starting install wizard in graphical mode
```

In this example, **`./pssetup`** is what you would type from the command line and the rest is what would appear as a result.

Italicized Font

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

Square or Straight Brackets

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

```
dpadmin [subcommands][options][arguments]
```

The presence of `[subcommands]`, `[options]`, and `[arguments]` indicates that there are optional parameters that may be added to the `dpadmin` command.

Command-Line Prompts

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

Common User Interface Verbs

Click instructs the user to press and release the mouse button when the onscreen pointer is on top of a UI element to invoke.

Double click instructs the user to click the mouse button twice in quick sucession.

Right click instructs the user to click the right mouse button when the pointer is on top of a UI element.

Select instructs the user to specify a choice among UI options (by highlighting, placing a checkmark in a checkbox, or clicking a radio button) in preparation for clicking OK or otherwise choosing to put your selections in action.

Choose instructs the user to pick a UI option that will immediately set the choice in motion, such as when you choose a menu item.

Type instructs the user to enter the appropriate typographic characters into a UI field.

Where to Find Related Information

In addition to this guide, Sun ONE Portal Server comes with supplementary information for administrators as well as documentation for developers. Use the following URL to see all the Sun ONE Portal Server documentation:

`http://docs.sun.com/prod/s1portalsrv`

Listed below are the additional documents released with the Sun ONE Portal Server 6.1 documentation suite:

- *Sun ONE Portal Server 6.1 Installation Guide*
- *Sun ONE Portal Server 6.1 Migration Guide*
- *Sun ONE Portal Server 6.1 Installation Guides*

- *Sun ONE Portal Server 6.1 Release Notes*
- *Sun ONE Portal Server, Secure Remote Access 6.1 Installation Guide*
- *Sun One Portal Server, Secure Remote Access 6.1 Administrator's Guide*
- *Sun One Portal Server, Secure Remote Access 6.1 Release Notes*

The following guides have not been updated for the Sun ONE Portal Server 6.1 release; however, the information contained in these documents is applicable to the Sun ONE Portal Server 6.1 product:

- *Sun ONE Portal Server 6.1 Desktop Customization Guide*
- *Sun ONE Portal Server 6.1 Developer's Guide*
- *Sun ONE Portal Server 6.1 Deployment Guide*

Where to Find This Guide Online

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

`http://docs.sun.com/prod/s1portalsrv`

Chapter 1

Introduction to Administering the Sun™ ONE Portal Server

Sun™ ONE Portal Server 6.2 product is a suite of integrated software products that allow enterprises to pull content from a variety of sources, personalize the content for a specific user or group of users, and aggregate content from these multiple sources into a single output format suitable for the specific user's device, such as a web browser.

This chapter provides basic information about the architecture of the product suite, the end user interface to the portal, the services implemented by the Sun ONE Portal Server software and how the are configured, and the tools used to administer the product. This chapter contains the following sections:

Architecture Overview

Sun ONE Portal Server is part of the Sun™ ONE architecture. Within the Sun ONE architecture, the Portal Server provides technologies that locate, connect, aggregate, present, communicate, personalize, notify, and deliver content. The content within Sun ONE is provided by web services. Portal Server does not provide web services itself. Rather, it is the mechanism by which a user interface is associated with web services and by which web services are made useful to people.

The Sun ONE Portal Server product architecture consists of a variety of integratablesoftware products.This allows the Sun ONE Portal Server to leverage functions and services from its internal components as well as external supporting products. The Sun ONE Portal Server itself includes the following internal components: Desktop, NetMail, Rewriter, and Search. External supporting products include the Sun™ONE Web Server, the Sun™ONE Directory Server, and Sun™ONE Identity Server (previously known as iPlanet™ Directory Server Access Management Edition). The Sun ONE Portal Server implements the web application container, user, service, and policy management, authentication and single sign-on, administration console, directory schema and data storage, and protocol support from these external products rather than implementing them in the Sun ONE Portal Server product itself. For example, the Sun ONE Portal Server product uses the Sun ONE Web Server as its default web container.

NOTE Although Sun ONE Portal Server uses the SunONE Web Server integrated with the Sun ONE Identity Server as its default web container (and uses its Java™ Development Kit for its Java™ run-time environment), the Sun™ ONE Application Server, IBM Websphere Application Server, and BEA Weblogic Application Server can also be used.

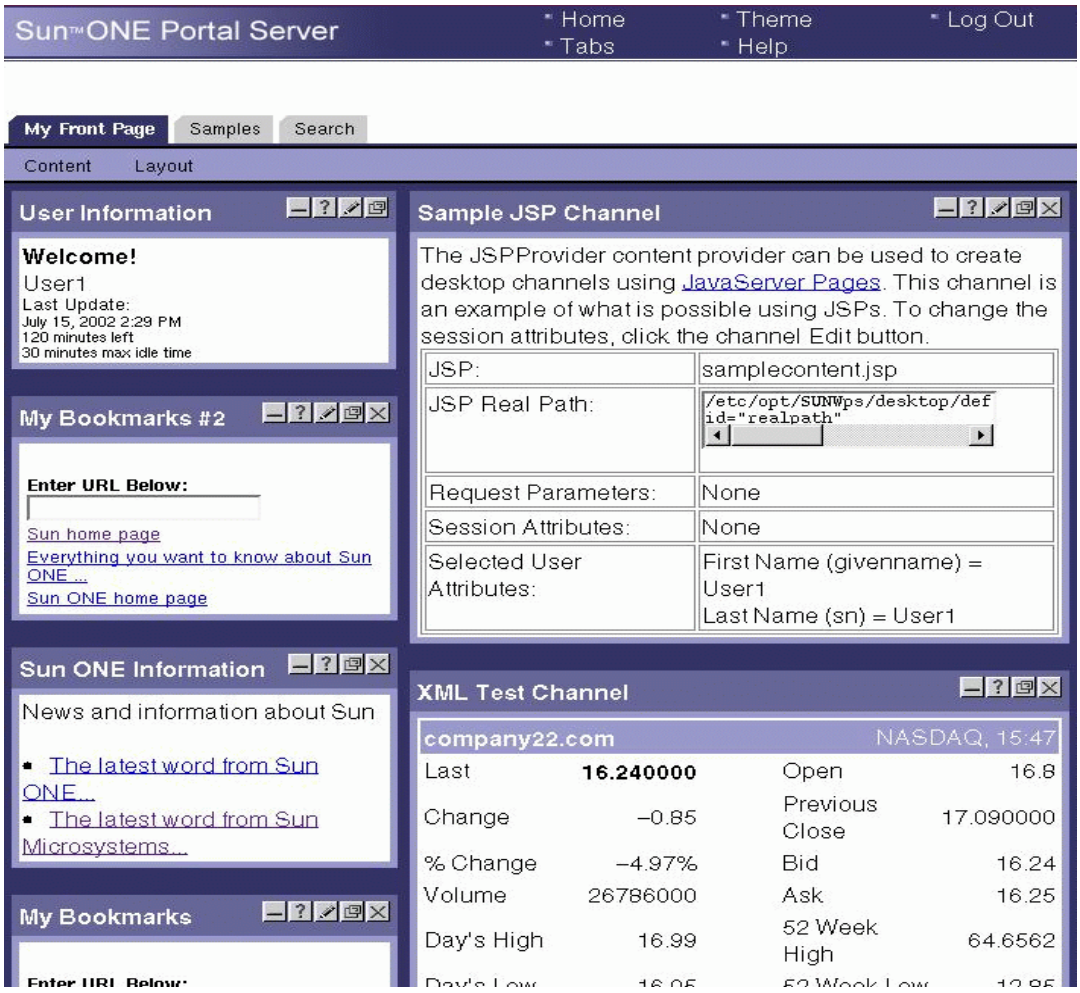
In addition, other Portal Server add-on software can be installed as well (for example, Sun™ ONE Portal Server: Secure Remote Access). Refer to the *Sun ONE Portal Server 6.1 Deployment Guide* for more information on the Sun ONE Portal Server architecture.

Portal Access Overview

Users typically access portal content through a web browser by requesting the URL for the portal's home page and authenticating through the Sun ONE Identity Server authentication service. Once authenticated, users are directed to the Sun ONE Portal Server Desktop.

Figure 1-1 on page 29 shows a sample Desktop from the Sun ONE Portal Server 6.2.

Figure 1-1 Sun ONE Portal Server Sample Desktop



The *Desktop* is the primary interface for the user to portal content. The Desktop service is implemented through a servlet, provider APIs, various channels, and various other support APIs and utilities. The Desktop uses programmatic entities called *providers* to generate content. A single unit of content is called a *channel.* Multiple channels of content can be aggregated together into *container channels* and arranged in a variety of formats such as tables or tabs on the Desktop. When a user accesses the portal, the Desktop references a *display profile* which stores content provider and channel data used to generate the user's content. As confusing as it may sound, the display profile does not actually define the overall layout, display, or organization of what users see on their Desktops. Fundamentally, the display

profile exists only to provide property values for channels. Actually, the Desktop uses multiple display profiles stored as LDAP attributes at various levels or nodes in the Sun ONE Directory Server (top-most, organization, role, and user levels) to determine the content for a user. XML documents are used to define the display profile properties for each level and upload the property values into the LDAP node. At runtime, a user's display profile is created by merging the display profile properties defined at each level. Although a display profile document can be defined at each level, you do not need to have a display profile document at each level.

To extend support to store and retrieve specific property values based on a given client type (such as HTML or MAPI), the Sun ONE Portal Server software includes:

- Conditional properties for defining the filtering criteria (see "Conditional Properties" on page 143) .
- The `authlessState` property for determining how the client is managed under authless authentication (see "Configuring Anonymous Authentication" on page 80).

Service Configuration Overview

The Sun ONE Portal Server is an Sun™ ONE application and, as such, its services are defined and managed using the Sun ONE Identity Server Service Management System (SMS). Service-related data that is not server-specific is defined using an Extensible Markup Language (XML) file that adheres to an SMS Document Type Definition (DTD). Server-specific data can be stored in properties files that are local to the specific server. Each Sun ONE Portal Server service (Desktop, Netmail, Rewriter, and Search) has its own XML and properties files for presenting and modifying service specific data.

Sun ONE Identity Server Services

As explained in Architecture Overview, the Sun ONE Portal Server implements many functions and services using supporting products from the Sun ONE architecture that are external to the Sun ONE Portal Server itself. In particular, while previous versions of the Portal Server implemented many administrative capabilities internally, integration with the Sun ONE Identity Server allows the Sun ONE Portal Server to leverage the following administrative tools and services from the Sun ONE Identity Server product:

- Administration Console

- Service Management
- User Management
- Authentication / Single Sign-On

See Chapter 2, "Administering Authentication, Users, and Services" for information on administering Sun ONE Identity Server services.

Sun ONE Portal Server Services

In addition to the standard Sun ONE Identity Server services, the Sun ONE Portal Server uses the Sun ONE Identity Server administration console to administer its internal services (Desktop, NetMail, Rewriter, and Search).

Desktop

As stated in the previous section, the Desktop provides the primary end-user interface for Sun ONE Portal Server. The Desktop is the mechanism for extensible content aggregation through the Provider Application Programming Interface (PAPI). The Desktop includes a variety of providers that enable container hierarchy and the basic building blocks for building some types of channels. For storing content provider and channel data, the Desktop implements a display profile data storage mechanism on top of an Sun ONE Identity Server service. You can edit the display profile and other Desktop service data through the administration console. Refer to Chapter 4, "Administering the Portal Desktop Service" and Chapter 5, "Administering the Display Profile" for information on administering the Desktop and the display profile.

Rewriter

The Rewriter provides a Java class library for rewriting URL references in various web languages such as HTML, JavaScript™, and WML, and in HTTP Location headers (redirections). The Rewriter defines an Sun ONE Identity Server service for storing rules that define how rewriting is to be done and the data to be rewritten. You can edit Rewriter rules through the administration console. Refer to Chapter 7, "Administering the Rewriter Service" for information on administering Rewriter.

Search Engine

The Search Engine service provides basic and advanced search and browse channels for the Desktop. It uses a robot to create resource descriptions for documents that are available in the intranet, and stores these resource descriptions in an indexed database. Resource descriptions (RDs) can also be imported from

another server or from a backup SOIF (Summary Object Interchange Format) file. The Search Engine includes Java and C APIs for submitting resource descriptions and for searching the database. The Search Engine database can also be used for storing other, arbitrary content, for example, a shared content cache for other content providers. You can edit Search Engine service data through the administration console. Refer to Chapter 8, "Administering the Search Engine Service" for information on administering Search.

NetMail

The NetMail service implements the NetMail (Java) and NetMail Lite email clients. These clients work with standard IMAP and SMTP servers. You can edit NetMail service data through the administration console. Refer to Chapter 6, "Administering the NetMail Service" for information on administering NetMail.

Configuration Mechanisms for Sun ONE Portal Server Services

The Sun ONE Portal Server uses a variety of configuration mechanisms to define, store and manage its services. This section contains five tables listing the configuration mechanisms used by each of the Sun ONE Portal Server internal services.

Table 1-1 on page 32 lists the configuration mechanisms for the Desktop service. The table is divided into two columns: Configuration Mechanism and Description. Configuration Mechanism lists the mechanisms and Description describes the purpose of the mechanism.

Table 1-1 Sun ONE Portal Server Desktop Configuration Mechanisms

Configuration Mechanisms	Description
Desktop Service Definition	Defines the Sun ONE Identity Server configuration attributes for the Desktop service. See Appendix B, "XML Reference" for more information.
Desktop Display Profile XML DTD	Defines the display configuration for the Desktop by defining provider and channel objects, and their properties. See Appendix B, "XML Reference" for more information.
Desktop Administration Console Module	Supplies the means by which you manage Sun ONE Portal Server services in the Sun ONE Identity Server framework. See Chapter 4, "Administering the Portal Desktop Service" for more information on administering the Desktop service configuration attributes. See Chapter 5, "Administering the Display Profile" for more information on administering the display profile.

Table 1-1 Sun ONE Portal Server Desktop Configuration Mechanisms

Configuration Mechanisms	Description
Desktop CLI	Supplies the `dpadmin` and `par` command utilities for product administration. See Chapter 14, "Command-Line Utilities" for more information.
Desktop Configuration Properties File	Defines the server-specific parameters for the Desktop service. See Appendix A, "Configuration Files" for more information.

Table 1-2 lists the configuration mechanisms for the Search service.The table is divided into two columns: Configuration Mechanism and Description. Configuration Mechanism lists the mechanisms and Description describes the purpose of the mechanism.

Table 1-2 Sun ONE Portal Server Search Configuration Mechanisms

Configuration Mechanisms	Description
Search Service Definition	Defines the Sun ONE Identity Server configuration attributes for the Search service. See Appendix I, "Schema Reference" for more information.
Search Administration Console Module	Supplies the means by which you manage Sun ONE Portal Server Search service data in the Sun ONE Identity Server framework. See Chapter 8, "Administering the Search Engine Service" for more information.
Search CLI	Supplies the `rdmgr`, `sendrdm`, and `StartRobot` command utilities for product administration. See Chapter 14, "Command-Line Utilities" for more information.
Search Configuration Properties File	Defines the server-specific parameters for the Search service. See Appendix A, "Configuration Files" for more information.
Robot Configuration Files	Define the behavior of the Search Engine robots. There are four robot configuration files. See Chapter 9, "Administering the Search Engine Robot" and Chapter 10, "The Pre-defined Robot Application Functions" for more information.

Table 1-3 lists the configuration mechanisms for the Rewriter service.The table is divided into two columns: Configuration Mechanism and Description. Configuration Mechanism lists the mechanisms and Description describes the purpose of the mechanism.

Table 1-3 Sun ONE Portal Server Rewriter Configuration Mechanisms

Configuration Mechanisms	Description
Rewriter Service Definition	Defines the Sun ONE Identity Server configuration attributes for the Rewriter service. See Appendix I, "Schema Reference" for more information.
Rewriter Rules XML DTD	See Appendix B, "XML Reference" for more information.
Rewriter Administration Console Module	Supplies the means by which you manage Sun ONE Portal Server Rewriter service data in the Sun ONE Identity Server framework. See Chapter 7, "Administering the Rewriter Service" for more information.
Rewriter CLI	Supplies the `rwadmin` command utility for product administration. See Chapter 14, "Command-Line Utilities" for more information.

Table 1-4 lists the configuration mechanisms for the NetMail service.The table is divided into two columns: Configuration Mechanism and Description. Configuration Mechanism lists the mechanisms and Description describes the purpose of the mechanism.

Table 1-4 Sun ONE Portal Server NetMail Configuration Mechanisms

Configuration Mechanisms	Description
NetMail Service Definition	Defines the Sun ONE Identity Server configuration attributes for the NetMail service. See Appendix I, "Schema Reference" for more information.
NetMail Administration Console Module	Supplies the means by which you manage Sun ONE Portal Server NetMail service data in the Sun ONE Identity Server framework. See the Chapter 6, "Administering the NetMail Service" for more information.

Administration Overview

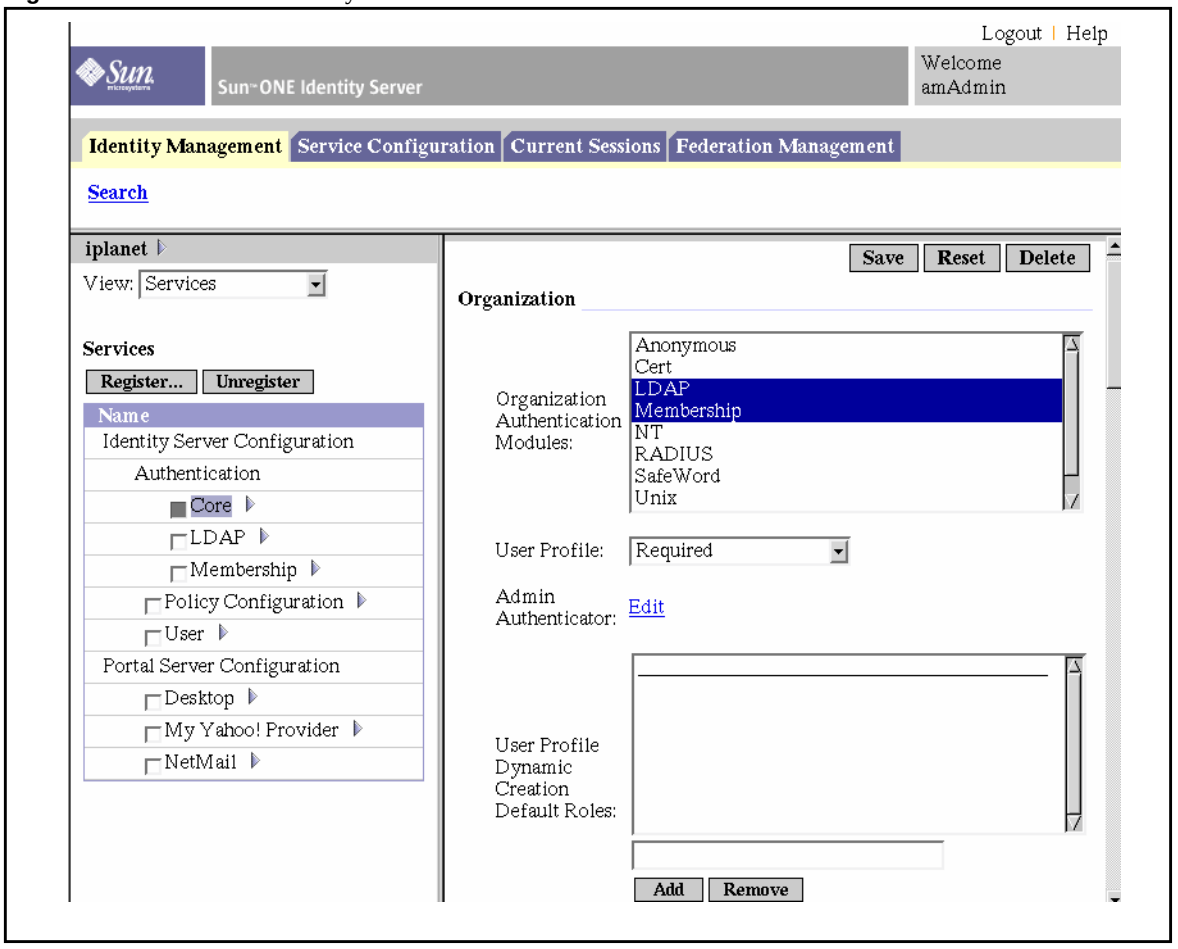
This section provides an overview of administering Sun ONE Portal Server, both from the Sun ONE Identity Server console and the command line.

Using the Sun ONE Identity Server Console

You administer Sun ONE Portal Server and Sun ONE Identity Server services through the HTML-based administration console provided by the Sun ONE Identity Server. Sun ONE Portal Server adds administration modules for Sun ONE Portal Server-specific services to extend the Sun ONE Identity Server console. See the individual chapters in this guide for information on the actual tasks you perform using the console.

The Sun ONE Identity Server console is divided into three sections: the location pane, the navigation pane and the data pane. Using all three panes, the administrator can navigate the directory, perform user and service configurations, and create policies. Figure 1-2 on page 36 shows the administration console.

Figure 1-2 Sun ONE Identity Server Administration Console



Location Pane

The location pane runs along the top of the console. The uppermost View menu allows the administrator to switch between the four different management views:

- Identity Management
- Service Configuration
- Current Session
- Federation Management

The Welcome field displays the name of the user that is currently running the console with a link to their user profile.

The Help link opens a browser window containing an HTML version of Appendixes C, D, E, and F of this documentation, the Attribute Reference Guide.

The Logout link enables the user to log out of the Sun ONE Identity Server console.

Navigation Pane

The navigation pane is the left portion of the console. The Directory Object portion is at the top of the pane and displays the name of the directory object that is currently open and its Properties link. The Show menu lists the directories under the selected directory object. Depending on the number of sub-directories, a paging mechanism is provided.

Data Pane

The data pane is the right portion of the console. Object attributes and their values are displayed and configured here. Entries are selected for their respective group, role or organization in this pane.

Using Command-Line Utilities

The Sun ONE Portal Server command-line interface consists of utilities provided by the Sun ONE Identity Server and Sun ONE Portal Server.

See Chapter 14, “Command-Line Utilities” for a complete list and syntax of Sun ONE Portal Server command-line utilities. Refer to the Sun ONE Identity Server product documentation for information on its command-line utilities

Chapter 2

Administering Authentication, Users, and Services

This chapter describes how to use Sun™ ONE Identity Server to administer authentication, users, and services. This chapter does not attempt to explain all aspects of Sun ONE Identity Server. Instead, it focuses on those aspects that pertain to Sun™ ONE Portal Server. See the Sun ONE Identity Server documentation for more information.

This chapter contains these sections:

Overview of Sun ONE Identity Server

In Sun™ ONE Portal Server 3.0 (formerly known as iPlanet™ Portal Server 3.0) implementationfs, you administer authentication methods, create domains, roles and users, and manage other data, such as profile attributes and logs, through the product itself. You also use the iPlanet Portal Server 3.0 APIs to develop custom applications.

Now, with Sun ONE Portal Server 6.2 product, you use Sun ONE Identity Server administrative capabilities and APIs formerly found within iPlanet Portal Server 3.0 itself. Sun ONE Identity Server is a set of tools that leverage the management and security potential of Sun™ ONE Directory Server. The goal of Sun ONE Identity Server is to provide an interface for managing user objects, policies, and services for organizations using the Sun ONE Directory Server.

Sun ONE Identity Server enables:

- Sun ONE Directory Server to perform user authentication and single sign-on, increasing data security.
- Administrators to initiate user entry management based on roles, an entry grouping mechanism which appears as an attribute in a user entry.
- Developers to define and manage the configuration parameters of a multitude of default and custom-made services.

You access all three of these functions through a graphical user interface, the web-based Sun ONE Identity Server admin console. In addition, the command-line interface, `amadmin`, enables you to perform batch administrative tasks on the directory server. For example, you can create, register, and activate new services; and create, delete, and read (get) organizations, people containers, groups, roles, and users.

Summary of Sun ONE Identity Server Features

Sun ONE Identity Server provides the following management components. Previously, these components resided within the Sun ONE Portal Server 3.0 framework itself.

- **User Management**—Creates and manages user-related objects (user, role, group, people container, organization, suborganization, and organizational unit objects). These can be defined, modified, or deleted using either the Sun ONE Identity Server console or the command-line interface.

- **Authentication**—Provides a plug-in solution for user authentication. The criteria needed to authenticate a particular user is based on the authentication service configured for each organization in the Sun ONE Portal Server enterprise. Before being allowed access to a Sun ONE Portal Server session, a user must pass through authentication successfully.
- **Single Sign-On**—Once the user is authenticated, the Sun ONE Identity Server API for Single Sign-On (SSO) takes over. Each time the authenticated user tries to access a protected page, the SSO API determines whether the user has the permissions required based on their authentication credentials. If the user is valid, access to the page is given without additional authentication. If not, the user will be prompted to authenticate again.
- **Service Management**—Specifies configuration parameters for default and custom-made services, including those for the Sun ONE Portal Server product itself (Portal Desktop, Rewriter, Search, and NetMail).
- **Policy Management**—Defines, modifies, or removes the rules that control access to business resources. Collectively, these rules are referred to as policy. Policies can be role-based or organization-based and can offer privileges or define constraints.

Comparison: Portal Server 3.0 and Portal Server 6.2

Table 2-1 on page 42 provides an overview to the major changes that have taken place to the Portal Server product. Many functions and features that previously were part of the Sun ONE Portal Server 3.0 (formerly iPlanet Portal Server 3.0) product are now part of Sun ONE Identity Server. In the table, the first column lists a concept or term, the second column defines the function or feature for that term in the Sun ONE Portal Server 3.0 product, the third column describes the corresponding feature or function in the Sun ONE Portal Server 6.2 product.

Table 2-1 Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2 Comparison

Concept or Term	Sun ONE Portal Server **3.0**	**Sun ONE Portal Server 6.2**
Role tree	A hierarchy you configure within Sun ONE Portal Server 3.0 to organize users and applications. The four levels of the role tree are: • root • domain • role • user	Concept of role tree no longer applies. Instead, because Sun ONE Identity Server leverages the capability of Sun ONE Directory Server, you use the Directory Information Tree (DIT) to organize your users, organizations, suborganizations, and so on.
Domain / Organization	A top-level grouping of users with common interests, such as employees or customers. Note that this is not a DNS domain, but a means that Sun ONE Portal Server 3.0 uses to group users into logical communities.	Concept of domain no longer applies. Instead, the Sun ONE Identity Server *organization* represents the top level of a hierarchical structure used by an enterprise to manage its departments and resources. Upon installation, Sun ONE Identity Server asks for the root suffix, and the default is derived from the domain name (for example, for the domain sun.com, the default is dc=sun, dc=com). Additional organizations can be created after installation to manage separate enterprises. All created organizations fall beneath the top-level organization. Within these sub organizations other suborganizations can be nested. There is no limitation on the depth to the nested structure.

Table 2-1 Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2 Comparison *(Continued)*

Concept or Term	Sun ONE Portal Server 3.0	Sun ONE Portal Server 6.2
Role	Divides the members of a domain according to function. The role contains a set of attributes and policies that define a user's Desktop policy.	Contains a privilege or set of privileges that can be granted to a user or users. This includes access and management of identity information stored in Sun ONE Directory Server and access to privileges protected by the Sun ONE Identity Server policy module. A Sun ONE Identity Server role also has associated with it a profile, which is stored in the class-of-service template. Role is defined differently in Sun ONE Identity Server and it includes the ability for a single user to have multiple roles, which was previously not supported. The privileges for a role are defined in access control instructions (ACIs). The Sun ONE Identity Server includes several predefined roles. The Sun ONE Identity Server Console allows you to edit a role's ACI to assign access privileges within the Directory Information Tree.

Table 2-1 Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2 Comparison *(Continued)*

Concept or Term	Sun ONE Portal Server 3.0	Sun ONE Portal Server 6.2
Attribute	Supports two types of attributes: global and user-configurable. Global attributes apply to the entire platform and are configured only by the Super Administrator. User-configurable attributes apply to underlying levels of the role tree, as described in the following sections. A delegated Domain Administrator can configure these attributes for the domain, parent role, child role, and user levels. At the user level of the role tree, some attributes can be customized for each user, as needed.	Makes use of Sun ONE Identity Server attributes, which can be one of the following types: • Global— The values applied to the global attributes are applied across the Sun ONE Identity Server configuration and are inherited by every configured organization. • Dynamic—A dynamic attribute can be assigned to an Sun ONE Identity Server configured role or organization. When the role is assigned to a user or a user is created in an organization, the dynamic attribute then becomes a characteristic of the user. • Organization—These attributes are assigned to organizations only. In that respect, they work as dynamic attributes. They differ from dynamic attributes, though, as they are not inherited by entries in the subtrees. • User—These attributes are assigned directly to each user. They are not inherited from a role or an organization and, typically, are different for each user. • Policy—Policy attributes are privilege attributes. Once a policy is configured, they may be assigned to roles or organizations. That is the only difference between dynamic and policy attributes; dynamic attributes are assigned directly to a role or an organization and policy attributes are used to configure policies and then applied to a role or an organization.

Table 2-1 Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2 Comparison *(Continued)*

Concept or Term	Sun ONE Portal Server 3.0	Sun ONE Portal Server 6.2
Policy	Configures portal access policies to applications, the Desktop, NetFile, Netlet, and so on.	Rules that define who can do what to which resource. The Sun ONE Identity Server Policy Service allows an organization to set up these rules or policies. In general, policy is created at the organization (or suborganization) level to be used throughout the organization's tree. In order to create a named policy, the specific policy service must first be registered to the organization under which the policy will be created. In Sun ONE Identity Server 6.0, the policy service consists only of lists of URLs that are allowed or denied. This is not sufficient for Portal Server to build a policy-based Desktop for content. This is why policy for channel access is built into the display profile for the Desktop. The Portal Server 6.2 Desktop supports a display profile that allows list of channels to be merged from several roles. If, for example, you have 25 roles, each with a handful of channels associated with that role, users can be configured to have any number of those roles, and the Desktop they get will then provide the aggregation of all those roles. Merge semantics control how channels from the various roles are aggregated or merged. For the purpose of merging display profiles, a hierarchical ordering is imposed on the roles in the Portal Server. The merge begins with the lowest priority document (lowest number) and proceeds in increasing priority number, until it arrives at the user (base), the highest priority profile. See Chapter 5, "Administering the Display Profile" for information on merging display profiles.

Table 2-1 Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2 Comparison *(Continued)*

Concept or Term	Sun ONE Portal Server 3.0	Sun ONE Portal Server 6.2
Component / Service	The four major components of Portal Server 3.0 are the server itself, the profile server, the gateway, and the firewall.	Component has been replaced by Sun ONE Identity Server service, which is group of attributes defined under a common name. The attributes define the parameters that the service provides to an organization. Sun ONE Identity Server is the service framework. Sun ONE Portal Server 6.2 relies on Sun ONE Identity Server to provide core services, such as authentication, user management, and policy management, as well as for the framework to run Portal Server specific services (Desktop, NetMail, Rewriter, and Search).
Administrative interfaces	Provides its own admin console to administer only Portal Server 3.0 components. The command-line interface is `ipsadmin`.	Uses the Sun ONE Identity Server admin console to administer Sun ONE Identity Server services, users, and policy, as well as Sun ONE Portal Server specific services (Desktop, NetMail, Rewriter, and Search.) The command-line interfaces that replace `ipsadmin` are `amadmin`, `dpadmin`, and `rwadmin`.

Comparison: Portal Server 6.0 and Portal Server 6.2

Table 2-2 on page 47 provides an overview to the changes that have taken place between the Portal Server 6.0 product and Portal Server 6.1 product. In the table, the first column lists a concept or term, the second column defines the function or feature for that term in the Sun ONE Portal Server 6.0 product, the third column describes the corresponding feature or function in the Sun ONE Portal Server 6.2 product.

Table 2-2 Sun ONE Portal Server 6.0 to Sun ONE Portal Server 6.2 Comparison

Concept or Term	Sun ONE Portal Server 6.0	Sun ONE Portal Server 6.2
Policy	Assign a policy to users. Once a policy has been named and created, it can be assigned to the organization or role. Assigning a policy at the organization level makes its attributes available to all entries in the organization. Assigning policy to a role makes its attributes available to all users who contain the role attribute.	Delegate an organization's policy definitions and decisions to another organization. (Alternately, policy decisions for a resource can be delegated to other policy products.) A referral policy controls this policy delegation for both policy creation and evaluation. Create a normal policy to define access permissions. A normal policy can consist of multiple rules, subjects, and conditions.
Authentication menu	The authentication menu configuration feature provided by the Sun ONE Identity Server 5.1 administration console supports a menu of authentication modules selected by the user.	If you need to configure a selectable list of valid authentication modules, use the Sun ONE Identity Server administration console to set each authentication module with the same value in the authentication level attribute. Refer to Chapter 2, "Administering Authentication, Users, and Services" for information on configuring authentication modules.

Sun ONE Identity Server Constraints

When using Sun ONE Identity Server, the following constraints apply:

- The predefined Sun ONE Identity Server roles cannot span multiple parallel organizations; however, a role can be assigned to a user who resides in a child organization of the organization that the role is associated with. In addition, access to resources in multiple domains can also be enabled by creating a custom role and defining the necessary Access Control Instructions (ACIs) to grant the role the privileges required.
- A user must belong to an organization and can only belong to that organization.
- Hierarchical roles are not supported. For example, you cannot create role C as equal to the sum of role A and role B, and have a user with role C have access to the resources in Role A, without being explicitly assigned to role A.
- The access permission for the `RoleAdministratorRole` can only be configured through editing corresponding ACIs directly.

- When role administrators (delegated administrators) log in to the Sun ONE Identity Server admin console, they can see all the roles and their associated services and properties under the same organization even if the role administrators don't have the permission to modify them.

Sun ONE Identity Server Interfaces

Sun ONE Identity Server Admin Console

This browser-based console provides a graphical user interface to manage the Sun ONE Identity Server enterprise, including Sun ONE Portal Server services. The admin console has default administrators with varying degrees of privileges used to create and manage the services, policies and users. (Additional delegated administrators can be created based on roles.) See Chapter 3, "Configuring Delegated Administration" for more information.

The Sun ONE Identity Server admin console is divided into three sections: the location pane, the Navigation pane and the Data pane. By using all three panes you navigate the directory, perform user and service configurations, and create policies.

See Chapter 1, "Introduction to Administering the Sun™ ONE Portal Server" for more information.

Sun ONE Identity Server Command-Line

The Sun ONE Identity Server command-line interface is `amadmin`, to administer the server, and `amserver`, to stop and start the server process. `amadmin` is also used to load XML service files into the directory server and perform batch administrative tasks on the directory tree.The Sun ONE Portal Server 3.0 command-line interfaces, `ipsadmin` and `ipsserver` are no longer used.

For more information on `amadmin`, see the Sun ONE Identity Server documentation.

Logging In to the Sun ONE Identity Server Admin Console

You can log in to the Sun ONE Identity Server console in two ways:

- Using a Specific URL

- Through HTTPS

When you log in to the admin console, the capabilities that are presented to you depend on your access permissions. Access permissions are determined based on the ACIs or roles assigned to you. For example, the superuser sees all of the admin console's functionality; a delegated administrator might only see a subset of this functionality, perhaps for a suborganization; end users see only the user attributes pertaining to their particular user ID.

Currently, there are two URLs available for logging in to the admin console:

- `http://`*`host`*`:`*`port`*`/amconsole/`
- `http://`*`host`*`:`*`port`*`/amserver/`

The `/amconsole` URL explicitly requests the HTML pages for the Sun ONE Identity Server admin console. If you log in using `/amconsole`, it brings up the admin console and then you'll see the URL change to `/amserver/UI/login` so the user can authenticate. Regardless of the configuration, this URL can be used to access the admin console.

The `/amserver` URL requests the HTML pages for the Sun ONE Identity Server service. Although the default set up when Sun ONE Portal Server is installed is to redirect this URL to log in to the admin console, because the `/amserver` URL accesses the Sun ONE Identity Server service this URL can be used to make other services besides the console available. For example,

- If a user accesses an application with an invalid session, an application may redirect the `/amserver` URL request to `amserver/UI/login` with the goto parameter. For example, the Sun ONE Portal Server Desktop does this as well as the Sun ONE Identity Server agent.
- A customer may direct users to `amserver/UI/login` as their starting point into some application or portal. Their default redirect URL could then be some portal application or custom application.
- A custom application could directly call the `amserver/UI/login` to authenticate.

To log in to the Sun ONE Identity Server admin console

- Using a specific URL:
  Type `http://`*`host`*`:`*`port`*`/amserver/`
  or
  Type `http://`*`host`*`:`*`port`*`/amconsole/`
- Using HTTPS:
  Type `https://`*`host`*`:`*`ssl_port`*`/amconsole/`

Configuring Log in to the Admin Console Using an IP Address

You cannot log in to the Sun ONE Identity Server admin console by using the server's IP address. This is because of the cookie domain settings in Sun ONE Identity Server.

However, you can add the local host's IP address to the list of Cookie Domains on the admin console.

1. Select Service Configuration from the location pane.
2. Click Platform.
3. Add your local host's IP address to Global.

You should now be able to access the admin console with IP address, rather than the domain name.

Viewing Basic Information

A script is available to enable you to display basic information about the product such as the version, build date of the Sun ONE Portal Server as well as the version and build date for the jar file. The version script is installed in *portal-server-installation-root*`/SUNWps/bin` directory where *portal-server-installation-root* is the base directory in which you installed the Sun ONE Portal Server. The default is `/opt`.

To view product information:

1. Change directories to the directory where the script is installed. That is:

   `cd` *portal-server-installation-root***`/SUNWps/bin`**

2. To view information about the Sun ONE Portal Server, type

   `./version`

3. To view information about the jar file on the Sun ONE Portal Server, type

   `./version` *jar-file*

   where *jar-file* is the name of the jar file.

Starting and Stopping Sun ONE Portal Server

This section describes how to stop and start Sun ONE Portal Server. Because Sun ONE Portal Server depends on Sun ONE Identity Server, you do not start and stop Sun ONE Portal Server directly. You need to restart the Sun ONE Identity Server server itself.

- To start Sun ONE Portal Server, enter:
  `/etc/init.d/amserver start`
- To start multiple instances Sun ONE Portal Servers, enter:
  `/etc/init.d/amserver startall`
- To stop Sun ONE Portal Server, enter:
  `/etc/init.d/amserver stop`

NOTE You do not need to stop the server to restart it. If you start a server that is already running, the server is stopped and restarted.

These instructions may vary with the web container. See the *Sun ONE Portal Server 6.1 Developer's Guide* for more information.

The Sun ONE Portal Server supports various platform locales. To start the Sun ONE Portal Server with a value other than the installed default see the *Sun ONE Portal Server 6.1 Developer's Guide.*

Managing Sun ONE Identity Server Services

This section provides an introduction to Sun ONE Identity Server services used by Sun ONE Portal Server. See the Sun ONE Identity Server documentation for complete information.

Installation and Sun ONE Web Server Packaging

- The Sun ONE Portal Server installer executes the Sun ONE Identity Server installer if the Sun ONE Identity Server has not previously been installed.

- Sun ONE Portal Server shares the web container with Sun ONE Identity Server. The web container specifies a runtime environment for Web components including concurrency, deployment, life cycle management, security, transaction, and other services.
- Sun ONE Portal Server uses the JVM™ and other components that Sun ONE Identity Server provides.

User Management

- Sun ONE Portal Server stores its profile information in Sun ONE Identity Server using the Sun ONE Identity Server APIs.
- Sun ONE Portal Server leverages multi-role support in Sun ONE Identity Server.
- Sun ONE Portal Server uses open and non-proprietary standard schema attributes, for example, `givenName`.
- Sun ONE Identity Server provides direct access to the LDAP directory.

Single Sign-On/Authentication

- In Sun ONE Portal Server 6.2, the authentication is managed by Sun ONE Identity Server.
- Sun ONE Identity Server provides all the authentication modules.
- Sun ONE Portal Server uses Sun ONE Identity Server policy attributes to restrict access.

Service Management

Sun ONE Portal Server 6.2 defines the following Sun ONE Identity Server services:

- Desktop—Provides the portal front-end and is the primary end user interface to the portal.See Chapter 4, “Administering the Portal Desktop Service” for information on setting up and administering the Portal Desktop.
- NetMail—Accesses the IMAP and SMTP mail servers in the Internet and allows users to access mail through the portal. See Chapter 6, “Administering the NetMail Service” for information on setting up and administering NetMail.

- Rewriter—Implements rules set up by the administrator to rewrite URLs to provide appropriate access. See Chapter 7, "Administering the Rewriter Service" for information on setting up and administering the Rewriter.
- Search—Provides a search capability for the Sun ONE Portal Server including basic and advanced search channels of the available documents. See Chapter 8, "Administering the Search Engine Service" for information on setting up and administering the Search service.

Managing Sun ONE Portal Server Users

The Directory Information Tree (DIT) organizes your users, organizations, suborganizations, and so on into a logical or hierarchical structure that enables you to efficiently administer and assign appropriate access to the users assuming those roles or contained within those organizations. This section provides information to help you plan the directory structure or tree underlying your portal server implementation by providing information about the functions and capabilities of organizations, suborganizations, and roles, and also providing procedures for creating and managing organizations, roles, and users.

NOTE Sun ONE Portal Server 6.2 supports organizations; previously, Sun ONE Portal Server 3.0 used the concept of domains.

The top of the organization tree in Sun ONE Identity Server is specified at install time. Additional organizations can be created after installation to manage separate enterprises. All created organizations fall beneath the top-level organization. Within these suborganizations other suborganizations can be nested. There is no limitation on the depth to the nested structure.

NOTE The top of the tree does not have to be called `isp`. It can be called anything. But with a tree organized with a generic top, for example, `isp`, then organizations within the tree can share roles.

Roles are a new grouping mechanism that are designed to be more efficient and easier to use for applications. Each role has members, or entries that possess the role. As with groups, you can specify role members either explicitly or dynamically. The roles mechanism automatically generates the `nsRole` attribute containing the DN of all role definitions in which the entry is a member. Each role contains a privilege or set of privileges that can be granted to a user or users. In Sun

ONE Portal Server 6.2, multiple roles can be assigned to a single user. The privileges for a role are defined in Access Control Instructions (ACIs). The Sun ONE Portal Server includes several predefined roles. The Sun ONE Identity Server console allows you to edit a role's ACI to assign access privileges within the Directory Information Tree. Built-in examples include `Top-level Admin Role` and `Top-level Help Desk Admin Role`. You can create other roles that can be shared across organizations.

Planning Organizations, Suborganizations, and Roles

As you plan your DIT structure, you need to decide whether to use a hierarchical or flat tree structure. As a general rule, you should strive to make your tree as flat as possible. However, as the size of your organization grows, a certain amount of hierarchy is important to facilitate granting and managing user access. The three key structural entities in Sun ONE Identity Server for building your DIT structure are organizations (or suborganizations), roles, and users. Before you plan your structure, you should understand the functions, characteristics, and interrelationships of each of these entities.

Organizations and Suborganizations

- Allow creation of hierarchical relationships that can represent or model your enterprise or organization's hierarchy.
- Can contain users created by its corresponding admin. This provides a method of grouping users together for administration and access control purposes. It is typically easier to administer and control access if users with similar needs are grouped together.
- Can be easily created or removed by an admin in a parent organization or suborganization via the admin console. However, when removed, all subordinate organizations and users are also removed, so not suitable when names or structure likely to change.

Roles

- Allow assignment of a privilege or set of privileges to a user or users. Within an organization, multiple roles can be defined to provide specific privilege sets to users.

- Define permissions via Access Control Instructions (ACI), which must be directly edited. Once defined, can be easily assigned or unassigned to an organization, a suborganization or a user. Unassigning a role from one entity only applies to that entity. Roles will still exist and remain assigned and be available for reassignment to other entities, so are more suited for organizations in which access changes will be frequently required.
- Can control visibility of channels and user's ability to overwrite channels. Settings within the XML Display Profile can make channels in the XML document visible or invisible by default. In addition, the default channels in the XML document can be prevented from being overridden.

Users

- Represent the identity of a person. Can be created within an organization or suborganization by its admin.
- Can be associated with multiple roles, but user must be within the roles' scope. In addition users inherit attributes from the suborganization.
- Belong to only one organization or suborganization; however, users can be easily moved from one organization to another if the admin has the privilege to do it.
- Can personalize visibility of channels.

Scenario 1: Hierarchical Structure with Suborganizations and Roles

Although you should strive for as flat a structure as possible, some hierarchy is useful to provide necessary groupings. The high-level steps to create a hierarchical structure are:

1. Creating a top-level organization.
2. Identifying all the functional or organizational groupings of users in your enterprise and determine for which ones you want to create a DIT structural entity, that is, ones that need to have specific privileges. Typically this should be only the largest subdivisions in your enterprise and the administrators for managing them. Use names that are generic or functional, so reorganizations and name changes will not be problematic.
3. For each DIT entity that has some affiliation with the top-level organization, creating either a *suborganization* (that is, an organization under another organization in the Sun ONE Identity Server world) or a *role* for that entity.

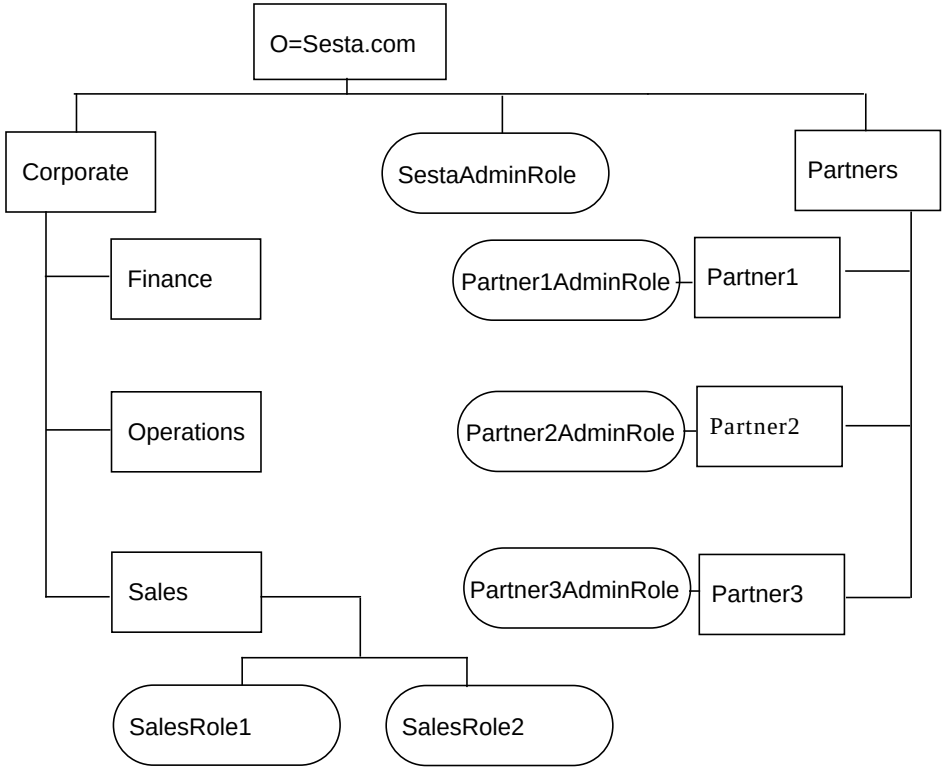
   Use the following guidelines to decide whether to use a suborganization or role:

- Define a suborganization for entities that contain groupings of users with similar access needs. Typically this will be broad functional or organizational entities for which a single set of permissions could be assigned.
- Define a role if it is possible that users in the child organizations need to have this role. All users belong to an organization or suborganization. If they do not have any roles assigned to them, they inherit their permissions from the organization in which they reside. Therefore, if you want a user to have attributes from both the organization they reside in and any parent organizations, you must use the role mechanism and assign them multiple roles.

4. For each role, defining a RoleAdministratorRole to manage the role. Then set the ACIs appropriately (management privileges: add or delete users, modify role attributes, and so on.)
5. Defining the users who will access your enterprise. If users are inheriting their privileges from their organization, place them in the appropriate organization. If users are receiving their privileges through role assignments, they must be placed so that they are within the role's scope, that is, within the organization or a child of the organization in which the role is defined.

Figure 2-1 illustrates a hierarchical directory structure. In this figure, the top-level organization is `Sesta.com`. Directly beneath the top-level is the `SestaAdminRole` to administer the organization and the `Corporate` and `Partners` suborganizations. The `Corporate` organization has three suborganizations: `Finance`, `Operations`, and `Sales`. Because there are multiple types of users within the `Sales` organization, two roles for are defined: `SalesRole1` and `SalesRole 2`. Within the `Partners` organization there are three suborganizations: `Partner1`, `Partner2`, and `Partner3`. Each of these organizations, requires its own administrator, so three roles are defined and each one is associated with the appropriate organization.The partner roles are `PartnerAdmin1`, `PartnerAdmin2`, and `PartnerAdmin3`.

Figure 2-1 Hierarchical Directory Structure

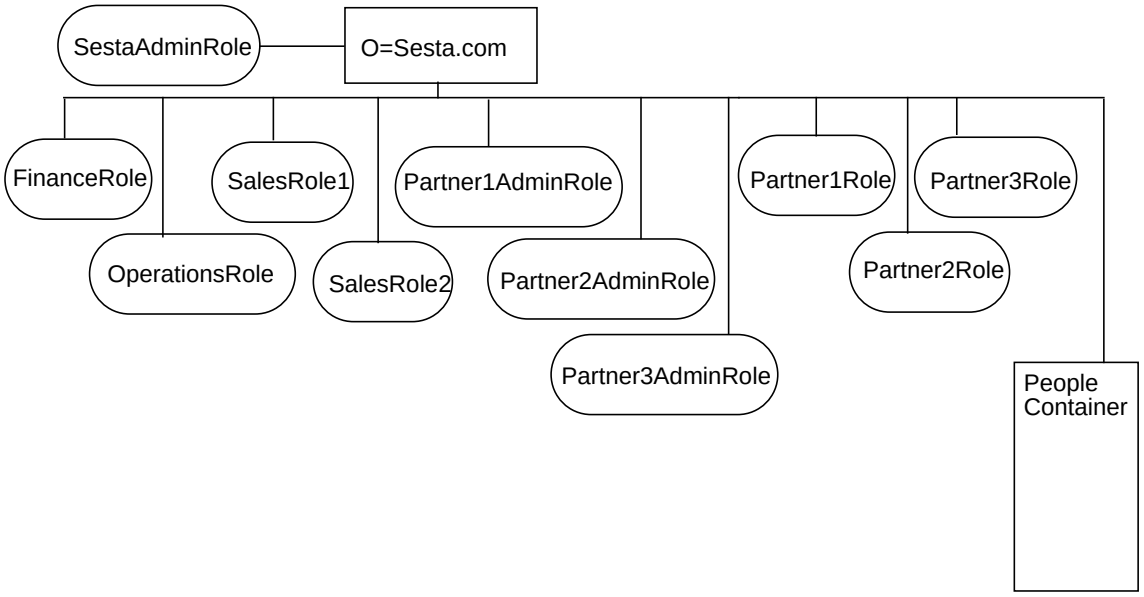


Scenario 2: Flat Tree Structure

If your organization changes often, a flatter or even totally flat tree structure may be appropriate. A structure with one organization, with one People container, and roles all at the same level is often useful if your enterprise changes frequently. With one organization, enterprise changes will not impact your DIT. All access privileges will be defined using roles and since all users are in the single People container and all roles are at the same level, any user can be assigned any role.

Figure 2-2 illustrates a flat directory structure. In this figure, the top-level and only organization is `Sesta.com`. All entities are defined directly beneath this top-level organization. They include the `SestaAdminRole` to administer the organization, four roles for the various corporate functions needed by the Finance, Operations, Sales1 and Sales2 users, and six roles for the user functions required by the partners: `Partner1Role`, `Partner2Role`, `Partner3Role`, `Partner1AdminRole`, `Partner2AdminRole` and `Partner3AdminRole`.

Figure 2-2 Flat Directory Structure



Creating New Organizations and Suborganizations

Organizations and suborganizations allow you to structure and group users for administration and access control purposes. Once you have determined the hierarchy or structure for your enterprise you must create the necessary organizations and suborganizations to implement it. By default, when you create a

new organization or suborganization, there are no services, policies, users, or roles defined for it. Therefore, whenever you create a new organization or suborganization, you need to perform the following high-level steps to configure it:

1. Registering all the services you want available to the organization. See To Register a Service for information. Typically, at a minimum you will want to register the following services:
   - Authentication. The Core authentication service and any authentication service with which users in the organization will use to authenticate (LDAP, anonymous). See Configuring Authentication for further information.
   - URL Policy Agent.
   - User.
   - Portal Server Configuration. Any Portal Server services you want to enable for users in the organization (Portal Desktop and NetMail).
2. Creating templates for each of the registered services. See To Create a Template for a Service for more information.
3. Creating the policies needed to grant users within the organization access privileges. See Overview of How Sun ONE Portal Server Uses Policy Management for more information on using policies.
4. Adding users to the organization. See To Add a New User for information.
5. Creating and assigning any roles you want in the organization. See To Create a New Role and To Assign a Role to a User for information.
6. Configuring the services enabled for your organization. To configure the Desktop, see Chapter 4, “Administering the Portal Desktop Service” for information. To configure NetMail, see Chapter 6, “Administering the NetMail Service”.

For a quickstart procedure to create a new organization and configure it to use portal, see Creating a New Portal Organization Quick Start.

To Create a New Organization or Suborganization

See Planning Organizations, Suborganizations, and Roles for recommendations on how to plan your organizations and suborganizations for use with Sun ONE Portal Server.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. If you are creating a suborganization, use the navigation pane to select the organization where the suborganization will be created.

3. Click New in the navigation pane.

   The New Organization page displays in the data pane.

4. Type a value for the name of the organization or suborganization in the New Organization page.

5. Choose a status of `Active` or `Inactive`.

   The default is `Active`. This can be changed at any time during the life of the organization or suborganization by selecting the properties arrow. Choosing `inactive` disables log in to the organization or suborganization.

6. Click Create.

   The new organization or suborganization displays in the navigation pane.

7. Choose Services from the View menu.

8. Click Register.

9. Enable the desktop service for the new organization.

   a. Select Identity Management from the location pane.

   b. Select Organizations from the View menu.

   c. Select the newly created organization.

   d. Select Services from the View menu.

   e. Select Portal Desktop

f. Change the valuef rom DummyChannel to JSPTabContainer (or the the name of the op-level container that will be used by the new organization) in Default Channel Name.

g. Change the value from default to sampleportal (or the desktop type that will be used by the new organization) in Portal Desktop Type .

To Register a Service

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization for which you want to register a service.

   Use the View menu in the navigation pane.

3. Choose Services from the View menu.

4. Click Register.

5. Select the service or services to register from the data pane and click Save.

To Create a Template for a Service

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization where the registered service exists.

   Use the View menu in the navigation pane

3. Choose Services from the View menu.

4. Click the properties arrow next to the registered service.

5. Accept or modify the default attribute values for the service and click Save.

   For information on setting Sun ONE Identity Server specific service attributes, see the *Sun ONE Identity Server Administrator's Guide*. For information on the setting Sun ONE Portal Server specific service attributes, see the appropriate appendix in this guide.

To Add a New User

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization where the user will be created.

3. Choose Users from the View menu and click New.

   The New User page appears in the data pane.

NOTE If you do not see Users but instead see People Containers in the drop-down menu, then make sure you have set the Show People Containers attribute for your organization, or up at the top level at some point. This is set in the Sun ONE Identity Server Services under Administration.

Users do always go into the People Container, but unless the Show People Containers attribute is selected you will just be able to see and interact with them directly under the organization. Show People Containers is not set by default.

---

4. Select the services to assign to the user and click Next.

   a. Select the user in the navigation pane and click the Properties arrow.

   b. Select Services from the View menu.

   c. Click Add to choose the services to assign to the users.

   d. Click Save,

   Typically, at a minimum you will want to register the Portal Desktop, Authentication Configuration, and Subscription services for most users.

5. Enter the user information and click Create.

   The new user appears in the navigation pane.

To Add a Service to a User

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization where the user will be created.
3. Choose Users from the View menu
4. Select the user in the navigation pane and click the Properties arrow.
5. Select Services from the View menu.
6. Click Add to choose the services to assign to the users.
7. Check the services and click Save,

   Typically, at a minimum you will want to register the Portal Desktop, and Subscription services for most users.

To Create a New Role

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization where the role will be created.
3. Choose Roles from the View menu and click New.

   The New Role page appears in the data pane.

4. Enter the role information (Name, Description, Role Type, Access Permissions) and click Create.

   The new role appears in the navigation pane.

NOTE If you are creating a customized role for delegated administration, you must have previously defined the ACI privileges for the role. See Chapter 3, “” for information.

To Assign a Role to a User

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.
2. Navigate to the organization or suborganization where the role will be created.
3. Choose Users from the View menu.
4. Click the properties arrow next to the user who will be assigned the role.

   The user profile information appears in the data pane.
5. Click Roles from the View menu in the data pane.

   The Add Roles page appears.
6. Check the box next to the roles to assign and click Save.

   The Roles for this User box is updated with the assigned roles.
7. Click Save to save the changes.

Enabling Existing Users to Access the Sun ONE Portal Server

When you install the Sun ONE Portal Server on an existing instance of Sun ONE Identity Server, users are not registered to use the Sun ONE Portal Server Desktop. In order to allow users to access the Desktop, you must enable them. Use the following procedures to enable users in the default organization or in another organization.

To Enable Users in the Default Organization

Before you start you will need the to obtain some configuration information. If you do not know all the details of the configuration, the information can be retrieved using a script from the `/var/sadm/pkg/SUNWps/pkginfo` file.

1. Determine or retrieve the following information from the `/var/sadm/pkg/SUNWps/pkginfo` file:

- The distinguished name for the directory manager (referred to as *DS_DIRMGR_DN/*). Default value is `cn=Directory Manager`.

- The directory manager password (referred to as *DS_DIRMGR_PASSWORD/*).
- The fully qualified domain name of the directory server (referred to as *DS_HOST/*).
- The port on which the directory server runs (referred to as *DS_PORT/*). Default value is 389.
- The root suffix of the directory tree (referred to as *DS_ROOT_SUFFIX/*). Default value is `dc=`*orgname*`,dc=com (such as dc=sun,dc=com)`.
- The default organization of the Sun ONE Portal Server installation (referred to as *DS_DEFAULT_ORG/*). Default value is `o=domain-name`.
- The base directory of the Sun ONE Portal Server installation (referred to as */BaseDir/*). Default value is `/opt`.

  If you do not know the configuration information, run the following script and refer to the output to obtain the information you will need to complete this procedure.

```
################################################
# Get configuration from file
################################################

GrabConfig() {
  GRABCONFIG_KEY=$1
  GRABCONFIG_FILE=$2
  GRABCONFIG_SEPARATOR=$3
  ANSWER_CONFIG=‘$GREP "^$GRABCONFIG_KEY$GRABCONFIG_SEPARATOR"
$GRABCONFIG_FILE | $UNIQ | $SED -e
"s/$GRABCONFIG_KEY$GRABCONFIG_SEPARATOR//" | $SED -e "s/^ //"‘
}

################################################
# Get PS6 Settings
################################################
```

```
GetPS6Settings() {
  if [ -f $PKGINFO ]; then
    # Ldap Settings

#
    GrabConfig "DS_HOST" $PKGINFO "="
    DS_HOST=$ANSWER_CONFIG
    echo "DS_HOST=$DS_HOST"
    GrabConfig "DS_PORT" $PKGINFO "="
    DS_PORT=$ANSWER_CONFIG
    echo "DS_PORT=$DS_PORT"

GrabConfig "DS_DIRMGR_DN" $PKGINFO "="
    DS_DIRMGR_DN=$ANSWER_CONFIG
    echo "DS_DIRMGR_DN=$DS_DIRMGR_DN"
    GrabConfig "DS_DIRMGR_PASSWORD" $PKGINFO "="
    DS_DIRMGR_PASSWORD=$ANSWER_CONFIG
    echo "DS_DIRMGR_PASSWORD=$DS_DIRMGR_PASSWORD"

################################################
# Get PS6 Settings
################################################

GetPS6Settings() {


  if [ -f $PKGINFO ]; then
```

```
 # Ldap Settings
    #
    GrabConfig "DS_HOST" $PKGINFO "="
    DS_HOST=$ANSWER_CONFIG
    echo "DS_HOST=$DS_HOST"
    GrabConfig "DS_PORT" $PKGINFO "="
    DS_PORT=$ANSWER_CONFIG
    echo "DS_PORT=$DS_PORT"
    GrabConfig "DS_DIRMGR_DN" $PKGINFO "="
    DS_DIRMGR_DN=$ANSWER_CONFIG
    echo "DS_DIRMGR_DN=$DS_DIRMGR_DN"
    GrabConfig "DS_DIRMGR_PASSWORD" $PKGINFO "="
    DS_DIRMGR_PASSWORD=$ANSWER_CONFIG
    echo "DS_DIRMGR_PASSWORD=$DS_DIRMGR_PASSWORD"

# Dsame Settings
    #
    GrabConfig "IDSAME_BASEDIR" $PKGINFO "="
    IDSAME_BASEDIR=$ANSWER_CONFIG
    echo "IDSAME_BASEDIR=$IDSAME_BASEDIR"
```

```
AMCONFIG="${IDSAME_BASEDIR}/SUNWam/lib/AMConfig.properties"

    if [ -f $AMCONFIG ]; then

      DS_ROOT_SUFFIX=‘$GREP "^com.iplanet.am.rootsuffix="
$AMCONFIG |

$SED -e "s/com.iplanet.am.rootsuffix=//"‘

      echo "DS_ROOT_SUFFIX=$DS_ROOT_SUFFIX"

      DS_DEFAULT_ORG=‘$GREP "^com.iplanet.am.defaultOrg="
$AMCONFIG | \
                      $SED -e "s/com.iplanet.am.defaultOrg=//"‘

      echo "DS_DEFAULT_ORG=$DS_DEFAULT_ORG"

 else

      print "‘$GETTEXT ’Error - Cannot find DSAME configuration
file,
please verify PS6 installation.’‘"

      exit 1

    fi

  else

    print "‘$GETTEXT ’Error - Cannot find SUNWps package
information
files, please verify PS6 installation.’‘"

    exit 1

  fi
```

2. Change directories to Sun ONE Identity Server utilities directory. For example, if the base directory is `/opt`, enter:

   `cd` */IDSAME_BaseDir*`/SUNWam/bin`

3. If the root suffix of the directory server and the default organization are not the same, execute the following command:

`./ldapsearch -h` */DS_HOST/* **`-p`** */DS_PORT/* **`-D`** */DS_DIRMGR_DN/* **`-w`** */DS_DIRMGR_PASSWORD/* **`\ -b "ou=People,`***/DS_DEFAULT_ORG/*`,`*/DS_ROOT_SUFFIX/***`" "(uid=*)" dn | \ /usr/bin/sed ’s/^version.*//’ > /tmp/.tmp_ldif_file1`**

4. If the root suffix of the directory server and the default organization are the same, execute the following command:

```
./ldapsearch -h /DS_HOST/ -p /DS_PORT/ -D /DS_DIRMGR_DN/ -w /DS_DIRMGR_PASSWORD/
\ -b "ou=People,/DS_ROOT_SUFFIX/" "(uid=*)" dn | \ /usr/bin/sed 's/^version.*//' >
/tmp/.tmp_ldif_file1
```

5. Execute the following command

```
grep "^dn" /tmp/.tmp_ldif_file1 | awk '{
print $0
print "changetype: modify"
print "add: objectclass"
print "objectclass: sunPortalDesktopPerson"
print "objectclass: sunPortalNetmailPerson\n" }' >
/tmp/.tmp_ldif_file2
```

6. Execute the following command.

```
./ldapmodify -c -h DS_HOST -p DS_PORT \ -D DS_DIRMGR_DN -w
DS_DIRMGR_PASSWORD -f /tmp/.tmp_ldif_file2
```

7. Remove all temporary files.

```
rm /tmp/.tmp_ldif_file1 /tmp/.tmp_ldif_file2
```

To Enable Users in a Non-Default Organization

1. Determine or retrieve the following information from the `/var/sadm/pkg/SUNWps/pkginfo` file:

- The distinguished name for the directory manager (referred to as *DS_DIRMGR_DN/*). Default value is `cn=Directory Manager`.
- The directory manager password (referred to as *DS_DIRMGR_PASSWORD/*)
- The fully qualified domain name of the directory server (referred to as *DS_HOST/*)
- The port on which the directory server runs (referred to as *DS_PORT/*). Default value is 389.
- The root suffix of the directory tree (referred to as *DS_ROOT_SUFFIX/*). Default value is `dc=`*orgname*`,dc=com (such as dc=sun,dc=com)`.
- The organization of the Sun ONE Portal Server installation for which you want to update the users (referred to as *DS_ORG_TO_UPDATE/*). Default value is ".

- The base directory of the Sun ONE Portal Server installation (referred to as */BaseDir/*). Default value is `/opt`.

2. Register services for the organization or suborganization containing the existing users you want to enable. See To Register a Service for information on the procedure.
3. Create a template for each service you register. See To Create a Template for a Service for information on the procedure.
4. Create and assign policies for each service. See To Register a Policy Service for a Peer or Suborganization, To Create a Referral Policy for a Peer or Suborganization, and To Create a Normal Policy for a Peer or Suborganization for information.
5. Set the URL to which to redirect successfully authenticated users from the organization. See To Redirect Successful Login User to the Portal Desktop URL.
6. Change directories to Sun ONE Identity Server utilities directory. For example, if the base directory is `/opt`, enter

   `cd` */IDSAME_BaseDir*`/SUNWam/bin`
7. Enable users within the organization or organizations, do one of the following:

   - To enable users only within a particular organization, defined as *DS_ORG_TO_UPDATE/*, then use the following command:

```
./ldapsearch -h /DS_HOST/ -p /DS_PORT/ -D /DS_DIRMGR_DN/ -w
/DS_DIRMGR_PASSWORD/ \ -b "ou=People,/DS_ORG_TO_UPDATE/,/DS_ROOT_SUFFIX/"
"(uid=*)" dn | \ /usr/bin/sed 's/^version.*//' > /tmp/.tmp_ldif_file1
```

   - To enable users in all organizations, then use the following command:

```
./ldapsearch -h /DS_HOST/ -p /DS_PORT/ -D /DS_DIRMGR_DN/ -w
/DS_DIRMGR_PASSWORD/ \ -b "/DS_ROOT_SUFFIX/" "(uid=*)" dn | \ /usr/bin/sed
's/^version.*//' > /tmp/.tmp_ldif_file1
```

8. Execute the following command:

```
grep "^dn" /tmp/.tmp_ldif_file1 | awk '{
print $0
print "changetype: modify"
print "add: objectclass"
print "objectclass: sunPortalDesktopPerson"
print "objectclass: sunPortalNetmailPerson\n" }' >
/tmp/.tmp_ldif_file2
```

9. Execute the following command:

   ./ldapmodify -c -h *DS_HOST* **-p** *DS_PORT* **\ -D "***DS_DIRMGR_DN***" -w** *DS_DIRMGR_PASSWORD* **-f /tmp/.tmp_ldif_file2**

10. Remove all temporary files.

    rm /tmp/.tmp_ldif_file1 /tmp/.tmp_ldif_file2

11. Change directory to Portal Server utilities directory.

    cd /*IDSAME_BASEDIR***/SUNWps/bin**

12. Execute the following to load the display profile for your non-default organization.

    ./dpadmin modify -u "uid=amadmin,ou=people,*DS_DEFAULT_ORG***,***DS_ROOT_SUFFIX***" -w** *DS_DIRMGR_PASSWOR***D -d "***NON_DEFAULT_ORG***,***DS_DEFAULT_ORG***,***DS_ROOT_SUFFIX***" ** *IDSAME_BASEDIR***/SUNWps/samples/desktop/dp-org.xml**

13. To enable users in another organization, repeat steps Step 7 through Step 13.

Creating a New Portal Organization Quick Start

The following task describes the steps to create a new organization and enable it for portal use. By default, when you log in, Identity Management is selected in the location pane, and Organizations is selected in the Navigation pane.

1. Create the new organization.

   e. Select Organizations from the View menu.

   f. Click New.

      The Create Organization page opens in the data pane.

   g. Type the new organization name. The Organization Status should be Active. Click Create.

      The newly created organization appears in the navigation page.

2. Register services for the new organization.

   a. Select Organizations from the View menu in the navigation pane and select the newly created organization from the Name menu.

   b. Select Services from the View menu.

c. Click Register.

The Register Services page appears in the data pane. Click the check box for the following minimum services, then click Register.

- LDAP
- Membership
- Policy Configuration
- Portal Desktop
- Subscriptions

The newly registered services appear in the navigation pane.

d. Configure each service by clicking the properties arrow. Click Create to modify the configuration attributes. See the *Sun ONE Identity Server Administration Guide* for a description of attributes that are not specific to Portal Server configuration

NOTE Suborganizations must register their services independently of the parent organization.

3. Create templates for the registered services if necessary.

a. Select Services from the View menu in the navigation pane.

b. One by one, click the properties arrow icon next to the services and create the templates.

4. Create the Desktop referral policy for the new organization.

The referral must define the parent organization as the resource in the rule, and it must contain a SubOrgReferral with the suborganization as the value in the referral

a. Select Identity Management from the location pane.

b. Select Policies from the View menu.

c. Click New to create new policy.

The Create Policy page appears in the data pane.

d. For Name, type SubOrgReferral_Desktop. Then click Create.

e. Select Portal Desktop in Service and click Next

f. Click Rules from the View menu in the data pane and click Add. Make sure Portal Desktop is selected and click Create.

g. Click Referrals from the View menu in the data pane and click Add. Make sure that the name of the suborganization is selected for Value in the data pane and click Create to complete the policy's configuration.

5. Create a normal Portal Desktop policy for the new organization.

a. Choose Policies from the View menu.

The policies for that organization are displayed.

b. Select New in the navigation pane. The New Policy page opens in the data pane.

c. Make sure you select Normal in Type of Policy.

d. Choose Rules from the View menu in the data pane and click Add. The Add Rule page opens in the data pane

e. Select Portal Desktop from the Service menu and click Next. Make sure Has Privilege to Execute Desktop is checked.

f. Choose Subjects from the View menu in the data pane and click Add. The Add Subject page opens in the data pane.

If LDAP Bind Password is not configured in the service template for the Policy Configuration Service, then clicking the Add Subject page results in a warning message "There are no matching entries. Please refine you search".

g. Select a subject that the Portal Desktop policy will be applied and choose Next to complete the subject configuration.

h. Click Create to complete the policy's configuration.

6. Create the Subscriptions referral policy for the new organization.

The referral must define the parent organization as the resource in the rule, and it must contain a SubOrgReferral with the suborganization as the value in the referral

a. Select Identity Management from the location pane.

b. Select Policies from the View menu.

c. Click New to create new policy.

The Create Policy page appears in the data pane.

d. For Name, type SubOrgReferral_Subscriptions. Then click Create.

e. Select Subscriptions in Service and click Next

f. Click Rules from the View menu in the data pane and click Add. Make sure Subscriptions is selected and click Create.

g. Click Referrals from the View menu in the data pane and click Add. Make sure that the name of the suborganization is selected for Value in the data pane and click Create to complete the policy's configuration.

7. Create a normal Subscriptions policy for the new organization.

a. Choose Policies from the View menu.

The policies for that organization are displayed.

b. Select New in the navigation pane. The New Policy page opens in the data pane.

c. Make sure you select Normal in Type of Policy.

d. Choose Rules from the View menu in the data pane and click Add. The Add Rule page opens in the data pane

e. Select Subscriptions from the Service menu and click Next. Make sure Has Privilege to Execute Desktop is checked.

f. Choose Subjects from the View menu in the data pane and click Add. The Add Subject page opens in the data pane.

g. Select a subject that the Subscriptions policy will be applied and choose Next to complete the subject configuration.

h. Click Create to complete the policy's configuration.

8. Create a new user in the new organizations.

a. Select Identity Management from the location pane.

b. Select Organizations from the View menu.

c. Select the newly created organization.

d. Select the user in the navigation pane and click the Properties arrow.

e. Select Services from the View menu.

f. Click Add to choose the services to assign to the users.

g. Click Save,

9. Enable the desktop service for the new organization.
   a. Select Identity Management from the location pane.
   b. Select Organizations from the View menu.
   c. Select the newly created organization.
   d. Select Services from the View menu.
   e. Select Portal Desktop
   f. Change the valuef rom DummyChannel to JSPTabContainer (or the the name of the op-level container that will be used by the new organization) in Default Channel Name.
   g. Change the value from default to sampleportal (or the desktop type that will be used by the new organization) in Portal Desktop Type .
10. Access the new organization's Desktop.
    a. Log out of the admin console.
    b. Open a browser page and type:

    `http://`*server*`:`*port*`/amserver/UI/login?org=`*neworg*

    The users's Desktop should appear.

Configuring Authentication

This section describes how to configure Sun ONE Portal Server authentication. Sun ONE Identity Server provides a framework for authentication. Authentication is implemented through plug-in modules that validate the user's identity. Sun ONE Identity Server provides seven different authentication modules as well as a Core authentication module. The Sun ONE Identity Server admin console is used to set the default values, to register authentication services, to create an organization's authentication template, and to enable the service. Because the Core authentication module provides the overall configuration for authentication, the Core authentication module must be registered and a template for it created for each organization before you can configure any of the specific authentication modules.

NOTE The authentication menu configuration feature provided by the Sun ONE Identity Server 5.1 administration console is not supported in the Sun ONE Identity Server 6.0 release. If you need to configure a selectable list of valid authentication modules, use the Sun ONE Identity Server administration console to set each authentication module with the same value in the authentication level attribute. Refer to To Configure the Authentication Menu for information on configuring authentication modules.

During installation the Core authentication is registered and a template is created for it in the default organization. In addition, the installation also registers and creates templates for the following authentication modules:

- LDAP—LDAP authentication allows any valid user within the search base of the directory tree to log in to the Sun ONE Portal Server. This will automatically assign a user to a specific role.
- Membership—Membership authentication allows a user to create an account and personalizes it without the aid of an administrator. With this new account, the user can access it as a registered user.

NOTE Although the installation configures a basic authentication implementation consisting of the Core, LDAP and Membership modules, you will need to configure authentication manually if you create new organizations or if you want to set up additional authentication functionality such as the ability to authenticate to an external LDAP directory or identity provider.

The high-level steps to configure an authentication module are as follows:

1. Registering the Core authentication service for each new organization. See To Register a Service for the steps to register a service.
2. Creating a template for the Core authentication service. See To Create a Template for a Service for the steps to create template for a service.
3. Registering the authentication services to support for each organization. See To Register a Service for the steps to register a service.

4. Creating service templates for the authentication services to support for the organization. See To Create a Template for a Service for the steps to create a template for an authentication service. For information on the setting the service attributes, see the *Sun ONE Identity Server Administrator's Guide*, Chapter 5, "Authentication Options."
5. Configuring the authentication menu. See To Configure the Authentication Menu for the steps to configure the authentication order.
6. Configuring the order to use authentication services. See To Configure Authentication Order for the steps to configure the authentication order.

Authentication By Authentication Level

Each authentication module can be associated with an integer value for its authentication level. Authentication levels can be assigned by clicking the authentication module's Properties arrow in Service Configuration, and changing the corresponding value for the module's Authentication Level attribute. Higher authentication levels define a higher level of trust for the user once that user has authenticated to one or more authentication modules.

To Configure the Authentication Menu

Users can access authentication modules with a specific authentication level. For example, a user performs a login as a user with the following syntax:

`http://`*`hostname:port/deploy_uri`*`/UI/Login?authlevel=`*`auth_level_value`*

All modules whose authentication level is larger or equal to *auth_level_value* will displayed as an authentication menu for the user to choose. If only one matching module is found, then the login page for that authentication module will be directly displayed.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, when you log in, Identity Management is selected in the location pane, and Organizations is selected in the Navigation pane.
2. Navigate to the organization or suborganization that you want to configure authentication for.

   Use the View menu in the navigation pan
3. Choose Services from the View menu and click Register.

4. Click the properties arrow next to Core.
5. Enable the appropriate authentication modules by selecting them in the Organization Authentication Modules field of the Organization section.

   By default, Sun ONE Portal Server installation enables LDAP and Membership.
6. Enter a value in the Default Auth Level for each authentication module (default is 0).

   The value for each authentication module must be the same in order to appear in the authentication menu.
7. Click Save.

To Configure Authentication Order

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, when you log in, Identity Management is selected in the location pane, and Organizations is selected in the Navigation pane.
2. Navigate to the organization or suborganization that you want to configure authentication for.

   Use the View menu in the navigation pan
3. Choose Services from the View menu and click Register.
4. Click the properties arrow next to Core.
5. Enable the appropriate authentication modules by selecting them in the Organization Authentication Modules field of the Organization section.

   By default, Sun ONE Portal Server installation enables LDAP and Membership.
6. Enter a value in the Default Auth Level for each authentication module (default is 0).

   The value for each authentication module must be the same in order to appear in the authentication menu.
7. Select Edit in Organization Authentication Configuration to specify the attribute information for each authentication module.

   a. Click Add to add an authentication module to the menu.

b. Click Reorder to change the order that the authentication modules will appear in the authentication module.

c. Click Save to save the attribute information.

8. Click Save

9. Use the following URL to verify that the authentication menu appears with the appropriate choices by logging in to the admin server.

`http://`*host*`:`*port*`/amserver/UI/login`

If this is not the default organization, use the following URL to verify the authentication menu for the organization:

`http://`*host*`:`*port*`/amserver/UI/login?org=`*org_name*

To Configure LDAP Authentication to an External Directory

When you install the Sun ONE Portal Server, the installation program configures LDAP authentication to directory instance automatically. The installation program allows you to install an internal instance of the directory on the local server and configure LDAP authentication to that internal directory or to configure LDAP authentication to a pre-existing external instance of the directory. Once you have your initial configuration, there are some scenarios where you might want to configure authentication to an external LDAP directory. For example, you may want to isolate authentication information for particular organization onto a dedicated LDAP server for performance or security reasons.

CAUTION Do not configure authentication to an external LDAP directory for the organization containing the `amadmin` user. This can prevent the `amadmin` user from authenticating and lock you out of the admin console. If you do inadvertently configure the organization containing the `amadmin` user, you will need to log in using the full DN of the `amadmin` and then correct the LDAP template. The `amadmin` DN is listed in the `com.sun.authentication.super.user` property in the `AMConfig.properties` file.

1. Log in to the Sun ONE Identity Server administration console as administrator.

By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization that you want to configure authentication for.

   Use the View menu in the navigation pane.

3. Choose Services from the View menu.

4. Click the properties arrow next to Core from Identity Server Configuration.

5. Check Dynamically Created from the Dynamic User Profile menu.

6. Click the properties arrow next to LDAP from the Identity Server Configuration menupeople,dc=sesta.dc=com.

7. Set the appropriate LDAP Attributes for your server. The following example sets up access to the LDAP server `ds-sesta1.sesta.com` on port 389 with a search start point of `ou=people,dc=sesta,dc=com` and using a root user bind to `cn=root,ou=people,dc=sesta,dc=com`:

   Primary LDAP Server and Port: `ds-sesta1.sesta.com`:389
   Secondary LDAP server and port: `ds-sesta1.sesta.com`:389
   DN to Start User Search: `ou=people,dc=sesta,dc=com`
   DN for Root User Bind: `cn=root,ou=people,dc=sesta,dc=com`
   Password for Root User Bind: `root password`
   User Naming Attribute: `uid`
   User Entry Search Attributes: `employeenumber`
   User Search Filter: blank
   Search Scope: `subtree`
   Enable SSL to LDAP Server: `off`
   Return User DN to Auth: `off`
   Authentication Level: `0`

8. Click Save.

Configuring Anonymous Authentication

The Sun ONE Portal Server supports two methods for implementing anonymous authentication:

- Using the Authentication-less User ID attributes. Users accessing the Desktop URL are automatically authenticated and granted access to the Desktop.
- Using an Anonymous user session. Users select Anonymous from the Authentication menu, log in as `anonymous`, and are granted access to the Desktop.

To support anonymous authentication, the Sun ONE Portal Server installation program creates a user account, `authlessanonymous`, and sets up access for this user within the following two Portal Desktop Services global attributes:

- Authorized Authentication-less User IDs
- Default Authentication-less User ID

Sun ONE Portal Server can support both authentication-less and anonymous authentication to be configured at the same in the sense that you can do the following:

1. Configure the Desktop to work in authentication-less mode.
2. Configure the authentication menu so that Anonymous is one of the displayed choices.
3. Access the Desktop with browser A, thereby accessing it in authentication-less mode.
4. Access `http:/`*server*`/amserver/UI/login` with browser B, and select Anonymous, and see the Desktop.

At this point you are using authentication-less mode in browser A and anonymous mode in browser B.

The way in which the Desktop is accessed occurs in two different ways. One, authentication-less access, was through a direct reference to `/portal/dt` and the other (anonymous) was indirectly through `/amserver/UI/login`.

The Sun ONE Identity Server Login menu could be avoided by configuring Sun ONE Identity Server to only have anonymous login in the menu.

Both authentication-less accessand anonymous authentication are not supported simultaneously in that when you access `/portal/dt` without an Sun ONE Identity Server session, only one of two things happens:

a. The Desktop will redirect to `/amserver/UI/login`, which may automatically do an Anonymous login and redirect you back to `/portal/dt`.

b. The Desktop will run in authentication-less access mode.

You do not have to disable anonymous authentication to use authentication-less access. But if you want the above item a to work, you have to disable authentication-less access mode.

To Configure Anonymous Authentication (Anonymous User Session Method)

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization that you want to configure authentication for.

   All created organizations are displayed in the navigation pane.

3. Select Service Configuration in the location pane.

4. Click the properties arrow next to the Portal Desktop service.

   The Portal Desktop attributes appear in the data pane.

5. Select the value listed in the Authorized Authentication-less User IDs attribute and click Remove.

6. Select the value listed in the Default Authentication-less User ID attribute and click Remove.

7. Click Save.

8. Choose Identity Management from the location pane.

9. Choose Organizations from the View menu.

   All created organizations are displayed in the navigation pane.

10. Navigate to the organization or suborganization that you want to configure authentication for.

    Use the View menu in the location pane.

11. Choose Services from the Show menu.

12. Register and configure the Anonymous service.

    See To Register a Service and To Create a Template for a Service for information.

13. Add Anonymous to the Authentication menu.

    See To Configure Authentication Order for information.

14. Create an anonymous user account.

    See To Add a New User for information.

To Configure Anonymous Authentication (Authentication-less Access)

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. By default, when you log in, Identity Management is selected in the location pane, and Organizations is selected in the Navigation pane.

   All created organizations are displayed in the navigation pane.

3. Navigate to the organization or suborganization that you want to configure authentication for.

   Use the View menu in the navigation pane.

4. Create an `authlessanonymous` user account with the password `authlessanonymous`.

   See To Add a New User for information.

5. Select Service Configuration in the location pane.

6. Select Portal Desktop in the navigation pane.

7. Add the fully distinguished name for the `authlessanonymous` user to the Authorized Authentication-less User IDs attribute. For example:

   ```
   uid=authlessanonymous, ou=People, dc=sesta, dc=com
   ```

8. Specify the fully distinguished name for the `authlessanonymous` user in the Default Authentication-less User ID attribute.

9. Click Save.

You must close and restart your browser to access the Desktop using the newly configured Authentication-less User ID method. The Authentication-less User ID method allows you to specify the UID of the user account in the query string. For example, to access the Desktop from the default organization of sestat.com, use the following URL:

```
http://server:port/portal/dt?dt.suid=uid= authlessanonymous,
ou=People,dc=sesta, dc=com
```

NOTE If a user logs in a browser with locale that is not the user's own language , all other users will share the same locale at the login prompt.

There are multiple options to get around this problem.

- Turn off caching by changing the value for `refreshTime` to 0 for `JSPTabContainer` in `dp-anon.xml`.
- You can specify multiple authentication-less users, one authentication-less user per locale and redirect the authentication-less desktop to the right user based on browser's locale.

Configuring Portal Server for Federated Users

The Sun ONE Portal Server software supports users that have federated identities conforming to the Liberty Alliance specification. A federated user that are Liberty single signed on can access a personalized desktop at a portal server without the need for further authentication.

See the *Sun ONE Identity Server Administrator's Guide* for more information about Liberty-enabled authentication services. Example configurations with Sun ONE Portal Server acting as a service provider can be found in the following location:

PortalServerBaseDir`/SUNWps/samples/liberty`

To Configure Federated Users

By default, federated users do not have permission to access the Sun ONE Portal Server acting as a service provider. The Sun ONE Portal Server can handle federated users as follows:

- Federated users who are Liberty single signed on can access a personalized portal desktop.
- Federated users that are not Liberty signle signed on are redirected to the authentication page of an identity provider

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Select Service Configuration in the location pane.
3. Select Portal Desktop in the navigation pane.

4. Check Enable Federation.
5. Specify the ID of the host provider.
6. Click Save.

To Configure Authentication-less Access for Federated Users

By default, federated users do not have permission to access the authentication-less portal desktop.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.
2. Navigate to the organization or suborganization that you want to configure authentication for.

   Use the View menu in the navigation pane.
3. Select Service Configuration in the location pane.
4. Select Portal Desktop in the navigation pane.
5. Uncheck Disable Authentication-less Access for Federated Users.
6. Click Save.

See To Configure Anonymous Authentication (Authentication-less Access) for more information on authentication-less access.

To Configure UNIX Authentication

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.
2. Choose Organizations from the View menu in Identity Management.

   All created organizations are displayed in the navigation pane.
3. Select Service Configuration in the location pane.
4. Click the properties arrow next to UNIX in the navigation pane (under Identity Server Configuration).
5. Set the appropriate UNIX Attributes for your server.

6. Click Save.
7. Navigate to the organization or suborganization that you want to configure authentication for.

   Use the View menu in the navigation pane.
8. Choose Services from the View menu.
9. Click Register in the navigation pane.
10. Click Core under Authentication in the data pane.
11. Select Unix from the Organization Authentication Modules menu in the data pane.
12. Click Save.

To Configure UNIX Authentication for the Organization Level

The UNIX authentication documented in To Configure UNIX Authentication is for configuring UNIX globally. This procedure is to configure at the organization level.

1. Log in to the Sun ONE Identity Server administration console as administrator (amadmin) by entering `http://`*`fullservername`*`:`*`port`*`/amconsole` in your browser's web address field.
2. At the logon screen, enter amadmin as the user ID and the passphrase you chose during installation.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.
3. Choose Organizations from the View menu in Identity Management.

   All created organizations are displayed in the navigation pane.
4. Choose Services from the View menu.
5. Select Register.
6. Check UNIX in the right pane and click Register.
7. Select the properties arrow next to UNIX.
8. Select Create in the right pane.
9. Set the appropriate UNIX Attributes for your server.

10. Select Save.
11. Select the properties arrow next to Core.
12. Highlight UNIX in Authentication Menu and select Save.

Overview of How Sun ONE Portal Server Uses Policy Management

This section describes how to use Sun ONE Identity Server Policy Management feature. See the Sun ONE Identity Server documentation for procedures to create, modify, and delete policies.

The Sun ONE Identity Server Policy Service enables you to define rules or access to resources. Policies can be role-based or organization-based and can offer privileges or define constraints. Sun ONE Portal Server ships with three policies:

- Ability to execute Portal Server Portal Desktop - Enables users to display the Desktop
- Ability to execute Portal Server NetMail - Enables user to run NetMail

NOTE Chapter 4, "Administering the Portal Desktop Service" and Chapter 6, "Administering the NetMail Service" provide detailed descriptions on assigning their specific policies.

By default, the Policy Configuration service is automatically registered to the top-level organization. Suborganizations must register their policy services independently of their parent organization. Any policy service you create must be registered to all organization. The high-level steps to use policies are:

1. Registering the Policy service for an organization. (This will be done automatically for the organization specified at installation.) Suborganizations do not inherit their parent's services, so you need to register a suborganization's Policy service. See To Register a Service for information.
2. Creating a referral policy for a peer or suborganization. You can delegate an organization's policy definitions and decisions to another organization. (Alternately, policy decisions for a resource can be delegated to other policy products.) A referral policy controls this policy delegation for both policy

creation and evaluation. It consists of a rule and the referral itself. If the policy service contains actions that do not require resources, referral policies cannot be created for suborganizations. See To Create a Referral Policy for a Peer or Suborganization for information.

3. Creating a normal policy for a peer or suborganization. You create a normal policy to define access permissions. A normal policy can consist of multiple rules, subjects, and conditions. See To Create a Normal Policy for a Peer or Suborganization for information.

To Register a Policy Service for a Peer or Suborganization

Peer or Suborganizations do not inherit their parent's services, so you need to register a peer or suborganization's Policy service.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization that you want to create a referral policy.

   All created organizations are displayed in the navigation pane.

3. Select Organizations from the View menu in the navigation pane and select desired organization from the Name menu.

4. Select Services from the View menu.

5. Click Register.

   The Register Services page appears in the data pane. Click the check box for the to the following minimum services, then click Register.

   - LDAP
   - Membership
   - Policy Configuration
   - Portal Desktop
   - NetMail

   The newly registered services appear in the navigation pane.

6. Configure each service by clicking the properties arrow. Click Create to modify the configuration attributes. See the Sun ONE Identity Server Administration Guide for a description of attributes that are not specific to Portal Server configuration

To Create a Referral Policy for a Peer or Suborganization

You can delegate an organization's policy definitions and decisions to another organization. A referral policy controls this policy delegation for both policy creation and evaluation. It consists of a rule and the referral itself. The referral must define the parent organization as the resource in the rule, and it must contain a SubOrgReferral or PeerOrgReferral with the name of the organization as the value in the referral

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization that you want to create a referral policy.

   All created organizations are displayed in the navigation pane.

3. Select Policies from the View menu.

4. Click New to create new policy.

   The Create Policy page appears in the data pane.

5. For Name, type either SubOrgReferral_*organization* or either PeerOrgReferral_*organization*. Make sure you select Referral in Type of Policy. Then click Create.

6. Select the type of service in Service and click Next

7. Click Rules from the View menu in the data pane and click Add. Then click Next.

   The Add Rule template appears in the data pane.

8. Enter the name of the rule in Rule Name and click Create.

9. Click Referrals from the View menu in the data pane and click Add.

   The Add Referral template appears in the data pane.

10. Enter SubOrgReferralName in Name.

    Make sure that the name of the suborganization is selected for Value in the data pane and click Create to complete the policy's configuration.

11. Click Save in the data pane.

    The message "The policy properties have been saved" is displayed when the data is saved.

To Create a Normal Policy for a Peer or Suborganization

You create a normal policy to define access permissions. A normal policy can consist of multiple rules, subjects, and conditions.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization or suborganization that you want to assign a policy.

   All created organizations are displayed in the navigation pane.

3. Choose Policies from the View menu.

   The policies for that organization are displayed.

4. Select New in the navigation pane. The New Policy page opens in the data pane.

5. For Name, type either SubOrgNormal_*organization* or either PeerOrgNormal_*organization*.Make sure you select Normal in Type of Policy. Click Create

6. Select a service from the Service menu and click Next. Enter the name of the rule in Rule Name. Make sure the appropriate checkbox is selected to grant execution privilege to the desired service.

7. Choose Rules from the View menu in the data pane and click Add. The Add Rule page opens in the data pane

8. Choose Subjects from the View menu in the data pane and click Add. The Add Subject page opens in the data pane.

9. Click Create to complete the policy's configuration.

   The message "The policy properties have been saved." is displayed when the data is saved.

Logging In to the Sun ONE Portal Server Portal Desktop

If you installed the sample portal, users will be able to log in to the sample Desktop. In addition, the Sun ONE Portal Server supports a variety of other user logins. This section describes some of the other user ways users can log in to the Sun ONE Portal Server.

To Log In to the Sample Portal Desktop

To access the sample Desktop, type the following URL:

`http://`*server:port***`/portal/dt`**

To Log In to a Suborganization

If users have access privileges to an organization, they can also log in to suborganizations within the organization. For example, if a user has access to the organization A which has a suborganization B, type the following URL to log in to suborganization B:

`http://`*server:port***`/amserver/UI/login?org=B`**

To Log On Using Anonymous Authentication

NOTE You must register the anonymous authentication module to support anonymous authentication. See Configuring Anonymous Authentication for information on registering and enabling anonymous authentication modules.

1. Log on using the following URL:

   `http://`*`server:port`***`/portal/dt`**

2. At the Sun ONE Identity Server authentication page, click Anonymous.
3. The sample Desktop appears.
4. If desired, and if the Membership authentication module has been register, use the Login screen to create and register a user ID.

Managing Logging

Sun ONE Portal Server uses the Sun ONE Identity Server logging and debugging APIs.

By default, the Sun ONE Portal Server log and debug files are located in:

- `/var/opt/SUNWam/logs`
- `/var/opt/SUNWam/debug`

The Sun ONE Identity Server admin console allows you to define the following logging attributes:

- Max Log Size
- Number Of History Files
- Log Location
- Logging Type
- Database User Name
- Database User Password
- Database Driver Name

See the *Sun ONE Identity Server Administrator's Guide* for further information.

Configuring Delegated Administration

This chapter describes how to configure delegated administration for Sun™ ONE Portal Server.

This chapter contains these sections:

Overview of Delegated Administration

As enterprises create larger and more complex portals, a centralized administration model is no longer viable. Delegated administration or Line of Business (LOB) administration addresses this issue by delegating or distributing the administration tasks to the actual portal users.

The Sun ONE Portal Server allows you to delegate administration functions to users by using roles. Role-based administration enables an enterprise to break its business into smaller organizations or lines of business (LOB) and then allows different users to administer the organizations, suborganizations, users, policy, roles, and channels of the LOB based on the user's roles.

Table 3-1 on page 94 lists and defines some important delegated administration terms as they apply in the Sun ONE Portal Server. The table contains two columns: the first column lists the term and the second column gives a brief description.

Table 3-1 Delegated Administration Terms

Term	Description
Privilege	The combination of a single resource and a single action that can be performed upon the resource (for example, view a static web page, view paystubs in a paycheck application, modify W-4 data in the paycheck application, and so on).
Action	Actions are a procedure or operation that can be performed on a resource (for example, read a catalog, write a catalog, get email using POP, get email using IMAP, and so on).
Resource	A resource is something that can be abstractly represented in software and whose access is controlled and protected. In Sun ONE Identity Server, the Resource refers to the URL Access only.
Top-level Admin role	A role that has complete management rights to all policy and identity settings.
Organization admin role	A role that has complete management rights to policy and identity settings for an organization.
Line of Business (LOB)	Line of business capabilities are administration capabilities that can be done by a business analyst or equivalent position. LOB administrators are able to perform administrative tasks that do not require Top-level Admin capabilities to complete. Typically, LOB capabilities, such as adding or removing users to and from roles that grant access to resources, would be available only within their sphere of interest.
Role administrator role	A role administrator role is a role with the access permissions to administer some other specific roles and a certain set of user objects. For example, adding or removing users from a role or editing role level attributes.
Role administrator	Role administrators are users to whom role administrator roles have been assigned.

Delegated Administration Roles

The Sun ONE Identity Server administration console provides role-based delegated administration capabilities to different kinds of administrators to manage organizations, users, policy, roles, and channels based on the given permissions.

Sun ONE Identity Server administration console provides a number of predefined administrator roles for delegating administration functions. They are as follows:

- Top-Level Admin
- Group Admin
- Organization Admin
- Organization Help Desk Admin
- People Container Admin
- Container Admin
- Container Help Desk Admin

For detailed information on these roles, refer to the Sun ONE Identity Server product documentation.

NOTE Sun ONE Identity Server also implements three other roles: Top-level Admin, Top-level Help Desk Admin, and Deny Write Access. These roles are created during installation and only exist at the root of the installation. Any new organizations created will not get these three roles. By default, when a new organization is created, three roles get created with it: Organization Admin, Organization Help Desk Admin, and People Admin.

You can use these predefined administrator roles to set up your delegated administration implementation if their function fits the need. For example, if the directory structure for your model comprises an organization with multiple sub-organizations, you could assign Organization Admin roles to users to create delegated administrators for each of the suborganizations. However, if the organizational structure of your enterprise is more complicated, you might want to create a delegated administration model that targets your specific needs. To do this, the Sun ONE Identity Server administration console allows you to define delegated administrator roles with privileges specific to your business needs.

To implement an enterprise-specific delegated administration model, there are three critical conceptual roles:

- Top-level Admin Role
- Organization Admin Role
- Role Administrator Role

The Top-level Admin Role is created when the system is set up, and the Organization Admin Role is created automatically when a new organization is set up. The Role Administrator Role is a role you create based on the requirements of the delegated administration model. The access permissions for the Role Administrator Role are defined by directly editing the corresponding Access Control Instructions (ACIs).

In a delegated administration, the following principles apply:

- User privileges are granted by the user's role.
- Privileges are granted on a per individual user basis by defining a role with desired privileges and assigning this role to the individual user.
- Sets of users can be grouped together by assigning them a specific role. These users will be granted the set of privileges and inherit the values for dynamic attributes that are defined for that role.
- Users can have multiple or aggregated roles. Users with multiple roles have access to combined features of all their roles. When there is a conflict in the features granted by aggregated roles, conflict resolution is based on the priority configured through Conflict Resolution Level defined for the each of the services for those roles. There are seven conflict resolution settings available ranging from Highest to Lowest. When an attribute conflict occurs as role templates from multiple rolers are merged, the attribute on the template set with the highest conflict resolution level is returned.

Developing a Delegated Administration Model

In order to delegate administration functions for the Sun ONE Portal Server appropriately, you should develop a delegated administration model to help determine the administration roles required for you enterprise. Consider the following when developing your model:

- Focus on the business requirements of your enterprise. In general, the proposed solution for the role-based delegated administration should be parallel with the business requirements.
- Develop a directory structure that enables users to be grouped so they can access their required resources and have their administration needs managed by a delegated administrator.

- Try to fit your business entities into a more standard tree structure as much as possible while still addressing all the business requirements. You can use a structure with a hierarchy of organizations and suborganizations or a flat directory tree structure. In a flat directory structure, all the entities are defined immediately beneath the top level organization and all the roles (including Role Administrator Roles) are “parallel” to each other in terms of the organizational hierarchy. For example, all the users who are affiliated with business unit would be created in people containers under the top-level organization. For each of the access roles and administrative roles needed in your model a corresponding role at the top-level would be created.

Configuring Delegated Administration

The high-level steps that you perform to configure a delegated administration implementation for the Sun ONE Portal Server are:

1. Defining the ACI settings for the Role Administrator Roles
2. Creating new Admin Roles for the delegation model
3. Assigning Role Administrator Roles to users
4. Configuring Additional Restrictions on a Role

Defining the ACI Settings for Role Administrator Roles

To configure the appropriate privileges for any of the role administrator roles you identified in your delegation model, you must define the appropriate permissions in an ACI for each unique role in your delegation model. You can define an ACI permission template for a role using the Sun ONE Identity Server administration console or the Directory Server console. You can also define an ACI for a specific role using the `ldapmodify` command.

Use the following format when defining ACI permission templates in the Sun ONE Identity Server administration console or with the Directory Server console:

permission_name | *aci_desc* | *dn:aci ## dn:aci ## dn:aci*

where:

permission_name is the name of the permssion.

aci_desc is a text description of the access these ACIs allow.

dn:aci represents pairs of DNs and ACIs separated by ##. Sun ONE Identity Server sets each ACI in the associated DN entry.

This format also supports tags that can be substituted for values that would otherwise have to be specified literally in an ACI: ROLENAME, ORGANIZATION, GROUPNAME, and PCNAME. Using these tags lets you define roles flexible enough to be used as defaults. When a role is created based on one of the default roles, tags in the ACI resolve to values taken from the DN of the new role.

For detailed information setting ACIs, refer to the *Sun ONE Identity Server Programmer's Guide*.

NOTE In these example ACI definitions, the root suffix is assumed to be `dc=sesta,dc=com`.

To Define an ACI Using the Command Line

1. Create a text file containing the ACI settings for use with the `ldapmodify` command. For example, the following file, `acis.ldif`, contains an ACI definition of two roles called `JDCAdmin1` and `JDCAdmin2`.

```
dn:dc=sesta,dc=com
changetype:modify
# aci for JDCAdmin1 role
# This role can add/delete users from JDC role
add:aci
aci: (target= "ldap:///ou=people,dc=sesta,dc=com") (targetattr = "*")(version
3.0; acl "Allow JDCAdmin1 Role to read and search users"; allow (read,search)
roledn = "ldap:///cn=JDCAdmin1,dc=sesta,dc=com";)
-
add:aci
aci: (target="ldap:///dc=sesta,dc=com")
(targetfilter="(entrydn=cn=JDC,dc=sesta,dc=com)")(targetattr="*")(version 3.0;
acl "Allow JDCAdmin1 Role to read and search JDC Role";allow (read,search)
roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com";)
-
add:aci
aci:
(target="ldap:///ou=people,dc=sesta,dc=com")(targetattr="nsroledn")(targetfilt
er="(!(|(nsroledn=cn=Top-level Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Help Desk Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Organization Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Policy Admin
Role,dc=sesta,dc=com)))")(targattrfilters="add=nsroledn:(nsroledn=cn=JDC,dc=se
sta,dc=com),del=nsroledn:(nsroledn=cn=JDC,dc=sesta,dc=com)")(version 3.0; acl
"Allow JDCAdmin1 Role to add/remove users to JDC Role"; allow
(write)roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com";)
-
# aci for JDCAdmin2 role
# This role can add/remove channels from the JDC role's display profile
add:aci
aci:
(target="ldap:///cn=SunPortalDesktopService,dc=sesta,dc=com")(targetfilter=(cn
=cn=JDC,dc=sesta,dc=com))(targetattr="*")(version 3.0; acl "Allow JDCAdmin2 to
edit display profile of JDC Role"; allow (all)
roledn="ldap:///cn=JDCAdmin2,dc=sesta,dc=com";)
-
add:aci
aci: (target="ldap:///dc=sesta,dc=com")(targetattr = "*") (version 3.0; acl
"Allow JDCAdmin2 to read and search all"; allow (read,search) roledn =
"ldap:///cn=JDCAdmin2,dc=sesta,dc=com";)
```

2. Change directories to Sun ONE Identity Server utilities directory. For example,

   `cd /`*`BaseDir`***`/SUNWam/bin`**

3. Set `LD_LIBRARY_PATH` to include `IS_BASEDIR/SUNWam/ldaplib/solaris/sparc/ldapsdk`

4. Execute the following command.

   `./ldapmodify -D "`*`DS_DIRMGR_DN`***`" -w`** *DS_DIRMGR_PASSWORD* **`-f /tmp/acis.ldif`**

5. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

6. Navigate to the organization or suborganization to create a new role (such as `JDCAdmin1` and `JDCAdmin2`).

   a. Choose Roles from the View menu and click New.

   b. The New Role page appears in the data pane.

   c. Enter the role information (Name, Description, Role Type,Access Permissions) and click Create (for example, a static role JDC with "Type=Service" and "Access Permissions=No Permissions").

      The new role appears in the navigation pane.

7. Create "Desktop" service template for role you created.

   a. Choose Services from the View menu.

   b. Click the properties arrow next to the Desktop service.

   c. Accept or modify the default attribute values for the Desktop service and click Save.

8. Create a tab in the role display profile (for example, the role display profile for JDC).

   a. Navigate to the role where the tab will be created.

   a. Choose Services from the View menu in the navigation pane.

   b. Click the properties arrow next to Desktop in the navigation pane.

   c. The Desktop attributes page appears in the data pane.

   d. In the Desktop page, click the Channel and Container Management link.

e. The Channels page appears, with the container path set at the root.

f. Click the Container that you want to add the channel or container to.

g. The top of the page displays the container path where the channel will be added. Defined channels and container, if any, appear in lists.

h. Click Add to add a container channel or channel.

i. To add a container channel, click Add under Container Channel. To add a channel, click Add under Channel.

j. The Add Channel page appears.

k. Type a channel name and select the type of provider from the menu.

l. Click Create.

Refer to Chapter 5, "Administering the Display Profile" for more information.

9. Create a user (such as `admin1` or `admin2`).

a. Navigate to the role where the user will be created.

b. Choose Users from the View menu and click New.

c. The New User page appears in the data pane.

d. Select the services to assign to the user and click Next.

e. Enter the user information and click Create.

f. The new user appears in the navigation pane.

10. Assign a role to a user (such as `JDCadmin1` to `admin1` or `JDCadmin2` to `admin2`).

a. Navigate to the organization or suborganization where the role will be assigned.

b. Choose Users from the View menu.

c. Click the properties arrow next to the user who will be assigned the role.

d. The user profile information appears in the data pane.

e. Click Roles from the View menu in the data pane.

f. The Add Roles page appears.

g. Check the box next to the roles to assign and click Save.

h. The Roles for this User box is updated with the assigned roles.

i. Click Save to save the changes.

11. Logout from the admin conole.

To Define an ACI Using the Admin Console

1. Log in to the Sun ONE Identity Server administration console as Top-level Admin.

   By default, Identity Management is selected in the location menu, and Organizations is selected in the navigation pane.

2. Click Service Configuration in the location pane.

3. Click the properties arrow next to the Administration service.

   The administration attributes appear in the data pane.

4. In the Default Role Permissions (ACIs) entry field, type in the ACI definition and click Add. For example, for the `JDCAdmin1 and JDCAdmin1` role defined previously, you would enter the following:

```
JDCAdmin1|Add/delete users from JDC role|dc=sesta,dc=com:aci:
(target= "ldap:///ou=people,dc=sesta,dc=com") (targetattr =
"*")(version 3.0; acl "Allow JDCAdmin1 Role to read and search
users"; allow (read,search) roledn =
"ldap:///cn=JDCAdmin1,dc=sesta,dc=com";)##dc=sesta,dc=com:aci:
(target="ldap:///dc=sesta,dc=com")
(targetfilter="(entrydn=cn=JDC,dc=sesta,dc=com)")(targetattr="*"
)(version 3.0; acl "Allow JDCAdmin1 Role to read and search JDC
Role";allow (read,search)
roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com";)
##dc=sesta,dc=com:aci:(target="ldap:///ou=people,dc=sesta,dc=com
")(targetattr="nsroledn")(targetfilter="(!(|(nsroledn=cn=Top-lev
el Admin Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Help Desk
Admin Role,dc=sesta,dc=com)(nsroledn=cn=Organization Admin
Role,dc=sesta,dc=com)(nsroledn=cn=Top-level Policy Admin
Role,dc=sesta,dc=com)))")(targattrfilters="add=nsroledn:(nsroled
n=cn=JDC,dc=sesta,dc=com),del=nsroledn:(nsroledn=cn=JDC,dc=sesta
,dc=com)")(version 3.0; acl "Allow JDCAdmin1 Role to add/remove
users to JDC Role"; allow
(write)roledn="ldap:///cn=JDCAdmin1,dc=sesta,dc=com";)


JDCAdmin2|Add/remove channels from the JDC
role|dc=sesta,dc=com:aci:(target="ldap:///cn=SunPortalDesktopSer
vice,dc=sesta,dc=com")(targetfilter=(cn=cn=JDC,dc=sesta,dc=com))
(targetattr="*")(version 3.0; acl "Allow JDCAdmin2 to edit
display profile of JDC Role"; allow (all)
```

```
roledn="ldap:///cn=JDCAdmin2,dc=sesta,dc=com";)##dc=sesta,dc=com
:aci: (target="ldap:///dc=sesta,dc=com")(targetattr = "*")
(version 3.0; acl "Allow JDCAdmin2 to read and search all"; allow
(read,search) roledn = "ldap:///cn=JDCAdmin2,dc=sesta,dc=com";)
```

The new ACI appears in the Default Role Permissions (ACIs) list.

5. Click Save.

To Create a New Admin Role for the Delegation Model

Once you have created an ACI defining the permissions for a delegated administration role, you must create a role for using that ACI definition.

1. Log in to the Sun ONE Identity Server administration console as Top-level Admin or Organization Admin.

   By default, Identity Management is selected in the location menu, and Organizations is selected in the navigation pane.

2. Navigate to the organization or suborganization where the role will be created.

   All created organizations are displayed in the navigation pane.

NOTE If this is a new organization, you must register all the services and create the appropriate templates. See Chapter 2, “Administering Authentication, Users, and Services” for more information.

3. Choose Roles from the View menu and click New.

   The New Role page appears in the data pane.

4. Enter a name, select static role, and click Next.

5. Enter the description and choose Administrative as the type.

6. Select the Access Permissions:

   a. If you created the ACI definition for the role using the Administration Console, select the role you created from the Access Permissions list.

   b. If you created the ACI definition for the role using the command line, select No Permissions as the role name will not be listed in the Access Permissions list.

7. Click Create.

   The new role appears in the navigation pane.

To Assign a Role Administrator Role

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location menu, and Organizations is selected in the navigation pane.

2. Navigate to the organization or suborganization where the role was created.

   All created organizations are displayed in the navigation pane.

3. Choose Roles from the View menu.
4. Click the properties arrow for the role to assign.
5. Choose Users from the View menu in the data pane and click Add.

   The Add Users page appears in the data pane.

6. Specify the values for the fields to find the user to assign and click Filter.

   A list of users displays.

7. Check the box next to the users to which to assign the role or click Select All to choose all the users.
8. Click Submit.

   The list of users for this role box is updated with the assigned users.

To Configure Additional Restrictions on a Role Administrator Role

You can configure a role with a restricted set of capabilities. One common restriction you might want is a role with permissions to modify the display profile and perform content management functions, but that is restricted from viewing the rest of the Desktop attributes.

You can also set up delegated administrators with a start DN view. The start DN view is the directory location below which the delegated administrator can see and modify entities.

To configure additional restrictions on a role:

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location menu, and Organizations is selected in the navigation pane.

2. Navigate to the organization or suborganization where containing the role to configure.

   All created organizations are displayed in the navigation pane.

3. Choose Roles from the View menu.

4. Select the role to configure.

5. Select Services from the View menu.

6. To restrict the role to only display profile or channel management capabilities, do the following:

   a. Click the Edit link for the Desktop service.

   b. Create a User service template at this role.

      The Desktop page appears in the data pane.

   c. Unselect the Show Desktop Attributes checkbox.

   d. Specify a DN in Admin DN Starting V.

   e. Click Save.

NOTE If the Show Desktop Attributes checkbox is unselected, when users with this role access the Desktop services, they will not be able to see the Desktop attributes; they will only see the Channel and Container Management link. In addition, they will only be able to see the channels and containers defined at the role level.

7. To restrict the role to a particular start DN, do the following:

   a. Click the Edit link for the User service.

   b. Create a User service template for the role.

      The User page appears in the data pane.

   c. Specify a DN in Admin DN Starting View. For example, `cn=JDC, dc=sesta, dc=com`.

d. Click Save.

Chapter 4

Administering the Portal Desktop Service

This chapter describes how to administer the Sun™ ONE Portal Server Desktop service.

This chapter contains these sections:

Overview of the Desktop

This section describes the Desktop component, its underlying structure, and how you administer it.

Desktop Glossary

Table 4-1 describes the pertinent Desktop terminology.

The first column of the table lists the term; the second column provides a definition of the term.

Table 4-1 Desktop Glossary

Term	Definition
Desktop	Provides the primary end user interface for Sun ONE Portal Server.
Provider	Adapts the interface of a generic resource for use use by the Sun ONE Portal Server. A JSP provider compiles and executes a JSP file to generate a markup. An XML provider translates an XML file to generate a markup The portal server can also query the provider for information to display a markup on a portal page.
Portlet	Pluggable web components that process requests and generate content within the context of a portal. Portlets are managed by the Portlet Container (an implementation of the Portlet Specification as defined by the JSR168 Expert Group). Conceptually they are equivalent to the software Providers.
Channel	Displays content in the Desktop, usually arranged in rows and columns. At runtime, a channel consists of a provider object, configuration, and any data files (JSP, HTML templates, and so on) required to support the channel.
Container or Container Channel	A channel that primarily generates its content by including or aggregating the content of other channels (referred to as child channels).

Portal Desktop Architecture and Container Hierarchy

The Desktop is the primary end-user interface for Sun ONE Portal Server. It is implemented through a servlet and is supported by various APIs and utilities (for example, Sun™ ONE Identity Server APIs, resource bundles, properties files, back-end servers such as mail, and so on).

The Desktop provides a mechanism for extending and aggregating content through the Provider Application Programming Interface (PAPI). Content providers, or providers, enable container hierarchy and the basic building blocks for building some types of channels. Usually, channels are arranged in rows and columns, but they can also be displayed in some other arrangement, depending on the implementation of the container channels. The provider is the programmatic entity responsible for the generation of content, which is displayed in the channel. Generated content can consist of entire pages, frames, or channels; any markup.

As the amount of content on a portal increases, a containment method for referencing or referring to groups of content can facilitate the portal configuration, development, and end-user experience.The Sun ONE Portal Server provides a flexible, extensible set of container providers to aggregate content.

Figure 4-1 provides an example of the Desktop container hierarchy. In this figure, a Tab container is the top-level container. The Tab Container contains two Tab Channels, Tab 1 and Tab 2. Tab 2 is a Table Container and contains five channels.

Figure 4-1 Sample Portal Desktop Container Hierarchy

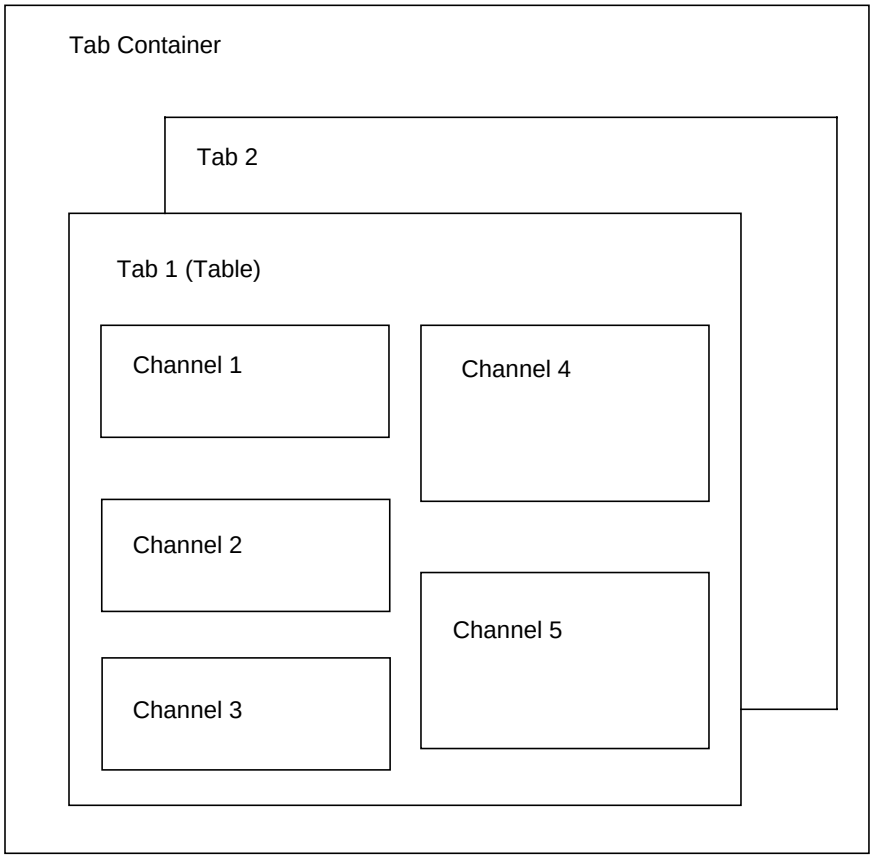


Figure 4-1 illustrates the following containment types:

- **Tab Container** - Contains any number of table, single or tab containers. This container also includes contains the banners, and menu bars for the portal as well.
- **Tab Channel** - Aggregates the output of other channels, providing a tabbed user interface to switch between them. Tab containers configuration are modified at runtime to vary which leaf channel is displayed.
- **Table Container** - Aggregates the content of other channels into rows and columns. This container functions much like the Sun™ Portal Server 3.0 front provider. It can be thought of as a bucket for the content of other channels.

User Defined Channels

Each tab in a tab container includes a Content link. If you select the Content link, a page where a user can select the channels they would like to appear in the current tab's container is displayed. In this release, an additional link on the top right of this page, Create New Channel link, is included. The Create New Channel link, when selected, presents a page where a user can create a new channel. However, the channels that can created by the user is definable by the administrator.

To create a new channel (from the page shown in), the user must specify the information outlined in Table 4-2 on page 110 in the form presented.

Table 4-2 User Defined Channels

Form Field	Field Type	Field Description
Channel Name	Text field	Channel name may contain only letters (a-z,A-Z) and digits (0-9).
Channel Title	Text field	This is the title that will appear in the Channel titlebar.
Channel Description	Text field	This is the description for the Channel that appears on the Content link page.
Channel Type	Combo box	This is a list of Providers that new Channels can be created from.
Channel Category	Combo box	This is a list of the Categories for the Tab's Container.

Table 4-2 User Defined Channels

Form Field	Field Type	Field Description
Display Channel	Radio buttons with "Yes" and "No"	Select Yes for Display Channel so that the new Channel will automatically be displayed when the Browser is refreshed after selecting the Create button. Select No so that the Channel will not automatically be displayed when the Browser is refreshed after selecting the Create button. Instead, the channel can be displayed in the Browser by selecting the Channel from the Content link. In either case, once the new Channel is selected and displayed in the Browser, it is necessary to update its properties by selecting the Edit button which is available in the newly created Channel's titlebar.
Create	Button	Select Create to create the new Channel.
Cancel	Button	Select Cancel to return the user to their Desktop display.

The Delete A Channel link is displayed on the Content page after a user has created a user-defined channel. When a user clicks on the link, a list of all of the channels that the user created is displayed for possible deletion.

Portal Desktop Providers

Sun ONE Portal Server uses two types of providers:

- Building Block Providers—Extendable providers whose interfaces are public. These providers connects to a generic resource (like a JSP file). These providers can generate more than one channel in the Portal Desktop, thus the relationship between the provider and the channel is one to many.
- Content Providers—Non-extendable providers expects a specific set of data in order to render (for example, a bookmark provider expects a specific template and data). These kind of providers are not building block providers.

The Portal Desktop uses a *display profile* for storing content, provider, portlet, and channel data. See Chapter 5, "Administering the Display Profile" for more information.

Portal Desktop Service

The Desktop service uses Sun ONE Identity Server services to store application and user-specific attributes for each organization or suborganization. You then create a display profile policy and assign it to users. You also use the Sun ONE Identity Server Sun ONE Identity Server administration console to modify Desktop attributes. See Appendix C, “Portal Desktop Attributes” for more information.

Sample Desktops

Within the sample Desktops, Sun ONE Portal Server includes the following channels:

- Bookmarks
- Applications
- User Information
- Search
- Notes
- Mail Check
- Login
- Simple Web Service
- Simple Web Service Configurable

These channels are customized and configured for the sample portal. They may require the modification of the user interface before they are deployed.

Portal Desktop Customization

When deploying Sun ONE Portal Server, one of your major tasks will be to develop, or customize your own portal. You will create create and extend providers, channels and container channels, deploy your own online help, come up with a look-and-feel, and so on. If desired, you can use the sample Desktops as a starting point in customizing your site’s portal. See the *Sun ONE Portal Server 6.1 Desktop Customization Guide* for more information on customizing your portal.

Overview of Hot Deployment of Channels

Sun ONE Portal Server enables you to deploy providers and channels on a live system without performing a restart, hence the “hot deployment.” You can do so without interrupting user sessions.

The three technologies that facilitate hot deployment are:

- Provider class loader—Reloads providers and classes used by providers. For the provider class loader to function properly, all classes (or JAR files) must reside in a well-defined directory.
- Display profile refresh—Updates the in-memory Desktop configuration, that is, the display profile, if it has been changed by an external source such as the Sun ONE Identity Server Sun ONE Identity Server administration console or the `dpadmin` command.
- Portal Desktop template and JSP reloading—Retrieves the appropriate template and JSP files for the Desktop type configured.

Overview of Provider Archives

The `par` utility enables you to package and transport channels, portlets, and providers, and all associated files, in and out of the Sun ONE Portal Server system. The channel, portlet, or provider is stored in the `.par` file format. Files included in the `.par` include:

- Display profile documents
- Class files
- Provider resource bundle files (property files)
- Templates and JSP files
- Static content files, that is, HTML and image files

Administering the Portal Desktop Service

The Desktop merges all of the documents in a user’s display profile merger set and uses the result to configure the user’s desktop. A display profile merger set consists of all the display profile documents associated with a user. Display profiles are defined at different levels in the Sun ONE Identity Server organization tree.

Display profile documents from the various levels of the tree are merged or combined to create the user's display profile. For example, the user's display profile document is merged with the role display profile documents (if any), the organization's display profile document, and the global display profile document to form the user's display profile.

The Desktop display profile and other configuration data are defined as service attributes of the Portal Desktop service under the Sun ONE Identity Server service management framework. When an organization registers for the Portal Desktop service from the Sun ONE Identity Server administration console, all users within the organization inherit the Portal Desktop service attributes in their user profiles. These attributes are queried by the Portal Desktop to determine how information will be aggregated and presented in the Portal Desktop.

By default, the Policy Configuration service is automatically registered to the top-level organization. Suborganizations must register their policy services independently of their parent organization. Any policy service you create must be registered to all organizations.

The following describes the high-level steps that you perform to configure the Portal Desktop service for users in an Sun ONE Identity Server organization:

1. Registering the Policy service for an organization.
2. Creating a referral policy for a peer or suborganization.
3. Creating a normal policy for a peer or suborganization.
4. Assigning a default redirect URL.
5. Customizing Desktop service attributes.

NOTE If you install the sample portal, the installer installs all the necessary display profile XML files for the sample. You can customize the profiles using the Sun ONE Identity Server console or the command-line interface. See Chapter 5, "Administering the Display Profile" for further information.

By default, the Policy Configuration service is automatically registered to the top-level organization. Suborganizations must register their policy services independently of their parent organization. Any policy service you create must be registered to all organization. The high-level steps to use policies are:

1. Registering the Policy service for an organization. (This will be done automatically for the organization specified at installation.) Suborganizations do not inherit their parent's services, so you need to register a suborganization's Policy service. See To Register a Policy Service for a Suborganization for information.
2. Creating a referral policy for a peer or suborganization. You can delegate an organization's policy definitions and decisions to another organization. (Alternately, policy decisions for a resource are delegated to other policy products.) A referral policy controls this policy delegation for both policy creation and evaluation. It consists of a rule and the referral itself. If the policy service contains actions that do not require resources, referral policies cannot be created for suborganizations. See To Create a Referral Policy for a Suborganization for information.
3. Creating a normal policy for a peer or suborganization. You create a normal policy to define access permissions. A normal policy can consist of multiple rules, subjects, and conditions. See To Create a Normal Policy for a Suborganization for information.

To Register a Policy Service for a Suborganization

Suborganizations do not inherit their parent's services, so you need to register a suborganization's Policy service.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose the organization for which you would like to register the Desktop service.
3. Choose Services from the View menu in the navigation pane.
4. Click Register in the navigation pane.

   A list of available services displays in the data pane.
5. Select the check box for Portal Desktop under Portal Server Configuration and click Register.

   The Navigation pane is updated with the registered Desktop service under Portal Server Configuration.

6. Choose Services from the View menu in the navigation pane.
7. Click the properties arrow next to Desktop in the navigation pane.
8. A question is displayed in a message box in the data pane to confirm if a service template should be created for the Desktop service. Click Create in the message box to create the template.
9. After the page is submitted and the template created, the data pane displays a list of Desktop service attributes and their default values, if any. Modify the values as needed. When done, click Save to store the final values in the service template.

   The display profile of a newly created service template takes on the value entered in the Dynamic section of the Portal Desktop service under Service Management. If those values were blank, the display profile in this new template is also blank.

NOTE The default value for the Conflict Resolution Interval attribute is "Highest." Setting up service templates at different levels (for example, organization and role) with the same priority for a registered service could lead to unexpected results.

To Create a Referral Policy for a Suborganization

You can delegate an organization's policy definitions and decisions to another organization. A referral policy controls this policy delegation for both policy creation and evaluation. It consists of a rule and the referral itself. The referral must define the parent organization as the resource in the rule, and it must contain a SubOrgReferral with the name of the organization as the value in the referral.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Select Identity Management from the navigation pane.
3. Select Policies from the View menu.
4. Click New to create new policy.

   The Create Policy page appears in the data pane.

5. For Name, type SubOrgReferral_Desktop. Make sure you select Referral in Type of Policy. Then click Create.
6. Select Desktop in Service and click Next
7. Click Rules from the View menu in the data pane and click New. Make sure Portal Desktop is selected and click Next.

   The New Rule template appears in the data pane.
8. Enter DesktopRule in Rule Name and click Create.
9. Click Referrals from the View menu in the data pane and click New.

   The New Referral template appears in the data pane.
10. Enter SubOrgReferral_Desktop in Name.

    Make sure that the name of the suborganization is selected for Value in the data pane and click Create to complete the policy's configuration.
11. Click Save in the data pane.

    The message "The policy properties have been saved" is displayed when the data is saved.

To Create a Normal Policy for a Suborganization

You create a normal policy to define access permissions. A normal policy can consist of multiple rules, subjects, and conditions.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Navigate to the organization or suborganization that you want to assign a policy.

   All created organizations are displayed in the navigation pane.
3. Choose Policies from the View menu.

   The policies for that organization are displayed.
4. Select New in the navigation pane. The New Policy page opens in the data pane.

5. Enter SubOrgNormal_Desktop in Name. Make sure you select Normal in Type of Policy. Click Create
6. Choose Rules from the View menu in the data pane and click New. The New Rule page opens in the data pane
7. Select Portal Desktop from the Service menu and click Next. Enter DesktopRule in Rule Name. Make sure Has Privilege to Execute NetMail is checked
8. Select Portal Desktop from the Service menu and click Next. Make sure Has Privilege to Execute NetMail is checked.
9. Select the type of subject from the Type menu and click Next to complete subject configuration.
10. Choose Subjects from the View menu in the data pane and click New. The New Subject page opens in the data pane.
11. Click Create to complete the policy configuration.

    The message “The policy properties have been saved.” is displayed when the data is saved.

To Redirect Successful Login User to the Portal Desktop URL

By default, users in an organization receive the Desktop service attributes and values after successfully logging in. These values are queried by the Desktop servlet to determine the Portal Desktop contents of any users in the organization. To instruct Sun ONE Identity Server to invoke the Portal Desktop servlet automatically after a user has successfully logged in, you can change the value of the Default Redirect URL to the Portal Desktop URL.

To set the default redirect for a specific organization to the Portal Desktop URL:

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Select the organization for which you want to set the Portal Desktop URL.
3. Choose Services from the View menu.
4. Click the properties arrow next to Core in the navigation pane.

5. In the data pane, search for an attribute named User's Default Redirect URL.
6. Set the value of the User's Default Redirect URL to the URL for the Portal Desktop servlet, for example, `/portal/dt` is the URL for the sample Desktop.
7. Click Save.
8. Verify the default redirect URL by logging in to the Portal Desktop.

To Redirect Successful Login User to the Portal Desktop URL (Global)

The values applied to the global attributes are applied across the Sun ONE Identity Server configuration and will be inherited by every newly created organization.

To set the Default Redirect URL to the Portal Desktop URL globally:

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Management in the location pane.
3. Click the properties arrow next to Core in the navigation pane.
4. In the data pane, search for an attribute named User's Default Redirect URL.
5. Set the value of the Default Redirect URL to the URL for the Portal Desktop Servlet, for example, `/portal/dt`.
6. Click Save.

To Modify the Values of Portal Desktop Service Attributes

You can customize the Portal Desktop service by modifying its service attributes.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Select the organization for which you want to modify the Desktop attributes.

3. Click the properties arrow next to Desktop in the navigation pane.

   A list of Portal Desktop service attributes, including the display profile XML, is displayed in the data pane.

4. Modify the service attribute values.

   See Appendix C, "Portal Desktop Attributes" for information on the attributes.

5. When done, click Save.

   The changes will affect only users in this particular suborganization or role.

To Modify the Values of Portal Desktop Service Attributes (Global)

Occasionally, you need to modify the global Desktop service attribute values that affect all organizations that want to register for the Desktop service in the future.

The values applied to the global attributes are applied across the Sun ONE Identity Server configuration and are inherited by every configured organization.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Management in the location pane.

3. Click the properties arrow next to Desktop in the navigation pane.

   A list of global Desktop service attributes, including the display profile XML, is displayed in the data pane.

4. Modify the service attribute values.

   See Appendix C, "Portal Desktop Attributes" for information on the attributes.

5. When done, click Save.

   The changes affect all organizations that register the Desktop service in the future.

To Access the Sample Portal Desktop

1. Log out from the Sun ONE Identity Server administration console.

2. Log on with a user account (not the `amadmin` user) using the following URL:

   `http://`*server*:*port*`/portal/dt`

   If you need to create a user account, see Chapter 2, “Administering Authentication, Users, and Services” on page 39 for information.

To Examine the Desktop Logs

Portal Desktop errors on the are logged to debug log files. By default, the location of these log files is as follows.

- `/var/opt/SUNWam/debug/desktop.debug`
- `/var/opt/SUNWam/debug/desktop.dpadmin.debug`

Examine these log files for errors. An example follows. This error indicates that an unauthenticated user attempted to execute the Portal Desktop.

```
06/20/2002 02:36:30:600 PM PDT: Thread[Thread-177,5,main]
ERROR: DesktopServlet.handleException()
com.sun.portal.desktop.DesktopException:
DesktopServlet.doGetPost(): no privilige to execute desktop
        at
com.sun.portal.desktop.DesktopServlet.doGetPost(DesktopServlet.j
ava:456)
        at
com.sun.portal.desktop.DesktopServlet.service(DesktopServlet.jav
a:303)
        at
javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
        at
com.sun.server.http.servlet.NSServletRunner.invokeServletService
(NSServletRunner.java:897)
        at
com.sun.server.http.servlet.WebApplication.service(WebApplicatio
n.java:1065)
        at
com.sun.server.http.servlet.NSServletRunner.ServiceWebApp(NSServ
letRunner.java:959)
```

Administering Portlets

Portlets are administered from the Sun™ ONE Identity Server administration console. The administration console includes pages for creating portlet channels from portlets and changing preferences of portlet channels. The `pdeploy` is a command line tool that can be used to deploy and undeploy the portlet web application into a web container (see Command-Line Utilities for more information).

NOTE If a client request accesses a portal page which contains at least one session-enabled portlet, it is strongly recommended that all the portlets on that portal page should be packaged within a single portlet application, otherwise the resulting behavior of the session creation may be nondeterministic."

To Create a Channel from a Portlet

1. Log in to the Sun ONE Identity Server administration console and select your organization.
2. Select Services under Show in the navigation menu.
3. Select the Desktop service from Portal Server Configuration.
4. Select Channel and Container Management link.
5. Select the Add Portlet Channel button under Channels.

   The page to create a portlet channel is displayed.
6. Specify in the Add Channel page,
   - The channel name.
   - Note that channel names can contain only letters (A through Z) and digits (0 through 9) and it is a required field.
   - The Portlet

   Only contains portlets that are deployed in the system are displayed.
7. Select the Create button to create the portlet channel.

Figure 4-2 Add Portlet Channel Page Sample

To Create a Channel from a Portlet for a Specific Container

1. Log in to the Sun ONE Identity Server administration console and select your organization.
2. Select Services under Show in the navigation menu.
3. Select the Desktop service under Portal Server Configuration.
4. Select Channel and Container Management.
5. Select the link to the Container where you wish to create a portlet channel.

   The page for managing the container is displayed.
6. Select the Add Portlet Channel button under Channels.

   The page for creating and adding a portlet channel is displayed. .
7. Specify, in the Add Channel page:
   - A name for the channel.
   - The Portlet from the pull-down list. The list only contains portlets that are deployed in the system.
8. Whether the channel will be available to end-users or whether it will be available and visible on the Desktop by selecting the appropriate radio button.
9. Select the OK button.

   Note that the channel is added to the list of channels under Channels and under Available and Visible in the Container Management page.

To Add the Portlet Channel to a Container

1. Log in to the Sun ONE Identity Server administration console and select your organization.
2. Select Services under Show in the navigation pane.
3. Select Desktop from Portal Server Configuration.
4. Select Channel and Container Management.

5. Select the link to the Container to which you wish to add the newly created portlet channel.

   The page for managing the container is displayed.

6. Select the portlet channel you wish to add Channel Management and select Add.

   This will add the selected portlet channel to the list of channels available and visible on the selected container.

7. Select Save button under Channel Management to save the new settings.

To Edit a Portlet Channel Preferences and Properties

The portlet preferences are defined in `portlet.xml`

```
<portlet-preferences>
           <preference>
               <name>foo</name>
               <value>apple</value>
           </preference>
           <preference>
               <name>bar</name>
               <value>orange</value>
               <value>grape</value>
               <read-only>true</read-only>
           </preference>
       </portlet-preferences>
```

is mapped to the following display profile:

```
<Collection name="__Portlet__AdditionalPreferences"/>
       <Collection name="__Portlet__PreferenceProperties">
           <Collection name="default">
               <String name="foo" value="|apple"/>
               <String name="bar" value="|orange|grape"/>
```

```
            </Collection>
            <Collection name="isReadOnly">
                <Boolean name="foo" value="false"/>
                <Boolean name="bar" value="true"/>
            </Collection>
        </Collection>
        <String name="__Portlet__foo" value="|apple"/>
        <String name="__Portlet__bar" value="|orange|grape"/>
```

There is an empty collection `__Portlet__AdditionalPreferences` created to hold the preferences added during runtime. The collection `__Portlet__PreferenceProperties` contains two collections, `default` and `isReadOnly`. The `default` collection stores the default values as defined in `portlet.xml`. Similar to the `default` collection, the `isReadOnly` collection stores the read-only flags of the preferences using Boolean properties.

Each preference in the `portlet.xml` has one corresponding `String` property in the `default` collection with the preference name as the property name. The value of the `String` property is to represent the default value defined in `portlet.xml` prepended and delimited by the character "|". Each preference is then represented by a `String` property which stores the current value of the preference. The name of the property is the name of the preference prepended by the string `__Portlet__`. The value of the property is the current preference values prepended and delimited by the character "|".

1. Log in to the Sun ONE Identity Server administration console and select your domain.
2. Select Services under Show in the navigation pane.
3. Select Desktop from Portal Server Configuration.
4. Select Edit link for the portlet channel you wish to edit.
5. The Edit Channel page is displayed. The channel edit page displays the portlet preferences for the portlet entity.
6. Modify the preferences and select Save to save the modifications.
7. To modify the default values of the preferences, select Edit link for the preference you wish to edit. Properties can be edited in the Edit Channel page.

Administering par Files

The par utility enables you to transfer or move providers or channels from one Sun ONE Portal Server to another. The par utility creates a specialized packaging mechanism called a .par file for transport of channels, portlets, and providers into and out of the server. A .par file is an extended form of the .jar file format, with added manifest information to carry the deployment information and an XML document intended for integration into the Sun ONE Portal Server display profile on the target server.

The par command line utility is used to create, modify, and deploy par files. The export subcommand allows you to create or modify a par file. The import subcommand allows you to import or deploy the provider, channel, or portlet on an Sun ONE Portal Server. The describe subcommand describes the contents of a par file. See par for detailed information on the syntax of the par command.

To use the par utility, you must be logged in as superuser to the Sun ONE Portal Server on which the files you want to export or import are resident. When you export you need to be sure to export all the required files for the channel, portlet, or provider. For example, with channels you must include the static content files and with providers you must include all the class files used by the provider. Because specifying all the data to be included in the par file on the command line can be cumbersome, a simple text file with lines indicating the data is created and this "export file" is called by the par utility. See Chapter 4, "Administering the Portal Desktop Service" for further information.

To Create a New par File

To create a new par file to export a channel, portlet, or provider:

1. Log in to the Sun ONE Portal Server from which to export the channel, portlet, or provider.
2. Change directories to the directory where the script is installed. That is:

   cd *BaseDir***/SUNWps/bin**

3. At the command line, enter the `par export` command and subcommand and include the following arguments: the name of the par file to create, a directory server name argument corresponding to the desired display profile document to export, and any number of (requires at least one) export files or `from` specifications. For example, to export the channel mychannel from `o=sesta.com,o=isp` to the `mychannel.par` file, enter

   `./par export mychannel.par "o=sesta.com,o=isp" from: channel mychannel`

   See Chapter 14, "Command-Line Utilities" for syntax information.

To Modify an Existing par File

To modify an existing par file to export a channel, portlet, or provider:

1. Log in to the Sun ONE Portal Server from which to export the channel, portlet, or provider.
2. Change directories to the directory where the script is installed. That is:

   `cd` *BaseDir*`/SUNWps/bin`

3. At the command line, `par export` command and subcommand with the `modify` option and include the following arguments: the name of the par file to modify, a directory server name argument corresponding to the desired display profile document to export, and any number of (requires at least one) export tiles or `from` specifications. For example, to modify the `mychannel.par` file to include the static content file `/mycontent.html`, enter

   `./par export --modify mychannel.par "dc=sesta,dc=com" "from= file /mycontent.html"`

To Deploy par Files

To import a par file to an Sun ONE Portal Server to deploy a provider or channel on the system:

1. Copy the par file for the provider or channel to import to the Sun ONE Portal Server on which to deploy the provider or channel.
1. Log in to the Sun ONE Portal Server on which to import the channel, portlet, or provider.

2. Change directories to the directory where the script is installed. That is:

   cd *BaseDir***/SUNWps/bin**

3. At the command line, `par import` command and subcommand and include the following arguments: the name of the par file to import, a directory server name argument corresponding to the desired display profile document to export, For example, to import the `mychannel.par` file, enter

   ```
   ./par import --auto myfile.par "do=sesta,dc=com"
   ```

Chapter 5

Administering the Display Profile

This chapter describes the Sun™ ONE Portal Server display profile component and how to administer it.

This chapter contains these sections:

Overview of Display Profile

This section describes the display profile component of Sun ONE Portal Server.

The display profile creates the display configuration for the Desktop by defining the following three items:

- **Provider definition—**Specifies the name and the Java™ class for the provider. A provider is a template used to generate content, which is displayed in the channel. See Provider Object for more information.
- **Channel definition—**Specifies the run-time configuration of an instance of the provider class. A channel is a unit of content, usually (but not necessarily) arranged in rows and columns. You can also have channels of channels, that is, container channels.

- **Provider and channel property definitions—**Specify the values for provider and channel properties. Properties defined in a provider usually specify default values for the channels that are derived from the provider. The display configurations for the channels include properties such as the title, description, channel width, and so on. The properties defined in the channel usually specify the specific value for that channel that is different from the default value.

  Container properties define the display definition about how to display the contained channels in the container, including: the layout of the container (thin-wide, wide-thin, or thin-wide-thin); a list of the contained channels; the position of the channel (the row and column number); and the window state of the contained channels (minimized or detached).

NOTE The display profile does not actually define the overall layout or organization of what users see on their Desktops. The display profile exists only to provide property values for channels. However, the display profile does indirectly control some aspects of channel presentation, such as column layout for a table container or how the table container draws channels in a table.

The display profile determines the layout in that channel properties determine layout. For example, the display profile for the sample portal's table provider definition contains the following statement:

```
<Integer name="layout" value="1"/>
```

This refers to thin-thick columns. However, there is nothing here in the structure of the display profile regarding actual layout.

The display profile does not control such things as how `XMLProvider` parses XML, it only has a definition of the kind of rules (XSL file) that are in it.

The Portal Desktop implements a display profile data storage mechanism on top of the Sun™ ONE Directory Server Access Management Edition service for storing content provider and channel data. In addition, properties are set for the channels and providers.

The user's display profile is a series of XML documents describing container management and properties for channels. (One display profile document is equivalent to one XML document.) The display profile documents are stored in their entirety as a single attribute in the Sun ONE Identity Server services layer. That is, the display profile documents are an LDAP attribute residing in an instance of Sun™ ONE Directory Server.

To change display profile property values, the providers use the provider APIs (PAPI) to get and set the values. When the channel values are set to the display profile, the PAPI internal implementation uses the Sun ONE Identity Server SDK to set the display profile document in the Sun ONE Identity Server Desktop service attribute.

CAUTION Though possible, you should not edit the display profile using the Sun ONE Identity Server SDK.

Display Profile and the Administration Console

You can edit the display profile and other Portal Desktop service data through the Sun ONE Identity Server administration console and the `dpadmin` command. When you edit the display profile, you add, modify, and remove providers, containers, and channels, and edit properties. The Upload XML and Download XML links allow you to upload and download the display profile document. In addition, the Sun ONE Identity Server administration console provides an Channel and Container Management link in the Portal Desktop attributes page to add channels and containers and edit existing properties. The Channel and Container Management link enables you to define properties when a new channel or container is created. You can also use the Channel and Container Management link to add, modify, and remove channels and containers. See Administering the Display Profile for more information.

NOTE As the Channel and Container Management link enables access to only a portion of the display profile, it is envisioned that delegated administrators will use it. See Chapter 3, “Configuring Delegated Administration” for more information on how to configure delegated administrators.

Display Profile Document Structure

This section describes the overall structure of the display profile documents. The underlying data format for a display profile document is XML. See Appendix B, “XML Reference” for information on the display profile DTD syntax.

The display profile format establishes the Desktop's display configuration by defining provider and channel objects and their properties. The display profile is stored in the Sun ONE Directory Server at the `isp` level (or the top most directory node), the organization level, the role level, or the user level. At run time, a user's display profile is a result of "merging" all the display profile documents from the the user's specific profile in the directory tree, and the value of a specific display profile object for that user is decided by the "merge" semantics of the display profile.

The display profile objects map directly to the XML tag that defines them. For example, the `<Channel name> </Channel>` XML tags define a channel object.

In general, the document structure of the display profile resembles the following:

```
<DisplayProfile>
 <Properties>...global properties...</Properties>
 <Channels>...channel definitions...</Channels>
 <Providers>...provider definitions...</Providers>
</DisplayProfile>
```

`<Properties>`, `<Channels>`, and `<Providers>` are mechanisms to do grouping. These mechanisms make the XML display profile document more structured, so that like objects are in each "bag." See Putting Together Display Profile Objects for more information on "bags."

The following sections describe the display profile objects in more detail.

DisplayProfile root Object

The DisplayProfile `root` container object enables the Desktop servlet to act as a container provider to get handles to providers, and so forth. There is no actual provider class associated with the channel. This channel should not be referenced by any other display profile object.

DisplayProfile root Object XML Syntax

```
<Container name="_desktopRoot" provider="none">
    <Properties />
    <Available />
    <Selected />
    <Channels/>
</Container>
```

Provider Object

A provider object is the software entity executed at run time when a channel is rendered. (Thus, a channel is the instance of a provider at run time.) The `<Provider>` display profile definition is a template from which display profile channels are defined. It sets up the class name for the Provider java object and default values for all required properties.

The `<Provider>` display profile definition contains the information necessary for a client of the display profile to construct the `provider` object, namely, the Java™ class name.

The `<Provider>` display profile definition sets default property values for all channels that point to this provider. Channel-specific properties are only necessary when the provider defaults need to be overwritten. The provider display profile object should contain default values for all properties that are used in the provider Java object. For example, if the provider Java code contains:

```
getStringProperty("color")
```

Channel Object

A `channel` object represents a single display element. The objects contained by a `channel` object can be thought of as properties for the channel. The `<Channel name>` definition includes a symbolic reference to the provider. In addition, you can define channel-specific properties to overwrite default values defined in the provider definition. A channel name needs to be unique for a given channel within a display profile document, but you can define the same name at different channel levels.

Example Channel Object XML Syntax

```
<Channel name="SampleXML" provider="XMLProvider">

    <Properties >
        <String name="refreshTime" value="600" advanced="true"/>
        <String name="title" value="XML Test Channel"/>
        <String name="description" value="This is a test of the XML Provider
system"/>
        <String name="url"
value="file:///etc/opt/SUNWps/desktop/default/SampleXML/getQuotes.xml"/>
        <String name="xslFileName"
value="/etc/opt/SUNWps/desktop/default/SampleXML/html_stockquote.xsl"/>
    </Properties>

</Channel>
```

Container Object

A `container` object is identical to a `channel` object, except that it a `container` object does not generate content. That is, a container is a channel that gets its content from other channels. A `container` object allows for available and selected channel lists and can contain child channel definitions. A child channel is typically aggregated on a page with other channels and generates its own content. A container channel primarily generates content by aggregating the content of one or more child channels.

Example Container Object XML Syntax

```
<Container
name="TemplateTableContainer"provider="TemplateTableContainerProvider">
  <Properties>
    <String name="title" value="Template Based Table Container"/>
    <String name="description"
        value="This is the channel for the front provider"/>
    <Collection name="channelsColumn" advanced="true">
        <String name="SampleJSP" value="2"/>
        <String name="SampleXML" value="2"/>
        <String name="Notes" value="2"/>
    </Collection>
    <Collection name="channelsRow" advanced="true">
        <String name="MailCheck" value="3"/>
        <String name="SampleRSS" value="2"/>
        <String name="SampleXML" value="2"/>
        <String name="App" value="5"/>
        <String name="SampleSimpleWebService" value="6"/>
        <String name="Bookmark" value="4"/>
        <String name="Notes" value="3"/>
    </Collection>
    <Collection name="channelsIsRemovable">
        <Boolean name="UserInfo" value="false"/>
    </Collection>
  </Properties>
  <Available>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleXML"/>
    <Reference value="App"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Bookmark"/>
    <Reference value="Notes"/>
  </Available>
```

```
  <Selected>
    <Reference value="UserInfo"/>
    <Reference value="MailCheck"/>
    <Reference value="SampleRSS"/>
    <Reference value="SampleJSP"/>
    <Reference value="SampleXML"/>
    <Reference value="App"/>
    <Reference value="SampleSimpleWebService"/>
    <Reference value="Bookmark"/>
    <Reference value="Notes"/>
  </Selected>

  <Channels>
  </Channels>

</Container>
```

Putting Together Display Profile Objects

The `root`, `provider`, and `channel` objects can have properties associated with them. The display profile groups properties inside of a properties "bag." The term bag is used to indicate that its only purpose is a holding place for properties. A property does not have a properties bag associated with it. See *Sun ONE Portal Server 6.1 Desktop Customization Guide* for property definitions.

Property bags in channels, providers, and the root level have different semantics. Global properties are shared for all channels. A property defined as a global property here can be accessed by any channel. Themes are an example of a global property. Theme data isb defined globally so they can be shared among all channels.

Properties defined in providers are defaults for channels based on that provider. If the property is not defined in the channel, then the default is used. The implication is that a provider must define every property used by a provider Java object. Thus, if the Java code contains:

```
String f = getStringProperty("color");
```

the corresponding `<Provider name>` definition in the display profile must define:

```
<String name="color" ... />
```

NOTE Do not use global properties as defaults for all channels. A display profile provider definition defines the property interface used by the `provider` object that will use the provider definition.

Channel properties override the defaults in the provider definition to customize the channel. For example, `URLScaperProvider` defines a `url` property. A default does not make sense here, as a channel would naturally override this value.

Display Profile Object Lookup

At runtime, the system never asks for properties directly from a provider. The request always goes to a channel. If a Java `provider` object requests a property, it searches the display profile in the following order until it finds the property, or until it reaches the top of the containment hierarchy:

1. Channel's properties
2. Channel's provider's properties
3. Channel's parent's properties
4. Channel's parent's provider's properties
5. Channel's parent's parent's properties (and so on)
6. The global properties bag defined in the display profile `root` definition

Therefore, when a channel asks for the names of its properties, it gets the set of the union of all the above.

Properties that exist in a `provider` object are intended to have the semantics of default values for the channel. For example, for a provider `xml` that defines property `title`, all channels that are derived from provider `xml` inherit the `title` property. If the channel wants to override this property, it can set the value within its own properties.

Display Profile Properties

This section describes display profile properties and how to specify them.

Display Profile Property Types

The display profile property types are:

- Boolean—An atomic object representing a Boolean value. Example:

  ```
  <Boolean name="isEditable" value="false"/>
  ```

- Collection—An object representing either a list or hash table. A collection is a type of property, or named bag, in which to put other properties. Example:

  ```
  <Collection name="channelsRow">
      <String name="MailCheck" value="4"/>
      <String name="App" value="5"/>
  </Collection>
  ```

- Integer—An atomic object representing an integer value. Example:

  ```
  <Integer name="numberOfHeadlines" value="7"/>
  ```

- String—An atomic object representing a string value. Example:

  ```
  <String name="title" value="Table Container Channel 1"/>
  ```

- Reference—An object representing a pointer to a channel definition (that is, to a channel name in a container's selected and available channel lists.) Reference is an unnamed string useful for design tools to be able to distinguish such things from strings. Example:

  ```
  <Reference value="UserInfo"/>
  ```

Atomic property values can also be specified as body content. Example:

```
<String name=”foo”>bar</String>
<Integer name=”aNumber”>1</Integer>
<Boolean name=”flag”>false</Boolean>
```

Document Type Definition Element Attributes

The Portal Desktop DTD defines element attributes that allow you to control usage of display profile and its properties. Table 5-1 on page 139 lists document type definition element attributes. This three column table lists the attributes in the first column, a brief description in the second column, and an example in the third.

.

Table 5-1 Display Profile Attributes

Attribute	Definition	Example
advanced	"Hides" the display profile property from users in the iPlanet Directory Server Access Management Edition administration console Channel and Container Management link when set to true. However, the property is not hidden when using the Edit XML or Download XML links. The advanced attribute is a Boolean attribute that can take a value of true or false. The default value is false.	<String name="refreshTime" value="0" advanced="true"/>
lock	Enables low-priority documents to prevent a higher-priority document from using merge semantics to change particular aspects of the display profile. When a display profile object is locked, it cannot be affected by merge semantics in lower priority documents. The lock attribute is a Boolean attribute that can take a value of true or false. The default value is false.	<Selected merge="fuse"> ... <Reference value="EmployeeNews" lock="true"/> ... </Selected>

Table 5-1 Display Profile Attributes

Attribute	Definition	Example
`merge`	Controls how properties are combined as display profile documents from different LDAP nodes (base DN, DN, and role DNs) are merged to form a single representation (that is, Portal Desktop). Allowable values are `replace`, `remove`, and `fuse`. The default value is `fuse`. Note that fuse is not valid for atomic properties (boolean int, stringv ref).	See Display Profile Merge Types for replace, remove, and fuse examples.
`priority`	Sets the priority of the display profile document. Display profile documents are merged from low priority to high priority. A lower number represents a lower priority. For example, a 1 is a lower priority than a 2. High priority documents override values set in lower priority documents using merge semantics (unless a lower priority document has locked the object for merging). Allowable values are integers and the keyword `user`. The priority `user` is the highest priority, and it should onle be set for user-level display profile documents.	`<DisplayProfile version="1.0" priority="10">`

Table 5-1 Display Profile Attributes

Attribute	Definition	Example
`propagate`	Controls how properties are treated when they are read non-locally but set locally. You can mark all display profile properties, including Boolean, Collection, Integer, Strings, and Reference, with the `propagate` attribute. The `propagate` attribute is a Boolean attribute that can take a value of `true` or `false`. The default value is `true`.	`<String name="color" value="blue" propagate="false"/>`

In the display profile XML, the following attributes are not listed in the XML file and displayed in the administration console unless the attribute's default value has been changed:

```
<advanced="false" lock="false" merge="fuse" propagate="true">
```

If a default value is reset, only the attribute whose default value has been changed is included in the XML fragment and displayed in the administration console. The default properties are inherited from the provider. If the default property is edited, it is displayed as customized.

Specifying Display Profile Properties

When you specify display profile properties, you need to consider how to "nest" them, how to use unnamed properties in collections, how to use conditional properties and how properties can be propagated.

Property Nesting

The display profile can contain nested properties (properties within properties) to any depth. This enables you to have collections of collections of collections of strings, or a collection of strings and collections, and so on. For example, here is a collection of collections:

```
<Collection name="people">
    <Collection name="john">
        <Integer name="age" value="31"/>
        <String name="eyes" value="hazel"/>
    </Collection>
    <Collection name="bob">
        <Integer name="age" value="35"/>
        <String name="eyes" value="blue"/>
    </Collection>
    ... etc ...
</Collection>
```

Unnamed Properties

Atomic property types (Boolean, Integer, and String) can be unnamed, for example:

```
<String value="apple"/>
```

is equivalent to

```
<String name=”apple” value=”apple”/>
```

That is, if an atomic property does not have a name then it is equivalent to the string value of that property.

For all practical purposes, this is useful only inside a collection, because it enables you to use collections to represent an ordered set (almost a list), instead of a table. For example, here is a collection representing a list of zip codes:

```
<Collection name="zipcodes">
    <Integer value="95112"/>
    <Integer value="95054"/>
    <Integer value="98036"/>
</Collection>
```

The key to using unnamed properties is that collections can represent tables (*name=value*) or lists.

NOTE Do not create an unnamed property with the same value as another unnamed property in the same collection. The property will be created, but the provider will not be able to access the value because of the duplicate name.

In addition, because the Sun ONE Portal Server treats a property that has the same name and value as equivalent to an unnamed Boolean property you may unintentionally create properties with duplicated names in the same collection. This again can result in all but one being inaccessible.

Conditional Properties

This provides a generic operation for retrieving conditional properties. The most common conditions are `locale` and `client`, but you can define properties on any sort of condition. See the *Sun ONE Portal Server 6.1 Desktop Customization Guide* for more information.

For instance, the implementation of the locale filter is:

```
public class LocalePropertiesFilter extends PropertiesFilter
{
    public LocaleProperties() {
    super();
    }
    String getCondition()
    return "locale";
    }
    public boolean match(ProviderContext pc, String  condition,
String value) {
    return condition.toLowerCase().equals("locale")     &&
        getValue().equals(value);
    }
}
```

A conditional property lookup involves one or more property filters. If a filter in the filter list is required, then it must match for the overall conditional lookup to succeed. If a filter is not required, then it can fail to match without causing the overall lookup to fail.

A chain of non-required filters can be used to implement a progressively less-specific filter lookup, similar to the semantics of Java resource bundle lookup. For instance, an optional filter would be useful in a case where a locale lookup is followed by a date lookup. Given the filter {locale=en, locale=US, date=03/03/2003}, you can get it to successfully match a property with the qualifier {locale=en; date=03/03/2003} even though it does not exactly match the filter specification. This is done by setting the locale filter to be optional.

In the administration console, the conditional properties are displayed as condition-value and can be edited like collections. The conditional properties can be nested and can be added to a channel or inside another conditional property. Use the Add Property page to add a new conditional property.

<ConditionalProperties> Tag

The <ConditionalProperties> tag must be used to define the filtering criteria. The tag contains the following required attributes:

- condition: Specifies condition on which the filter should operate
- value: Specifies the value to be used in the filter

In the display profile, the <ConditionalProperties> tag can be defined as outlined in Code Example 5-1 on page 144.

Code Example 5-1 <ConditionalProperties> Tag Usage Sample

```
<Properties>
    <String name="foo" value="bar">
    <ConditionalProperties condition="locale" value="de">
        <String name="foo" value="german bar">
        <String name="baz" value="a german baz value">
    </ConditionalProperties>
    <ConditionalProperties condition="client" value="nokia">
        <ConditionalProperties condition="locale" value="de">
            <String name="foo" value="nokia german bar">
        </ConditionalProperties>
    </ConditionalProperties>
</Properties>
```

Display Profile Property Propagation

You can mark all display profile properties, including Boolean, Collection, Integer, Strings, and Reference, with the `propagate` attribute. The `propagate` attribute is a Boolean attribute that can take a value of `true` or `false` (the default is `true`). The `propagate` attribute controls how properties are treated when they are read non-locally but set locally.

For example, the set of properties for a channel consists of the set that is the union of:

- The set of properties existing locally in the channel's properties (`<Properties>`) bag
- The set of properties existing locally in the channel's provider (specified by the `provider` attribute on the channel)
- The set of properties existing locally in each ancestor channel of the channel (channel's parent, channel's parent's parent, and so on)
- The set of properties existing locally in each ancestor channel provider of the channel (channel's parent provider , channel's parent's parent provider, and so on)
- The set of global properties existing under the display profile `root` object

When a channel requests a property value, it can be read from any of these "remote" locations. When a property value is set, there are two options where to store the property value:

1. The channel's property bag
2. The remote location

The `propagate` attribute controls the location. When you set the `propagate` attribute to `true`, a property is stored locally to the object that set the property (in most cases, a channel). When you set the `propagate` attribute to `false`, the property is set in place (wherever it was read from). That is, when set to `false`, the existing value is changed, but when `true`, a new property is created and stored locally (unless it was already local).

Consider the following display profile XML fragment:

```
<DisplayProfile>
  <Properties>
    <String name="color" value="blue"/>
  </Properties>
...
<Channel name="testchannel" provider="..."/>
  <Properties/>
</Channel>
...
</DisplayProfile>
```

The property `color` lives in the global properties bag. Because `propagate` is not set (and is `true` by default), the following results if channel `testchannel` sets property `color`:

```
<DisplayProfile>
  <Properties>
    <String name="color" value="blue"/>
  </Properties>
...
<Channel name="testchannel" provider="..."/>
  <Properties/>
    <String name="color" value="new value"/>
</Channel>
...
</DisplayProfile>
```

The property is propagated to the local object that set it (the channel). On the other hand, if `propagate` were set to `false` in the global properties bag, for example:

```
<String name="color” value="blue" propagate="false"/>
```

The result of channel `testchannel` setting property `color` would be:

```
<DisplayProfile>
  <Properties propagate="false">
    <String name="color" value="new value"/>
  </Properties>
...
<Channel name="testchannel" provider="..."/>
  <Properties/>
</Channel>
...
</DisplayProfile>
```

In addition to individual properties, a properties bag can also be marked with the `propagate` attribute, for example:

```
<Properties propagate="false">
...
</Properties>
```

For a property to be considered as `propagate=false`, the following must be true:

- The property's `propagate` attribute must be `false`, or the property's properties bag's `propagate` attribute must be set to `false`.
- The above statement must be true for all mergers of the property.

For anything else, `propagate` is considered to be `true`.

You can only mark top-level properties with the `propagate` attribute. The display profile DTD does not disallow this but the display profile code ignores it. A top-level property is defined directly inside the properties bag.

Display Profile Document Priorities

At runtime, when a user logs in, the system determines the set of documents that makes up the user's display profile document set. The Desktop internal implementation of the display profile (the part that interprets the display profile) determines this set by looking at all of the LDAP nodes that the user belongs to. This can be the organization DN (`o=sesta.com`), suborganizations, role DNs

(cn=Role1,o=sesta.com), and uid (uid=jtb,ou=People,cn=Role1,0=sesta.com), as well as the global display profile. The display profile documents from each of these LDAP nodes and global display profile are then read (if it exists there), and all of the documents are put into a set. The system sorts the set according to the document priorities. A lower number represents a lower priority. For example, a 1 is a lower priority than a 2. The documents are then sorted from lower number to higher number. See How the Merge Process Works for more information on this process.

The user level document (uid=jtb,ou=People,. . .) is a special case referred to as the *base document*. Think of the base document as a priority equal to infinity. Thus, it is always the highest number (and hence highest priority). All of the mergers are associated with the base document in sorted order, and the priority setting on a user document is always the highest. The priority attribute used in the <DisplayProfile> tag takes the special keyword user to indicate that the current display profile is the user level display profile.

When a merge occurs, it starts at the lowest priority document (lowest number) and proceeds in increasing priority number, until it arrives at the user (base) document.

Thus, the implication of display profile document priorities is that what really matters is the priority number. For example, an organization level document can have a higher priority than a role level document, but it does not have to. It depends on how you need to prioritize these documents for your site.

You specify the display profile document priority in the XML file with the <DisplayProfile priority= *syntax*> tag. You can change the priority by directly editing the display profile XML by using the Sun ONE Identity Server administration console or by using the dpadmin command to load the display profile. See Chapter 14, "Command-Line Utilities" for more information on the dpadmin command.

NOTE Do not assign the same priority to two display profile documents. Doing so causes the Desktop to not appear properly. However, the product does not check for duplicate document priorities.

Document Priority Example 1

This example uses two display profiles, one for the organization acme and one for the uid bill. When Bill logs in (uid=bill) to the Desktop, the bookmark channel titled "Bill's Bookmarks" is displayed with the following three bookmarks (in that order):

- ACME
- Amazon
- EBay

```
display profile @ o=acme.com
<DisplayProfile version="1.0" priority="10">
...
  <Channel name="Bookmark" provider="BookmarkProvider" merge="fuse">
    <Properties>
      <String name="title" value="My Bookmarks" merge="replace" lock="false"
propagate="true"/>
      <String name="refreshTime" value="600" merge="replace" lock="false"
propagate="true"/>
        <Collection name="targets" merge="fuse" lock="false"
propagate="true">
          <String value="ACME home page|http://www.acme.com" merge="replace"
lock="false" propagate="true"/>
        </Collection>
    </Properties>
  </Channel>
...
</DisplayProfile>
```

```
dp @ uid=bill,ou=people,o=acme.com
<DisplayProfile version="1.0" priority="1">
...
  <Channel name="Bookmark" provider="BookmarkProvider" merge="fuse">
    <Properties>
      <String name="title" value="Bill's Bookmarks" merge="replace"
lock="false" propagate="true"/>
        <Collection name="targets" merge="fuse" lock="false" propagate="true">
          <String value="Amazon|http://www.amazon.com" merge="replace"
lock="false" propagate="true"/>
          <String value="EBay|http://www.ebay.com" merge="replace"
lock="false" propagate="true"/>
        </Collection>
    </Properties>
  </Channel>
...
</DisplayProfile>
```

Document Priority Example 2

This example uses three display profiles, the global display profile, the display profile for the organization `acme`, and the display profile for the role `hradmin`. When the user who is assigned to the `hradmin` role logs in to the Desktop, the `TemplateTableContainer` appears with the following channels:

- `UserInfo`
- `MailCheck`
- `SampleSimpleWebService`

```
dp @ global:
<DisplayProfile version="1.0" priority="0">
...
  <Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties>
      ...
    </Properties>
    <Available>
      ...
    </Available>
    <Selected merge="fuse" lock="false">
      <Reference value="UserInfo"/>
    </Selected>
    <Channels/>
  </Container>
...
</DisplayProfile>
```

```
dp @ o=acme.com:
<DisplayProfile version="1.0" priority="10">
...
  <Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties>
      ...
    </Properties>
    <Available>
      ...
    </Available>
    <Selected merge="replace" lock="false">
      <Reference value="Bookmark"/>
      <Reference value="Notes"/>
    </Selected>
    <Channels/>
  </Container>
...
</DisplayProfile>
```

```
dp @ cn=hradmin,o=acme.com:
<DisplayProfile version="1.0" priority="5">
...
  <Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties>
      ...
    </Properties>
    <Available>
    <Selected merge="fuse" lock="true">
      <Reference value="MailCheck"/>
      <Reference value="SampleSimpleWebService"/>
    </Selected>
    <Channels/>
  </Container>
...
</DisplayProfile>
```

Display Profile Document Priority Summary

A display profile document has a low or high priority depending on whether you consider the merge order or the ability to lock as the defining factor.

Without considering locking, the lower numbered display profile document has a lower priority. The lower numbered display profile document gets merged first so the value of a higher priority document overrides the value of a lower priority document. In this sense, the lower numbered document has a lower priority.

However, the lower numbered display profile document can also lock an object so it cannot be affected by a higher numbered document. In this sense, the lower numbered document has a higher priority.

Display Profile Merge Semantics

The display profile is composed of a hierarchy of XML documents. Sun ONE Portal Server could store a display profile document for the user, each role the user belongs to, and the user's organization or suborganization. At runtime, the system merges these multiple display profile documents to deliver a particular portal desktop to the user. This process of merging display profile documents affects the final display profile by potentially changing channel, provider, and property definitions.

The display profile data format contains syntax that defines how these documents are combined. This definition is commonly known as *merge semantics*.

Merge semantics control how attributes are combined as display profile documents from different LDAP nodes (base DN, DN, and role DNs) are merged to form a single representation (that is, Desktop). Merge semantics assume an ordering to display profile documents. Sun ONE Identity Server does not provide hierarchical structure of roles. Instead, the users' role structure is flat. All roles are peers. Because of this, Sun ONE Portal Server imposes an additional ordering on Sun ONE Identity Server roles to simulate a hierarchical structure.

The set of display profile documents for a user consists of: the documents that exist at the user's LDAP organization an suborganization nodes; the documents that exist at each of the user's role nodes; and the document that exists at the user's entry node. Documents do not need to be defined at each of these nodes, but there must be at least one document defined at a node. The set of documents is sorted according to a priority value that the display profile document defines. See Display Profile Document Priorities for more information.

You can visualize the process of document merging as laying one display profile document on top of another. A merge happens where like named channels, providers, and properties fall on top of one another. Merging is based on the name of the display profile object, not the XML structure defined in the display profile document. Like named channels can exist in different containers within the containment hierarchy in the display profile to be merged.

How the Merge Process Works

When a user logs on to Sun ONE Portal Server, and after authentication takes place, the system determines the user's display profile by:

1. Locating all the display profile documents for that user by searching through the global display profile, and LDAP organization, suborganization, role, and user nodes that the user belongs to.
2. Placing the retrieved display profile documents in a temporary area, which you can visualize as a bag.
3. Sorting the display profile documents in the bag based on priority, starting at the lowest priority. (The node at which the document was retrieved does not influence the priority sorting. Also, the user display profile document always has the highest priority.)
4. Taking the documents out of the bag, lowest priority first, then placing the next higher level priority document over this document, and applying merge and lock semantics.
5. Continuing Step 4 until all the documents have been taken out of the bag so that the system returns a value to the user that is a merge of the objects found in the documents.

Display Profile Merge Types

The display profile uses the following three types of merges to determine how to combine display profile documents:

- `replace`—All the display profile objects defined in the higher priority document completely override the ones defined at the lower one. If the object does not exist in the lower priority document, it is added to the merge result (the object replaces the value in the merge results).
- `remove`—The named object is removed from the merge up to this point (the object is removed from the merge results). It no longer exists in the display profile (but it can be re-introduced by another document to be merged). It can be redefined by a higher priority document.
- `fuse`—The object from the lower priority document is combined with one from the higher priority document (the object is merged with the value in the merge results).

NOTE The exact meaning of each merge type depends on the display profile object they are applied to.

For channels and providers, `fuse` has special meaning. The channels themselves are not actually fused together. Rather, `fuse` indicates that the channel's or provider's properties should be combined. The `replace` semantic replaces the entire channel or provider, including all properties. The `remove` semantic removes the entire channel or provider from the merge up to that point.

The display profile `<DisplayProfile>` root node can also have merge semantics. The `replace` semantic means that all the DP objects defined in the higher priority document completely override the ones defined at the lower one. All merges up to that point are negated and the higher priority document is used as the new base for merging. The `remove` semantic indicates that all merge results up to the point of this document are to be discarded. The merge begins with the next display profile document found in the sorted set. As with channels and providers, the `fuse` semantic means that the contained objects (channels and providers) should be combined.

Atomic display profile properties (those that cannot contain other properties) cannot use the `fuse` semantic. This includes the String, Integer, Boolean, and Reference properties.

The set of properties for a channel consists of the channel's properties plus the channel's provider's properties plus the channel's parent's properties, and so on. You can think of this total set of properties as the channel's single document properties. An implication of document merging is that the total set of properties for a document consists of the set union of the channel's single document properties for all documents in the user's merge set.

Remove Example: Using remove Merge to Modify Container's Selected Channel List

The following example shows how all users' merge set can consist of an organizational level document that has the following display profile fragment.

```
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties> ... </Properties>

    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="UnixTipoftheDay"/>
    </Selected>
</Container>
```

The “unix tip of the day” describes ways to use UNIX. It is likely that users that belong to the admin role would not find this channel helpful. To remove this channel from everyone with the admin role, define the `TemplateTableContainer` channel in the admin role document as follows:

```
admin role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties> ... </Properties>

    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="Outages"/>
        <Reference value="SolarisAdmin"/>
        <Reference value="AdminTipoftheDay"/>
        <Reference value="UnixTipoftheDay" merge="remove"/>
    </Selected>
</Container>
```

The preceding sample snippet causes the `Reference value="UnixTipoftheDay"` to be removed from the admin role display profile.

Replace Example: Using replace Merge to Remove Channel from All Users' Display

The following example shows how for a particular container, a role admin can ignore all of the channels defined in the organization level. The organization definition resembles the following:

```
organization display profile
<Container name=...>
    ...
    ...
    <Selected>
        <Reference name="X"/>
        <Reference name="Y"/>
        <Reference name="Z"/>
    </Selected>
</Container>
```

Because the role admin does not want any of the users under that role to have the X, Y, or Z channels, the container is defined as follows:

```
admin role
<Container name=...>
    ...
    ...
    <Selected merge="replace">
        <Reference name="A"/>
        <Reference name="B"/>
        <Reference name="C"/>
    </Selected>
</Container>
```

The selected list in the role document's container replaces the selected list in the organization document's container.

Fuse Example: Using fuse Merge to Create Role-based Channel List

You commonly use the `fuse` merge semantic to combine non-atomic display profile objects. These objects include `Collection` and the available or selected channel lists. Here, `fuse` indicates that all the properties contained in the non-atomic property should also be merged. Using `fuse` in this way enables the final non-atomic property presented to the user to be build up from various documents.

The following example display profile documents are for a user who belongs to the admin, employee, and movieFreak roles. The selected channels for the user appear at the end.

```
admin role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties> ... </Properties>

    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="Outages"/>
        <Reference value="SolarisAdmin"/>
        <Reference value="AdminTipoftheDay"/>
    </Selected>
</Container>
```

```
employee role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties> ... </Properties>

    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="Benefits"/>
        <Reference value="EmployeeNews"/>
    </Selected>
</Container>
```

```
movieFreak role
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties> ... </Properties>

    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="NewMoviesReleases"/>
        <Reference value="MovieShowTimes"/>
    </Selected>
</Container>
```

The resultant list of selected channels for the user is as follows, with the available channel list ordered in the same way that the merging was applied, from lower to higher priority:

```
<Container name="TemplateTableContainer"
provider="TemplateTableContainerProvider" merge="fuse">
    <Properties> ... </Properties>

    <Available> ... </Available>
    <Selected merge="fuse">
        <Reference value="Outages"/>
        <Reference value="SolarisAdmin"/>
        <Reference value="AdminTipoftheDay"/>
        <Reference value="Benefits"/>
        <Reference value="EmployeeNews"/>
        <Reference value="NewMoviesReleases"/>
        <Reference value="MovieShowTimes"/>
    </Selected>
</Container>
```

Merge Locking

Any display profile object that is able to be merged can also be locked. When an object is locked, it cannot be affected by merge semantics in higher priority documents. This enables low-priority documents to prevent a high-priority document from using the merge semantics to change particular aspects of the display profile.

Merge Locking Example: Using lock Merge to Force Property Value for All Users

The following example shows how to ensure that for a particular organization, all users see the “employee news” channel. The users cannot remove this channel from their display. At the organization level document, the container channel’s selected list is defined as follows:

```
<Selected merge="fuse">
    ...
    <Reference value="EmployeeNews" lock="true"/>
    ...
</Selected>
```

Merge Locking Example: Using lock Merge to Force-remove Channel from All Users’ Display

The following example shows how to force the “online games” channel to be removed. In this scenario, users have added this channel to the selected channels list in their user document, so simply removing it from the organization level document’s selected channel’s list will not work. Instead, the employee and organization lists will be merged together resulting in the “online games” channel being present. To forcibly remove the channel from all users under the organization, the selected channels list is defined as follows:

```
<Selected merge="fuse">
    ...
    <Reference value="OnlineGames" merge="remove" lock="true"/>
    ...
</Selected>
```

The `remove` semantic removes the channel from merged result, and `lock` prevents lower priority documents from merging the value back in.

Display Profile and Sun ONE Identity Server

The set of display profile documents for a user can consist of:

- The document that exists at the user's LDAP organization (or suborganization) node
- The documents that exist at each of the user's role nodes
- The document that exists at the user's entry node
- The document that exists at the global display profile

Documents do not need to be defined at each of these nodes, but there must be at least one document defined at a node. The set of documents is sorted according to a priority value that the display profile document defines. See Display Profile Document Priorities for more information. Merge semantics control how attributes are combined as display profile documents from different nodes are merged to form a single representation or Desktop. See Display Profile Merge Semantics for more information.

Administrators can edit the display profile using the Sun ONE Identity Server administration console. You can set up delegated administrators so that they do not see the display profile in the Sun ONE Identity Server administration console. You do this when you create the Desktop service template. When you create the template for the Desktop service, if you unselect the "Show Desktop Service Attributes" box, you can hide the display profile text from a delegated administrator.

TIP The organization administrator can define a container (or container hierarchy) associated with certain roles through the Portal Desktop service. Then, the delegated administrator (role administrator) can define the necessary channels and containers under this container through the Channel and Container Management link in the Portal Desktop attributes page. See Using the Channel and Container Management Link to Administer Channels for more information.

Administering the Display Profile

You edit the display profile (and other Portal Desktop service data) through the Sun ONE Identity Server administration console and `dpadmin` command. When you edit the display profile, you add, modify, and remove providers, containers, and channels from the display profile, and edit properties.

In addition, the Sun ONE Identity Server administration console provides the Channel and Container Management link in the Portal Desktop attributes page to add channels and edit properties. This link also enables you to modify properties when a new channel is created.

NOTE The Channel and Container Management link is suited for delegated administration and allows the administrator to add and modify attributes of containers and channels. The overall system administrator should be responsible for adding the container and providers available to the delegated administrator.

Table 5-2 on page 161 explains the different types of display profiles and how to use the Sun ONE Identity Server administration console to administer them. This three column table lists the types of display profiles in the first column, how to access that display profile using the Sun ONE Identity Server administration console, and a brief description in the third column.

Table 5-2 Types of Display Profile Documents

Type of Display Profile Document	How to View Using the Sun ONE Identity Server Administration Console	Description
Global Display Profile Document	Choose View Service Management. Click the properties arrow next to Portal Desktop. In the Desktop Global attributes section, click Edit XML.	Defines display profile elements that are inherited by all users on the system, regardless of the organization or role to which they belong. (Although currently not enforced, you might also want to use the display profile XML document to define the common providers that will be used by everyone.)
Dynamic Display Profile Document	Choose View Service Management. Click the properties arrow next to Portal Desktop. In the Desktop Dynamic attributes section, click Edit XML.	Describes container management and properties for channels. This display profile is not ‘used’ to generate a user’s Desktop at runtime, but becomes the default for each newly created organization and role. By default, the dynamic display profile document is blank. To use the dynamic display profile, you need to first populate it.

Table 5-2 Types of Display Profile Documents

Type of Display Profile Document	How to View Using the Sun ONE Identity Server Administration Console	Description
Organization, Suborganization, or Role Display Profile	Choose View User Management. Select the appropriate organization, suborganization, and if necessary, select Roles from the Show menu. Select Services from the Show menu. Click the properties arrow next to Portal Desktop. In the Desktop page, click Edit XML.	Shows the display profile for the selected organization, suborganization, or role. When you create a new organization, suborganization, or role, you create a template for this entity. When you create the template for the Desktop service, the initial display profile is set to the dynamic display profile document as mentioned above. Thus, if the dynamic display profile is blank, nothing is filled in. Most likely, you use this display profile document to customize container management and channel properties to fit the needs of different organizations and roles.

When you install Sun ONE Portal Server, you create an initial organization. The installer then imports the display profile global level document, and the default display profile, based on what you specify.

After that, each time you create a new organization, suborganization, or role, the display profile is not automatically loaded. You must manually load the display profile for a newly created organization, suborganization, or role. See To Load the Display Profile (Administration Console) for more information.

The high-level steps to administer the display profile are:

1. Loading the display profile for any newly created organization, suborganization, or role. (You do not need to perform this step for the organization that is created during the installation process.)
2. Modifying the display profile using the `dpadmin` command, the Edit XML link, or as a file that has been saved and then loaded using the Download XML and Upload XML links.
3. Adding channels and containers, and adding, deleting, and modifying their properties using the Channel and Container Management link.

Default Display Profile Documents

Table 5-3 explains the display profile documents that the Sun ONE Portal Server Desktop supplies in the `/opt/SUNWps/samples/portal desktop` directory at the time the sample portal is installed. This two column table lists the display profile documents in the first column and a brief description in the second column.

Table 5-3 Display Profile Documents Supplied with Sample Portal

Display Profile Document	Description
`dp-anon.xml`	Used by the authless anonymous user.
dp-org.xml	Sample display profile loaded at the default organization level. It defines all the global properties that are used for the organization and the channel definitions that are used by the organization.
`dp-org-final.xml`	A copy of `dp-org.xml`, with NetMail links defined in the Bookmark and Applications channels. This display profile document is used when the NetMail service is created.
`dp-providers.xml`	Sample display profile loaded at the global display profile level. This document defines all the provider definitions. Because these providers are going to be used by all organizations, the system loads this display profile at the top level, and every organization is able to use them. If a provider definition is used only by one organization, define it in the organization level display profile.

See the *Sun ONE Portal Server 6.1 Desktop Customization Guide* for information on customizing these sample display profiles.

Loading the Display Profile

When you first install Sun ONE Portal Server, the installer create an initial organization. The installer also imports the display profile global level document, and the default display profile, based on what you specify. If you decide not to install the sample portals, the sample display profile documents are not installed.

After that, when you create a new organization, suborganization or role, the display profile is not automatically loaded. You must manually load the display profile for a newly created organization, suborganization, or role.

There are three basic methods for loading the display profile:

- Using the Edit XML link of the Sun ONE Identity Server administration console. With this method you use the Edit XML link and an existing display profile in an organization, which you copy then paste into the blank display profile of the newly created organization, suborganization, or role. See To Load the Display Profile (Administration Console).
- Using the command line. With this method you use the `dpadmin` command to load the display profile. See To Load the Display Profile (Command Line). Before using the `dpadmin` command, see Guidelines for Using the dpadmin Command.
- Using the Download and Upload links of the Sun ONE Identity Server administration console. With this method you download a display profile to a file and then upload a display profile from a file. See To Download and Upload a Display Profile.

NOTE You cannot edit the display profile XML directly through the administration console if your browser is Netscape 4.x.

To Load the Display Profile (Administration Console)

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Select the organization or suborganization from which you want to copy the display profile.
3. Choose Services from the View menu.
4. Click the properties arrow next to Desktop in the navigation pane.

   The Portal Desktop attributes appear in the data pane.

TIP You might have to scroll down to see the Desktop service.

5. Copy the Display Profile.

   Click Edit XML then select and copy the entire text of the display profile.
6. Select the organization, suborganization, or role for which you want to load the display profile.
7. Choose Services from the View menu in the navigation pane.

8. Click the properties arrow next to Desktop in the navigation pane.

   A list of Portal Desktop service attributes, including the display profile XML, is displayed in the data pane.

9. Click Edit XML.

   The display profile XML document appears in a text window.

10. Paste the copied display profile into the display profile window.

11. When done, click Save.

    The changes affect only users in this particular organization.

To Load the Display Profile (Command Line)

Use the `modify` subcommand of the `dpadmin` command to load a display profile.

For example, the following command loads the display profile (`dp-org.xml`):

```
dpadmin add -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d
"o=sesta.com,o=isp" dp-org.xml
```

NOTE You can add the `-r` or `--dryrun` option to the end of the command before the file name to verify that the command will be successful before actually writing any changes to LDAP.

To Download and Upload a Display Profile

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Select the organization or suborganization from which you want to copy the display profile.

3. Choose Services from the View menu in tyhe navigation pane.

4. Click the properties arrow next to Desktop in the navigation pane.

   The Portal Desktop attributes appear in the data pane.

TIP You might have to scroll down to see the Desktop service.

5. Click Download XML in the Global attributes section and save the display profile to a file.
6. Select the organization, suborganization, or role for which you want to upload the display profile in the navigation pane.
7. Choose Services from the View menu in the navigation pane.
8. Click the properties arrow next to Desktop in the navigation pane.
9. Click Upload XML and specify the file to load.
10. Click Upload.

    A message indicating that the display profile upload was a success appears.
11. Click Close.

    The changes affect only users in this particular organization.

To View the Entire Display Profile

- Run the `dpadmin` command with the `list` subcommand to view the entire display profile, for example:

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

To Remove a Display Profile

If you need to remove a display profile for some reason, for example if it is corrupted, you can use the `dpadmin` command with the `remove` subcommand.

For example, to remove the entire display profile (`dp-org.xml`) from the root:

```
dpadmin remove -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d
"o=sesta.com,o=isp" -t root
```

If you remove a display profile from the root or from a node at which you require a display profile, you must load a new one. For example, if you removed the `dp-org.xml` display profile as shown above, you will have to load another similar display profile such as the `dp-org-final.xml` display profile. See To Load the Display Profile (Command Line) for information on loading a display profile.

Using the Channel and Container Management Link to Administer Channels

You use the Channel and Container Management link to administer:

- Properties—You can define and add global display profile properties.
- Containers—You can add or remove a content container to or from a container. You can also modify a content container's properties.
- Channels—You can add or remove a channel to or from a container
- You can also modify a channel's properties.

NOTE Currently, you can work with channels and containers and their properties using the Channel and Container Management link. This link does not work with providers.

When using the Desktop attributes page, delegated administrators see only the Channel and Container Management link. All other display profile attributes are hidden, and thus made secure.

Channel and Container Management Default Providers

The Portal Desktop Channel and Container Management link displays a management screen that allows you add or remove container channels or content channels.

Add Channels

The Add link for the Channels list allows you to select a content provider to add from a list of defined content providers. Table 5-4 on page 168 shows the provider channels that are available to use as a basis to create new channels. This two-column table lists the providers in the first column and a brief description of the provider in the second column. For more information on defined content providers, see the *Sun ONE Portal Server Desktop Customization Guide*.

Table 5-4 Defined Provider Channels

Provider	Description
`AppProvider`	Lists links to web applications (users can customizedlist).
`BookmarkProvider`	Allows users to manage a list of bookmarks displayed on a portal page.
`JSPProvider`	Obtains content from one or more JSP™ files.
LoginProvider	Allows users to authenticate to a Sun ONE Identity Server from an anonumous portal page.
`MailCheckProvider`	Gives information about a user's mail status.
`NotesProvider`	ELists system-wide messages and allows users to post such messages
SearchProvider	Supplies a search function using the Sun ONE Portal Server Search Engine.
`URLScraperProvider`	Obtains content from a given URL and uses the Sun ONE Portal Server to format the content.
`UserInfoProvider`	Collects information from the display profile and iPlanet Portal Server Access Management Edition. It displays a greeting, the user's name, timezone, locale and has access to the user's IMAP and SMTP data.
`XMLProvider`	Obtains XML content from a given URL and uses XSLT to translate the content to markup language..

Simple Web Services Provider

The Simple Web Services (SWS) Provider provides the ability to access data-oriented Web Services. Based on this provider, a sample channel demonstrates Web Services' implementation by accessing a currency conversion rate service.

There are two types of simple web service channels:

- Pre-Configured Web Service Channel
- New Container Channels

The sample pre-configured web service channel is available on the sample portal desktop by default. The sample configurable web service channel can be added by the administrator using the Identity Server admin console.

Either web service channels are best suited for use with relatively simple web services; for example, web services that have non-complex input parameters and user interface presentation requirements. If the Simple Web Service Provider detects that it is not equipped to handle a particular web service, it will display a suitable message to the user.

At any given time, a channel based on this provider can be bound to a single web service and associated method. The Simple Web Service Provider will support simple data types, such as integer, string, double. In this release, the simple web service provider:

- Will also support arrays of simple and complex types in the input and output parameters.
- Will not support the use of fault data in the binding operations in the WSDL Definition.

The Simple Web Service Provider will support the following WSDL configuration property types:

- SOAP Binding Style: rpc & document
- SOAP Encoding Type: encoded & literal

NOTE The rpc/literal combination is not supported. Support for `.Net` based web services might be limited.

Pre-Configured Web Service Channel

The sample pre-configured web service channel provides the means to interace with sample currency converter service.

To set up a pre-configured web service channel, you will be required to specify the WSDL URL and method name via the administration console.

Configurable Web Service Channel

The configurable web service channel allows the user to switch the channel to point to a user specified web service. This is achieved by giving the user the ability to modify values for the WSDL URL and the method name belonging to the web service. However, unlike the pre-configured channel type, the configurable web service channel will not allow the user any facility to store default values for the web service input parameters.

New Container Channels

The New link for the Container Channels list allows you to select a container provider to create from a list of defined container providers. Table 5-5 shows the shows them defined provider channels that are available to use as a basis to create new channels. This two-column table lists the providers in the first column and a brief description of the provider in the second column. For more information on defined content providers, see the *Sun ONE Portal Server 6.1 Desktop Customization Guide*

Table 5-5 Defined Provider Container Channels

Provider	Description
`JSPFrameCustomTableContainerProvider`	Create a new frame on a user's JSP frameset-based Portal Desktop.
`JSPSingleContainerProvider`	Displays a single channel.
`JSPTabContainerProvider`	Displays a channel that is made up of a number of tabs with titles on them.
`JSPTabCustomTableContainerProvider`	Creates a new tab on a user's JSP tab-based Portal Desktop.
`JSPTableContainerProvider`	Displays the content channels in a table.
`TemplateEditContainer`Provider	Draws the frame for the Edit page.
TemplateTabContainerProvider	Supports multiple tabs.
`TemplateTabCustomTableContainerProvider`	Creates a new tab.
TemplateTableContainerProvider	Displays content channels in a table.

To Create a Channel or Container Channel

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Select the organization, suborganization, or role to which you want to add a channel.

   When you log in as a delegated administrator, you are automatically taken to the organization, suborganization, or role to which you have administrative access.

3. Choose Services from the View menu in the navigation pane.

4. Click the properties arrow next to Portal Desktop in the navigation pane.

   The Desktop attributes page appears in the data pane.

5. In the Desktop page, click the Channel and Container Management link.

   The Channels page appears, with the container path set at the root.

6. Click the Container that you want to add the channel or container to.

   The top of the page displays the container path where the channel will be added. Defined channels and container, if any, appear in lists.

7. Click New to add a container channel or channel.

   To add a container channel, click New under Container Channel. To add a channel, click New under Channel.

   The New Channel page appears.

8. Type a channel name and select the type of provider from the menu.

   See Table 5-4 on page 168 for the available providers.

9. Click Create.

To Modify a Channel or Container Channel Property

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Select the organization, suborganization, or role in which you want to modify a channel.

   When you log in as a delegated administrator, you are automatically taken to the organization, suborganization, or role to which you have administrative access.

3. Choose Services from the View menu.

4. Click the properties arrow next to Portal Desktop in the navigation pane.

   The Desktop attributes page appears in the data pane.

5. In the Desktop page, click the Channel and Container Management link.

   The Channels page appears. At the top is the container path. The defined channels appear in a list.

6. Click the Edit Properties link beside the channel or container channel to be modified.

   The Properties page appears.

7. Modify the properties as needed.

   See the *Sun ONE Portal Server 6.1 Desktop Customization Guide* for more information on channel properties.

8. When done, click Save.

To Remove a Channel or Container Channel

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Select the organization, suborganization, or role in which you want to modify a channel.

   When you log in as a delegated administrator, you are automatically taken to the organization, suborganization, or role to which you have administrative access.

3. Choose Services from the View menu in the navigation pane.

4. Click the properties arrow next to Portal Desktop in the navigation pane.

   The Desktop attributes page appears in the data pane.

5. In the Desktop page, click the Channel and Container Management link.

   The Channels page appears. At the top is the container path. The defined channels appear in a list.

6. Click the checkbox beside the channel or container channel to be removed. Then click Delete.

7. The channel is deleted and the Channels list is updated to show its removal.

Administering Containers

When administering containers, you can use the Sun ONE Identity Server administration console to directly edit the display profile XML. You can also use the `dpadmin` command, which for the most part this section describes by using various examples.

These examples include:

- To View a Display Profile Object
- To Replace a Channel in a Container
- To Replace a Property in a Channel
- To Add a Channel to a Container
- To Add a Property to a Collection
- To Add a Collection Property
- To Remove a Property from a Channel or Container
- To Remove a Provider
- To Remove a Channel from a Container

- To Change a Display Profile Document Priority
- To Make a Channel Available for a Container
- To Make a Channel Unavailable for a Container
- To Select a Channel from a Container's Available Channel List
- To Unselect a Channel from a Containers Available Channel List

See Using the Display Profile Text Window for information on editing the display profile through the Sun ONE Identity Server administration console.

Using the dpadmin Command

The syntax of the `dpadmin` command is:

```
$ dpadmin list|merge|modify|add|remove [command-specific options] -u uid -w password
{-g|-d dn} [-l locale] [-r] [-b] [-h] {-v|--version} [file]
```

See Chapter 14, "Command-Line Utilities" for the complete syntax of the `dpadmin` command. When running the `dpadmin` command, note the following:

- `file` argument—If present, the `file` argument must also be the last argument on the command line. It specifies a path to an XML file that contains an XML fragment that conforms to the display profile DTD. Subcommands that require XML input include `modify` and `add`.

  When adding or modifying an entire display profile, always include a proper XML header, for example

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
```

- `list` subcommand—Retrieves and displays display profile node objects. Objects are displayed in their native XML format. The object to be displayed is sent to standard out. If you do not use the `-n` or `--name` option, the entire display profile document is displayed. If you do not use the `-n` or `--name` option does not specify a DP node object, then the entire DP document is displayed.
- `merge` subcommand—retrieves and displays the merged result of the specified DP node objects. Objects are displayed in their native XML format. The object to be displayed is sent to standard out. If you do not use the `-n` or `--name` option, then an error is reported.
- `modify` subcommand— Changes the value of an existing display profile object. This command assumes that the object already exists in the display profile. The `modify` subcommand reads data for the new object from either standard input or the file specified as an argument. Data for the new object must be XML and conform to the display profile DTD. Specifically, the object data must be a well-formed XML fragment.
- `add` subcommand—Adds a new object to the display profile. This subcommand assumes that the object to be added does not exist in the display profile. The `add` subcommand reads data for the new object from either standard input or the file specified as an argument. Data for the new object must be XML and conform to the display profile DTD. Specifically, the object data must be a well-formed XML fragment.
- `remove` subcommand—Removes an existing object from the display profile.
- `-g` option—Specifies the global level display profile document.
- `-d` *dn* option—Designates the DN where `dpadmin` will execute. The `-d` and `-g` options are mutually exclusive.
- `-r` or `--dryrun` option—Reports whether the current command will succeed or fail, and does not write any changes into LDAP. This is useful to ensure that a particular `dpadmin` command is correctly formatted.
- `-n` or `--name` option—Specifies a fully qualified name of the display profile container, channel or provider object, or of the parent of the display profile object. If the name agrument does not specify a DP node object, then an error is reported.
- `-p` or `--parent` options—Specify a fully qualified name of the parent display profile container, channel or provider object, or of the parent of the display profile object.

- `-v` or `--version` options—Print the version number of the `dpadmin` command to standard output.

Guidelines for Using the dpadmin Command

Use the following guidelines when running the `dpadmin` command to update the display profile:

- Make sure no other administrator is currently using the Sun ONE Identity Server administration console or `dpadmin` command to make display profile modifications. Such a situation could cause changes to be lost, as there is no locking mechanism to prevent `dpadmin` and the administration console from accessing the display profile at the same time.
- The preferred sequence when using `dpadmin` is to put your modifications into a file as an XML "fragment" then run the `dpadmin` command with the `add` subcommand. For example,

  `/opt/SUNWps/bin/dpadmin add -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w` *password* **`-d "uid=anonymous,ou=people,o=sesta.com,o=isp" newtheme.xml`**

  In this example, `newtheme.xml` is a file containing the XML "fragment" to be added to the display profile.
- If you edit a display profile document directly, first use the `dpadmin` command with the `list` subcommand to obtain the latest contents of the display profile, make your edits, then run the `dpadmin` command with the `modify` subcommand. For example,

  `/opt/SUNWps/bin/dpadmin list -u"uid=amAdmin,ou=People,o=sesta.com,o=isp" -w` *password* **`-d "o=sesta.com,o=isp" > dp-org.xml`**

  (Edit the `dp-org.xml` file.)

  `/opt/SUNWps/bin/dpadmin modify -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w` *password* **`-d "o=sesta.com,o=isp" dp-org.xml`**

CAUTION Between the time you run the `dpadmin list` and `dpadmin modify` commands, do not change the display profile document in the LDAP server in any way (by using the administration console, `dpadmin`, or `ldapmodify` commands). Otherwise, those changes will be overwritten by the latest `dpadmin modify`.

Modifying the Display Profile

You can modify display profile objects by performing one of the following:

- Manually editing an existing display profile document then loading it at the appropriate LDAP node or global level by using the `dpadmin modify` command.
- Running the `dpadmin` command with the specified changes, in XML text, on standard input. When adding a new object you use the `add` subcommand. When modifying an existing object, you use the `modify` subcommand.
- Creating a new display profile document from scratch then loading it at the appropriate LDAP node or global level by using the `dpadmin modify` command.

Understanding Display Profile Error Messages

The system reports errors when you try and save a display profile document containing invalid XML. The error messages appear as a title, a message, and a sub-message. The title of the message box is “Invalid XML document.” The message appears as one of the following:

- Failed to parse XML...
- Missing doctype in the XML
- Failed to sore DP...
- Invalid XML input...

If you receive an “Invalid XML document” error, you need to correct the error to be able to save the XML document you are working on.

To View a Display Profile Object

- Use the `list` subcommand to view a display profile object.

  For example, the following command gets the channel, container, or provider named `TemplateTableContainer` and prints it to standard output.

```
dpadmin list -n "TemplateTableContainer" -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
```

NOTE You can view the entire display profile document by omitting the `-n` option.

To Replace a Channel in a Container

1. Use the `modify` subcommand to replace a channel in a container with a value specified on standard input.

   For example, this command replaces the channel `Test` in the container `TemplateTableContainer` with value specified on standard input.

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Channel name="Test" provider="testprovider">
    <Properties>
        <String name="title" value="Test Channel"/>
        <String name="description" value="This channel is a test."/>
    </Properties>
</Channel>
EOF
```

2. Use the `list` subcommand to verify that the channel was replaced.

   See To View a Display Profile Object for information.

To Replace a Property in a Channel

1. Use the `modify` subcommand to replace a property in a channel with a value specified on standard input.

   For example, the following command acts upon the channel `NewChannel` to replace the property named in the `new.xml` with the new object in said file, where `new.xml` is:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<String name="welcome" value="Hi, welcome to your desktop!"/>
```

```
dpadmin modify -p TemplateTableContainer/NewChannel -u
"uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
new.xml
```

2. Use the `list` subcommand to verify that the property was replaced.

   See To View a Display Profile Object for more information.

To Add a Channel to a Container

1. Modify your display profile input XML file to include only the new `<Channel>` definition, for example, create the following file `testadd.xml`:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Channel name="TestChannel" provider="testprovider">
    <Properties>
      <String name="teststring" value="sfds"/>
    </Properties>
</Channel>
```

2. Use the add subcommand to add the channel to a container.

   For example, the following command adds a new channel defined in `testadd.xml` to the display profile. In this example, the new channel must be added in the `TemplateTableContainer` level. If you do not specify a parent object with the `-p` option, the channel is added at the root level.:

```
dpadmin add -p "TemplateTableContainer" -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" testadd.xml
```

NOTE When you add a new channel to `JSPTabContainer`, you actually add a new tab. `JSPTabContainer` requires `TabProperties` defined for all its available and selected tabs. Thus, for any new container or channel added to the `JSTTabContainer`, add the following XML snippet inside the `TabProperties Collection` in the `JSPTabContainer` for which the new channel or container is added.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Collection name="<New Channel Name>">
            <String name="title" value="<New Channel Title>"/>
            <String name="desc" value="<New Channel Description>"/>
            <Boolean name="removable" value="false"/>
            <Boolean name="renamable" value="true"/>
            <Boolean name="predefined" value="true"/>
</Collection>
```

3. Use the `list` subcommand to verify that the channel was added.

   See To View a Display Profile Object for information.

To Add a Property to a Collection

1. Use the `combine` (`-m`) option to add a new property to a collection.

   For example, the following command adds a new property `msg2` to the collection `bar`. The existing property, `msg`, still remains in the result. The `list` subcommand is used before and after to show the property values.

```
dpadmin list -n TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp
...
<Collection name="foo">
    <Collection name="bar">
        <String name="msg" value="hi"/>
    </Collection>
</Collection>
...
```

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" -m <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Collection name="foo">
    <Collection name="bar">
        <String name="msg2" value="woo hoo"/>
    </Collection>
</Collection>
EOF
```

```
dpadmin list -n TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp”
...
<Collection name="foo">
    <Collection name="bar">
        <String name="msg" value="hi"/>
        <String name="msg2" value="woo hoo"/>
    </Collection>
</Collection>
...
```

To Add a Collection Property

1. Use the add subcommand to add a collection with a value specified on standard input.

   For example, the following command adds the collection property `zipCodes` specified on standard input to the channel, container, or provider named `Postal`.

```
dpadmin add -p SamplesTabPanelContainer/Postal -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Collection name="zipCodes">
    <Integer value="98012"/>
    <Integer value="98036"/>
    <Integer value="94025"/>
    <Integer value="95112"/>
</Collection>
EOF
```

2. Use the `list` subcommand to verify that the collection property was added.

   See To View a Display Profile Object for information.

To Remove a Property from a Channel or Container

1. Use the `remove` subcommand to remove a property from a channel or container.

   For example, the following command removes the property `locations` from the `Bookmarks` channel (or container) at the global level.

```
dpadmin remove -t property -p Bookmarks -n locations -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -g
```

2. Use the `list` subcommand to verify that the property was removed.

   See To View a Display Profile Object for information.

To Remove a Provider

1. Use the `remove` subcommand to remove a provider.

   For example, the following command removes the provider `NotesProvider`.

```
dpadmin remove -t provider -n "NotesProvider" -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp"
```

2. Use the `list` subcommand to verify that the provider was removed.

   See To View a Display Profile Object for information.

To Remove a Channel from a Container

1. Use the `remove` subcommand to remove a channel from a container.

   For example, the following command removes the channel `Test` that exists in the parent container `TemplateTableContainer`.

```
dpadmin remove --type channel --parent TemplateTableContainer --name "Test"
--runasdn "uid=amAdmin,ou=People,o=sesta.com,o=isp" --password password --dn
"o=sesta.com,o=isp"
```

2. Use the `list` subcommand to verify that the channel was removed.

   See Chapter 5, “Administering the Display Profile” for information.

To Change a Display Profile Document Priority

1. Use the `modify` subcommand to change the priority of a display profile document.

   For example, the following command changes the document priority from the original priority to 10 for the organization.

```
dpadmin modify -m -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w password -d
"o=sesta.com,o=isp" <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<DisplayProfile priority=”30” version=”1.0”
<Properties/>
<Channels/>
<Providers/>
EOF
```

2. Use the `list` subcommand to verify that the priority change was made.

   See To View a Display Profile Object for information.

To Make a Channel Available for a Container

1. Use the `modify` subcommand with the `combine` (`-m`) option to add a channel specified on standard input to a container’s existing Available list.

   For example, the following command adds the `BookMark` channel to the `Available` list of the `TemplateTableContainer`.

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" -m <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Available>
    <Reference value="BookMark">
</Available>
EOF
```

2. Use the `list` subcommand to verify that the priority change was made.

   See To View a Display Profile Object for information.

To Make a Channel Unavailable for a Container

1. Use the `remove` subcommand to remove a channel from a container's `Available` list.

   For example, the following command removes the channel `Test` from the `Available` list in the parent container `TemplateTableContainer`.

```
dpadmin remove --type available --parent TemplateTableContainer --name "Test"
--runasdn "uid=amAdmin,ou=People,o=sesta.com,o=isp" --password password --dn
"o=sesta.com,o=isp"
```

2. Use the `list` subcommand to verify that the channel was removed.

   See To View a Display Profile Object for information.

To Select a Channel from a Container's Available Channel List

1. Use the `modify` subcommand with the `combine` (`-m`) option to add a channel specified on standard input to a container's existing `Selected` list.

   For example, the following command adds the `BookMark` channel to the `Selected` list of the `TemplateTableContainer`.

```
dpadmin modify -p TemplateTableContainer -u "uid=amAdmin,ou=People,
o=sesta.com,o=isp" -w password -d "o=sesta.com,o=isp" -m <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<Selected>
    <Reference value="BookMark">
</Selected>
EOF
```

2. Use the `list` subcommand to verify that the priority change was made.

   See To View a Display Profile Object for information.

To Unselect a Channel from a Containers Available Channel List

1. Use the `remove` subcommand to remove a channel from a container's `Selected` list.

   For example, the following command removes the channel `Test` from the `Selected` list of the parent container `TemplateTableContainer`.

```
dpadmin remove --type selected --parent TemplateTableContainer --name "Test"
--runasdn "uid=amAdmin,ou=People,o=sesta.com,o=isp" --password password --dn
"o=sesta.com,o=isp"
```

2. Use the `list` subcommand to verify that the channel was removed.

   See To View a Display Profile Object for informaton.

Using the Display Profile Text Window

The Sun ONE Identity Server provides a text window for viewing and directly editing the display profile text. As long as you have administrative access to an organization, suborganization, or role, you can use the Sun ONE Identity Server administration console to navigate to this text window and view or edit the display profile.

NOTE You cannot edit the display profile XML directly through the administration console if your browser is Netscape 4.x.

To Access the Display Profile Text Window

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Select the organization, suborganization, or role for which you want to modify the display profile document.

   When you log in as a delegated administrator, you are automatically taken to the organization, suborganization, or role to which you have administrative access.

3. Choose Services from the View menu in the navigation pane.

4. Click on the properties arrow next to Portal Desktop in the navigation pane.

   The Desktop attributes page appears in the data pane.

5. In the Desktop page, click the Display Profile Edit XML link.

   The display profile appears in a text window.

NOTE By default, the display profile priority level is set to the keyword `user`, indicating that the current display profile is the user level display profile. Other allowable values are integers with lower numbers representing lower priorities. For example, a 1 is a lower priority than a 2.

6. Make your changes and click Save.

NOTE Changes to global, organization, suborganization, or role level documents are effectively immediately. Changes to user level documents are effectively after users log out and log in.

Chapter 6

Administering the NetMail Service

This chapter describes how to administer the NetMail service. This chapter contains these sections:

Overview of the NetMail Service

NetMail service implements the NetMail (Java™) and NetMail Lite email clients. These clients work with standard IMAP and SMTP servers. NetMail allows users to access one or more mail servers to read, compose and delete emails, and create, access and delete folders.

In Sun™ ONE Portal Server 6.2, you define and manage the NetMail service through the iPlanet™ Directory Server Access Management Edition administration console. The NetMail service defines the service attributes and default values for the NetMail client for managing email messages and its configuration. You define and customize service attribute values for an organization and its users to control how the NetMail client behaves.

Administering the NetMail Service

The Sun ONE Identity Server Policy Service enables you to define rules or access to resources. Policies can be role-based or organization-based and can offer privileges or define constraints.

By default, the Policy Configuration service is automatically registered to the top-level organization. Suborganizations must register their policy services independently of their parent organization. Any policy service you create must be registered to all organization. The high-level steps to use policies are:

1. Registering the Policy service for an organization.
2. Creating a referral policy for a suborganization. You can delegate an organization's policy definitions and decisions to another organization. (Alternately, policy decisions for a resource can be delegated to other policy products.) A referral policy controls this policy delegation for both policy creation and evaluation. It consists of a rule and the referral itself. If the policy service contains actions that do not require resources, referral policies cannot be created for suborganizations. See "To Create a Referral Policy for a Suborganization" on page 191" for information.
3. Creating a normal policy for a peer or suborganization. You create a normal policy to define access permissions. A normal policy can consist of multiple rules, subjects, and conditions. See "To Create a Normal Policy for a Suborganization" on page 192" for information.

To Register a Policy Service for a Peer or Suborganization

Suborganizations do not inherit their parent's services, so you need to register a suborganization's Policy service.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.
2. Navigate to the organization or suborganization that you want to create a referral policy.

   All created organizations are displayed in the navigation pane.
3. Select Organizations from the View menu in the navigation pane and select desired organization from the Name menu.
4. Select Services from the View menu.

5. Click Register.

   The Register Services page appears in the data pane. Click the check box for the NetMail service, then click Register.

   The newly registered service appear in the navigation pane.

6. Configure the NetMail service by clicking the properties arrow.

7. The following message appears in the data pane:

   No template available for this service. Do you want to create it?

8. Click Create in the message box to create the template.

   The NetMail attributes appear.

9. Make any changes to the NetMail attributes.

   See Appendix D, "NetMail Attributes"," for information on the NetMail attributes.

10. Click Save to store the final values in the service template.

NOTE When you create a new organization, you need to create and assign a NetMail policy for that organization. You do not need to do so for the sample portal as NetMail is already enabled by default.

To Create a Referral Policy for a Suborganization

You can delegate an organization's policy definitions and decisions to another organization. A referral policy controls this policy delegation for both policy creation and evaluation. It consists of a rule and the referral itself. The referral must define the parent organization as the resource in the rule, and it must contain a SubOrgReferral with the name of the organization as the value in the referral

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Navigate to the organization that contains the suborganization where you want to create a referral policy.

   All created organizations are displayed in the navigation pane.

3. Select Policies from the View menu.
4. Click New to create new policy.

   The Create Policy page appears in the data pane.
5. For Name, type SubOrgReferral_NetMail. Make sure you select Referral in Type of Policy. Then click Create.
6. Click Rules from the View menu in the data pane and click Add. Make sure NetMail is selected and click Next.

   The Add Rule template appears in the data pane.
7. Select NetMail in Service and click Next
8. Enter NetMailRule in Rule Name and click Create.
9. Click Referrals from the View menu in the data pane and click Add.

   The Add Referral template appears in the data pane.
10. Enter SubOrgReferral_*suborg_name* in Name.

    Make sure that the name of the suborganization (is selected for Value in the data pane and click Create to complete the policy's configuration.
11. Click Save in the data pane.

    The message "The policy properties have been saved" is displayed when the data is saved.

To Create a Normal Policy for a Suborganization

You create a normal policy to define access permissions. A normal policy can consist of multiple rules, subjects, and conditions.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.
2. Navigate to the organization or suborganization that you want to assign a policy.

   All created organizations are displayed in the navigation pane.
3. Choose Policies from the View menu.

   The policies for that organization are displayed.

4. Select New in the navigation pane. The New Policy page opens in the data pane.
5. Enter SubOrgNormal_NetMail in Name. Make sure you select Normal in Type of Policy. Click Create
6. Choose Rules from the View menu in the data pane and click Add. The Add Rule page opens in the data pane
7. Select NetMail from the Service menu and click Next. Enter NetMailRule in Rule Name. Make sure Has Privilege to Execute NetMail is checked
8. Select NetMail from the Service menu and click Next. Make sure Has Privilege to Execute NetMail is checked.
9. Select the type of subject from the Type menu and click Next to complete subject configuration.
10. Choose Subjects from the View menu in the data pane and click Add. The Add Subject page opens in the data pane.
11. Click Create to complete the policy configuration.

    The message “The policy properties have been saved.” is displayed when the data is saved.

To Modify NetMail Service Attributes (Specific Organization)

You can customize the NetMail service by modifying the attributes for the service.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.
2. Choose the organization.
3. Choose Services from the View menu.
4. Click the properties arrow next to NetMail in the navigation pane.

   A list of NetMail service attributes appears in the data pane.

5. Modify the service attribute values and then click Save to save the changes.

   The changes affect only users in the selected organization.

   See Appendix D, “NetMail Attributes”,” for more information.

To Modify NetMail Service Attributes (All Organizations)

Occasionally, you need to modify the global NetMail service attribute values that affect all organizations that want to register for the NetMail service in the future.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Choose Service Management in the location pane.

3. Click the properties arrow next to NetMail in the navigation pane.

   A list of NetMail service attributes appears in the data pane.

4. Modify the service attribute values then click Save to save the changes.

   The changes affect all organizations that register the NetMail service in the future.

To Configure NetMail Lite to Open a New Window

In the default configuration, if users click on the NetMail Lite link on the Desktop when they have NetMail Lite running and are composing a message, their current NetMail Lite window is replaced with a new instance of NetMail Lite and they lose the text in the message. To avoid this issue, you can configure NetMail Lite to open in a new window each time a user clicks on NetMail Lite link on the Desktop.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Choose the organization.

3. Click the properties arrow next to Desktop in the navigation pane.

   A list of Desktop service attributes appears in the data pane.

4. Click Channel and Container Management link in the data pane

5. Click the Edit link of App channel under Channels.

6. Choose the organization, and choose Services from the View menu.

7. Click the Edit link of `targets` property.

8. Replace the NetMail Lite property with the following:

```
NetMail Lite| ^javascript:var nmServerURL = document.URL; nmDestURL
=nmServerURL.split('dt')[0];nmAdjustedURL = nmDestURL
+'NetMailServlet?nsid=newHTMLSession';
openAppURL(nmAdjustedURL,'_blank');return false;
```

9. Click Save.

10. Verify the change.

    Log in as a test user within the organization. Access NetMail Lite and start composing a message. Click the NetMail Lite link. A new window containing NetMail Lite should open.

Using the Remote Address Book (LDAP)

To enable the remote address book feature for NetMail, you configure the LDAP server list attribute in the NetMail service.

NOTE The address book search capability enables users to search for names based on user specified text compared using the following criteria if supported by the search engine: containing, equal to, beginning with, ending with, and sounding like.

The personal address book only supports searching by contain. If you add an LDAP address book, you will see these other options enabled.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and Organizations is selected in the Navigation pane.

2. Choose the organization.
3. Choose Services from the View menu.
4. Click the properties arrow next to NetMail in the navigation pane.

   A list of NetMail service attributes appears in the data pane.
5. Modify the LDAP Server Details to Use in Address Book Search value. Each entry is a comma separated list of `name="`*value*`"` pairs where the valid names are:

   - `name`—The name that is shown in the Address page of NetMail (default: none)
   - `server`—The fully qualified domain name of the LDAP server (default: none)
   - `base`—The distinguished name (DN) that is used to start the search (default: "")
   - `searchin`—A comma separated list of attributes to look in (default: "cn,gn,sn")
   - `result`—The attribute that contains the email address (default: "`mail`")
   - `filter`—An additional LDAP filter to use for the search (default: ""). The syntax of the filter uses LDAP filter syntax.
   - `referral`—Value defining whether to follow LDAP referrals. The default is "`follow`"; use "" to define not to follow referrals.

   For example, to search the `Sesta` LDAP directory, use the following entry:

   ```
   name="Sesta
   LDAP",server="ldap-server.sesta.com",base="dc=sesta,dc=com"
   ```
6. Click Save.

Administering the Rewriter Service

This chapter describes how to administer the Rewriter service of the Sun™ ONE Portal Server.

This chapter includes the following sections:

Overview of the Rewriter Service

The Sun™ ONE Portal Server Rewriter provides an engine for performing URL translation in markup languages and JavaScript™ code. The `URLScraperProvider` and the `XMLProvider` in the Desktop and the Sun™ ONE Portal Server: Secure Remote Access gateway service all use the Rewriter service.

Rewriter scans the content of web pages and identifies the URLs it finds on those web pages. It uses a collection of rules defined in a ruleset to determine the elements of a web page to rewrite. Once Rewriter identifies a URL it can rewrite the URL by:

Expanding Relative URLs to Absolute URLs

The URLScraperProvider is part of the core Sun ONE Portal Server product. In a non-gateway scenario, the URLScraperProvider can be used to expand relative URLs to absolute URLs. For example, if a user is trying to access the site:

```
<a href="../mypage.html">
```

The Rewriter translates this to:

```
<a href="http://www.yahoo.com/mail/mypage.html">
```

where http://www.yahoo.com/mail/ is the base URL of the page scraped.

URLScraperProvider Limitations

The URLScraperProvider simply tries to display a designated URL in a channel. There's no way to specify parts of a document URL (document) to display. The URLScraperProvider acts much like an HTTP client, in that it makes a request for the content of the specified URL. Just like in a browser, the target URL to scrape must be network visible, or you must have a proxy configured.

The resultant URL scraper channel, however, is not a mini-browser nor is it a frame. Therefore, if you have a link in the content, it effects the whole page, not just the channel. You should not browse inside the URL scraper channel. If you select a link within the channel the browser can interpret the link and replace the currently displayed page (your portal server Desktop) with the contents of the link location.

The appearance of the scraped channel is controlled by whatever is producing the original content. The URLScraperProvider does not modify the content at all and only displays whatever is available through the URL. Since the channel is essentially a cell in an HTML table, it can only display HTML content that is legal to appear in table cells. That is, a frameset cannot be scraped using the URLScraperProvider because a <FRAMESET> tag cannot appear within a <BODY> tag. The URLScraperProvider will also not execute JavaScript code in <HEAD> tags. Because of this, the following scraping scenarios are inappropriate for the URLScraperProvider:

- When an Edit function of some sort is required so that the user can customize the channel.
- When the data comes from a non-HTML, non-web server source, that is, a database or mail server.
- When the data needs to be reformatted in some way for the channel.
- When a more efficient solution is required as the URLScraperProvider will do a request and look up for every Desktop display and user.

When cookies are sent by the origin server, they are forwarded back everytime web content is re-scraped. So the origin should get the cookies it sent as the web content scraped the first time, when portal desktop is updated or reloaded. But those cookies are not expected to be sent back when user clicks on any links in the url scraper channel.

Prefixing the Gateway URL to an Existing URL

In an implementation with a gateway such as the Sun ONE Portal Server: Secure Remote Access, the gateway acts as a proxy for the client and accesses intranet sites and returns responses to the client. The Rewriter translates URLs in downloaded pages so that they point back to the gateway rather than to the original site by prefixing the gateway URL to the existing URL.

For example, if a user tries to access an HTML page on `mymachine` using the following URL:

```
<a href="http://mymachine.intranet.com/mypage.html>
```

The Rewriter prefixes this URL with a reference to the gateway as follows:

```
<a
href="https://gateway.company.com/http://mymachine.intranet.com/mypage.html>
```

When a user selects a link associated with this anchor, the browser contacts the gateway. The gateway fetches the content of `mypage.html` from `mymachine.intranet.com`.

See the *Sun ONE Portal Server: Secure Remote Access 6.0 Administrator's Guide* for more information on using the Rewriter to prefix a gateway URL to an existing URL.

Supported URLs

Rewriter supports rewriting of all standard URLs as specified by RFC-1738. These URLs are supported whether the protocol is HTTP or HTTPS and regardless of the capitalization of the protocol. For example, hTtP, HTtp, and httP are all valid. Some sample standard URLs are listed below:

```
http://www.my.sesta.com
http://www.example.org:8000/imaginary/test
http://www.example.edu/org/admin/people#andy
http://info.example.org/AboutUs/Index/Phonebook?dobbins
http://www.example.org/RDB/EMP?*%20where%20name%%3Dobbins
http://info.example.org/AboutUs/Phonebook
http://user:password@example.com
```

Rewriter supports rewriting of some basic non-standard URLs. The information to convert non-standard URLs to a standard format is taken from the base URL of the page where the URL appears and can include the protocol, host name, and path. The back slash (\) is supported only when it is part of a relative URL and not part of an absolute URL. For example, `http://sesta.com\index.html` is rewritten, but `http:\\sesta.com` is not.

In addition, URLs with a single slash (/) after the protocol or scheme such as `http:/sesta.com` are not rewritten.

Defining Rewriter Rules and Rulesets

The Rewriter modifies the URL portions of various elements that appear on a web page. The Rewriter comes with a default set of rules to determine the elements of a web page to rewrite. A collection of rules for various categories and subcategories is stored in a `.dtd` file and is called a ruleset. The Rewriter rulesets are defined in XML.

The DTD is located in `/opt/SUNWps/web-src/WEB-INF/lib/rewriter.jar` (`resources/RuleSet.dtd`). Rulesets are used to identify URLs. By default, all strings in web content starting with characters such as "`/`", `../`, "`http`" and "`https`" are considered to be URLs and are candidates for rewriting.

To configure the Rewriter for your implementation, you create a ruleset and define rules in the Rewriter section of the Portal Server Configuration in the administration console. See "Administering the Rewriter Service" for details on creating and modifying rulesets. You define multiple rules based on the content type in the web pages. For example, the rule required to rewrite HTML content would be different from the rule required to rewrite JavaScript content. Rewriter rules fall into the following broad categories:

- Rules for HTML Content
- Rules for JavaScript Content

- Rules for XML Content

NOTE As Wireless Markup Language (WML) is similar to HTML, HTML rules are applied for WML content.

No rules are required for CSS content.

The ruleset is an XML document and the XML within it must be properly formed. When defining rules in a ruleset, keep the following guidelines in mind:

- All rules need to be enclosed within the `<ruleset> </ruleset>` tags.
- Include all rules to rewrite HTML content in the `<HTML> </HTML>` section of the ruleset.
- Include all rules to rewrite JavaScript content in the `<JSRules> </JSRules>` section of the ruleset.
- Include all rules to rewrite XML content in the `<XML> </XML>` section of the ruleset.

Rules for HTML Content

HTML content in web pages can be classified into attributes, JavaScript tokens, forms, and applets. Accordingly, the rules for HTML content are classified as:

- Attribute Rules for HTML Content
- JavaScript Token Rules for HTML Content
- Form Rules for HTML Content
- Applet Rules for HTML Content

Attribute Rules for HTML Content

Attribute rules identify the basic attribute tags in HTML pages to rewrite. Rewriter modifies the various occurrences of the defined tags by expanding or prefixing the existing URL. The default ruleset rewrites the following attribute tags:

- action
- background
- codebase
- code

- href
- src
- value
- imagePath
- lowsrc
- archive

The syntax for attribute rules is:

```
<Attribute name="name" [tag="tag" valuePatterns="patterns"]
```

where *name* specifies the attribute, *tag* specifies the tag to which the attribute belongs (set to * to match all tags), and *patterns* specifies the possible patterns to match with the attribute. The `tag` and `valuePatterns` parameters are optional.

JavaScript Token Rules for HTML Content

Web pages can contain pure JavaScript code within the JavaScript tags, or they can contain JavaScript tokens or functions. For example, a web page can contain an `onClick()` function that causes a jump to a different URL. In order for the page to function properly, the value of the `onClick()` function needs to be translated and rewritten. In most cases, the rules provided in the default ruleset are sufficient to rewrite the URLs in JavaScript tokens. The default ruleset rewrites the following JavaScript tokens:

- `onAbort`
- `onBlur`
- `onChange`
- `onClick`
- `onDblClick`
- `onError`
- `onFocus`
- `onKeyDown`
- `onKeyPress`
- `onKeyUp`
- `onLoad`
- `onMouseDown`

- onMouseMove
- onMouseOut
- onMouseOver
- onMouseUp
- onReset
- onSelect
- onSubmit
- onUnload

The syntax for JavaScript Token rules is:

<JSToken>*javascript_function_nam*e</JSToken>

where *javascript_function_nam*e is the name of the function such as onLoad or onClick.

Form Rules for HTML Content

Users can browse HTML pages that contain forms. Form elements, such as input, can take a URL as a value. The default ruleset does not rewrite any form elements. The syntax for form rules is:

<Form source="/*source.html*" name="*form1*" field="*field1*">
[valuePatterns="*pattern*"] />

where */source.html* is the URL of the HTML page containing the form, *form1* is the name of the form, *field1* is the field of the form to be rewritten, and *pattern* indicates the part of the field to be rewritten. All content that follows the pattern specified is rewritten.

The valuePatterns parameter is optional.

Applet Rules for HTML Content

A single web page can contain many applets, and each applet can contain many parameters. The Rewriter rule for URLs in applets should contain pattern matching information for the following:

- source, such as filename.htm
- code, such as classname.class
- parameter name, such as servername
- parameter value, such as some_url

Rewriter matches the values specified in the rule with the content of the applet and modifies the URLs as required. This replacement is carried out at the server and not when the user is browsing the particular web page. A wildcard character (*) can also be used as part of the rule. For example, the parameter name could be *, in which case, the Rewriter does not compare the parameter name in the applet.

The default ruleset does not rewrite any applet parameters.

The syntax for applet rules is:

```
<Applet source="sourcehtml.jsp" code="class" param="parameter_name"
[valuePatterns="pattern"]
```

where */sourcehtml.jsp* is the URL containing the applet, *class* is the name of the applet class, *parameter_name* is the parameter whose value needs to be rewritten, and *pattern* indicates the part of the field to be rewritten. All content that follows the pattern specified is rewritten. The valuePatterns parameter is optional.

Rules for JavaScript Content

URLs can occur in various portions of JavaScript code. The Rewriter cannot directly parse the JavaScript code and determine the URL portion. A special set of rules needs to be written to help the JavaScript processor translate the URL.

JavaScript elements that contain URLs are classified as follows:

- JavaScript Variables
- JavaScript Function Parameters

JavaScript Variables

JavaScript variables are again classified into five categories:

- JavaScript URL Variables
- JavaScript EXPRESSION Variables
- JavaScript DHTML Variables
- JavaScript DJS (Dynamic JavaScript) Variables
- JavaScript System Variables

JavaScript URL Variables

URL variables have a URL string on the right hand side. The default ruleset rewrites the following JavaScript URL variables:

- `imgsrc`
- `location.href`
- `_fr.location`
- `mf.location`
- `parent.location`
- `self.location`

The syntax of URL variables in JavaScript content rules is:

`<Variable type="URL">`*variable_name*`</Variable>`

where *variable_name* is the name of the variable to be rewritten.

JavaScript EXPRESSION Variables

EXPRESSION variables have an expression on the right hand side. The result of this expression is a URL. The Rewriter appends a JavaScript function for converting the expression to the HTML page as it cannot evaluate such expressions. This function takes the expression as a parameter and evaluates it at the client browser.

The default ruleset rewrites the `location` JavaScript EXPRESSION variable.

The syntax of EXPRESSION variables in JavaScript content rules is:

`<Variable type="EXPRESSION">`*variable_exp*`</Variable>`

where *variable_exp* is the expression variable.

JavaScript DHTML Variables

DHTML variables are JavaScript variables that hold HTML content. The default ruleset rewrites the following JavaScript DHTML variables:

- `document.write`
- `document.writeln`

The syntax of DHTML variables in JavaScript content is:

`<Variable type="DHTML">`*variable*`</Variable>`

where *variable* is the DHTML variable.

JavaScript DJS (Dynamic JavaScript) Variables

DJS (Dynamic JavaScript) variables are JavaScript variables that hold JavaScript content.

The syntax of DJS variables in JavaScript content is:

```
<Variable type="DJS">variable</Variable>
```

where *variable* is the DJS variable.

The JavaScript code contained in the variable needs another rule to translate it.

JavaScript System Variables

System variables are variables that are not declared by the user, but that are available as a part of the JavaScript standard.

The default ruleset rewrites the `window.location.pathname` JavaScript system variable.

The syntax of system variables in JavaScript content is:

```
<Variable type="SYSTEM">variable</Variable>
```

where *variable* is the system variable.

JavaScript Function Parameters

Function parameters are classified into four categories:

- JavaScript URL Parameters
- JavaScript EXPRESSION Parameters
- JavaScript DHTML Parameters
- JavaScript DJS Parameters

JavaScript URL Parameters

URL parameters are string parameters that directly contain the URL.

The default ruleset rewrites the following JavaScript URL parameters:

- `openURL`
- `openAppURL`
- `openNewWindow`
- `parent.openNewWindo`
- `window.open`

The syntax for URL parameters is:

```
<Function type = "URL" name = "function" [paramPatterns="y,y,"] />
```

where *function* is the name of the function to be evaluated and *y* indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but the third parameter should not be rewritten.

JavaScript EXPRESSION Parameters

EXPRESSION parameters are variables within a function that result in a URL when they are evaluated. The syntax for EXPRESSION parameters is

```
<Function type = "EXPRESSION" name = "function" [paramPatterns="y,y,"]
/>
```

where *function* is the name of the function to be evaluated and *y* indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but the third parameter should not be rewritten.

JavaScript DHTML Parameters

DHTML parameters are native JavaScript methods that generate an HTML page dynamically. For example, the `document.write()` method falls under this category.

The default ruleset rewrites the following JavaScript DHTML parameters:

- `document.write`
- `document.writeln`

The syntax for DHTML parameters is:

```
<Function type = "DHTML" name = "function" [paramPatterns="y,y,"] />
```

where *function* is the name of the function to be evaluated and *y* indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but not the third parameter should not be rewritten.

JavaScript DJS Parameters

Dynamic JavaScript (DJS) parameters such as Cascading Style Sheets (CSS) in HTML are also translated. There are no rules defined for this translation as the URL appears only in the `url()` function of the CSS. The syntax for DJS parameters is:

```
<Function type = "DJS" name = "function" [paramPatterns="y,y,"] />
```

where *function* is the name of the function to be evaluated and *y* indicates the position of the parameter(s) that need to be rewritten. Parameter positions are delimited by commas. For example, in the syntax line the first and second parameters need to be rewritten, but not the third parameter should not be rewritten.

Rules for XML Content

Web pages can contain XML content which in turn can contain URLs and Rewriter can rewrite URLS in XML content.

XML content that contains URLs is classified as follows:

- Tag Text in XML
- Attributes in XML

Tag Text in XML

Rewriter translates XML content based on the tag name.

The default ruleset rewrites the following tags in XML:

- `baseroot`
- `img`

The syntax for tag text is:

`<TagText tag ="`*attribute*`" attributePatterns="name=`*src*`"/>`

where *attribute* is the name of the tag and *src* is the name of the attribute.

Attributes in XML

The rules for attributes in XML are similar to the rules for attributes in HTML. See "Attribute Rules for HTML Content" on page 201" for additional information Rewriter translates attribute values based on the attribute and tag names.

The default ruleset rewrites the following attributes in XML:

- `xmlns`
- `href`

The syntax for attributes in HTML is:

```
<Attributes>
   <Attribute name="attribute" [valuePatterns="name=src"/>
</Attributes>
```

where *attribute* is the name of the tag and *src* is the name of the attribute.

Administering the Rewriter Service

In Sun ONE Portal Server 6.2, the Rewriter service uses iPlanet™ Directory Server Access Management Edition attributes to provide persistent storage for the Rewriter rulesets. A Rewriter ruleset defines how contents in a web page should be rewritten by the Rewriter. Multiple Rewriter rulesets can be defined and stored as Sun ONE Identity Server service attribute values through the Sun ONE Identity Server administration console.

You can also administer the Rewriter using the command line. See Chapter 14, "Command-Line Utilities" for more information on the `rwadmin` command.

Because the Sun ONE Identity Server administration console does not have any concept of a rewriter ruleset, Sun ONE Portal Server uses a customized service management plug-in module to manage them. All Rewriter rulesets are global to the organizations in Sun ONE Identity Server. There is no provision to enable the creation of ruleset at any particular organization level.

NOTE The `URLScraperProvider` can only scrape content that is valid inside of an HTML table cell. If the HTML markup to scrape contains markup that cannot be rendered in a table cell, such as `<body>`, `<base>`, and certain JavaScript procedures, that cannot be rendered within a table cell, the display of the Desktop page can be corrupted. When defining content to scrape, try to confirm the content is valid HTML. See "URLScraperProvider Limitations" for further information.

To Configure the Rewriter URLScraperProvider for SSL

You can use the Rewriter's `URLScraperProvider` to scrape SSL pages and rewrite the URLs for access over a secure session.

1. Initialize the trust database in the web server administration console for the server on which you installed Sun ONE Portal Server as follows:

   a. From a browser, enter the following URL to access the Web Server admin page:

      `http://`*servername*`:8088`**

b. Log in as Admin and click the Security tab.

c. Enter the Database password twice and select OK.

2. Create a password file as follows:

a. Change directories to */BaseDir*/SUNWam/config.

b. Create a hidden text file .wtpass.

c. Type the password that you gave when you initialized the trust database.

3. Add the following line to the */BaseDir*/SUNWam/lib/AMConfig-*instance_nickname*.properties file if the root CA is not installed for the certificates used by the Web servers accessed using the URLScraperProvider.

```
com.sun.am.jssproxy.trustAllServerCerts=true
```

This option tells JSS to trust the certificate.

4. Restart Sun ONE Portal Server.

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

To Create a New Ruleset from the Default Template

1. Log in to the Sun ONE Identity Server administration console as administrator.

2. Select Service Configuration from the location pane.

3. Click the properties arrow next to Rewriter in the navigation pane.

A list of currently defined rulesets appears in the data pane.

4. Click New.

This displays a ruleset template for possible modifications.

5. Edit the <RuleSet id="ruleset_template"> line, replacing default_ruleset with the name for the new ruleset.

6. Add or modify the rules within the ruleset template to rewrite URLs as necessary.

7. Click Save to create the new ruleset.

   Upon success, you see the initial page and the list of all currently defined rulesets, which should include the one you just created.

To Edit an Existing Ruleset

1. Log in to the Sun ONE Identity Server administration console as administrator.
2. Select Service Configuration from the location pane.
3. Click the properties arrow next to Rewriter in the navigation pane.

   A list of currently defined rulesets appears in the data pane.
4. Click the Edit link for the ruleset to edit.

   This displays the XML for the ruleset to edit.
5. Add or modify the rules within the ruleset template to rewrite URLs as necessary.
6. If you would like to change the name of the ruleset, edit the `<RuleSet id="ruleset_template">` line, replacing name with a name for the ruleset.
7. Click Save.

To Download a Ruleset

Rulesets can be downloaded and saved to a file.

1. Log in to the Sun ONE Identity Server administration console as administrator.
2. Select Service Configuration from the location pane.
3. Click the properties arrow next to Rewriter in the navigation pane.

   A list of currently defined rulesets appears in the data pane.
4. Clickt the Download link for the ruleset to save to a file.
5. Specify a name for the file and save it.

To Upload a Ruleset

A ruleset file can be uploaded into the system.

1. Log in to the Sun ONE Identity Server administration console as administrator.
2. Select Service Configuration from the location pane.
3. Click the properties arrow next to Rewriter in the navigation pane.

   A list of currently defined rulesets appears in the data pane.
4. Click the Upload link next to any ruleset in the list.
5. Browse to or type the file name for the ruleset to upload.
6. Click Upload.

   If the name defined in the `<RuleSet id="ruleset_template">` line within the file matches a ruleset name on the system that ruleset file will be replaced with the contents of the file. If the name defined in the `<RuleSet id="ruleset_template">` line is unique, a new ruleset will be created with that name and added to the list.

To Delete an Existing Ruleset

1. Log in to the Sun ONE Identity Server administration console as administrator.
2. Select Service Configuration from the location pane.
3. Click the properties arrow next to Rewriter in the navigation pane.

   A list of currently defined rulesets appears in the data pane.
4. Click the checkbox next to the ruleset to be deleted.

   You can select more than one ruleset.
5. Click Delete.

   A confirmation message appears.
6. Click Yes to delete the selected rulesets.

To Restore the Default Ruleset

In case you accidentally delete the default ruleset, you can restore it as follows:

```
rwadmin store --runasdn "uid=amadmin, ou=people, o=sesta.com, o=isp"
--password "testing123" /resources/DefaultRuleSet.xml
```

where "/resources/DefaultRuleSet.xml" is the location of the ruleset stored in the rewriter.jar file.

NOTE The default ruleset packaged from the installation is restored. If you have customized the default ruleset, the changes are not restored.

Chapter 8

Administering the Search Engine Service

This chapter describes how to configure and administer the Sun™ ONE Portal Server Search Engine service.

This chapter contains these sections:

- Overview of the Search Engine Service
- Configuring the Search Channel
- Administering the Search Engine
- Administering the Robot
- Administering the Database
- Administering the Database Taxonomy

Overview of the Search Engine Service

The Sun ONE Portal Server Search Engine is a taxonomy and database service designed to support search and browse interfaces similar to popular internet search engines such as Google, Alta Vista, and so on. Search Engine includes a robot to discover, convert, and summarize document resources. In Sun ONE Portal Server 6.2, the interface is provided by the Desktop exclusively, using JSP™ providers. Search Engine includes administration tools for configuration editing

and command-line tools for system management. Configuration settings can be defined and stored as iPlanet™ Directory Server Access Management Edition service attribute values through the Sun ONE Identity Server administration console.

NOTE Although the administration console permits an administrator to configure a majority of the Search Engine options, the administration console does not perform all the administrative functions available through the command line.

Search Database

Search users search through a database to locate particular resources or kinds of resources. The individual entries in the database are called resource descriptions (RDs). An Resource Description is a specific set of information about a single resource. The fields of each Resource Description are determined by the database schema.

To get RDs into the database, you can use two approaches:

- Creating RDs—This is by far the most common method, using a robot process to locate resources and generate their descriptions.
- Exchanging RDs—This method is appropriate for large, distributed network indexes. A remote system generates RDs, and the Search Engine imports those into its database.

The RDs in the Sun ONE Portal Server Search Engine are based on open Internet standards, such as the Summary Object Interchange Format (SOIF) and resource description messages (RDM). This ensures that the Search Engine can operate in a cross-platform enterprise environment.

Search Robots

One method of filling the database is via robots. The Search Engine uses robots to find and report on the resources in their domains. A *robot* is a small program that does two things:

- Extracts and follows links to resources (also called enumeration or crawling)
- Describes those resources and puts the descriptions in the database (also called generation or indexing)

As the system administrator, you control every aspect of these processes in a number of ways, including the following:

- When the robot runs by starting, stopping, and scheduling the robot.
- Where the robot looks for resources by defining the sites the robot visits.
- How aggressively it searches by defining the crawling attributes.
- What types of resources the robot indexes by defining filters.
- What kind of entries it creates for the database by defining the indexing attributes.

The Search Engine also provides utilities to ensure that the robot has done what you wanted.

Database Taxonomy Categories

Users interact with the Search system in two distinct ways: they can type direct queries to search the database, or they can browse through the database contents using a set of categories you design. A hierarchy of categories is sometimes called a taxonomy. Categorizing resources is like creating a table of contents for the database.

Browsing is an optional feature in a Search system. That is, you can have a perfectly useful Search system that does not include browsing by categories. You need to decide whether adding browsable categories will be useful to the users of your index, and then what kind of categories you want to create.

The resources in a Search database are assigned to categories to clarify complexity. If there is a large number of items in the database, it is helpful to group related items together. This allows users to quickly locate specific kinds of items, compare similar items, and choose which ones they want.

Such categorizing is common in the product and service indexes. Clothing catalogs divide men's, women's, and children's clothing, with each of those further subdivided for coats, shirts, shoes, and so on. An office products catalog could separate furniture from stationery, computers, and software. And advertising directories are arranged by categories of products and services.

The principles of categorical groupings in a printed index also apply to online indexes. The idea is to make it easy for users to locate resources of a certain type, so that they can choose the ones they want. No matter what the scope of the index you design, the primary concern in setting up your categories should be usability. That is, you need to know how users will use the categories. For example, if you were

designing an index for a company that has three offices in different locations, you might make your top-level categories correspond to each of the three offices. But if users are more interested in, say, functional divisions that cut across the geographical boundaries, it might make more sense to categorize resources by corporate divisions.

Once the categories are defined, you must set up rules to assign resources to categories. These rules are called *classification rules*. If you do not define your classification rules properly, users will not be able to locate resources by browsing in categories. You need to avoid categorizing resources incorrectly, but you also should avoid failing to categorize documents at all.

Documents can be assigned to multiple categories, up to a maximum number defined in the settings. Classification rules are simpler than filter rules because they don't involve any flow-control decisions. In classification rules you determine what criteria to use to assign specific categories to a resource as part of its Resource Description. A classification rule is a simple conditional statement, taking the form "if <some condition> is true, assign the resource to <a category>."

Configuring the Search Channel

This section describes how to initially configure the Search Engine service. Configuration settings can be defined and stored as Sun ONE Identity Server service attribute values through the Sun ONE Identity Server administration console.

The Search service is registered globally and its configuration applies to the entire Portal Server. By default, the organization you specify during the Sun ONE Portal Server installation will have the Search service registered. If you install the sample portal, the Search tab on the sample portal Desktop contains the search channel. This is configured for you during the Sun ONE Portal Server installation. However, for new organizations and for new instances you must define the Search URL.

The default behavior for a search provider user is that "No document matches" found will be displayed when the user enters a query.

You need to configure the Search server and create the document database to get search results.

To Initially Configure the Search Server

Use these steps to configure the Search provider. This is a sample way to fill in the database. You can also use the import function.

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

3. Click the properties arrow next to Search in the navigation pane.

4. Create a new site.

   a. Click Robot.

   b. Click Sites.

   c. Click New under Manage Sites to define sites for the Robot to index.

   d. Specify the type of site (URL or domain), the site to index, and the depth for the robot to crawl.

   e. Click Create Site to use the default Search attributes or select Create and Edit Site to define the search site more completely.

      See Appendix F, “Search Attributes” for more information on the attributes that define the site.

5. Create a taxonomy.

   You can create a taxonomy using the Category Editor under Categories or by copying a sample taxonomy SOIF file to `config/taxonomy.rdm`).

6. Disable any of the default filters that you do not want to use.

   Click Robot and then Filters. Turn off any filters in the Filter Rule list you do not want to use.

7. (Optional) Create robot classification rules if you need to get document results under categories.

   You can create a create robot classification rulesusing the Classification Rules Editor under Categories

8. Start the robot.

   Click Robot, Overview, and then Start to start the robot.

9. Reindex the categories

   Click Categories then Reindex to reindex.

To Define the Search URL

The `searchServer` property defines the Search URL. It is automatically configured for the default organization; however, this value is not defined when new organizations are created, when new `SearchProvider` instances are created, or when the sample `dp-org.xml` is loaded manually. If users search when value is not defined, the following error message is displayed on the user's Desktop:

```
You got a
com.sun.portal.search.providers.taglib.SearchTaglibException:
SearchRequest Error: search server is not defined.
```

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose the organization for which you want to define the Search URL.

3. Choose Services from the View menu

   Use the Show menu in the navigation pane and the Location path in the location pane.

4. Click the properties arrow next to Desktop in the navigation pane.

   The Desktop attributes page appears in the data pane.

5. In the Desktop page, click the Channel and Container Management link.

   The Channels page appears. At the top is the container path. The defined channels appear in a list.

6. Click the Edit Properties link beside the Search channel to be modified.

   The Properties page appears.

7. Specify the SearchURL in the Search Server property in the format:

   `http://`*portal_server_name*`:`*port*`/portal/search`**

8. Click Save.

9. To verify the Search URL, do the following:

a. Log in to the organization for which you configured the Search URL. For example, log in to an organization named B as follows:

   `http://`*portal_server_name*`:`*port*`/amserver/ui/login?org=B`

b. Perform a search from the Search channel.

Administering the Search Engine

Once you have initially configured the Search Engine and generated a database, you can view and manage the Search Engine from the Sun ONE Identity Server administration console.

Viewing, Managing, and Monitoring Search Engine Operations

Search Engine operational attributes have two levels: basic and advanced.The basic settings page appears by default when the Search service is selected from the administration console. The basic settings displayed include the server root, the location of the temporary files, and the document level security. The advanced settings include the log locations for various Search Engine components and the configured log level.

In addition, the administration console allows administrators to view the log files or specific information extracted from the log files.

To View or Manage the Basic Settings

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Click Server then Settings from the menu bar.
5. View or specify the Server Root directory for the Search Engine.
6. View or specify the Temporary Files directory for the Search Engine.

7. View or specify the Document Level Security attribute.

   Off means all users have access to the RDs in the database. On indicates that the ReadACL field in the RD must be evaluated to determine if the user has permission to access the RD.

8. Click Save to record any altered attributes.

To View or Manage the Advanced Settings

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

3. Click the properties arrow next to Search in the navigation pane.

4. Click Server then Advanced from the menu bar.

5. View or specify the Advanced attributes

   The attributes available are: Search (rdm), Disable Search Log, Index Maintenance, RD Manager, RDM Server, and Log Level.

6. Click Save to record any altered settings.

To Monitor Search Engine Activity

The Search Engine provides a number of reports to allow you to monitor the search activity.

To view the various reports:

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

3. Click the properties arrow next to Search in the navigation pane.

4. Click Reports.

5. Click on a link in the menu bar to view a specific report.

   The following report options are available: Starting Points, Excluded URLs, Robot Advanced Reports, Log Files, and Popular Searches.

Administering the Robot

The following are some configuration and maintenance tasks you might need to do to administer the robot:

- Defining Sites
- Controlling Robot Crawling
- Filtering Robot Data
- Defining the Indexing Attributes
- Using the Robot Utilities
- Scheduling the Robot

Defining Sites

The robot finds resources and determines if (and how) to add descriptions of those resources to the database. The determination of which servers to visit and what parts of those servers to index is called site definition.

Defining the sites for the Search Engine is one of the most important jobs of the server administrator. You need to be sure you send the robot to all the servers it needs to index, but you also need to exclude extraneous sites that can fill the database and make it hard to find the correct information.

To Define Sites for the Robot to Index

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

3. Click the properties arrow next to Search in the navigation pane.

4. Click Robot then Sites from the menu bar.
5. To create a site:
   a. Click New.
   b. Select the type of site (url or domain).
   c. Specify the site and depth.
   d. Click Save.
6. To edit the site attributes, click the Edit link.

   This displays a form containing site attributes. See Appendix F, “Search Attributes”,” for information on the Site attributes.

   e. Edit the attributes.
   f. Click Save.

Controlling Robot Crawling

The robot crawls to the various sites selected for indexing. Administrators can control how the robot searches sites by defining crawling operational parameters. Crawling parameters allow you to define the speed, completion actions, logging level, standards complaince, autentication parameters, proxy settings, maximum number of links to follow, and other settings. See Appendix F, “Search Attributes”,” for desciptions of the robot crawling attributes.

To Control Robot Crawling

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Click Robot and then Crawling from the menu bar.

   This displays a form containing attributes that define the Robot Crawling operational parameters and their settings. See Appendix F, “Search Attributes”,” for information on the Robot Crawling attributes.

5. Modify the Robot Crawling attributes as necessary.

NOTE If the `jvm12.conf` file has a proxy set up (using the `http.proxyHost=` and `http.proxyPort=` options) you must check Accepts Commands from Any Host for the Robot to run.

6. Click Save.

Filtering Robot Data

Filters allow an attribute of a resource to be compared against a filter definition to identify a resource so that it can be excluded or included by the Site definitions. The Robot comes with a number of predefined filters some of which are enabled by default. The following filters are predefined; files marked with an asterisk are enabled by default:

- Archive Files*
- Audio Files*
- Backup Files*
- Binary Files*
- CGI Files*
- Image Files*
- Java, JavaScript, Style Sheet Files*
- Log Files*
- Power Point Files
- Revision Control Files*
- Source Code Files*
- Temporary Files*
- Video Files*
- Spreadsheet Files
- Plug-in Files
- Lotus Domino Documents

- Lotus Domino OpenViews
- System Directories (UNIX)
- System Directories (NT)

To manage the filtering process, you can create new filter definitions, modify a filter definition, or enable or disable filters.

To Create a New Filter Definition

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Robot then Filters from the menu bar.
5. Click New and specify a Nick Name for the new filter.
6. In the Filter Definition, check the checkbox and specify the Filter Source, Filter by and Filter String values. You may specify as many Filter Definitions as necessary.
7. Type a description of the filter.
8. Check New Site if you would like this filter to be used when creating new sites.
9. Click the button to indicate whether to include or exclude resources that match this filter.
10. Click Save.

To Modify an Existing Filter Definition

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.

4. Select Robot then Filters from the menu bar.
5. Locate the Filter to modify from the Filter Rules list and click the Edit link.
6. Modify the Filter as necessary.
7. Type a description of the filter.
8. Click Save.

To Enable or Disable a Filter

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Robot then Filter from the menu bar.
5. Locate the Filter to modify from the Filter Rules list.
6. Select the button to indicate whether to turn the filter on or off.
7. Click Save.

Defining the Indexing Attributes

For each resource that passes through the robot's filters, the robot generates an RD that it places in the database. The choices you make in setting up the generation of RDs determine what users will see when they search the database. For example, you can choose to index the full text of each document or only some fixed portion of the beginning of the document.

To Define the Indexing Attributes:

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.

3. Click the properties arrow next to Search in the navigation pane.
4. Select Robot then Indexing from the menu bar.

   This displays a page containing attributes that define the Robot Indexing operational parameters and their settings. See Appendix F, "Search Attributes"," for information on the Robot Indexing attributes.
5. Modify the Robot Indexing attributes as necessary.
6. Click Save.

Using the Robot Utilities

The Robot includes two debugging tools or utilities:

- Site Probe—Checks for DNS aliases, server redirects, virtual servers, and the like.
- Simulator—Performs a partial simulation of robot filtering on a URL. Type one or more URLs to check and select OK. The simulator will indicate whether the listed sites would be accepted by the robot.

To Run the Site Probe Utility

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Robot then Site Probe from the menu bar.
5. Type the URL of the site to probe.
6. Click Show Advanced DNS information if you want the probe to return DNS information.
7. Click OK to start the Site Probe.

To Run the Simulator

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Robot then Simulator from the menu bar.
5. Type in one or more URLs on which to perform the simulation.
6. Select Check for DNS aliases if you would like the Simulator to check for aliases.
7. Select Check for Server Redirects (302) if you would like the Simulator to check for redirects.
8. Click OK to start the Simulator.

Scheduling the Robot

In order to keep the search data timely, the robot should search and index sites regularly. Because robot crawling and indexing can consume processing resources and network bandwidth. To avoid these resource constraints, you should schedule the robot to run during non-peak days and times. The administration console allows administrators to set up a `cron` job with the time and days to run the robot.

To Schedule the Robot

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Robot then Schedule from the menu bar.
5. Select the time (hour and minutes) and days to start the robot.

6. Select the time and days to stop the robot.
7. Click Save.

Administering the Database

The Search Engine stores its descriptions of resources in a database. The following are some configuration and maintenance tasks you may need to perform to administer the database:

- Importing to the Database
- Editing Resource Descriptions
- Editing the Database Schema
- Defining Schema Aliases
- Viewing Database Analysis
- Reindexing the Database
- Expiring the Database
- Purging the Database
- Partitioning the Database

Importing to the Database

Normally, the items in your Search database come from the robot. You tell the robot which sites to visit, and it locates and describes all the resources it finds there. But you can also import databases of existing items, either from other Sun ONE Portal Server Search Engines, from iPlanet Web Servers or Netscape™ Enterprise Servers or from databases generated from other sources. Import existing databases of RDs instead of sending the robot to create them anew helps reduce the amount of network traffic and also enables large indexing efforts to be completed more quickly by breaking the effort down into smaller parts. If the central database is physically distant from the servers being indexed, it can be helpful to generate the RDs locally, then have the central database import the various remote databases periodically.

The Search Engine uses an import agent to import RDs from another server or from a database. An *import agent* is a process that retrieves a number of RDs from an external source and merges that information into the local database. It contains parameters that tell it where to go to import RDs, what to ask for when it gets there, and some other information that fine-tunes the way it goes about the job.

Before you can import a database, you must create an import agent. Once an agent is created, you can start the import process immediately or schedule a time to run the import process.

To Create an Import Agent

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then click the Import Agents link..
5. Click New.

   The attributes page for the import agent appears.

6. Specify the appropriate attributes for the import agent.

   See Appendix F, “Search Attributes”,” for information on the Database Import attributes.

   a. Indicate whether the source is a local file or search server.
   b. If the source is a file, specify the local file path.
   c. If the source is another search server, specify the URL for the remote server, the instance name, and the search URI.
   d. Specify the name of the database to import.
   e. Specify the character set for the import agent.

7. Click Save.

To Edit an Existing Import Agent

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration om the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then click the Import Agents link.
5. Click the Edit link to the right of the agent to edit.
6. Specify the appropriate attributes for the import agent.

   See Appendix F, "Search Attributes"," for information on the Database Import attributes

7. Click Save.

Editing Resource Descriptions

At times you will find it necessary to change the contents of one or more Resource Descriptions. For example, you might need to correct a typographical error copied into an Resource Description from an original document.

To Edit the Resource Descriptions

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then Resource Descriptions from the menu bar.
5. Select the type of Resource Description to search for to edit.

   The following types are available: All RDs, Uncategorized RDs, Categorized RDs, RDs by category, Specific RD by URL, RDs that contain.

6. For Resource Descriptions that contain, specify a text string to search for in the Resource Description.
7. Click Search.
8. From the list of Resource Descriptions found, select the Resource Description to edit.
9. Edit the appropriate Resource Description attribute.
10. Click Save.

Editing the Database Schema

A schema determines what information your Search Engine maintains on each resource, and in what form. The design of your schema determines two factors that affect the usability of your index:

- The way users can search for resources
- The ways users view resource information

The schema is a master data structure for Resource Descriptions in the database. Depending on how you define and index the fields in that data structure, users will have varying degrees of access to the resources.

The schema is closely tied to the structure of the files used by the Search Engine and its robot. You should only make changes to the data structure by using the schema tools in administration console. You should never edit the schema file (`schema.rdm`) directly, even though it is a text file.

You can edit the database schema of the Search Engine to add a new schema attribute, edit a schema attribute, or delete attributes.

The schema includes the following attributes:

- Editable—If checked, this attribute indicates that the attribute appears in the Resource Description Editor, so you can change its values. The Resource Description Editor is explained in “Editing Resource Descriptions” on page 232.
- Indexable—This attribute indicates that the field appears in the pop-up menu in the Advanced Search screen. This allows users to search for values in that particular field.
- Description—This is a text string to use to describe the schema. You can use it for comments or annotations.

- Aliases—This attribute allows you to define aliases to convert imported database schema names into your own schema.

To Edit the Database Schema

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then Schema from the menu bar.

   The Schema attributes page appears.

5. To add a new attribute to the schema:
   a. Select New under Schema List.
   b. Type a name and description for the new attribute in the Name and Description fields.
   c. Check Editable to allow the attribute to be edited.
   d. Check Indexable to make the attribute indexed.
6. To make an existing schema attribute editable or indexable:
   a. Click the Edit link next to an attribute from the schema list.

      The Schema attributes page appears.

   b. Check Editable to allow the attribute to be edited.
   c. Check Indexable to make the attribute indexed.
   d. Click Update
7. To delete an attribute:
   a. Check an attribute from the schema list.
   b. Click Delete.

NOTE Changes to the search engine schema may require that the entire database be reindexed and the server restarted. This is because the search engine highlighting functions are sensitive to the order and types of the schema fields. Adding or removing (or even removing and then adding backagain) a text field has a high likelihood of causing search result highlighting to be incorrect.

Defining Schema Aliases

There are several instances where you might encounter discrepancies between the names used for fields in database schemas. One is when you import Resource Descriptions from one server into another. You cannot always guarantee that the two servers use identical names for items in their schemas. Similarly, when the robot converts HTML META tags from a document into schema fields, the document controls the names.

The Search Engine allows you to define schema aliases for your schema attributes, to map these external schema names into valid names for fields in your database.

To Define Schema Aliases

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then click the Schema link.

   The Schema attributes page appears.
5. Click the attribute for which to define an alias.
6. Specify the field name of the alias as it is used in the imported database.
7. Click Update.

8. Click Reindex.

   The reindexing process may take several hours for a large database.

Viewing Database Analysis

The Search Engine provides a report with information about the number of sites indexed and the number of resources from each in the database.

To View Database Analysis Information

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then Analysis from the menu bar.

   A sorted list of all sites and the number of resources from that site currently in the search database.

5. To generate a up-to-date list, click Save.

Reindexing the Database

In certain instances, you might need to reindex the Resource Description database for the Search Engine. One obvious instance is if you have edited the schema to add or remove an indexed field.

You might also need to reindex the database if a disk error corrupts the index file. It's also a good idea to reindex after adding a large number of new Resource Descriptions.

Reindexing the database can take several hours.

The time required to reindex the database is proportional to the number of records in the database, so if you have a large database, you should perform reindexing at a time when the server is not in high demand.

To Reindex the Database

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then Management.
5. Click Reindex under Database List.
6. Check the Reindex the database? checkbox and click OK.

   The Search Engine rebuilds the search collection and its index files.

Expiring the Database

Expiring the database will expire Resource Descriptions deemed out of date. Resource Descriptions will expire ONLY when you run the expiration. The expired Resource Descriptions will be deleted, however the database size will not decreased.

To Expire the Database:

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then Management.
5. Select Expire under Database List.
6. Check the Expire RDs? checkbox and click OK.

Purging the Database

One attribute of an Resource Description is its expiration date. Your robots can set the expiration date from HTML META tags or from information provided by the resource's server. By default, Resource Descriptions expire in three months from creation unless the resource specifies a different expiration date. Periodically your Search Engine should purge expired Resource Descriptions from its database.

Purging allows you to remove the contents of the database. Disk space used for indexes will be recovered, but disk space used by the main database will not be recovered, instead, it is reused as new data are added to the database.

To Purge Expired Resource Descriptions from a Server:

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Database then Management.
5. Select Purge under Database List.
6. Check the Purge the database? checkbox and click OK.

   When the purge is complete, the system displays the message "The database contents were successfully purged."

Partitioning the Database

The Search Engine allows you to split the physical files that contain the search database across multiple disks, file systems, directories, or partitions. By spreading the database across different physical or logical devices, you can create a larger database than would fit on a single device.

By default, the Search Engine sets up the database to use only one directory. The command-line interface allows you to perform two kinds of manipulations on the database partitions:

- Adding New Partitions
- Moving Partitions

The Search Engine does not perform any checking to ensure that individual partitions have space remaining. It is your responsibility to maintain adequate free space for the database.

You can add new database partitions up to a maximum of 15 total partitions.

NOTE Once you increase the number of partitions, you will need to delete the entire database if you later want to reduce the number again.

You can change the physical location of any of your database partitions by specifying the name of the new location. Similarly, you can rename an existing partition. Use the `rdmgr` command to manipulate the partitions. See Chapter 14, "Command-Line Utilities"," for information on the `rdmgr` command.

Administering the Database Taxonomy

The following are some configuration and maintenance tasks you may need to perform to administer the database taxonomy:

- Configuring Categories
- Defining Classification Rules

Configuring Categories

Using the Sun ONE Identity Server administration console you can perform the following procedures to configure the database taxonomy:

- To Create a Child Category (a sub-category of a parent category)
- To Create a Sibling Category (categories at the same level)
- To Update a Category
- To Delete a Category

To Create a Child Category

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Categories then Category Editor from the menu bar.
5. Select a category in which to create a child category.

   If you have not previously defined any categories, only the root category titled "Search" is listed. Click the lower Search link to expand the root category.
6. In the Name field, specify a name for the category.
7. In the Description field, specify a description for the category (optional).
8. Click Add as a Child to create the category.
9. Click Save.

NOTE The Category Editor has a go-to list that appears whenever the list of visible categories spans multiple pages. Use the the page-up and page-down buttons to scroll up or down one page from the current page. Use the go-to button to access more than one page.

To Create a Sibling Category

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Categories then Category Editor from the menu bar.
5. Select a category for which to create a sibling.
6. In the Name field, specify a name for the category.

7. In the Description field, specify a description for the category (optional).
8. Click Add as a Sibling to create the category.
9. Click Save.

NOTE The Category Editor has a go-to list that appears whenever the list of visible categories spans multiple pages. Use the the page-up and page-down buttons to scroll up or down one page from the current page. Use the go-to button to access more than one page.

To Update a Category

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Categories then Category Editor from the menu bar.
5. Select a category to update.
6. To change the name of the category, specify a new name for the category in the Name field.
7. To change the description of the category, specify a description for the category In the Description field.
8. Click Update.
9. Click Save.

NOTE The Category Editor has a go-to list that appears whenever the list of visible categories spans multiple pages. Use the the page-up and page-down buttons to scroll up or down one page from the current page. Use the go-to button to access more than one page.

To Delete a Category

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Categories then Category Editor from the menu bar.
5. Select the category to delete.

   When a category is deleted, all its child categories will also be deleted.
6. Click Delete.
7. Click Save.

NOTE The Category Editor has a go-to list that appears whenever the list of visible categories spans multiple pages. Use the the page-up and page-down buttons to scroll up or down one page from the current page. Use the go-to button to access more than one page.

Defining Classification Rules

A classification rule is a simple conditional statement. Its form is "if <some condition> is true, assign the resource to <a category>".

To Define a Classification Rule

1. Log in to the Sun ONE Identity Server administration console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.
2. Choose Service Configuration in the location pane.
3. Click the properties arrow next to Search in the navigation pane.
4. Select Categories then Classification Rules Editor from the menu bar.
5. f you are creating a new rule, click New.

6. If you are editing an existing rule, select the rule.
7. Click the element type or attribute to use to classify the resource from the drop-down menu.
8. Click the comparison test in the drop-down menu.

   Comparison tests available are is, contains, begins with, ends with, or regular expression.
9. Define a text string to compare.
10. Click the category in which to classify the resource if the comparison is true.
11. Click save.

Chapter 9

Administering the Search Engine Robot

This chapter describes the Sun™ ONE Portal Server Search Engine robot and its corresponding configuration files. The following topics are discussed:

Search Engine Robot Overview

A Search Engine robot is an agent that identifies and reports on resources in its domains. It does so by using two kinds of filters: an enumerator filter and a generator filter.

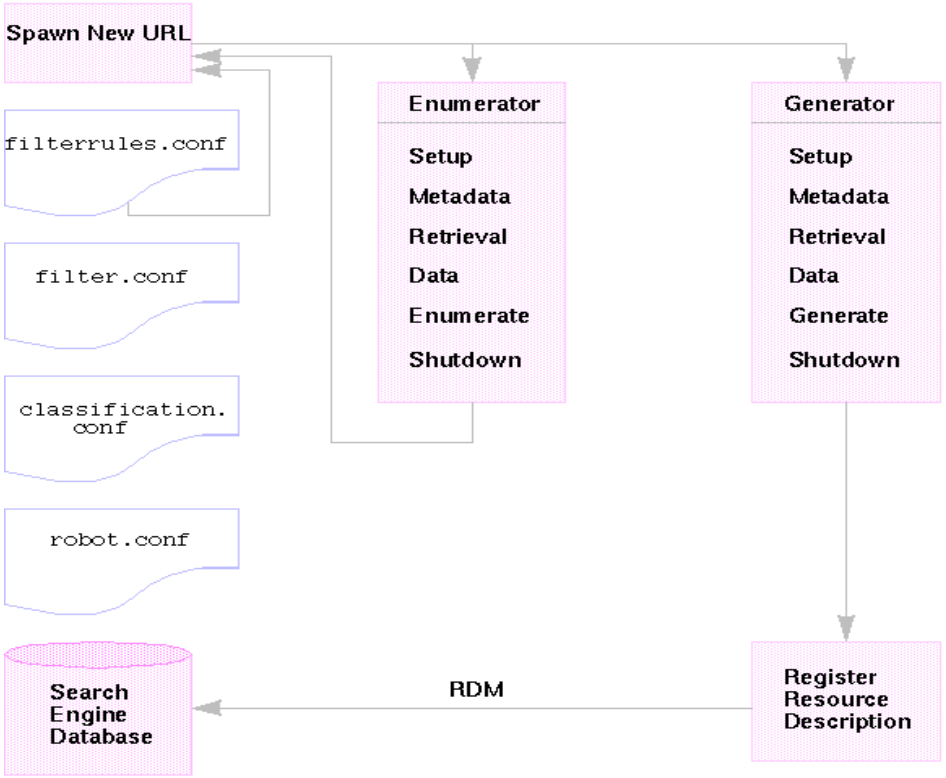
The enumerator filter locates resources by using network protocols. It tests each resource, and, if it meets the proper criteria, it is enumerated. For example, the enumerator filter can extract hypertext links from an HTML file and use the links to find additional resources.

The generator filter tests each resource to determine if a resource description (RD) should be created. If the resource passes the test, the generator creates an RD which is stored in the Search Engine database.

How the Robot Works

Figure 9-1 illustrates how the Search Engine robot works. In Figure 9-1, the robot examines URLs and their associated network resources. Each resource is tested by both the enumerator and the generator. If the resource passes the enumeration test, the robot checks it for additional URLs. If the resource passes the generator test, the robot generates a resource description that is stored in the Search Engine database.

Figure 9-1 How the Robot Works



Robot Configuration Files

Robot configuration files define the behavior of the Search Engine robots. These files reside in the directory `/var/opt/SUNWps/http-`*hostname-domain*`/portal/config`. Table 9-1 provides a description for each of the robot configuration files. The table contains two columns. The first column lists configuration file and the second column describes contents of the file.

Table 9-1 Robot Configuration Files

Robot Configuration File	Description
classification.conf	Contains rules used to classify RDs generated by the robot.
`filter.conf`	Contains all the filters available to the Search Engine robot for enumeration and generation. Including the same filtering rules for both the enumeration and generation filters ensures that a single rule change can be made to both types of filters. By reference, this file also includes the filtering rules stored in `filterrules.conf`.
`filterrules.conf`	Contains the starting points (also referred to as seed URLS) and filtering rules.
`robot.conf`	Defines most of the operating parameters for the robot. In addition, this file points the robot to applicable filters in the file `filter.conf`.

NOTE The Search service uses two other configuration files: `convert.conf` and `import.conf`. These files are generated by the Search server and in general should not be manually edited

Because you can set most parameters by using the Search Engine Administration Interface, you typically do not need to edit the `robot.conf` file.

However, advanced users might manually edit this file in order to set parameters that cannot be set through the interface.

Setting Robot Process Parameters

The file `robot.conf` defines many options for the robot, including pointing the robot to the appropriate filters in `filter.conf`. (For backwards-compatibility with older versions, `robot.conf` can also contain the seed URLs.)

The iPlanet™ Directory Server Access Management Edition administration console is used to edit the file `robot.conf`. Note that the few parameters you might manually edit by hand are described in detail in the "User-Modifiable Parameters" on page 253"section.

The most important parameters are `enumeration-filter` and `generation-filter`, which determine the filters the robot uses for enumeration and generation. The default values for these are `enumeration-default` and `generation-default`, which are the names of the filters provided by default in the `filter.conf` file.

All filters must be defined in the file `filter.conf`. If you define your own filters in `filter.conf`, you must add any necessary parameters to `robot.conf`.

For example, if you define a new enumeration filter named `my-enumerator`, you would add the following parameter to `robot.conf`:

```
enumeration-filter=my-enumerator
```

The Filtering Process

The robot uses filters to determine which resources to process and how to process them. When the robot discovers references to resources as well as the resources themselves, it applies filters to each resource in order to enumerate it and to determine whether or not to generate a resource description to store in the Search Engine database.

The robot examines one or more seed URLs, applies the filters, and then applies the filters to the URLs spawned by enumerating the seed URLs, and so on. The seed URLs are defined in the `filterrules.conf` file.

A filter performs any required initialization operations and applies comparison tests to the current resource. The goal of each test is to either allow or deny the resource. A filter also has a shutdown phase during which it performs any required cleanup operations.

If a resource is allowed, that means that it is allowed to continue passage through the filter. If a resource is denied, then the resource is rejected. No further action is taken by the filter for resources that are denied. If a resource is not denied, the robot will eventually enumerate it, attempting to discover further resources. The generator might also create a resource description for it.

These operations are not necessarily linked. Some resources result in enumeration; others result in RD generation. Many resources result in both enumeration and RD generation. For example, if the resource is an FTP directory, the resource typically will not have an RD generated for it. However, the robot might enumerate the individual files in the FTP directory. An HTML document that contains links to other documents can receive an RD and can lead to enumeration of the linked documents as well.

The following sections detail the filter process:

- Stages in the Filter Process
- Filter Syntax
- Filter Directives
- Writing or Modifying a Filter

Stages in the Filter Process

Both enumerator and generator filters have five phases in the filtering process. They both have four common phases: Setup—Performs initialization operations. Occurs only once in the life of the robot., Metadata—Filters the resource based on metadata that is available about the resource. Metadata filtering occurs once per resource before the resource is retrieved over the network. Table 9-2 lists examples of common metadata types. The table contains three columns. The first column lists the metadata type, the second column provides a description, and the third column provides an example.,Data—Filters the resource based on its data. Data filtering is done once per resource after it is retrieved over the network. Data that can be used for filtering include:, and Shutdown—Performs any needed termination operations. Occurs once in the life of the robot. If the resource makes it past the Data phase, it is either in the Enumerate—Enumerates the current resource in order to determine if it points to other resources to be examined. or Generate—Generates a resource description (RD) for the resource and saves it in the Search Engine database. phase, depending on whether the filter is an enumerator or a generator.

The phases are as follows:

- Setup—Performs initialization operations. Occurs only once in the life of the robot.
- Metadata—Filters the resource based on metadata that is available about the resource. Metadata filtering occurs once per resource before the resource is retrieved over the network. Table 9-2 lists examples of common metadata types. The table contains three columns. The first column lists the metadata type, the second column provides a description, and the third column provides an example.

Table 9-2 Common Metadata Types

Metadata	Description	Example
Complete URL	The location of a resource	`http://home.siroe.com/`
Protocol	The access portion of the URL	http, ftp, file
Host	The address portion of the URL	`www.siroe.com`
IP address	Numeric version of the host	198.95.249.6
PATH	The path portion of the URL	`/index.html`
Depth	Number of links from the seed URL	5

- Data—Filters the resource based on its data. Data filtering is done once per resource after it is retrieved over the network. Data that can be used for filtering include:
  - content-type
  - content-length
  - content-encoding
  - content-charset
  - last-modified
  - expires
- Enumerate—Enumerates the current resource in order to determine if it points to other resources to be examined.
- Generate—Generates a resource description (RD) for the resource and saves it in the Search Engine database.
- Shutdown—Performs any needed termination operations. Occurs once in the life of the robot.

Filter Syntax

The `filter.conf` file contains definitions for enumeration and generation filters. This file can contain multiple filters for both enumeration and generation. Note that the robot can determine which filters to use because they are specified by the `enumeration-filter` and `generation-filter` parameters in the file `robot.conf`.

Filter definitions have a well-defined structure: a header, a body, and an end. The header identifies the beginning of the filter and declares its name, for example:

```
<Filter name="myFilter">
```

The body consists of a series of *filter directives* that define the filter's behavior during setup, testing, enumeration or generation, and shutdown. Each directive specifies a function, and if applicable, parameters for the function.

The end is marked by `</Filter>`.

Code Example 9-1 on page 251 shows a filter named `enumeration1`

Code Example 9-1 Enumeration File Syntax

```
<Filter name="enumeration1>
   Setup fn=filterrules-setup config=./config/filterrules.conf
#  Process the rules
   MetaData fn=filterrules-process
#  Filter by type and process rules again
   Data fn=assign-source dst=type src=content-type
   Data fn=filterrules-process
#  Perform the enumeration on HTML only
   Enumerate enable=true fn=enumerate-urls max=1024
type=text/html
#  Cleanup
   Shutdown fn=filterrules-shutdown
</Filter>
```

Filter Directives

Filter directives use Robot Application Functions (RAFs) to perform operations. Their use and flow of execution is similar to that of NSAPI directives and Server Application Functions (SAFs) in the file `obj.conf`. Like NSAPI and SAF, data are stored and transferred using parameter blocks, also called *pblocks*.

There are six robot directives, or RAF classes, corresponding to the filtering phases and operations listed in "The Filtering Process" on page 248:

- Setup
- Metadata
- Data
- Enumerate
- Generate
- Shutdown

Each directive has its own robot application functions. For example, use filtering functions with the Metadata and Data directives, enumeration functions with the Enumerate directive, generation functions with the Generate directive, and so on.

The built-in robot application functions, as well as instructions for writing your own robot application functions, are explained in the *Sun ONE Portal Server 6.1 Developer's Guide*.

Writing or Modifying a Filter

In most cases, you should not need to write filters from scratch. You can create most of your filters using the administration console. You can then modify the `filter.conf` and `filterrules.conf` files to make any desired changes. These files reside in the directory `/var/opt/SUNWps/http-`*hostname-domain*`/portal`.

However, if you want to create a more complex set of parameters, you will need to edit the configuration files used by the robot.

Note the following points when writing or modifying a filter:

- The order of execution of directives (especially the available information at each phase)
- The order of rules

For a discussion of the parameters you can modify in the file `robot.conf`, the robot application functions that you can use in the file `filter.conf`, and how to create your own robot application functions, see the *Sun ONE Portal Server 6.1 Developer's Guide*.

User-Modifiable Parameters

The `robot.conf` file defines many options for the robot, including pointing the robot to the appropriate filters in `filter.conf`. For backwards-compatibility with older versions, `robot.conf` can also contain the seed URLs.

Because you can set most parameters by using the administration console, you typically do not need to edit the `robot.conf` file. However, advanced users might manually edit this file in order to set parameters that cannot be set through the administration console. See "Sample robot.conf File" on page 260 for an example of this file.

Table 9-3 on page 253 lists the user-modifiable parameters in the `robot.conf` file. The first column of the table lists the parameter, the second column provides a description of the parameter, and the third column provides an example.

Table 9-3 User-Modifiable Parameters

Parameter	Description	Example
auto-proxy	Specifies the proxy setting for the robot. It can be a proxy server or a JavaScript file for automatically configuring the proxy. For more information see, the *Sun ONE Portal Server 6.2 Administrator's Guide*.	`auto-proxy="http://proxy_server/proxy.pac"`
bindir	Specifies whether the robot will add a bind directory to the `PATH` environment. This is an extra `PATH` for users to run an external program in a robot, such as those specified by `cmd-hook` parameter.	`bindir=path`

Table 9-3 User-Modifiable Parameters

Parameter	Description	Example
cmd-hook	Specifies an external completion script to run after the robot completes one run. This must be a full path to the command name. The robot will execute this script from the `/var/opt/SUNWps/` directory. There is no default. There must be at least one RD registered for the command to run. For information about writing completion scripts, see the *Sun ONE Portal Server 6.1 Developer's Guide*.	`cmd-hook="command-string"` There is no default.
command-port	Specifies the socket that the robot listens to in order to accept commands from other programs, such as the Administration Interface or robot control panels. For security reasons, the robot can accept commands only from the local host unless `remote-access` is set to `yes`.	command-port=port_number
connect-timeout	Specifies the maximum time allowed for a network to respond to a connection request. The default is 120 seconds.	command-timeout=seconds
convert-timeout	Specifies the maximum time allowed for document conversion. The default is 600 seconds.	convert-timeout=seconds

Table 9-3 User-Modifiable Parameters

Parameter	Description	Example
depth	Specifies the number of links from the seed URLs (also referred to as starting point) that the robot will examine. This parameter sets the default value for any seed URLs that do not specify a depth. The default is 10. A value of negative one (depth=-1) indicates that the link depth is infinite.	`depth=integer`
email	Specifies the email address of the person who runs the robot. The email address is sent with the user-agent in the HTTP request header, so that Web managers can contact the people who run robots at their sites. The default is *user@domain*.	`email=user@hostname`
enable-ip	Generates an IP address for the URL for each RD that is created. The default is *true*.	`enable-ip=[true | yes | false | no]`
enable-rdm-probe	Determines if the server supports RDM, the robot decides whether to query each server it encounters by using this parameter. If the server supports RDM, the robot will not attempt to enumerate the server's resources, since that server is able to act as its own resource description server. The default is *false*.	`enable-rdm-probe=[true | false | yes | no]`
enable-robots-txt	Determines if the robot should check the `robots.txt` file at each site it visits, if available. The default is yes.	`enable-robots-txt=[true | false | yes | no]`

Table 9-3 User-Modifiable Parameters

Parameter	Description	Example
engine-concurrent	Specifies the number of pre-created threads for the robot to use. The default is 10. This parameter cannot be set interactively through the administration console.	`engine-concurrent=[1..100]`
enumeration-filter	Specifies the enumeration filter that is used by the robot to determine if a resource should be enumerated. The value must be the name of a filter defined in the file `filter.conf`. The default is `enumeration-default`. This parameter cannot be set interactively through the administration console.	`enumeration-filter=enumfiltername`
generation-filter	Specifies the generation filter that is used by the robot to determine if a resource description should be generated for a resource. The value must be the name of a filter defined in the file `filter.conf`. The default is `generation-default`. This parameter cannot be set interactively through the administration console.	`generation-filter=genfiltername`
index-after-ngenerated	Specifies the number of minutes that the robot should collect RDs before batching them for the Search Engine. If you do not specify this parameter, it is set to 256 minutes.	`index-after-ngenerated=30`

Table 9-3 User-Modifiable Parameters

Parameter	Description	Example
loglevel	Specifies the levels of logging. The `loglevel` values are as follows: • Level 0: log nothing but serious errors • Level 1: also log RD generation (default) • Level 2: also log retrieval activity • Level 3: also log filtering activity • Level 4: also log spawning activity • Level 5: also log retrieval progress The default value is 1.	`loglevel=[0...100]`
max-connections	Specifies the maximum number of concurrent retrievals that a robot can make. The default is 8.	`max-connections=[1..100]`
max-filesize-kb	Specifies the maximum file size in kilobytes for files retrieved by the robot.	`max-filesize-kb=1024`
max-memory-per-url / max-memory	Specifies the maximum memory in bytes used by each URL. If the URL needs more memory, the RD is saved to disk. The default is 1. This parameter cannot be set interactively through the administration console.	`max-memory-per-url=n_bytes`

Table 9-3 User-Modifiable Parameters

Parameter	Description	Example
max-working	Specifies the size of the robot working set, which is the maximum number of URLs the robot can work on at one time. This parameter cannot be set interactively through the administration console.	`max-working=1024`
onCompletion	Determines what the robot does after it has completed a run. The robot can either go into idle mode, loop back and start again, or quit. The default is `idle`. This parameter works with the `cmd-hook` parameter. When the robot is done, it will do the action of `onCompletion` and then run the `cmd-hook` program.	`OnCompletion=[idle | loop | quit]`
password	Specifies the `password` is used for httpd authentication and ftp connection.	`password=string`
referer	Specifies the parameter sent in the HTTP request if it is set to identify the robot as the referer when accessing Web pages	`referer=string`
register-user and register-password	Specifies the user name used to register RDs to the Search Engine database. This parameter cannot be set interactively through the Search Engine Administration Interface.	`register-user=string`
register-password	Specifies the password used to register RDs to the Search Engine database. This parameter cannot be set interactively through the administration console.	`register-password=string`

Table 9-3 User-Modifiable Parameters

Parameter	Description	Example
remote-access	This parameter determines if the robot can accept commands from remote hosts. The default is `false`.	`remote-access=[true | false | yes | no]`
robot-state-dir	Specifies the directory where the robot saves its state. In this working directory, the robot can record the number of collected RDs and so on.	`robot-state-dir="/var/opt/SUNWps/instance/portal/robot"`
server-delay	Specifies the time period between two visits to the same web site, thus preventing the robot from accessing the same site too frequently.	`server-delay=delay_in_seconds`
site-max-connections	Indicates the maximum number of concurrent connections that a robot can make to any one site. The default is 2.	`site-max-connections=[1..100]`
smart-host-heuristics	Enables the robot to change sites that are rotating their DNS canonical host names. For example, `www123.siroe.com` is changed to `www.siroe.com`. The default is `false`.	`smart-host-heuristics=[true | false]`
tmpdir	Specifies a place for the robot to create temporary files. Use this value to set the environment variable `TMPDIR`.	`tmpdir=path`
user-agent	Specifies the parameter sent with the email address in the `http-request` to the server.	`user-agent=iPlanetRobot/4.0`
username	Specifies the user name of the user who runs the robot and is used for httpd authentication and ftp connection. The default is `anonymous`.	`username=string`

Sample robot.conf File

This section describes a sample `robot.conf` file. Any commented parameters in the sample use the default values shown. The first parameter, `csid`, indicates the Search Engine instance that uses this file; it is important not to change the value of the this parameter. See "User-Modifiable Parameters" on page 253 for definitions of the parameters in this file.

NOTE This sample file includes some parameters used by the Search Engine that you should not modify such as the `csid` parameter.

```
<Process csid="x-catalog://budgie.siroe.com:80/jack" \
   auto-proxy="http://sesta.varrius.com:80/"
   auto_serv="http://sesta.varrius.com:80/"
   command-port=21445
   convert-timeout=600
   depth="-1"
   # email="user@domain"
   enable-ip=true
   enumeration-filter="enumeration-default"
   generation-filter="generation-default"
   index-after-ngenerated=30
   loglevel=2
   max-concurrent=8
   site-max-concurrent=2
   onCompletion=idle
   password=boots
   proxy-loc=server
   proxy-type=auto
   robot-state-dir="/var/opt/SUNWps/https-budgie.siroe.com/ \
   ps/robot"
   server-delay=1
   smart-host-heuristics=true
   tmpdir="/var/opt/SUNWps/https-budgie.siroe.com/ps/tmp"
   user-agent="iPlanetRobot/4.0"
   username=jack
</Process>
```

Chapter 10

The Pre-defined Robot Application Functions

This chapter provides descriptions, parameter specifications, and examples of pre-defined Robot Application Functions (RAFs) in the Sun™ ONE Portal Server Search Engine. You can use these functions in the `filter.conf` file to create and modify filter definitions. The file `filter.conf` is located in the directory `/var/opt/SUNWps/http-`*hostname-domain*`/portal/config`.

The file `filter.conf` contains definitions for the enumeration and generation filters. Each of these filters invokes a set of rules which are stored in the file `filterrules.conf`. The filter definitions contain instructions that are specific to each filter while the filter rules contain the rules used by both filters.

To understand how filter rules are defined, examine the file `filterrules.conf`. Note that you typically need not manually edit this file since you create filter rules by using the administration console.

To see an example of filter definitions, you should examine the file `filter.conf`. You only need to edit the `filter.conf` file to modify the filters in a way that is not accommodated in the administration console, such as instructing the robot to enumerate some resources without generating resources for them.

This chapter contains the following sections:

- Sources and Destinations
- Setup Functions
- Filtering Functions
- Filtering Support Functions
- Enumeration Functions
- Generation Functions

- Shutdown Functions

Sources and Destinations

Most of the Robot Application Functions (RAFs) require sources of information and generate data that goes to destinations. The sources are defined within the robot itself and are not necessarily related to the fields in the resource description it ultimately generates. Destinations, on the other hand, are generally the names of fields in the resource description, as defined by the resource description server's schema.

For details on using the administration console to determine the database schema, see Chapter 8, "Administering the Search Engine Service."

The following sections describe the different stages of the filtering process, and the sources available at those stages.

Sources Available at the Setup Stage

At the Setup stage, the filter is set up and cannot yet get information about the resource's URL or content.

Sources Available at the MetaData Filtering Stage

At the MetaData stage, the robot encounters a URL for a resource, but it has not downloaded the resource's content, thus information is available about the URL as well as data that is derived from other sources such as the `filter.conf` file. At this stage, however, information is not available about the content of the resource.

Table 10-1 lists the sources available to the RAFs at the MetaData phase. The table contains three columns. The first column lists the source, the second column provides a description, and the third column provides an example.

Table 10-1 Sources Available to the RAFs at the MetaData Phase

Source	Description	Example
csid	Catalog Server ID	`x-catalog//budgie.siroe.com:8086/alexandria`

Table 10-1 Sources Available to the RAFs at the MetaData Phase *(Continued)*

Source	Description	Example
depth	Number of links traversed from starting point	`10`
enumeration filter	Name of Enumeration filter	`enumeration1`
generation filter	Name of Generation filter	`generation1`
host	Host portion of URL	`home.siroe.com`
IP	Numeric version of host	`198.95.249.6`
protocol	Access portion of the URL	`http, https, ftp, file`
path	Path portion of the URL	`/, /index.html, /documents/listing.html`
URL	Complete URL	`http://developer.siroe.com/docs/manuals/`

Sources Available at the Data Stage

At the Data stage, the robot has downloaded the content of the resource at the URL, and can access data about the content, such as the description, the author, and so on.

If the resource is an HTML file, the Robot parses the `<META>` tags in the HTML headers. Consequently, any data contained in `<META>` tags is available at the Data stage.

During the data phase, the following sources are available to RAFs, in addition to those available during the MetaData phase. The table contains three columns. The first column lists the source, the second column provides a description, and the third column provides an example.

Table 10-2 Sources Available to the RAFs at the Data Phase

Source	Description	Example
`content-charset`	Character set used by the resource	
`content-encoding`	Any form of encoding	
`content-length`	Size of the resource in bytes	

Table 10-2 Sources Available to the RAFs at the Data Phase *(Continued)*

Source	Description	Example
`content-type`	MIME type of the resource	`text/html`, `image/jpeg`
`expires`	Date the resource itself expires	
`last-modified`	Date the resource was last modified	
`data in <META> tags`	Any data that is provided in `<META>` tags in the header of HTML resources	Author Description Keywords

All these sources (except for the data in `<META>` tags) are derived from the HTTP response header returned when retrieving the resource.

Sources Available at the Enumeration, Generation, and Shutdown Stages

At the Enumeration and Generation stages, the same data sources are available as the Data stage.

At the Shutdown stage, the filter completes its filtering and is shuts down. Although functions written for this stage can use the same data sources as those available at the Data stage, the shutdown functions typically restrict their operations to shutdown and cleanup activities.

Enable Parameter

Each function can have an enable parameter. The values can be `true`, `false`, `on`, or `off`. The administration console uses these parameters to turn certain directives on or off.

The following example enables enumeration for `text/html` and disables enumeration for `text/plain`:

```
#  Perform the enumeration on HTML only
Enumerate enable=true fn=enumerate-urls max=1024 type=text/html
Enumerate enable=false fn=enumerate-urls-from-text max=1024
type=text/plain
```

Adding an `enable=false` parameter or an `enable=off` parameter has the same effect as commenting the line. Because the administration console does not write comments, it writes an `enable` parameter instead.

Setup Functions

This section describes the functions that are used during the setup phase by both enumeration and generation filters. The following functions are described:

filterrules-setup

When you use the `filterrules-setup` function, `logtype` is the type of log file to use. The value can be `verbose`, `normal`, or `terse`.

Parameters

Table 10-3 lists the parameter used with the `filterrules-setup` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-3 `filterrules-setup` Parameters

Parameter	Description
`config`	Path name to the file containing the filter rules to be used by this filter.

Example

```
Setup fn=filterrules-setup config=./config/filterrules.conf
logtype=normal
```

setup-regex-cache

The `setup-regex-cache` function initializes the cache size for the `filter-by-regex` and `generate-by-regex` functions. Use this function to specify a number other than the default of 32.

Parameters

Table 10-4 lists the parameter used with the `setup-regex-cache` function. The table contains three columns. The first column lists the parameter, the second column provides a description, and the third column provides an example.

Table 10-4 `setup-regex-cache` Parameter

Parameter	Description
`cache-size`	Maximum number of compiled regular expressions to be kept in the regex cache.

Example

```
Setup fn=setup-regex-cache cache-size=28
```

setup-type-by-extension

The `setup-type-by-extension` function configures the filter to recognize file name extensions. It must be called before the `assign-type-by-extension` function can be used. The file specified as a parameter must contain mappings between standard MIME content types and file extension strings.

Parameters

Table 10-5 lists the parameter used with the `setup-type-by-extension` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-5 `setup-type-by-extension` Parameter

Parameter	Description
`file`	Name of the MIME types configuration file.

Example

```
Setup fn=setup-type-by-extension file=./config/mime.types
```

Filtering Functions

The following functions operate at the Metadata and Data stages to allow or deny resources based on specific criteria specified by the function and its parameters.

These functions can be used in both Enumeration and Generation filters in the file `filter.conf`.

Each “filter-by” function performs a comparison, then either allows or denies the resource. Allowing the resource means that processing continues to the next filtering step. Denying the resource means that processing should stop, because the resource does not meet the criteria for further enumeration or generation. The following functions are described:

- filter-by-exact
- filter-by-max
- filter-by-md5
- filter-by-prefix
- filter-by-regex
- filterrules-process

filter-by-exact

The `filter-by-exact` function allows or denies the resource if the allow / deny string matches the source of information exactly. The keyword `all` matches any string.

Parameters

Table 10-6 lists the parameters used with the `filter-by-exact` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-6 `filter-by-exact` Parameter

Parameter	Description
`src`	Source of information.
`allow/deny`	Contains a string.

Example

The following example filters out all resources whose content-type is `text/plain`. It allows all other resources to proceed:

```
Data fn=filter-by-exact src=type deny=text/plain
```

filter-by-max

The `filter-by-max` function allows the resource if the specified information source is less than or equal to the given value. It denies the resource if the information source is greater than the specified value.

This function can be called no more than once per filter.

Parameters

Table 10-7 lists the parameters used with the `filter-by-max` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-7 `filter-by-max` Parameters

Parameter	Description
`src`	Source of information. It must be one of the following: hosts, objects, or depth.
`value`	Specifies a value for comparison.

Example

This example allows resources whose content-length is less than 1024 K:

```
MetaData fn-filter-by-max src=content-length value=1024
```

filter-by-md5

The `filter-by-md5` function only allows the first resource with a given MD5 checksum value. If the current resource's MD5 has been seen in an earlier resource by this robot, the current resource is denied. As a result, duplication of identical resources or single resources with multiple URLs is prevented.

You can only call this function at the Data stage or later. It can be called no more than once per filter. The filter must invoke the `generate-md5` function to generate an MD5 checksum before invoking `filter-by-md5`.

Parameters

none

Example

The following example shows the typical method of handling MD5 checksums by first generating the checksum and then filtering based on it:

```
Data fn=generate-md5
Data fn=filter-by-md5
```

filter-by-prefix

The `filter-by-prefix` function allows or denies the resource if the given information source begins with the specified prefix string. The resource doesn't have to match completely. The keyword `all` matches any string.

Parameters

Table 10-8 lists the parameters used with the `filter-by-prefix` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-8 `filter-by-prefix` Parameters

Parameter	Description
`src`	Source of information.
`allow/deny`	Contains a string for prefix comparison.

Example

The following example allows resources whose content-type is any kind of text, including `text/html` and `text/plain`:

```
MetaData fn=filter-by-prefix src=type allow=text
```

filter-by-regex

The `filter-by-regex` function supports regular expression pattern matching. It allows resources that match the given regular expression. The supported regular expression syntax is defined by the `POSIX.1` specification. The regular expression `\\*` matches anything.

Parameters

Table 10-9 lists the parameters used with the `filter-by-regex` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-9 `filter-by-regex` Parameters

Parameter	Description
`src`	Source of information.
`allow/deny`	Contains a regular expression string.

Example

The following example denies all resources from sites in the government domain:

```
MetaData fn=filter-by-regex src=host deny=\\*.gov
```

filterrules-process

The `filterrules-process` function handles in the rules in the `filterrules.conf` file.

Parameters

none

Example

```
MetaData fn=filterrules-process
```

Filtering Support Functions

The following functions are used during filtering to manipulate or generate information on the resource. The robot can then process the resource by calling filtering functions. These functions can be used in Enumeration and Generation filters in the file `filter.conf`. The following functions are described:

- assign-source
- assign-type-by-extension
- clear-source
- convert-to-html

- copy-attribute
- generate-by-exact
- generate-by-prefix
- generate-by-regex
- generate-md5
- generate-rd-expires
- generate-rd-last-modified
- rename-attribute

assign-source

The `assign-source` function assigns a new value to a given information source. This permits editing during the filtering process. The function can assign an explicit new value, or it can copy a value from another information source.

Parameters

Table 10-10 lists the parameters used with the `assign-source` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-10 `assign-source` Parameters

Parameter	Description
`dst`	Name of the source whose value is to be changed.
`value`	Specifies an explicit value.
src	Information source to copy to `dst`

You must specify either a `value` parameter or a `src` parameter, but not both.

Example

```
Data fn=assign-source dst=type src=content-type
```

assign-type-by-extension

The `assign-type-by-extension` function uses the resource's file name to determine its type and assigns this type to the resource for further processing.

The `setup-type-by-extension` function must be called during setup before `assign-type-by-extension` can be used.

Parameters

Table 10-11 lists the parameter used with the `assign-type-by-extension` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-11 `assign-type-by-extension` Parameter

Parameter	Description
`src`	Source of file name to compare. If you do not specify a source, the default is the resource's path.

Example

```
MetaData fn=assign-type-by-extension
```

clear-source

The clear-source function deletes the specified data source. You typically do not need to perform this function. You can create or replace a source by using the `assign-source`.

Parameters

Table 10-12 lists the parameter used with the `clear-source` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-12 `clear-source` Parameter

Parameter	Description
`src`	Name of source to delete.

Example

The following example deletes the path source:

```
MetaData fn=clear-source src=path
```

convert-to-html

The convert-to-html function converts the current resource into an HTML file for further processing, if its type matches a specified MIME type. The conversion filter automatically detects the type of the file it is converting.

Parameters

Table 10-13 lists the parameter used with the `convert-to-html` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-13 `convert-to-html` Parameter

Parameter	Description
`type`	MIME type from which to convert.

Example

The following sequence of function calls causes the filter to convert all Adobe Acrobat PDF files, Microsoft RTF files, and FrameMaker MIF files to HTML, as well as any files whose type was not specified by the server that delivered it.

```
Data fn=convert-to-html type=application/pdf
Data fn=convert-to-html type=application/rtf
Data fn=convert-to-html type=application/x-mif
Data fn=convert-to-html type=unknown
```

copy-attribute

The `copy-attribute` function copies the value from one field in the resource description into another.

Parameters

Table 10-14 lists the parameters used with the `copy-attribute` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-14 `copy-attribute` Parameters

Parameter	Description
`src`	Field in the resource description from which to copy.

Table 10-14 `copy-attribute` Parameters

Parameter	Description
`dst`	Item in the resource description into which to copy the source.
`truncate`	Maximum length of the source to copy.
`clean`	Boolean parameter indicating whether to fix truncated text (such as not leaving partial words). This parameter is `false` by default.

Example

```
Generate fn=copy-attribute \
  src=partial-text dst=description truncate=200 clean=true
```

generate-by-exact

The `generate-by-exact` function generates a source with a specified value, but only if an existing source exactly matches another value.

Parameters

Table 10-15 lists the parameters used with the `generate-by-exact` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-15 `generate-by-exact` Parameter

Parameter	Description
`dst`	Name of source to generate.
`value`	Value to assign `dst`.
`src`	Source against which to match.

Example

The following example sets the classification to Siroe if the host is `www.siroe.com`.

```
Generate fn="generate-by-exact" match="www.siroe.com:80" src="host"
value="Siroe" dst="classification"
```

generate-by-prefix

This `generate-by-prefix` function generates a source with a specified value, but only if the prefix of an existing source matches another value.

Parameters

Table 10-16 lists the parameters used with the `generate-by-prefix` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-16 `generate-by-prefix` Parameters

Parameter	Description
`dst`	Name of the source to generate.
`value`	Value to assign to `dst`.
`src`	Source against which to match.
`match`	Value to compare to `src`.

Example

The following example sets the classification to Compass if the protocol prefix is HTTP:

```
Generate fn="generate-by-prefix" match="http" src="protocol"
value="World Wide Web" dst="classification"
```

generate-by-regex

The `generate-by-regex` function generates a source with a specified value, but only if an existing source matches a regular expression.

Parameters

Table 10-17 lists the parameters used with the `generate-by-regex` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-17 `generate-by-regex` Parameters

Parameter	Description
`dst`	Name of the source to generate.
`value`	Value to assign to `dst`.

Table 10-17 `generate-by-regex` Parameters

Parameter	Description
`src`	Source against which to match.
`match`	Regular expression string to compare to `src`.

Example

The following example sets the classification to Siroe if the host name matches the regular expression `*.siroe.com`. For example, resources at both `developer.siroe.com` and `home.siroe.com` will be classified as `Siroe`:

```
Generate fn="generate-by-regex" match="\\*.siroe.com" src="host"
value="Siroe" dst="classification"
```

generate-md5

The `generate-md5` function generates an MD5 checksum and adds it to the resource. You can then use the `filter-by-md5` function to deny resources with duplicate MD5 checksums.

Parameters

none

Example

```
Data fn=generate-md5
```

generate-rd-expires

The `generate-rd-expires` function generates an expiration date and adds it to the specified source. The function uses metadata such as the HTTP header and HTML `<META>` tags to obtain any expiration data from the resource. If none exists, it generates an expiration date three months from the current date.

Parameters

Table 10-18 lists the parameter used with the `generate-rd-expires` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-18 `generate-rd-expires` Parameters

Parameter	Description
`dst`	Name of the source. If you omit it, it defaults to `rd-expires`.

Example

```
Generate fn=generate-rd-expires
```

generate-rd-last-modified

The `generate-rd-last-modified` function adds the current time to the specified source.

Parameters

Table 10-19 lists the parameter used with the `generate-rd-last-modified` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-19 `generate-rd-last-modified` Parameter

Parameter	Description
`dst`	Name of the source. If you omit it, it defaults to `rd-last-modified`.

Example

```
Generate fn=generate-last-modified
```

rename-attribute

The `rename-attribute` function changes the name of a field in the resource description. It is most useful in cases where, for example, `extract-html-meta` copies information from a `<META>` tag into a field, and you want to change the name of the field.

Parameters

Table 10-20 lists the parameter used with the `generate-rd-last-modified` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-20 `generate-rd-last-modified` Parameter

Parameter	Description
`src`	String containing a mapping from one name to another.

Example

The following example renames an attribute from author to author-name:

```
Generate fn=rename-attribute src="author->author-name"
```

Enumeration Functions

The following functions operate at the Enumerate stage. These functions control if and how a robot gathers links from a given resource in order to use as starting points for further resource discovery. The following functions are described in this section:

- enumerate-urls
- enumerate-urls-from-text

enumerate-urls

The `enumerate-urls` function scans the resource and enumerates all URLs found in hypertext links. The results are used to spawn further resource discovery. You can specify a content-type to restrict the kind of URLs enumerated.

Parameters

Table 10-21 lists the parameters used with the `enumerate-urls` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-21 `enumerate-urls` Parameters

Parameter	Description
`max`	The maximum number of URLs to spawn from a given resource. The default, if max is omitted, is 1024.

Table 10-21 `enumerate-urls` Parameters

Parameter	Description
`type`	Content-type that restricts enumeration to those URLs that have the specified content-type. `type` is an optional parameter. If omitted, it will enumerate all URLs.

Example

The following example enumerates HTML URLs only, up to a maximum of 1024:

```
Enumerate fn=enumerate-urls type=text/html
```

enumerate-urls-from-text

The `enumerate-urls-from-text` function scans text resources, looking for strings matching this regular expression: `URL:.*`. It spawns robots to enumerate the URLs from these strings and generate further resource descriptions.

Parameters

Table 10-22 lists the parameter used with the `enumerate-urls-from-text` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-22 `enumerate-urls-from-text` Parameter

Parameter	Description
`max`	The maximum number of URLs to spawn from a given resource. The default, if max is omitted, is 1024.

Example

```
Enumerate fn=enumerate-urls-from-text
```

Generation Functions

The following functions are used in the Generate stage of filtering. Generation functions can generate information that goes into a resource description. In general, they either extract information from the body of the resource itself or copy information from the resource's metadata. The following functions are described in this section:

- extract-full-text
- extract-html-meta
- extract-html-text
- extract-html-toc
- extract-source
- harvest-summarizer

extract-full-text

The `extract-full-text` function extracts the complete text of the resource and adds it to the resource description.

NOTE The `extract-full-text` function should be used with caution, because it can significantly increase the size of the resource description, thus causing database bloat and overall negative impact on network bandwidth.

Parameters

Table 10-23 lists the parameters used with the `extract-full-text` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-23 `extract-full-text` Parameters

Parameter	Description
`truncate`	The maximum number of characters to extract from the resource.
`dst`	Name of the schema item that will receive the full text.

Example

```
Generate fn=extract-full-text
```

extract-html-meta

The `extract-html-meta` function extracts any `<META>` or `<TITLE>` information from an HTML file and adds it to the resource description. A content-type may be specified to restrict the kind of URLs that are generated.

Parameters

Table 10-24 lists the parameters used with the `extract-html-meta` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-24 `extract-html-meta` Parameters

Parameter	Description
`truncate`	The maximum number of bytes to extract.
`type`	Optional parameter. If omitted, it will generate all URLs.

Example

```
Generate fn=extract-html-meta truncate=255 type=text/html
```

extract-html-text

The `extract-html-text` function extracts the first few characters of text from an HTML file, excluding the HTML tags, and adds the text to the resource description. This permits the first part of a document's text to be included in the RD. A content-type may be specified to restrict the kind of URLs that are generated.

Parameters

Table 10-25 lists the parameters usedwith the `extract-html-text` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-25 `extract-html-text` Parameters

Parameter	Description
`truncate`	The maximum number of bytes to extract.
`skip-headings`	Set to `true` to ignore any HTML headers that occur in the document.
`type`	Optional parameter. If omitted, it will generate all URLs.

Example

```
Generate fn=extract-html-text truncate=255 type=text/html
skip-headings=true
```

extract-html-toc

The `extract-html-toc` function extracts the table-of-contents from the HTML headers and add it to the resource description.

Parameters

Table 10-26 lists the parameters used with the `extract-html-toc` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-26 `extract-html-toc` Parameters

Parameter	Description
`truncate`	The maximum number of bytes to extract.
`level`	Maximum HTML header level to extract. This parameter controls the depth of the table of contents.

Example

```
Generate fn=extract-html-toc truncate=255 level=3
```

extract-source

The `extract-source` function extracts the specified values from the given sources and adds them to the resource description.

Parameters

Table 10-27 lists the parameter used with the `extract-source` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-27 `extract-source` Parameter

Parameter	Description
`src`	List of source names; you can use the `->` operator to define a new name for the RD attribute, for example, `type->content-type` would take the value of the source named `type` and save it in the RD under the attribute named `content-type`.

Example

```
Generate fn=extract-source
src="md5,depth,rd-expires,rd-last-modified"
```

harvest-summarizer

The `harvest-summarizer` function runs a Harvest summarizer on the resource and adds the result to the resource description.

To run Harvest summarizers, you must have `$HARVEST_HOME/lib/gatherer` in your `path` before you run the robot.

Parameters

Table 10-28 lists the parameter used with the `harvest-summarizer` function. The table contains two columns. The first column lists the parameter, and the second column provides a description.

Table 10-28 `harvest-summarizer` Parameter

Parameter	Description
`summarizer`	Name of the summarizer program.

Example

```
Generate fn-harvest-summarizer summarizer=HTML.sum
```

Shutdown Functions

The following function can be used during the shutdown phase by both enumeration and generation functions.

filterrules-shutdown

After the rules are run, the `filterrules-shutdown` function performs clean up and shutdown responsibilities.

Parameters

none

Example

```
Shutdown fn=filterrules-shutdown
```

Chapter 11

Administering the Subscriptions Service

This chapter contains the following sections:

Overview

The Subscriptions service enables users to create a profile of interest covering many sources of information. In this release, the sources of information supported include categories, discussions, and searchable documents. The profile is updated with the latest information every time the user accesses the Subscriptions channel. The Subscriptions channel summarizes the number of hits (relevant information) that matches each profile entry the user defined for categorized document and/or discussions.

The Search service is used to:

- Match and count the number of new documents in a target category from a specified range of days
- Match and count the number of new relevant comments within a discussion from a specified range of days
- Match and count the number of document hits against saved searches

The result is displayed as a link that shows the number of matching information to the profile entry. This link redirects the user to a more detailed view of the match itself.

In case of a category subscription, the link redirects the user to the search channel where the specific documents of interest are summarized in a standard category search result format. The Subscriptions channel acts as the doorway to a more detailed view for the user.

Administering the Subscriptions Service

The administrator can enable or disable subscriptions service. The Subscriptions service can be administered at the:

Root Level

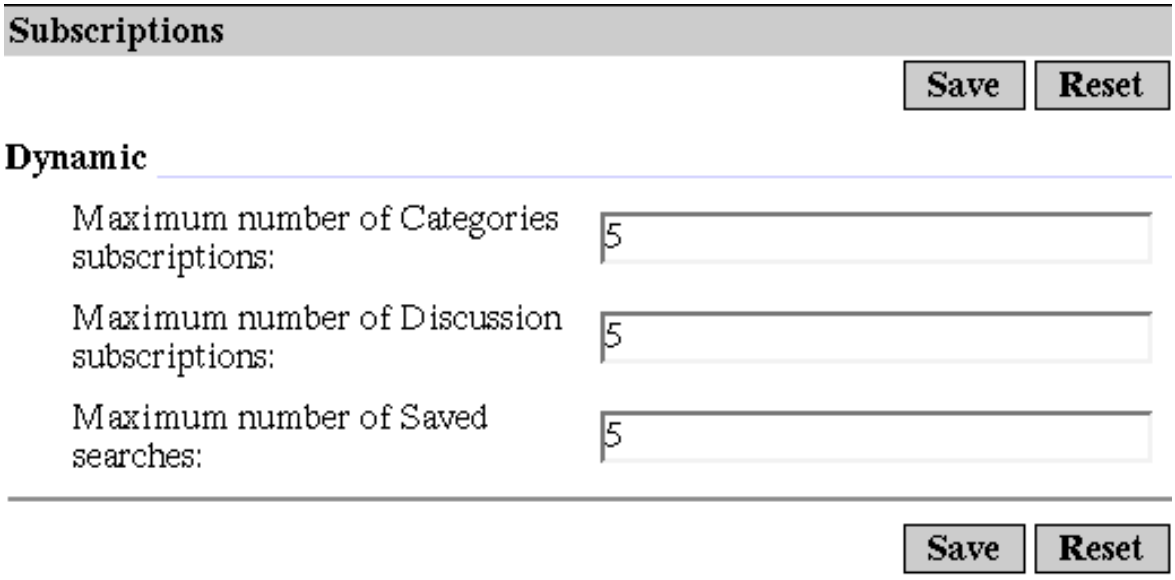
Organization level

Organization User level

Root Level

Administering the Subscriptions service at the Root level sets the system wide default maximum number of subscriptions per type (that is, for categories, discussions, and for saved searches). Figure 11-1 contains the interface for administering the Subscriptions service at the root level. See "To Define the Subscriptions Service at the Root Level" on page 290 for information on defining Subscriptions service at root level.

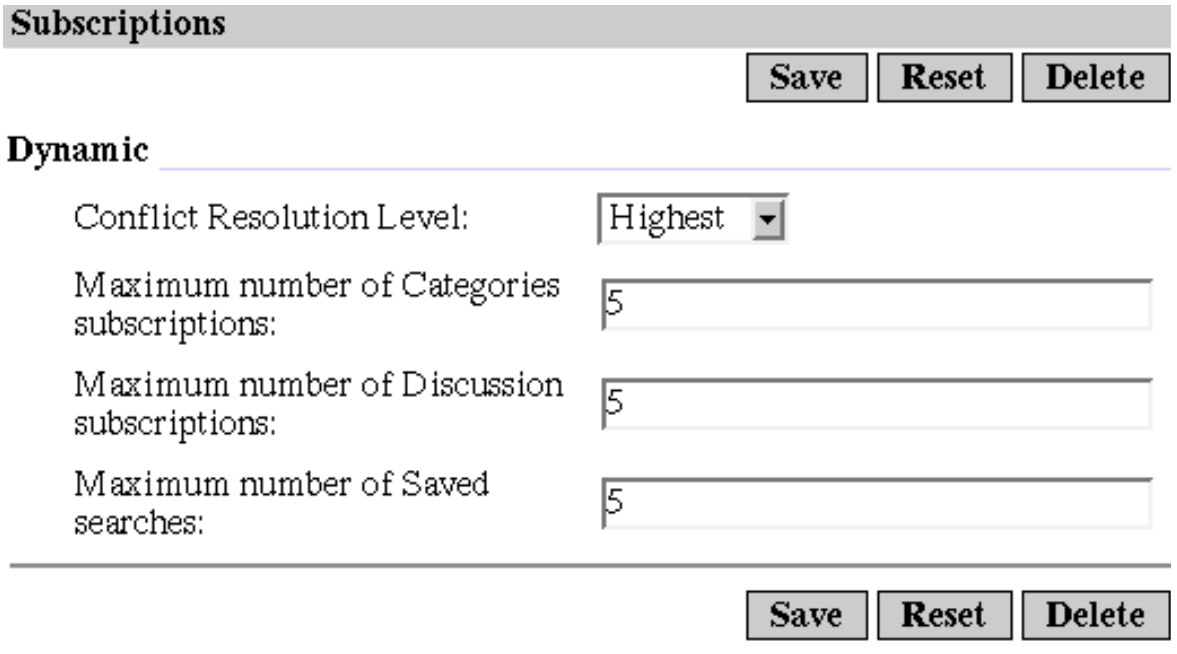
Figure 11-1 Root Level Subscriptions Administration Interface



Organization level

Administering the Subscriptions service at the Organization level overwrites the system wide default maximum number of subscription per type (that is, for categories, discussions, and for saved searches). Figure 11-2 on page 287 contains the interface for administering the Subscriptions service at the organization level. See "To Define the Subscriptions Service at the Organization Level" on page 290 for information on defining Subscriptions service at organization level.

Figure 11-2 Organization Level Subscriptions Administration Interface



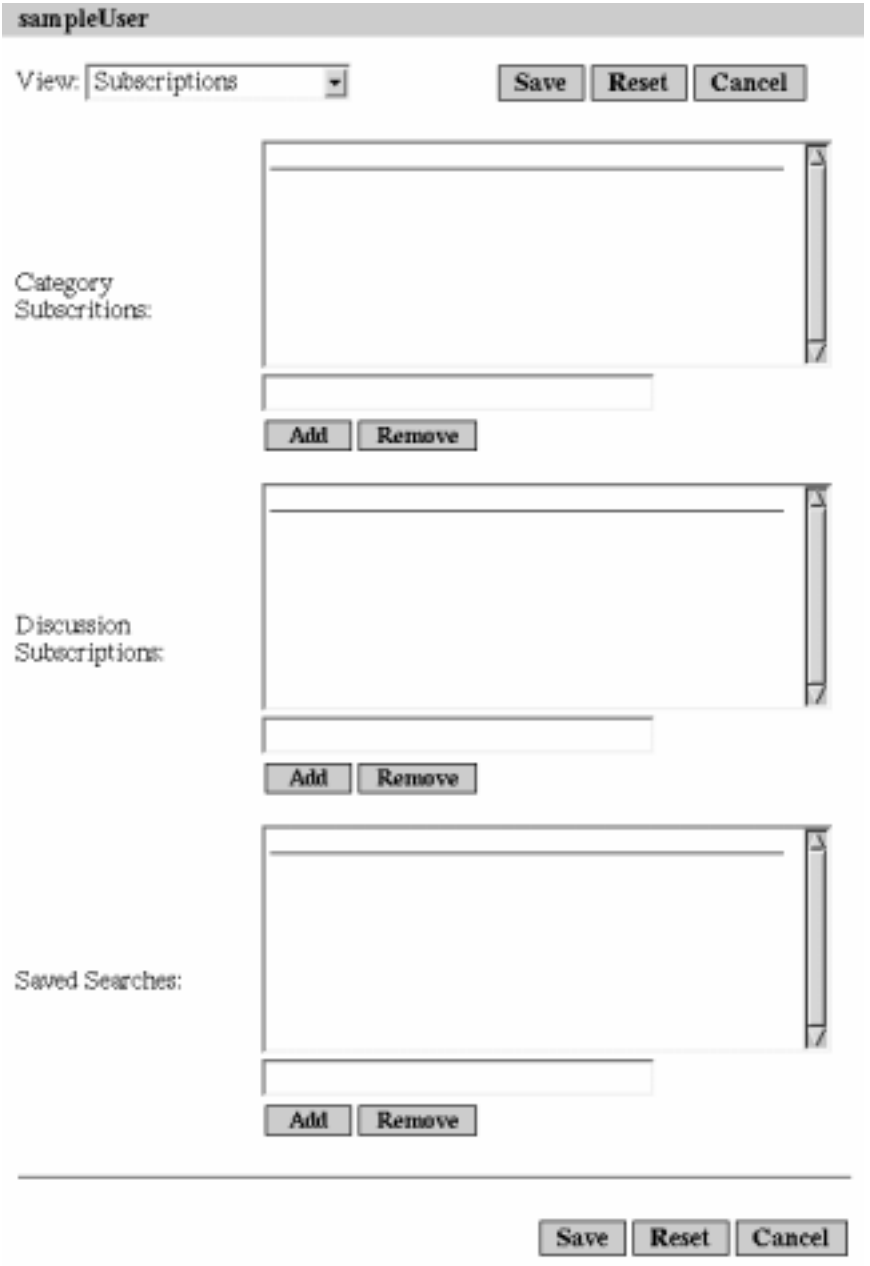
Organization User level

Administering the Subscriptions service at the Organization User level edits user's Subscriptions service settings. The administrator can maintain the user's service data, such as:

- Update user subscriptions
- Delete user subscriptions

Figure 11-3 on page 288 contains the interface for administering the Subscriptions service at the user level. See "To Manage the Subscriptions Service for the User" on page 291 for information on administering Subscriptions service for a user.

Figure 11-3 User Level Subscriptions Administration Interface


sampleUser
View: Subscriptions
Save
Reset
Cancel
Category Subscritions:
Add
Remove
Discussion Subscriptions:
Add
Remove
Saved Searches:
Add
Remove
Save
Reset
Cancel

To Define the Subscriptions Service at the Root Level

1. Log in to the Sun ONE Identity Server administration console and select the Service Configuration tab.
2. Select the Subscriptions service from the Portal Service Configuration menu on the left pane.
3. Modify the default values (see Figure 11-1 on page 286) for:
   - **Maximum number of Categories subscriptions** specifies the maximum number of categories that a user can subscribe to.
   - **Maximum number of Discussion subscriptions** specifies the maximum number of discussions that a user can subscribe to.
   - **Maximum number of Saved searches** specifies the maximum number of searches that can be saved.
4. Select:
   - Save to save your values.
   - Reset to reset the values if you modified them.

To Define the Subscriptions Service at the Organization Level

1. Log in to the Sun One Identity Server administration console and select Services from the View pull-down menu for your organization.
2. Select the Subscriptions service from the Portal Service Configuration menu on the left pane.
3. Modify the default values (see Figure 11-2 on page 287) for:
   - **Conflict Resolution Level** can be set to Highest, Higher, High, Medium, Low, Lower, and Lowest.
   - **Maximum number of Categories subscriptions** specifies the maximum number of categories that a user can subscribe.
   - **Maximum number of Discussion subscriptions** specifies the maximum number of discussions that a user can subscribe.

- **Maximum number of Saved searches** specifies the maximum number of searches that can be saved.

4. Select:
   - Save to save your values.
   - Reset to reset the values if you modified them.
   - Delete.

To Manage the Subscriptions Service for the User

1. Log in to the administration console and select Users from the View pull-down menu for your organization.
2. Select the User.

   The user information is displayed on the right pane.
3. Select Subscriptions from the View pull-down menu.

   A page to edit the user's subscriptions is displayed.
4. Edit the subscriptions definition (see Figure 11-3 on page 288.)

   For each type of subscription, add or remove subscriptions. The format of:
   - Category subscription is:

   ```
   label | target category | scope | lapsed time
   ```

   where:

`label`	Refers to a logical reference given to the edited subscription and it must be a string. This is a required field.
`target category`	Must be of the string format *ABC:DEF:GHI*
`scope`	Refers to a search query and it must be of a string format that is a valid search string, including search operators.

`lapsed time`	Must be one of the following numbers: •0=forever •7=since last week •30=since last month •180=since last 6 months •365=since last year

- Discussions subscriptions is:

```
label | target discussion RD's URL | scope | lapsed time |
minimum rating
```

where:

`label`	Refers to a logical reference given to the edited subscription and it must be a string. This is a required field.
`target discussion RD's URL`	Must be of string format matching the Discussion's URL. This cannot be edited by the user using the subscriptions channel for editing the discussion.
`scope`	Refers to a search query and if must be of a string format that is a valid search string, including search operators.
`lapsed time`	Must be one of the following numbers: •0=forever •7=since last week •30=since last month •180=since last 6 months •365=since last year
`minimum rating`	Refers to a filter base on a minimum rating.

- Saved searches is:

```
label | target category | scope | lapsed time
```

where:

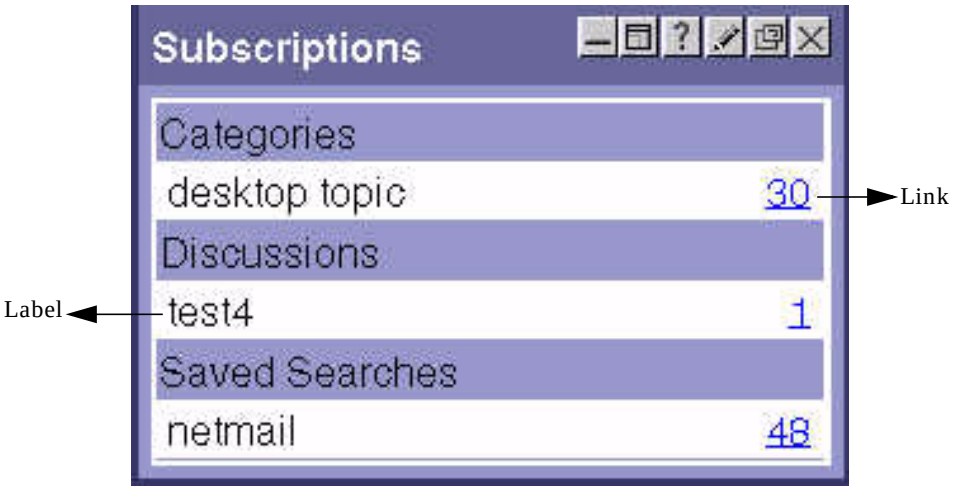
`label`	Refers to a logical reference given to the edited subscription and it must be a string. This is a required field.
`target category`	Must be of the string format *ABC:DEF:GHI*
`scope`	Refers to a search query and if must be of a string format that is a valid search string, including search operators.
`lapsed time`	Must be one of the following numbers: •0=forever •7=since last week •30=since last month •180=since last 6 months •365=since last year

Using the Subscriptions Channel

The Subscriptions channel (as shown in Figure 11-4) shows subscriptions by types which can be category subscriptions, discussion subscriptions, and saved searches. For each type of subscription, the following is displayed:

- The subscription label
- A link to the subscription detail representing the number of hits for that particular subscription

Figure 11-4 Sample Subscriptions Channel on the Desktop

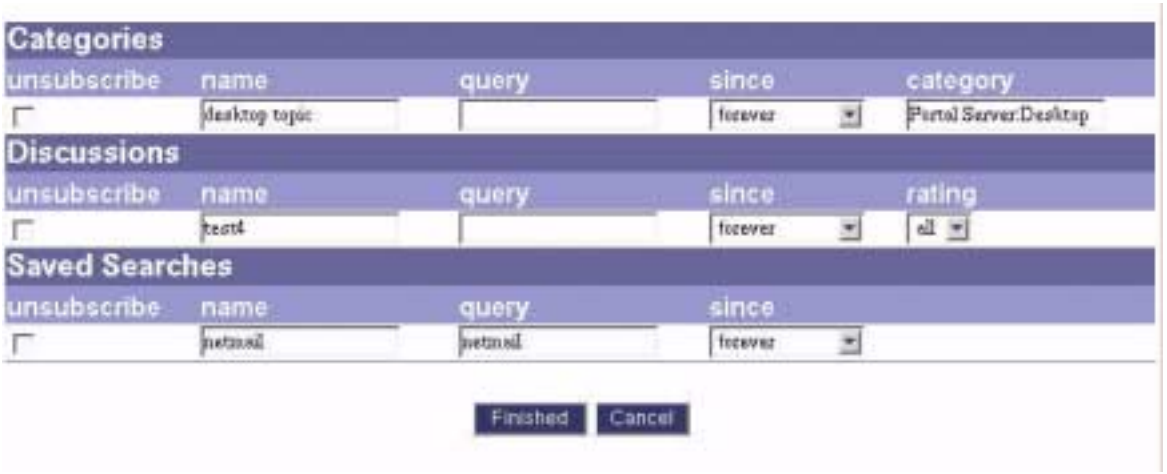


An end user can update all the subscriptions and unsubscribe via the subscriptions channel Edit button (see Figure 11-5 on page 294). End user alerts for matching subscriptions is grouped in the Subscriptions channel. The alerts are generated upon the subscriptions channel's refresh time. The administrator can set the `refreshTime` property for the channel that make the actual rendering of the content cache for a certain period of time. When the end user tries to refresh the content of the subscriptions channel more than once within less time than the refreshTime parameter, then the content would be read from the cache instead of being generated from the actual data. The `refreshTime` channel property value can be specified in seconds.

An end user is alerted of a new document when the document:

- Is categorized in a subscribed category and matches the scope and time criteria.
- Is a comment on a subscribed discussion and matches the scope and time criteria.
- Matches saved basic or advanced search criteria and time criteria.

Figure 11-5 Sample Subscriptions Channel Edit Page on the Desktop



To Subscribe to a Category

1. Log in to the sample Desktop.

   You can subscribe to categories via:

   - Browse categories - this includes a Subscribe link
   - Search results that show categories - this includes a Subscribe link
   - Results of search within a category - this includes a Subscribe to Category link (as shown in Figure 11-6 on page 296)

2. Select the subscribe link next to the category you wish to subscribe.

   The page to specify subscription information is displayed.

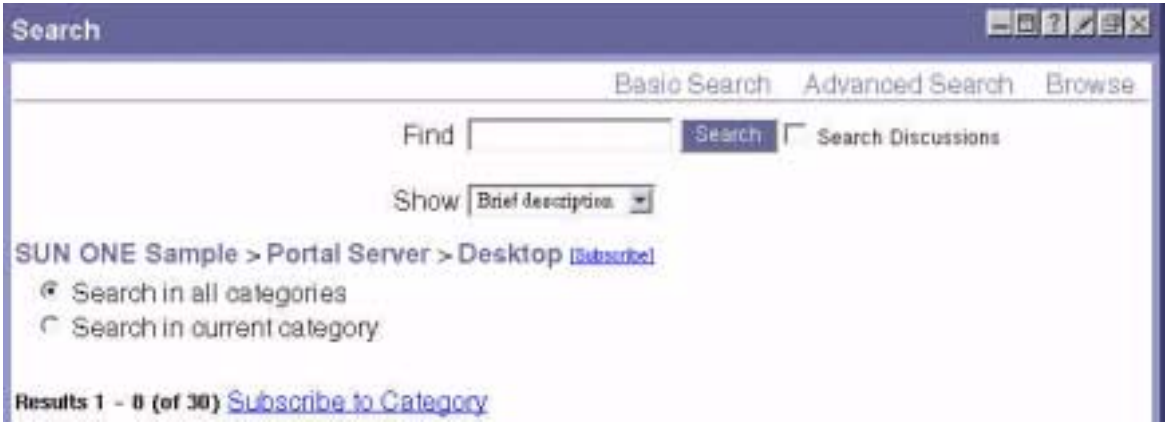
3. Specify:

   - Subscription Name - A name for the category
   - Target Category- Name of the category
   - Scope of Search- A query string, similar to the Search text field
   - Since - Amount of time you wish to be subscribed to the specified category. It can forever, since last week, since last month, since last 6 months, since last year

4. Select the Finished button.

   The category is added to your list of subscriptions.

Figure 11-6 Sample Page to Subscribe to a Category



To Subscribe to a Discussion

1. Log in to the sample Desktop.

   You can subscribe to discussions via the view discussions link - this includes a Subscribe link.

2. Select the subscribe link for the discussion you wish to subscribe.

   The page to specify subscription information is displayed.

3. Specify:

   - subscription name - A name for the category
   - target category - Name of the category
   - scope of search - A query string, similar to the Search text field
   - since - Amount of time you wish to be subscribed to the specified category. It can forever, since last week, since last month, since last 6 months, since last year.
   - rating - Threshold rating above which subscription is valid

4. Select the Finished button.

   You are now subscribed to the discussion.

To Save a Search

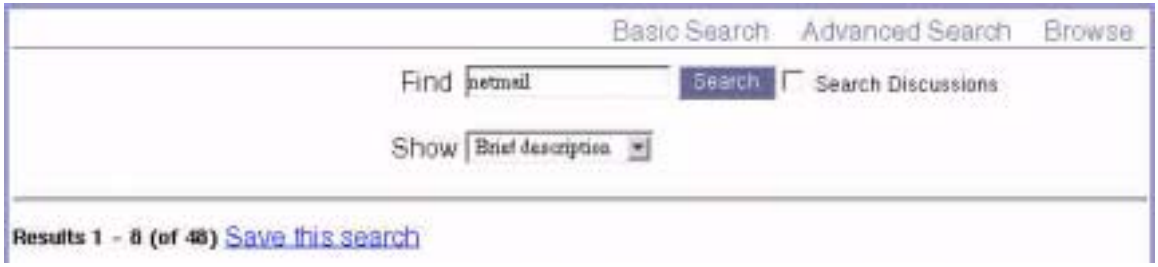
1. Log in to the sample Desktop.
2. Access the Search tab and search for a document.

   The search result page is displayed.
3. Select the subscribe link at the top of the result list (as shown in Figure 11-7 on page 297).

   The page to specify subscription information is displayed.
4. Specify:
   - Label - Save search label
   - Scope of Search- A query string, similar to the Search text field
   - Since - Amount of time you wish to save the specified search result. It can be forever, since last week, since last month, since last 6 months, since last year.
5. Select the Finished button.

   Your search result is now saved.

Figure 11-7 Sample Page to Subscribe to Search Results



Discussions

This section contains the following:

- Discussions Overview

Discussions Overview

Discussions are tied to topics and specific documents. It is a powerful way for people to add and talk about existing documents or create their own. It provides an easy way to share information about specific documents or new topics.

The Sun ONE Portal Server software discussions feature includes discussion threads, starting discussions based on documents or new topics, searching discussions, and rating discussions. By default, the Discussions channel is available on the sample portal for anonymous users. However, an anonymous user cannot subscribe to a discussion or edit the Discussion channel.

The DiscussionLite channel and the Discussions channel are based on the DiscussionProvider. Similar to the search channel JSPs, they have a query portion, a display portion, and use Desktop themes.

DiscussionProvider

The DiscussionProvider is JSP provider that uses the Desktop themes. It retrieves data from the backend Search service using search tag libraries and API. The discussions and comments are stored as different Resource Descriptors (RDs) in the discussion database. The DiscussionProvider supports:

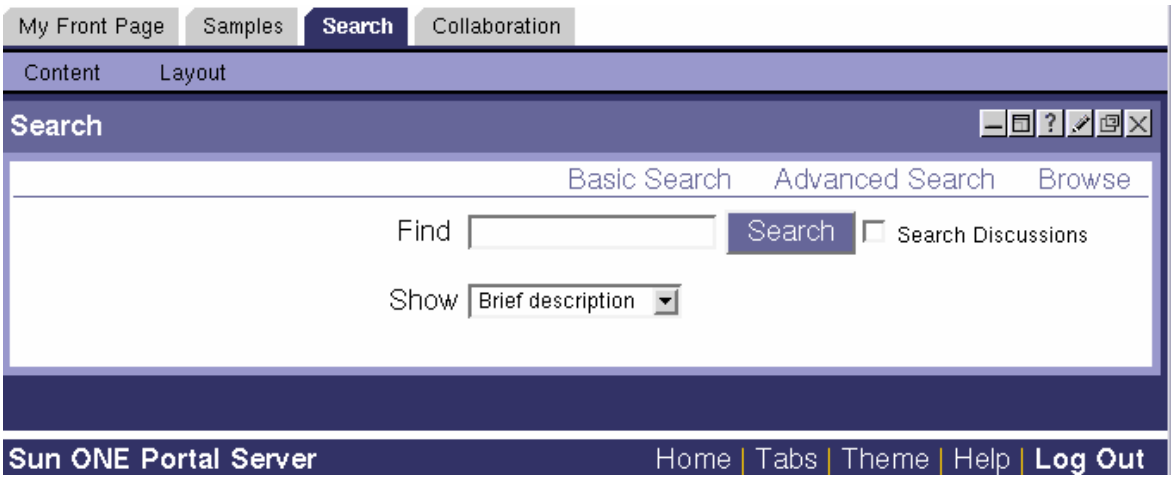
- A full view (via the Discussions channel) and an abbreviated view (via the DiscussionLite channel) that:
  - Starts a new discussion from the discussion channel.
  - Posts reply to an existing discussion.
  - Starts a new discussion based on web documents from the search channel.
- A Discussion List that:
  - Retrieves main posts sorted by last-modified date.
  - Has pagination so users can access older discussions.
- a discussion view that displays each discussion subtree. The main item is displayed in detail and the subtree is displayed below the main item. View discussion includes:

- Several filters on the page. A document display can be based on filters such as document rating (irrelevant, routine, interesting, important, and must read).
- Display preference can be set to threaded or flat display.
- Expansion threshold helps to control displayed items in the subtree. The users can choose to expand only highly rated documents, or expand all or collapse all. Default value is collapse all. Expand all will expand all the filtered comments. It will also show a description of the discussion, provide a menu for rating the discussion, and allow the user to post a reply.
- Support to search within a discussion.

The user also has the option to set these preferences through the channel edit page.

- Commenting and rating a discussion. For example, users can:
  - Add a comment on an existing discussion.
  - Rate all discussions and comments. However user rating is not immediately visible. The rating calculation is based on an algorithm such that the rating for any comment goes up gradually. For example, a comment has to be rated important three times before it is marked as important.
- Searching all discussions (see Figure 11-8) and within a discussion. These functions are routed to the search provider. Users can also search by rating in Advance Search.

Figure 11-8 Search Page on the Desktop that Allows Searching All Discussions



- Subscriptions. Authenticated users can choose to subscribe to a particular discussion by selecting the subscribe link. The request is handled by the SubscriptionProvider. The `displaySubscription` property (see Code Example 11-1) can be disabled if the feature is not required. By default, the value is true.

Display Profile XML Fragment for DiscussionProvider

Code Example 11-1 shows the DiscussionProvider provider XML fragment in the display profile.

Code Example 11-1 DiscussionProvider Provider Display Profile XML Fragment

```
<Provider name="DiscussionProvider"
class="com.sun.portal.providers.jsp.JSPProvider">
    <Properties>
        <String name="title" value="*** Discussions Provider ***"/>
        <String name="description" value="*** DESCRIPTION ***"/>
        <String name="refreshTime" value="0" advanced="true"/>
        <String name="helpURL" value="en/desktop/discussions.htm"
advanced="true"/>
        <String name="fontFace1" value="Sans-serif"/>
        <String name="productName" value="Sun ONE Portal Server"/>
        <String name="contentPage" value="discussionContent.jsp"/>
        <String name="editPage" value="discussionEdit.jsp"/>
        <String name="processPage" value="discussionDoEdit.jsp"/>
        <Boolean name="isEditable" value="true" advanced="true"/>
        <String name="editType" value="edit_subset" advanced="true"/>
        <Boolean name="showExceptions" value="false"/>
        <Boolean name="showErrors" value="true"/>
        <String name="width" value="thick"/>
```

Code Example 11-1 DiscussionProvider Provider Display Profile XML Fragment

```
        <String name="column" value="2"/>
        <String name="searchServer" value=""/>
        <String name="dbname" value=""/>
        <Integer name="viewHits" value="8"/>
        <String name="defaultDiscussionDisplay" value="Threaded"/>
        <String name="defaultFilter" value="Irrelevant"/>
        <String name="defaultExpansionThreshold" value="Collapse all"/>
        <Boolean name="viewDiscussionWindow" value="false"/>
        <String name="anonymousAuthor" value="anonymous"/>
        <Boolean name="displaySearch" value="true"/>
        <Boolean name="showDescription" value="false"/>
        <String name="ratingText"
value="Irrelevant,Routine,Interesting,Important,Must Read"/>
    </Properties>
</Provider>
```

Administering the DiscussionProvider

The DiscussionProvider administration is distributed between:

- Channel edit page (this is user configurable)
- Desktop Channel and Container Management link on the administration console for the DiscussionProvider's channel
- Search Service

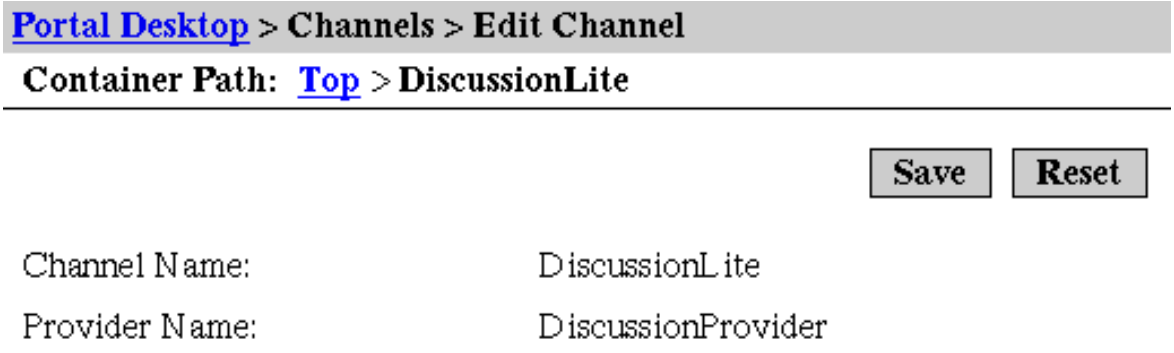
DiscussionLite Channel

The DiscussionLite channel (as shown in Figure 11-9 on page 301) displays the top twenty discussion titles (which can be reconfigured) and the date. The discussions are sorted by creation date (last modified) and the newest discussion is displayed first. The DiscussionLite channel view has links to view each discussion, view all discussions which target the Discussions Channel, and start a discussion. By default, the channel is displayed in a single container and all links are brought up in a JSPDynamicSingleContainer.

Figure 11-9 Sample Discussion Lite Channel on the Desktop

Properties can be configured from the administration console (see Figure 11-10). By default, there are no user editable properties for this channel.

Figure 11-10 Sample DiscussionLite Channel Edit Page on the Sun One Identity Server Admin Console



Discussions are stored in the discussion database specified in the `dbname` property in the display profile. Search server host (`searchServer` property), database name (`dbname` property), and the number of discussions to be displayed (`viewHits` property) can be configured in the display profile (see Code Example 11-2 on page 303.)

Code Example 11-2 DiscussionLiteProvider Channel Display Profile XML Fragment

```
<Channel name="DiscussionLite" provider="DiscussionProvider">
    <Properties>
        <String name="title" value="Recent Discussions"/>
        <String name="description" value="This is a DiscussionLite provider
example"/>
        <String name="contentPage" value="discussionLiteContent.jsp"/>
        <String name="editPage" value=""/>
        <String name="processPage" value=""/>
        <String name="width" value="thin"/>
        <String name="searchServer" value=""/>
        <String name="db" value="discussion"/>
        <Integer name="viewHits" value="20"/>
    </Properties>
</Channel>
```

The following JSPs are used by the DiscussionLite channel:

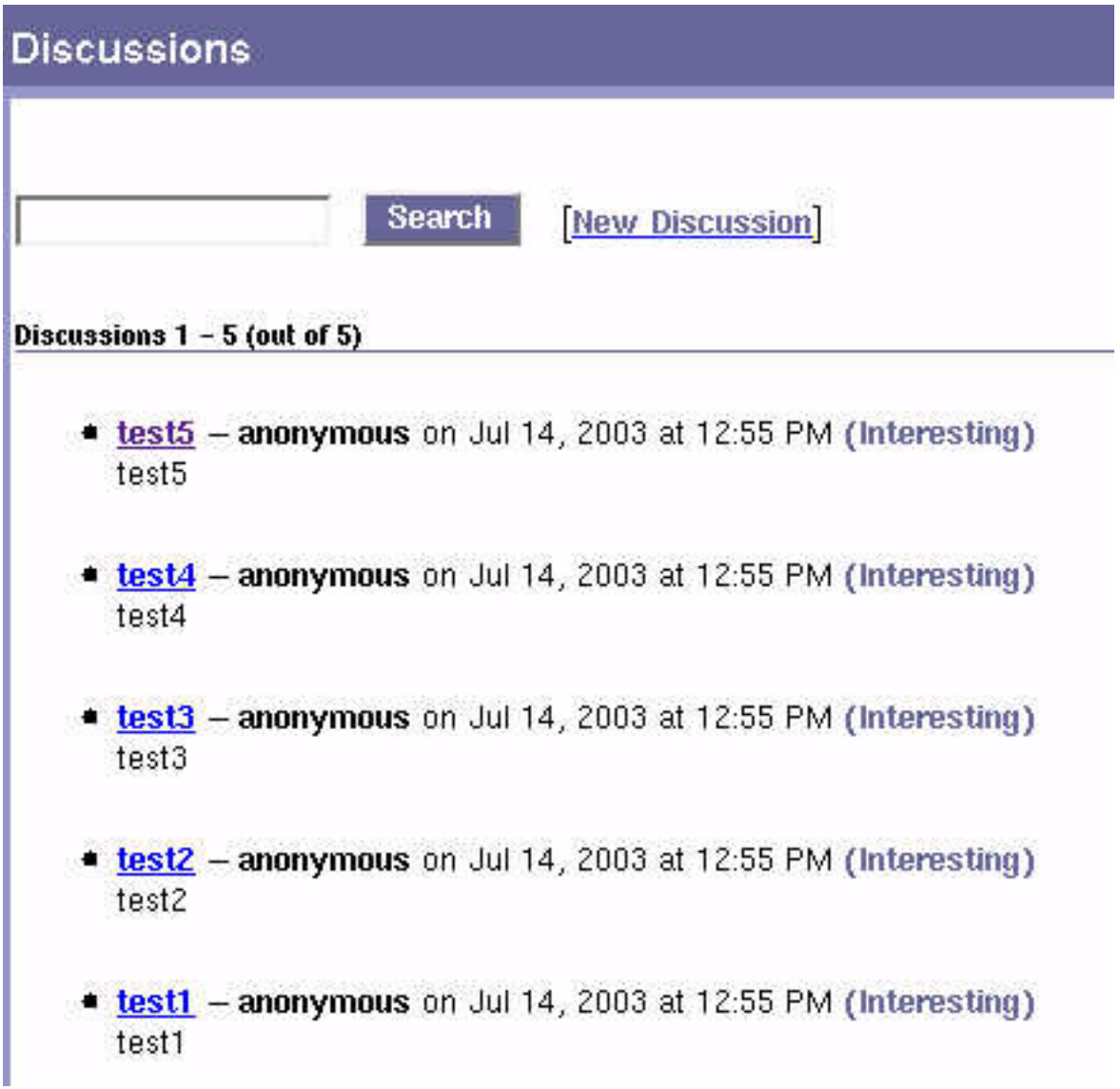
`discussionLiteContent.jsp`	JSP content page.
`query.jsp`	Sets and executes search query.
`display.jsp`	Displays results.
`error.jsp`	Displays exceptions and error messages.

Discussions Channel

The Discussions channel includes a full view that:

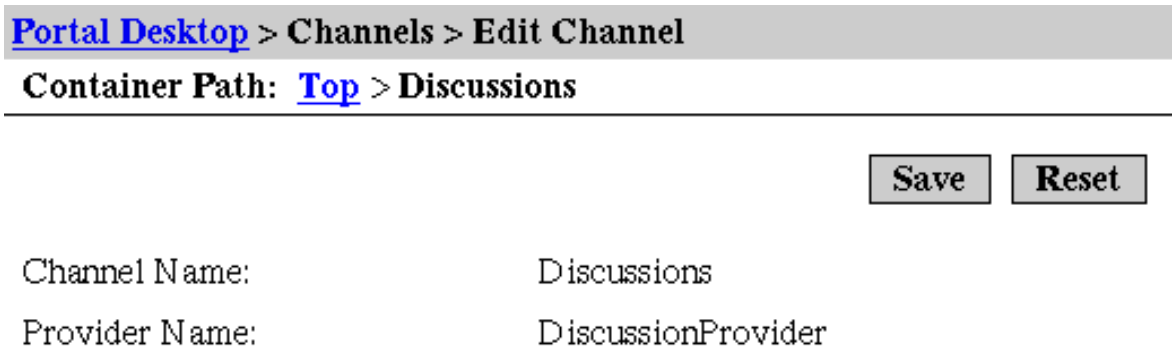
- Shows detailed descriptions for the top eight discussions sorted in descending order. This can be reconfigured via the channel edit page.
- Includes pagination so that users can see all the discussions.
- Supports search. The search returns discussion and comment results.

Figure 11-11 Sample Discussion Full View on the Desktop



The Discussions channel properties can be configured from the Sun ONE administration console.

Figure 11-12 Sample Discussions Channel Edit Page on the Sun One Identity Server Admin Console



Discussions are stored in the discussion database specified in the `dbname` property in the display profile. Search server host (`searchServer` property), database name (`dbname` property), and the number of discussions to be displayed (`viewHits` property) can be configured in the display profile (see Code Example 11-3.)

Code Example 11-3 Discussions Channel Display Profile XML Fragment

```
<Channel name="Discussions" provider="DiscussionProvider">
    <Properties>
        <String name="title" value="Discussions"/>
        <String name="description" value="This is a Discussion provider
example"/>
        <String name="searchServer" value=""/>
        <String name="dbname" value="discussions"/>
        <Integer name="viewHits" value="8"/>
    </Properties>
</Channel>
```

The following JSPs are used by Discussions channel:

`discussionContent.jsp`	JSP content page
`discussionEdit.jsp`	The edit page
`discussionDoEdit.jsp`	The process edit page
`declare.jsp`	
`portal.jsp`	Extracts the display profile data
`fullDiscussion.jsp`	handles the full view presentation

fullDiscussionDisplay.jsp	User interface for the all discussions page
searchUI.jsp	Search form displayed on the all discussions page
viewDiscussion.jsp	View discussion
viewDiscussionBar.jsp	Center horizontal bar with all the filters on the view discussion page
viewDiscussionDisplay.jsp	User interface for the view discussion page
viewDiscussionHeader.jsp	Header comment display on the view discussion page
viewDiscussionNavigation.jsp	Navigation bar displayed above and below the header on the view discussion page
feedback.jsp	Provides comment, feedback, and rating functionality
feedbackDisplay.jsp	Displays the feedback
feedbackForm.jsp	Provides the feedback form
feedbackProcess.jsp	Processes the feedback
error.jsp	Displays exceptions and error messages
query.jsp	Formats and executes the search query
pageFooter.jsp	Provides pagination

Managing and Using the Channels

Administering the DiscussionProvider Channel

Administration of the DiscussionProvider channel is distributed between the Desktop display profile and the Search service in the Sun ONE Identity Server administration console. Provider specific information is stored in the display profile. Discussion document and database related administration must be done in the Search service.

Discussions are stored in the discussion database. The discussion database expects a specific schema for discussions and comments. New schema fields have been added for this feature in the`schema.rdm` file.The search CLI `rdmgr` can be used for database management and debugging. For example, to dump all the comments, type:

```
./run-cs-cli rdmgr -y discussion
```

The sample DiscussionProvider channels are configured to use the default search server. Some sample discussions which are imported in the discussion database and channel is ready for use.

The samples are located at *S1PSBaseDir*/SUNWps/samples/discussions/ directory. They are:

discussions.soif	A sample SOIF file loaded in the discussion database.
dp-org.xml	Contains the discussion channel display profile XML fragments.
dp-providers.xml	Contains the discussion provider display profile XML fragments.
dp-anon.xml	Contains XML fragment for the authlessanonymous user, loaded at sample portal install time.

Access to discussions can be controlled (to read only or totally hidden) by the administrator.

To Create a Channel from DiscussionProvider

1. Log in to the Sun ONE Identity Server administration console and select Services from the View pull-down menu.

   The list of services is displayed in the left frame.

2. Select Desktop and Channel and Container Management.

   Note that the Channel and Container Management link is available in the right frame.

3. Select the Add button under Channels.

   The page to specify the type of channel to add is displayed.

4. Specify a name for the channel in the Channel Name text box and select DiscussionProvider from the Provider pull-down menu.

5. Select Create.

   This action creates a channel based on the specified provider. The Cancel button returns you to the Channel and Container Management page without creating any new channels.

6. Select the Edit link next to the newly created channel in the Channels table.

   The page to edit the default values of the channel is displayed.

7. Edit the properties and select the Save button to save the modified values.

   The following display profile properties are specific to this provider:

`searchServer`	Path to the search server. By default, `portal/portal/search`.
`dbname`	Any valid database.
`viewHits`	Number of discussions to display.
`defaultDiscussionDisplay`	This can be set to flat or threaded to allow the comment subtree to be displayed as flat or threaded.
`defaultFilter`	Filter for searching and displaying discussions and control display of the subtree. It can be based on ratings such as irrelevant, routine, interesting, important, or must read. By default, its value is irrelevant; so all comments rated irrelevant and above are displayed. The Must read filter will highlight the highly rated comments.
`defaultExpansionThreshold`	This can be set to expand all or collapse all. By default, its value is set to collapse all. If set to expand all, it will expand all the filtered comments, show description, rating menu, and allow user to post reply via links.
`anonymousAuthor`	
`viewDiscussionWindow`	
`displaySearch`	
`showDescription`	For the Discussions channel, this is configurable.
`ratingText`	By default, discussions can be rated at irrelevant, routine, interesting, important, or must read.

Using the DiscussionProvider Sample Channels

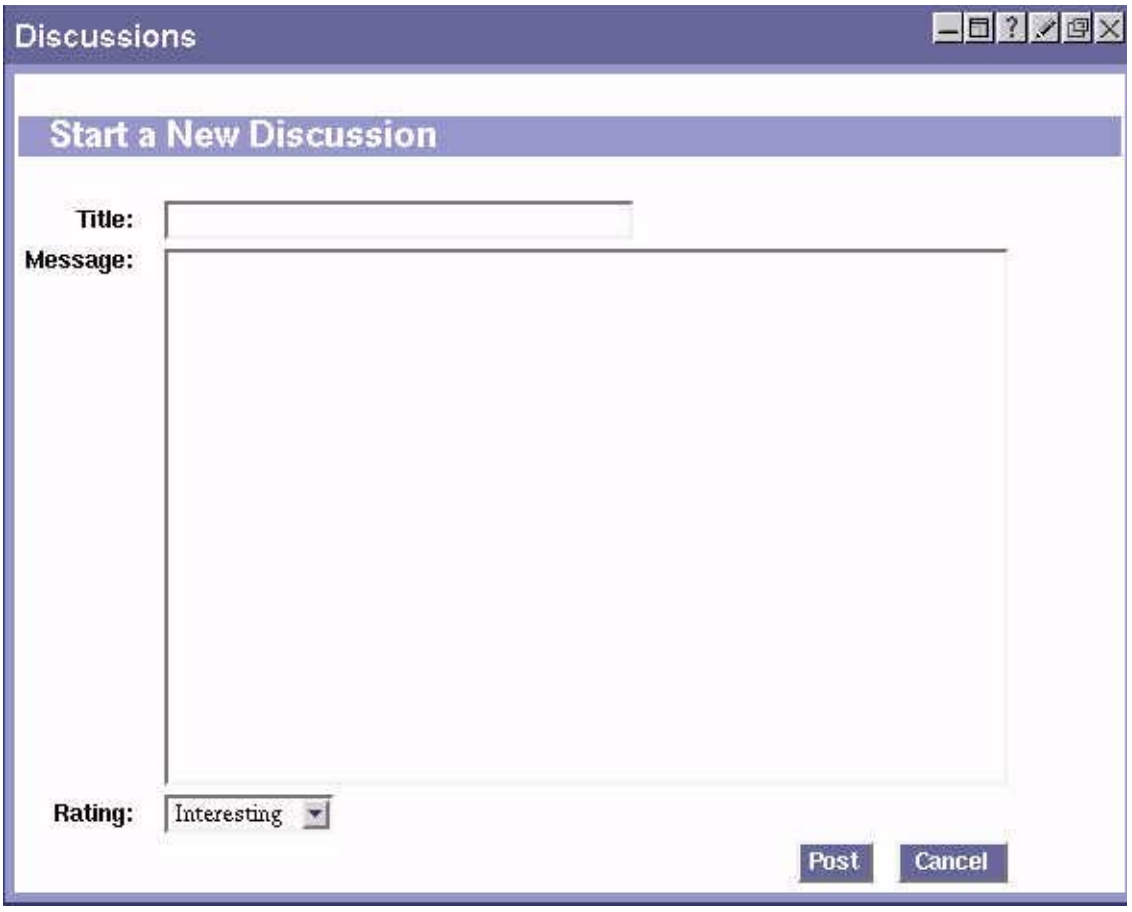
To Start a New Discussion

1. Log in to the sample Desktop.
2. To start a new discussion from the:

- Channel, select the Collaborations tab and select the link to Start A New Discussion.
- Search channel, select the Start A New Discussion link next to the document.

3. Specify:
   - Title - A title for the discussion
   - Message - Content for discussion
   - Rating - Rate the discussion. It can be routine, interesting, important, or must read.
4. Select Submit Feedback button.

Figure 11-13 Sample Start New Discussion Page on the Desktop


Discussions
Start a New Discussion
Title:
Message:
Rating:
Interesting
Post
Cancel

Chapter 12

Configuring the Communication Channels

This chapter provides information on the communication channels for Sun™ ONE Portal Server 6.2, starting with general descriptive information, moving to an explanation of the state of the communication channels after installation but before configuration, and finally leading into a description of various steps for configuring the communication channels according to a site's needs.

The information provided on configuration makes up the bulk of this chapter and includes administrator and end user configuration. End users have the ability to edit the configuration of each channel directly from the Portal Desktop by clicking the edit button accessible in each channel. This gives end users access to an edit page (or edit pages) that allows editing of specific server configuration information and that allows editing of specific features visible to the end user in the channel, such as the number of address book entries visible in the Address Book channel.

Administrators can limit or extend end users' editing options. Administrators can even preconfigure channels to work without the need for end user server configuration; for more information see "Administrator Proxy Authentication: Eliminating End-User Credential Configuration" on page 339.

Since administrators can design each channel's edit page, they can select which specific features end users will be able to edit; for more information see "Application Preference Editing: Configuring Communication Channel Edit Pages" on page 334.

Furthermore, if a site has more than one instance of a particular application available—for example, two or more instances of a mail application—administrators can allow end users to configure a second Mail channel on their Portal Desktops; For more information, see "Enabling End-Users to Set Up Multiple Instances of a Communication Channel Type" on page 338.

This chapter includes the following sections:

Overview of the Communication Channels

The Sun ONE Portal Server 6.2 product offers four communication channels that are accessible by end users directly in Portal Desktop. These channels allow end users access to corresponding applications—such as a mail application— which enable end users to organize, schedule, and communicate more effectively and efficiently.

The four communication channels are:

Address Book Channel The Address Book channel displays address book entries for end users to view. To access the address book in order to create and edit address book entries, first click Launch Address Book.

Calendar Channel The Calendar channel displays calendar events and tasks for end users to view. To access the calendar application in order to create new tasks and events, first click Launch Calendar.

Instant Messaging Channel The Instant Messaging Channel displays the presence status of other users with access to Sun™ ONE Instant Messenger. These contacts are from a list end users have created within the Instant Messenger application. Initiate a chat from the channel by clicking a presence status icon, which is one method of invoking Instant Messenger. To get presence updates directly from the channel, reload Portal Desktop. To receive presence updates as they occur, view contacts' presence status from Instant Messenger by invoking the application; therefore, click Instant Messenger.

Mail Channel The Mail channel displays mail messages sent to end users for them to view. To access the mail application in order to read and compose messages, click Launch Mail.

Supported Software for the Communication Channels

The Sun ONE Portal Server software supports the following resource server platforms for the Communication Channels:

- Sun™ ONE Messaging Server 5.2, 6.0
- Sun™ ONE Calendar Server 5.1.1, 6.0
- Sun™ ONE Instant Messaging Server 6.1
- IBM Lotus Notes 5.0.6
- Microsoft Exchange Server 5.5, 2000

The Installer and the Communication Channels

The Sun ONE Portal Server installer performs several tasks involving the communication channels. General communication channel configuration tasks are also handled by the installer. More detailed configuration is then required by administrators and end users depending up the needs of the site and of the individuals.

Sun ONE Portal Server Installer Tasks

The Sun ONE Portal Server Installer:

- installs the following packages, `SUNWpsso`, `SUNWpsap`, `SUNWpsmp`, `SUNWpscp`, and `SUNWiimps` which are deployed to the default Sun ONE Portal Server instance. Therefore, the installer does not install the communication channels on all of the Sun ONE Portal Server instances. For information on multi-server deployments, see “Multiple Instance Deployments” on page 314.
- creates the channels, Address Book, Calendar, Instant Messaging, and Mail. The installer places channels for Sun ONE servers into the My Front Page Tab panel container for the sample organization. Therefore, the communication channels are installed only when the sample portal is installed. Microsoft Exchange Server and IBM Lotus Notes server are not automatically placed in a container. An administrator would need to add these channels to a container, if desired.

The default configurations for the Calendar and Mail channels work after only basic configuration by end users; therefore, they do not require further configuration by administrators. The Address Book and Instant Messaging channels require further configuration by both administrators and end users.

- creates and configures the single sign-on (SSO) Adapter service which enables single sign-on with the Sun ONE Calendar Server and Sun ONE Messaging Server.

Multiple Instance Deployments

If you have a multi Sun ONE Portal Server deployment, manually deploy the communication channels to each additional instance of Sun ONE Portal Server and restart each instance. To deploy, type:

portal-server-install-dir*/SUNWps/bin/deploy redeploy -instance *instancename* -deploy_admin_password *deployadminpassword

Where *instancename* is the name for that particular non-default instance and *deployadminpassword* is the administrator password for the web container (web server or application server). The web container administrator password is only needed when the web container is Sun™ ONE Application Server or BEA WebLogic Server™. It is not problematic if you include the password when using one of the other acceptable web containers: Sun™ ONE Web Server or IBM WebSphere® Application Server; however, in those cases the password will be ignored.

Code Example 12-1 lists the commands for manually deploying communication channels to two non-default Sun ONE Portal Server instances and for restarting those instances, where `myinstance1` and `myinstance2` are non-default Sun ONE Portal Server instance names and `Admin` is the web container administrator password.

Code Example 12-1 Deploying Communication Channels to a Non-Default Instance

```
portal-server-install-dir/SUNWps/bin/deploy redeploy -instance myinstance1
    -deploy_admin_password Admin
portal-server-install-dir/SUNWps/bin/deploy redeploy -instance myinstance2
    -deploy_admin_password Admin

portal-server-install-dir/SUNWam/bin/amserver stopall
portal-server-install-dir/SUNWam/bin/amserver startall
```

Configuration Tasks for the Communication Channels

The following are the high-level tasks involved in setting up the communication channels. Not all tasks are applicable to all sites. You need to determine if a task is applicable to your site according to your site's business requirements.

- Configuring the Services for the Default Organization
- Configuring End-User Channel Settings
- Application Preference Editing: Configuring Communication Channel Edit Pages
- Enabling End-Users to Set Up Multiple Instances of a Communication Channel Type
- Administrator Proxy Authentication: Eliminating End-User Credential Configuration
- Configuring a Read-Only Communication Channel for the Authentication-Less Portal Desktop
- Configuring Microsoft Exchange Server or IBM Lotus Notes
- Creating a New User Under the Default Organization
- Configuring the Mail Provider to Work with an HTTPS Enabled Messaging Server

If you already have Sun ONE Messaging Server and Sun ONE Calendar Server installed either on the same server or on different servers, specify the respective URL when you create a channel.

Configuring the Services for the Default Organization

After the communication channels have been installed, the Instant Messaging and Address Book channels require more detailed configuration as explained subsequently. However, the Calendar and Mail channels have sample or default settings that can work without further configuration by an administrator. Site-specific issues can exist for any of the communication channels—including the Calendar and Mail channels—that deserve attention and might require configuration by an administrator before the channels will work according to the needs of the site.

The following sections provide important information relating to the configuration of the communication channels.

Communication Channel Configuration Information

Regarding All the Communication Channels

End-User Configuration

Unless you configure the communication channels with proxy authentication—see "Administrator Proxy Authentication: Eliminating End-User Credential Configuration" on page 339 for more information—end users will still need to go to each channel's edit page by clicking the edit button in the respective communication channel to further configure the channel.

CAUTION—Undetected Error: Missing Launch Button

If a client port number is entered incorrectly for any of the communication channels, end users will not receive an error message. The error manifests itself by not displaying the launch button for the respective channel, which does not aid end users in identifying the root cause of the problem. Both administrators and end users can enter an incorrect client port number, but since end users can only edit the client port number for the Calendar and Mail channels, those are the only channels where they can create this problem.

CAUTION—Undetected Error: Missing Channel

Various situations can cause end users *not* to see a communication channel and *not* to see an error message explaining the problem. The cause might be a misconfigured template or configuration name, which doesn't allow the template or configuration to be found. A communication channel does not display when any of the following conditions is true:

- The SSOAdapter template is not found.
- The SSO Adapter configuration is not found.
- The `display.template` file is not found.

Regarding the Mail Channel

HTTPS Enabled Messaging Server

If the Mail channel is connected to—a more secure—HTTPS enabled messaging server instead of the basic HTTP enabled messaging server, then you will need to make some security-related adjustments for the Mail channel to work as intended. For more information, see "Configuring the Mail Provider to Work with an HTTPS Enabled Messaging Server" on page 358.

Configuring the Instant Messaging Channel

Sun ONE Instant Messaging Server is installed during the installation of Sun ONE Portal Server if the Enable IM in Portal Server option is selected during the Sun ONE Instant Messaging Server installation.

While the Instant Messaging Portal channel is designed to work right out of the box, other configuration might be necessary depending upon your site's needs. Therefore, after following the steps in "To Configure the Instant Messaging Channel," see "Additional Configuration for the Instant Messaging Channel," to determine if any of that section's subsections apply to your installation.

The Instant Messaging channel is based on a Portal Server content provider called `IMProvider`. The `IMProvider` is an extension of the `JSPProvider` in the Portal Server. As an extension of the `JSPProvider`, `IMProvider` uses the JSP files to generate the content page and the edit page for the Instant Messaging channel. The JSP files are also used to generate the pages used to launch the Instant Messenger. The `IMProvider` also defines an instant messaging-specific tag library and this tag library is used by the JSP files. The JSP files and the tag library use the channel properties that are defined by the `IMProvider`.

For more information on Sun ONE Instant Messaging Server, see *Instant Messaging Administrator's Guide*. For information specific to the Sun ONE Portal Server Instant Messaging Channel tag library and the customization of Instant Messaging Channel through the editing of JSP files, see *Sun ONE Portal Server 6.2 Desktop Customization Guide*. Furthermore, administrators and end users can access information about Sun ONE Instant Messaging Server by visiting the URL used in the codebase property for the Instant Messaging Channel configuration.

To Configure the Instant Messaging Channel

1. From an Internet browser, log into the Sun™ ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Identity Management tab to display the View drop down list in the navigation pane (the lower left frame).

3. Select Services in the View drop down list to display the list of configurable services.
4. Under the Portal Server Configuration heading, click the arrow next to Portal Desktop to bring up the Portal Desktop page in the data pane (the lower right frame).
5. Click the Channel and Container Management link.
6. Scroll down to the Channels heading and click Edit Properties next to IMChannel to display the Instant Messaging service panel, which includes Basic Properties.

   The following is a partial list of the properties displayed in the Edit IMChannel page with example values provided for each property.

Property	Value
server	imserver.example.com
port	49999
mux	imserver.example.com
muxport	49909
codebase	imapplet.example.com
netletRule	IM
clientRunMode	plugin
authMethod	idsvr
authUsernameAttr	uid
username	(not applicable when idsvr is used for authmethod)
password	(not applicable when idsvr is used for authmethod)
contactGroup	My Contacts

7. In the text field next to each property you want to input, enter the desired value. Table 12-1 describes the properties and the type of information to enter as a value.

Table 12-1 Property and Value Description for Edit IMChannel Page

Property	Value
server	Enter the hostname of the Sun ONE Instant Messaging Server to be used by the channel.
port	Enter the port number associated with the Sun ONE Instant Messaging Server to be used by the channel. The default port number is 49999.

Table 12-1 Property and Value Description for Edit IMChannel Page

Property	Value
mux	Enter the hostname of the Sun ONE Instant Messaging Multiplexor to be used when the Instant Messaging client is launched by the channel.
muxport	Enter the port number associated with the Sun ONE Instant Messaging Multiplexor. The default port number is 49909.
codebase	Enter the URL prefix from which the Instant messaging client is downloaded.
netletRule	Enter the name of the netlet rule that is used with the Instant Messaging client when in secure mode via the Secure Remote Access (SRA) gateway.
clientRunMode	Enter the method for running the Instant Messaging client: plugin or jnlp (which is used for Java Web Start).
authMethod	It is usually preferable to enter `idsvr` as the value, which indicates that the authentication method to be used is the Sun ONE Identity Server authentication method. Two values are possible, `idsvr` or `ldap`. The `idsvr` value enables Single Sign-On to work. It also removes the `username` and `password` fields from the Instant Messaging channel edit page.
authUsernameAttr	Enter the name of the attribute to use for the user name when authenticating using the `idsvr` authentication method.
username	Enter the username to use when authenticating using the LDAP method.
password	Enter the password to use when authenticating using the LDAP method. When stored in the display profile, this property is obfuscated using the `AMPasswordUtil` class.
contactGroup	Enter the name of the contact group that is displayed in the Instant Messaging channel.

8. Scroll as needed and click Save.

Additional Configuration for the Instant Messaging Channel

Steps Might be Required to Allow Multiple Organizations

When a Portal Server instance serves multiple organizations but uses a single Instant Messaging server additional steps must be taken.

Identity Server and Portal Server allow administrators to set up users with the same User ID (uid) across an organization. For example, an organization could have two suborganizations that each have an end user named `enduser22`. This creates a conflict when these two end users attempt to access their respective accounts through the Instant Messaging channel.

To avoid this potential conflict, one set of JSP launch pages per organization must be created to contain a pass-in-the-parameter domain set to the value of the organization's attribute `sunPreferredDomain`. The default launch pages are:

`/etc/opt/SUNWps/desktop/default/IMProvider/jnlpLaunch.jsp`

`/etc/opt/SUNWps/desktop/default/IMProvider/pluginLaunch.jsp`

Inserting Instant Messenger Links in an Organization

By default the Instant Messenger links are added to the Application channel—which provides the links to launch various applications—in the default organization. The Instant Messenger links allow the Instant Messenger to be launched from the Application channel. You need to add Instant Messenger links manually, if:

- you want to add these links for another organization.
- you do not have the sample portal installed.
- you are using the `AppProvider` for another channel.

The contents for the Instant Messenger links are in the file `/opt/SUNWps/samples/desktop/dp-IMChannel.xml`. The `dp-IMChannel.xml` file also contains the sample `IMChannel`.

Edit a copy of the file `dp-IMChannel.xml` to add the Instant Messenger links information to the display profile for another organization and install the file using the `dpadmin` command as follows:

1. Change to the following directory:

   portal-server-install-dir`/SUNWps/bin/`

2. Create a copy of the `dp-IMChannel.xml` file as follows:.

   ```
   cp dp-IMChannel.xml newfile.xml
   ```

3. To modify the Application channel, type the following `dpadmin` command:

```
dpadmin modify -u ADMIN_DN -w PASSPHRASE -d ORG_DN -m
newfile.xml
```

Where,

ADMIN_DN - Replace with LDAP administrator DN. For example: `amadmin`

PASSPHRASE - Replace with the administrator's password.

ORG_DN - Replace with the DN of the Organization where the links are to be added. For example: `o=example.com, o=isp`

The URL for launching the Instant Messenger using Java Plug-in will be a reference to the Instant Messaging channel with a launch argument. For example:

```
/portal/dt?action=content&provider=IMChannel&launch=plugin&usern
ame=sam
```

The URL for launching the Instant Messenger applet with Java Web Start will be:

```
/portal/imlaunch?channel=IMChannel&launch=jnlp&username=sam
```

Enabling Secure Mode for Sun ONE Instant Messenger in Sun ONE Portal Server

Netlet facilitates secure communication between the Instant Messenger and the server.

NOTE	The Instant Messaging channel automatically uses the secured mode when accessed through the Secure Remote Access gateway. The Instant Messaging channel does not use the secured mode when it is not accessed through the gateway.

To enable the secure mode, you need to add the Netlet Rule.

To add the Netlet Rule:

1. From an Internet browser, log into the Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Identity Management tab to display the View drop down list in the navigation pane.

3. Select Services in the View drop down list to display the list of configurable services.
4. Scroll down to SRA Configuration and select Netlet.
5. Click the arrow icon beside Netlet. The Netlet Rules are displayed in the right panel.
6. Click Add under Netlet Rules.
7. Type `IM` in the Rule Name field.

NOTE The Netlet rule name can be different. You can configure the Instant Messaging channel to use a different Netlet rule.

8. Remove the default value in the URL field and leave the field blank.
9. Select the Download Applet check box and enter the following string:

   ```
   $IM_DOWNLOAD_PORT:$IM_WEBSERVER_HOST:$IM_WEBSERVER_PORT
   ```

   For example:

   ```
   49916:company22.example.com:80
   ```

   where,

   IM_DOWNLOAD_PORT. The port on which Instant Messaging resources are downloaded using Netlet.

   IM_WEBSERVER_HOST. The host name of the web server serving the Instant Messenger. For example. `company22.example.com`

   IM_WEBSERVER_PORT. The port number of the web server serving the Instant Messenger. For example, `80`.
10. Select the default value in the Port-Host-Port List and click Remove.
11. Enter the local host port on which Netlet will run in the Client Port field. For example: `49916`.
12. Enter the Instant Messenger host name in the Target Host(s) field.
13. Enter the Instant Messenger port in the Target Port(s) field.

NOTE	The values for Netlet Port, Instant Messaging Host, and Instant Messaging Port should be the same as the Instant Messaging service attributes mentioned in the Instant Messaging service panel as discussed in the final steps of "To Configure the Instant Messaging Channel" on page 317.

14. Click Add to List.
15. Click Save to save the Netlet Rule.

Disallowing Users from Launching Instant Messenger

You can remove the ability for users to use the Instant Messaging channel by removing the channel from the user's display profile. For example, to remove the sample IMChannel that is automatically installed, do the following:

1. From an Internet browser, log into the Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Identity Management tab to display the View drop down list in the navigation pane.
3. Select Services in the View drop down list to display the list of configurable services.
4. Click the arrow icon next to the Portal Desktop service.
5. Click the Container and Channel Management Link.
6. Select the check box to the left of the IMChannel channel.
7. Scroll as needed and click Delete to delete the channel.

Configuring the Address Book Channel

For the Address Book channel to work, you need to configure the defaults for the Address Book service. For the Address Book channel to run as efficiently as possible, you need to set the appropriate minimum and maximum LDAP connection pool size, according to the needs of your site.

Configuring the Address Book Service Defaults

Setting the Minimum and Maximum LDAP Connection Pool Size

Configuring the Address Book Service Defaults

This section provides information about single sign-on (SSO) Adapter templates. These templates globally affect the display of the communication channels on users' portal Desktops. To alter the display profile of users for the communication channels, you will need to edit or create SSO Adapter templates and configurations.

This chapter only discusses templates for Address Book. Even for Address Book, the discussion here is very specific. For a broader explanation of SSO Adapters, SSO Adapter templates, and SSO Adapter configurations, see Appendix H, "SSO Adapter Templates and Configurations" on page 537.

To Configure the Address Book Service Defaults

1. From an Internet browser, log into the Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Service Configuration tab to display the list of configurable services in the navigation pane.
3. Scroll down the navigation pane to the Single Sign-on Adapter Configuration heading and click the arrow next to the item SSO Adapter, which brings up the SSO Adapter page in the data pane.
4. Click the string with the protocol Lightweight Directory Access Protocol (LDAP) following "`default|`". Find this string in the box labeled SSO Adapter Templates under the heading Global as opposed to Dynamic:

   `default|ldap://`...

   This string appears among other strings, such as "`default|pop3:///`...","`default|imap:///`...", and "`default|http:///`...."

   Clicking the "`default|ldap://...`" string selects the string and places a copy of it in the field below—the configuration description field—allowing you to edit the string. The configuration description field is above the Add and Remove buttons.
5. With the "`default|ldap://...`" string showing in the configuration description field, click inside the field.

   Code Example 12-2 displays the complete default SSO Adapter Template string as it appears in the configuration description field before editing. This description appears in the field as one long string; however, for readability purposes, it has been divided here into separate lines where line breaks have been added preceding each ampersand (&).

Code Example 12-2 Address Book SSO Adapter Template Before Editing

```
default|ldap://[SERVER-NAME:PORT]/?configName=[SUN-ONE-ADDRESS-BOOK]
    &pabSearchBase=[PAB-SEARCH-BASE]
    &userSearchBase=[USER-SEARCH-BASE]
    &aid=[ADMIN-ID]
    &adminPassword=[ADMIN-PASSWORD]
    &imapHost=[IMAP-HOST]
    &imapPort=[IMAP-PORT]
    &clientPort=[CLIENT-PORT]
    &enableProxyAuth=false
    &proxyAdminUid=[PROXY-ADMIN-UID]
    &proxyAdminPassword=[PROXY-ADMIN-PASSWORD]
    &userAttribute=uid
    &type=AB-TYPE
    &subType=sun-one
    &ssoClassName=com.sun.ssoadapter.impl.LDAPABSSOAdapter
    &encoded=password
    &default=ssoClassName
    &default=host
    &default=port
    &default=pabSearchBase
    &default=userSearchBase
    &default=aid
    &default=adminPassword
    &default=imapHost
    &default=imapPort
    &default=clientPort
    &default=type
    &default=subType
    &default=enableProxyAuth
    &default=proxyAdminUid
    &default=proxyAdminPassword
    &default=userAttribute
    &merge=uid
    &merge=password
    &default=enablePerRequestConnection
    &enablePerRequestConnection=false
```

6. In the configuration description field, replace the bracketed values in the string as detailed in Table 12-2 on page 326 by selecting a bracketed value, such as `[SERVER-NAME:PORT]` and typing the specific replacement information, such as `psserver.company22.example.com:389`.

7. After replacing all the bracketed values in the string, click Add.

   This action places your newly edited "`default|ldap://...`" string in the SSO Adapter Template box among the other strings, including the original "`default|ldap://...`" string.

8. If the original "`default|ldap://...`" string—the string with the bracketed values—is not currently selected, select it now. Ensure that it is the only string selected.
9. Click Remove to remove the original "`default|ldap://...`" string.
10. Scroll down the SSO Adapter page and click Save.

    For more information about the attributes in an SSO Adapter template string, see Appendix H, "SSO Adapter Templates and Configurations" on page 537.

Table 12-2 Details of the Address Book SSO Adapter Template String Example

Parameter	Value

SERVER-NAME: PORT	Replace this string with the name and port number of the user or group directory server associated with the messaging server. For example:
	`psserver.company22.example.com:389`
	The server name you enter to replace `SERVER-NAME` in the bracketed value `[SERVER-NAME:PORT]` is usually the same server name you enter to replace the bracketed value `[IMAP-HOST]`.
	Though unlikely, it is possible for these two hosts to be different. They might be different if a different IMAP host has been designated as one whose authentication applies to Personal Address Book (PAB).
	To change the port number from `389` to another number, such as `390`:
	1. Enter `390` to replace PORT in the bracketed value [SERVER-NAME:PORT]. The server name and port example given at the beginning of this table would then appear as follows:
	`psserver.company22.example.com:390`
	2. Append the following to the Address Book SSO Adapter template string:
	`&default=port&port=390`
	This action would change the template string shown in Code Example 12-2 on page 325 to end as follows:
	`...merge=uid&merge=password&default=port&port=390`

Table 12-2 Details of the Address Book SSO Adapter Template String Example

Parameter	Value
SUN-ONE-ADDRESS-BOOK	Replace this string with the following: `sunOneAddressBook` It is the same value that appears in the Dynamic SSO Adapter configuration as: `configDesc`=SUN-ONE-ADDRESS-BOOK Specifically, It appears in the following string: `undef:///?configName=sunOneAddressBook&configDesc`=SUN-ONE-ADDRESS-BOOK
PAB-SEARCH-BASE	Replace this string with the PAB search base. The search base is the point from which the Personal Address Book search should begin. For example: `o=pab`.
USER-SEARCH-BASE	Replace this string with the user search base. For example: `o=example.com`
ADMIN-ID:	Replace this string with the PAB LDAP administrator's distinguished name (DN). For example: `uid=msg-admin,ou=People, o=company22.example.com,o=example.com`
ADMIN-PASSWORD	Replace this string with the password for the PAB Admin ID. For example: `admin` However, this is not an encrypted password. For information on how to use an encrypted password for the `adminPassword`, see Appendix A "SSO Adapter Templates and Configurations," the entry titled "encoded" in Table 14-27 on page 543.
IMAP-HOST	Replace this string with the Internet Messaging Access Protocol (IMAP) host name of the messaging server with the appropriate value. For example: `psserver.company22.example.com` The name of this server is usually the same as the one used for [SERVER-NAME: PORT].
IMAP-PORT	Replace this string with the IMAP port number. For example: `143`

Table 12-2 Details of the Address Book SSO Adapter Template String Example

Parameter	Value
CLIENT-PORT:	Replace this string with the HTTP port number on which the messaging solution server is running. For example: `1080`
PROXY-ADMIN-UID	Replace this string with the proxy administrator's User ID. For example: `msg-admin` If you do not have proxy authorization enabled, you can keep this bracketed value in the string as a placeholder, as shown in Code Example 12-3 on page 328.
PROXY-ADMIN-PASSWORD	Replace this string with the proxy administrator's password. For example: `mailpwd` If you do not have proxy authorization enabled, you can keep this bracketed value in the string as a placeholder, as shown in Code Example 12-3 on page 328.

Code Example 12-3 on page 328 displays the Address Book SSO Adapter Template string after the configuration details have been added. In this example, proxy authentication is not enabled. For more information on proxy authentication, see "Administrator Proxy Authentication: Eliminating End-User Credential Configuration" on page 339.

Code Example 12-3 Address Book SSO Adapter Template After Editing

```
default|ldap://company22.example.com/?configName=sunOneAddressBook
    &pabSearchBase=o=pab
    &userSearchBase=o=example.com
    &aid=uid=msg-admin,ou=People,o=company22.example.com,o=example.com
    &adminPassword=admin
    &imapHost=imserver.company22.example.com
    &imapPort=143
    &clientPort=1080
    &enableProxyAuth=false
    &proxyAdminUid=[PROXY-ADMIN-UID]
    &proxyAdminPassword=[PROXY-ADMIN-PASSWORD]
    &userAttribute=uid
    &type=AB-TYPE
    &subType=sun-one
    &ssoClassName=com.sun.ssoadapter.impl.LDAPABSSOAdapter
    &encoded=password
    &default=ssoClassName
    &default=host
    &default=port
```

Code Example 12-3 Address Book SSO Adapter Template After Editing

```
&default=pabSearchBase
&default=userSearchBase
&default=aid
&default=adminPassword
&default=imapHost
&default=imapPort
&default=clientPort
&default=type
&default=subType
&default=enableProxyAuth
&default=proxyAdminUid
&default=proxyAdminPassword
&default=userAttribute
&merge=uid
&merge=password
&default=enablePerRequestConnection
&enablePerRequestConnection=false
```

Setting the Minimum and Maximum LDAP Connection Pool Size

This section provides information about the connection pool for the Address Book provider and applies to all of the supported providers: Sun ONEAddress Book Provider, Microsoft Exchange Address Book Provider, and Lotus Notes Address Book Provider.

The connection pool maintains a set number of LDAP connections, which is never less than the value for the `connPoolMin` property and never more than the value for the `connPoolMax` property. The default values for these two properties are set in the `LdapABConstants.java` file as follows: `connPoolMin=5` and `connPoolMax=20`. However, to adjust these values, you should edit the SSO Adapter. Since it is most likely that you will want to set the LDAP connections at the global level, you should edit these properties in an Address Book SSO Adapter template as described subsequently in "To Set the Minimum and Maximum LDAP Connection Pool Size".

When an end user launches the address book, the connection pool looks to see if it has an available LDAP connection. If a connection is available, it is given to the end user. If a connection is not available and the value for the `connPoolMax` property has not been met, then a new LDAP connection is created and given to the end user. If a connection is not available and the value for the `connPoolMax` property has been met, then the end user must wait until a connection opens in the pool. If the timeout value is reached before a connection opens, then the end user fails to contact the address book server on that attempt.

Therefore, set the minimum number of connections in the pool to a number that is large enough to avoid the constant creation of new LDAP connections, since creating connections slows performance. However, do not set the minimum number to a number that is so large it significantly reduces the overall availability of hardware resources.

Furthermore, set the maximum number of connections in the pool to a number that is large enough to ensure that most end users get an LDAP connection without a wait, but small enough so as not to unnecessarily reduce the availability of hardware resources. The goal is to find a happy medium that doesn't strain the hardware, but allows end users quick access to their address books.

To Set the Minimum and Maximum LDAP Connection Pool Size

1. From an Internet browser, log into the Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Service Configuration tab to display the list of configurable services in the navigation pane.
3. Scroll down the navigation pane to the Single Sign-on Adapter Configuration heading and click the arrow next to the item SSO Adapter, which brings up the SSO Adapter page in the data pane.
4. Click the string with the protocol Lightweight Directory Access Protocol (LDAP) following "`default|`". Find this string in the box labeled SSO Adapter Templates under the heading Global as opposed to Dynamic:

   `default|ldap://`...
5. With the "`default|ldap://...`" string showing in the configuration description field, click inside the field.
6. Navigate to the end of the string and enter the proper settings for your site. Code Example 12-4 provides an example, where the minimum number of LDAP connections is set to 7, the maximum number of connections is set to 1,000, and the timeout is set to 180 seconds.

Code Example 12-4 LDAP Connection Code to Add to SSO Adapter Template

```
default=connPoolMin
    &connPoolMin=7
    &default=connPoolMax
    &connPoolMax=1000
    &default=timeout
    &timeout=180
```

7. Click Add.

   This action places your newly edited "`default|ldap://...`" string in the SSO Adapter Template box among the other strings, including the original "`default|ldap://...`" string.

8. If the original "`default|ldap://...`" is not currently selected, select it now. Ensure that it is the only string selected.

9. Click Remove to remove the original "`default|ldap://...`" string.

Configuring End-User Channel Settings

1. Log into the Desktop as the new user:

   a. From an Internet browser, go to:

      `http://`*`hostname.domain`*`:port/portal/dt, for example`
      `http://psserver.company22.example.com:80/portal/dt`

   b. Enter the user ID and password.

   c. Click Login.

2. Click the Edit button of each channel to configure the server settings.

   - To configure the Mail channel settings:

     Server Name. Enter the host name of the mail server. For example, `mailserver.example.com`.

     IMAP Server Port. Enter the mail server port number.

     SMTP Server Name. Enter the name of the Domain Name Server (DNS) of the outgoing mail—Simple Mail Transfer Protocol (SMTP)—server.

     Client Port. Enter the port number configured for HTTP service.

     User Name. Enter the mail server user name.

     User Password. Enter the mail server user password.

     When sending a message place a copy in Sent Folder. Check this box to store copies of your outgoing messages in the Sent folder.

     Finished. Click this button to save the mail configuration.

     Cancel. Click this button to close the window without saving the configuration details.

- To configure Address Book channel settings:

  The IMAP user ID and Password are the same as the User Name and User Password entered when configuring the mail channel settings. For details, refer to the previous bulleted item,"To configure the Mail channel settings:"

  IMAP User ID. Enter your IMAP User ID.

  IMAP Password. Enter you IMAP Password.

  Finished. Click this button to save the server information.

  Cancel. Click this button to close the window without saving the details.

- To configure the Calendar channel settings:

  Server Name. Enter the calendar server host name. For example, `Calserver.example.com`.

  Server Port. Enter the calendar server port number.

  User Name. Enter the calendar server user name.

  User Password. Enter the calendar server user password.

  Finished. Click this button to save the calendar configuration.

  Cancel. Click this button to close the window without saving the details.

- To configure the Instant Messaging channel settings:

  Contact List. Select the desired contact list whose contacts will be displayed in the Instant Messaging Channel.

  Launch Method. Select the desired launch method:
  `Java Plugin` or `Java Web Start`.

  Server. Enter the Sun ONE Instant Messaging Server name. For example:
  `IMserver.example.com`

  Server Port. Enter the Sun ONE Instant Messaging Server port number. For example:
  `49999`

  Multiplexor. Enter the Multiplexor name, which must be the same machine as the Sun ONE Instant Messaging server. For example:
  `IMserver.example.com`

Multiplexor Port. Enter the Multiplexor port number. For example: `49909`

User Name. (This field only appears when the authentication method is set to the Sun ONE Identity Server authentication method, `idsvr`) Enter the Sun ONE Instant Messaging user name.

User Password. (This field only appears when the authentication method is set to the Sun ONE Identity Server authentication method, `idsvr`) Enter the Sun ONE Instant Messaging user password.

Finished. Click this button to save the Sun ONE Instant Messaging Server configuration.

Cancel. Click this button to close the window without saving the details.

NOTE The Address Book, Calendar, and Mail channels each have display options that can be set by the user and by default cannot be overwritten by an administrator. After logging into the Portal Desktop, the user can change the display options for a channel by clicking the edit button in the panel for that channel. The display options are clearly marked and easily changed.

In Address Book, a display option that users can change is the Number of Entries option; in Calendar, a display option that users can change is the Display Day View option; in Mail, a display option that users can change is the Number of Headers option.

Changes made by users to the default communication channels display options take precedence. Any future changes made by administrators will not automatically take effect and a new channel added by administrators will not automatically be accessible by users. To make administrators' changes visible and accessible by users, see "Some Users Won't See Configuration Changes" on page 557 for more information.

Application Preference Editing: Configuring Communication Channel Edit Pages

You can configure the edit pages that end users will see after they click the edit button in a communication channel's toolbar for the Address Book, Calendar, and Mail channels. The Instant Messaging channel does not use application preference editing. For information about configuring the Instant Messaging Channel's edit page, see *Sun ONE Portal Server 6.2 Desktop Customization Guide*.

For the three communication channels that allow application preference editing, you can change which options are available for end users to edit, the names and wording that accompany those options, and the way the options are formatted. Configuration of the communication channels edit pages can be performed in the display profile, various HTML templates, and an SSO Adapter template. You might also need to access an SSO Adapter configuration. These items together are involved in the configuration of the edit pages.

This section gives only a brief explanation of application preference editing. Other chapters in this guide and the *Sun ONE Portal Server 6.2 Desktop Customization Guide* provide a more complete explanation of the template files and the display profile, including how they interact with each other and how you can access and edit them.

Display Profile Attributes for the Edit Pages

The communication channels have two collections in their display profile that drive the creation of the edit pages, `ssoEditAttributes` and `dpEditAttributes`.

You can edit these collections by accessing the Sun ONE Identity Server admin console. Either download the display profile—to edit the XML code before uploading it back to the directory server—or edit specific properties in these collections using only the admin console.

The `ssoEditAttributes` collection controls the editing of the attributes contained by the SSO Adapter service—such as `user name` and `user password`—while `dpEditAttributes` controls the editing for the display profile attributes—such as `sort order` and `sort by`, which are options that by default are editable by end users.

Therefore, these collections list the attributes that can be edited and also contain information on the type of input and the header for the input string to use. For example:

```
<String name="uid" value="string|User Name:"/>
<String name="password" value="password|User Password:"/>
```

The name in the collection must match the name of the corresponding display profile SSO Adapter attribute. The value portion of the item contains two pieces of information separated by the "|" character. The first part of the value string specifies what the display type is for the attribute. The second part of the attribute's value string specifies the text that will be displayed next to the item. The list below specifies how the type relates to a corresponding HTML GUI item:

- string - Creates a text field where alphanumeric characters can be entered
- password - Creates a password field where the input is replaced with "*"
- check - Creates a checkbox
- select - Creates a select box. Every select item must have a corresponding collection with a list of values and display text

For every select display type you must have a corresponding collection that lists the value to be returned and the display value for the option. The collection name must be made up of the name value for the attribute and the text `SelectOptions`. For example, for the `sortOrder` attribute in the `MailProvider`, the collection name is `sortOrderSelectOptions`:

```
<Collection name="sortOrderSelectOptions" advanced="false"
merge="replace" lock="false" propagate="true">
    <String name="top" value="Most recent at top"/>
    <String name="bottom" value="Most recent at bottom"/>
</Collection>
```

HTML Templates for the Edit Pages

There are nine HTML templates used to create the edit pages for the communication channel providers. The templates were created to be very generic in order to correspond to specific browser GUI types. They mostly relate to specific HTML inputs in the edit pages. The `edit-start.template` and the `edit-end.template` are exceptions in that they contain most of the HTML that is

used for page layout. Table 12-3 on page 336 contains a description of each template name and how it relates to the GUI types. Some of the templates are used to start, end and separate the attributes. These templates are available for each of the communication channels at:

`/etc/opt/SUNWps/desktop/default/`*`ChannelName_Provider`*`/html`

For example, the templates for the Calendar channel edit pages can be accessed at:

`/etc/opt/SUNWps/desktop/default/CalendarProvider/html`

Table 12-3 Templates for the Communication Channel Edit Pages

Template	Description
edit-start.template	Provides the starting HTML table for the edit page.
edit-checkbox.template	Provides a generic template for checkbox items.
edit-separate.template	Separates the display profile attributes from the SSO attributes.
edit-end.template	Ends the HTML table for the edit page.
edit-password.template	Provides a generic template for password items.
edit-string.template	Provides a generic template for text items.
edit-select.template	Provides a generic template for a select item.
edit-selectoption.template	Provides a generic template for a select option. This way the option can also be generated dynamically from the display profile.
edit-link.template	Provides a template to generate the link so the user can edit their client's display attributes.

A Display Profile Example

This example demonstrates how certain SSO Adapter attributes work together with their corresponding display profile attributes to give end users the ability to change the entries for specific features in a communication channel's edit page, thereby changing how the communication channels are configured and displayed on their Portal Desktops.

The SSO Adapter template in Code Example 12-5 on page 337 is for a sample mail channel. The SSO Adapter template contains two merged attributes:

- uid - User ID
- password - User password

A merged attribute is an attribute that end users can specify. Administrators decide which attributes are merged, therefore, which attributes they want end users to be able to edit.

Code Example 12-5 Sample SSO Adapter Template

```
default|imap:///&configName=MAIL-SERVER-TEMPLATE
    &encoded=password
    &default=protocol
    &default=clientProtocol
    &default=type
    &default=subType
    &default=ssoClassName
    &default=smtpServer
    &default=clientPort
    &default=host
    &default=port
    &merge=username
    &merge=userpassword
    &clientProtocol=http
    &type=MAIL-TYPE
    &subType=sun-one
    &ssoClassName=com.sun.ssoadapter.impl.JavaMailSSOAdapter
    &smtpServer=example.sun.com
    &clientPort=80
    &host=company22.example.com
    &port=143
```

Code Example 12-6 on page 338 contains the channel's display profile XML fragment for the channel's `ssoEditAttributes`.

After administrators have set an attribute to `merge` in an SSO Adapter template, they can then edit that attribute in the display profile in order to reconfigure how the attribute is displayed to end users in an edit page and how end users can edit it. Administrators can decide how end users are queried for the necessary information by editing the proper display profile collection. For example, in this example, administrators could replace `User Name` with the question, `What is your user name`? The use of the `string` attribute display type before the "`|`" symbol is the most likely choice. However, it's possible for an administrator to change this to the `password` type or to another type.

Code Example 12-6 Sample Mail Channel Display Profile XML Fragment

```
<Channel name="SampleMailChannel" provider="MailProvider">
<Properties>
<Collection name="ssoEditAttributes">
    <String name="username" value="string|User Name:"/>
    <String name="userpassword" value="password|User Password:"/>
</Collection>
```

For this example, in the Mail channel edit page, end users will see text fields titled:

- User Name:
- User Password:

Enabling End-Users to Set Up Multiple Instances of a Communication Channel Type

Multiple types of communication channels can be created by end users or administrators. For end users to create multiple types of communication channels, they will need to utilize the Create a new channel link found on the Content page.

Administrators can create multiple channels for an organization, role, or group. After administrators have made multiple instances of a particular component available—for example, a second instance of the address book component—they can allow end users to configure a second Address Book channel on their Portal Desktops.

Administrators can create an SSO Adapter template for each new communication channel type or they can use one SSO Adapter template and create multiple SSO Adapter configurations for each channel. For more information, see the SSO Adapter documentation in Appendix H, “SSO Adapter Templates and Configurations” on page 537.

Depending on the amount of configuration done by the administrator, the end users may not need to enter as many configuration settings. Administrators can configure these settings by utilizing the application preference editing feature (see “Application Preference Editing: Configuring Communication Channel Edit Pages” on page 334).

To create two Address Book channels, you make each refer to a different SSO adapter template. You can then add both Address Book channels to the visible page you just came from. Likewise, you can create one SSO Adapter template and two SSO Adapter configurations (dynamic). The SSO Adapter template would define the server settings as user definable values (`merge`) and the SSO Adapter configuration would then specify those server settings.

To configure the address book for different servers where end users can configure the servers as needed:

1. Specify the server information as user definable, merge, in the SSO Adapter template. For more information, see Appendix H, "SSO Adapter Templates and Configurations" on page 537.
2. Specify which attributes are editable in the channel's display profile `ssoEditAttributes` collection. For more information, see "Application Preference Editing: Configuring Communication Channel Edit Pages" on page 334 and for specific information about the display profile, see the *Sun ONE Portal Server 6.2 Desktop Customization Guide*.

Administrator Proxy Authentication: Eliminating End-User Credential Configuration

You can enable administrator proxy authentication for the Address Book, Calendar, and Mail channels. Extending support for proxy authentication between the Sun ONE Portal Server and Sun ONE Messaging Services (Messaging Server and Calendar Server) eliminates the need for end users to visit a channel's edit page in order to enter their credentials: user name and user password. An administrator's credentials are used instead of end-users' credentials and they are stored in the SSO Adapter template. Within the template, the administrator's User ID is stored as a value for the `proxyAdminUid` attribute while the administrator's password is stored as a value for the `proxyAdminPassword` attribute. Every time a user launches a channel, these values are used to make a connection between a channel and its respective back-end server. A naming attribute for the user is also sent to the back-end server. For more information on the use of naming attributes for administrator proxy authentication, see the `userAttribute` property in Table 12-4 on page 341.

Proxy authentication cannot be configured for Sun ONE Instant Messaging Server, Microsoft Exchange Server, or IBM Lotus Notes server.

CAUTION—Potential for Multiple End Users to be Directed to One Mail Account
Identity Server and Portal Server allow administrators to set up users with the same User ID across an organization. For example, the organization could have two suborganizations that each have an end user named `enduser22`. If administrator proxy authentication is enabled for a Sun ONE communication channel and the end user naming attribute is set to the default, `uid`, then both users could potentially access the same back-end user account. Administrator proxy authentication enables administrators to change the user naming attribute in the SSO Adapter template. For example, you can change the attribute to an attribute that is unique for each employee, such as employee number, to ensure that portal end users access the correct back-end server account.

Overview of How to Configure Proxy Authentication

In order to enable administrator proxy authentication for the Address Book, Calendar, and Mail channels, you need to access the SSO Adapter templates through the Sun ONE Identity Server admin console and you need to access the Sun ONE communication servers. More specifically, you need to:

- Edit SSO Adapter Templates.

  In the SSO Adapter Templates, you need to edit the strings that apply to the Address Book, Calendar, and Mail channels. One of the distinguishing factors of the strings is the protocol used:

  - The Address Book channel uses the LDAP protocol
  - The Calendar channel uses the HTTP protocol
  - The Mail channel uses the IMAP or POP protocol.

- access Sun ONE Messaging Server to enable proxy authentication for the Address Book and Mail channels
- access Sun ONE Calendar Server to enable proxy authentication for the Calendar channel.

Proxy Authentication and Single Sign-On (SSO) Adapter Templates

To Edit SSO Adapter Templates For Enabling Administrator Proxy Authentication

1. From an Internet browser, log into the Sun ONE Identity Server admin console at `http://`*`hostname:port`*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

2. Click the Service Configuration tab to display the list of configurable services in the navigation pane.
3. Select SSO Adapter under Single Sign-On Adapter Configuration to display the page for configuring the SSO Adapter in the data pane.
4. Click the string for the channel that you want to enable with administrator proxy authentication. The strings are in the box labeled SSO Adapter Templates under the heading Global as opposed to Dynamic. For more information about the relationship between the template strings and the communication channels, see "About SSO Adapter Templates" on page 539. Clicking a string makes it appear in the configuration description field—which is just above the Add and Remove buttons.
5. Click in the configuration description field.
6. Delete and key in the necessary information for administrator proxy authentication:

   Table 12-4 describes the properties that need to be edited in the SSO Adapter Template to enable support for administrator proxy authentication.

Table 12-4 SSO Adapter Template Properties for Administrator Proxy Authentication

Property	Value	Description
`enableProxyAuth`	`true` | `false`	The value associated with this attribute is a flag to indicate if proxy authentication is enabled or not. If true the SSO Adapter and Application Adapter will perform proxy authentication. For example, `&enableProxyAuth=true`
`proxyAdminUid`	(configurable)	The value associated with this attribute is the administrator's user name. For example, `&proxyAdminUid=ServiceAdmin`
`proxyAdminPassword`	(configurable)	The value associated with this attribute is the administrator's user password. For example, `&proxyAdminPassword=mailpwd`

Table 12-4 SSO Adapter Template Properties for Administrator Proxy Authentication

Property	Value	Description
`userAttribute`	(configurable)	The value associated with this attribute is the user's naming attribute. This value is mapped to an attribute on the user's record (the user's entry in the directory). A typical record has several attributes, including the User ID (uid) and employee number. By default, the naming attribute is set to uid. For example, &userAttribute=uid By editing the SSO Adapter template, you can map the naming attribute to another attribute, such as employee number.

The preceding four properties appear in the SSO Adapter template string again, as shown subsequently. The configuration of the properties can be set to either `default` or `merge`. In the following examples, they are all set to default.

Property	Value	Example
`enableProxyAuth`	default	&default=`enableProxyAuth`
`proxyAdminUid`	default	&default=proxyAdminUid
`proxyAdminPassword`	default	&default=proxyAdminPassword
`userAttribute`	default	&default=userAttribute

Code Example 12-7 contains a fully configured example of a mail SSO Adapter template for proxy authentication.

Code Example 12-7 Sample Mail SSO Adapter Template with Proxy Authentication

```
default|imap:///?configName=SUN-ONE-MAIL
    &encoded=password
    &default=protocol
    &default=clientProtocol
    &default=type
    &default=subType
    &default=enableProxyAuth
    &default=proxyAdminUid
    &default=proxyAdminPassword
    &default=ssoClassName
    &default=host
    &default=port
    &merge=uid
    &default=smtpServer
    &default=clientPort
```

Code Example 12-7 Sample Mail SSO Adapter Template with Proxy Authentication

```
&clientProtocol=http
&enableProxyAuth=true
&proxyAdminUid=ServiceAdmin
&proxyAdminPassword=mailpwd
&host=example.sun.com
&port=143
&smtpServer=example.sun.com
&clientPort=80
&type=MAIL-TYPE
&subType=sun-one
&ssoClassName=com.sun.ssoadapter.impl.JavaMailSSOAdapter
&default=enablePerRequestConnection
&enablePerRequestConnection=true
&default=userAttribute
&userAttribute=uid
```

Proxy Authentication and Communication Servers

Setting Up Sun ONE Messaging Server for Administrator Proxy Authentication

1. Log in to the Sun ONE Messaging Server software host and become super user.
2. Type the following code:

   ***messaging-server-install-dir*/*msg-instance-name*/`configutil -o service.http.allowadminproxy -v yes`**
3. Restart the Messaging Server.

   See the *Sun ONE Messaging Server Administrator's Guide* for detailed instructions on running `configutil` and restarting the server.

Setting Up Calendar Server for Administrator Proxy Authentication

1. Log in to the Sun ONE Calendar Server software host and become super user.
2. Open the following file with the editor of your choice:

   calendar-server-install-dir`/cal/bin/config/ics.conf`
3. Set the following attribute as shown:

   ```
   service.http.allowadminproxy = "yes"
   ```
4. Restart the calendar server.

   See the *Calendar Server Administrator's Guide* for detailed instructions on restarting the server.

Configuring a Read-Only Communication Channel for the Authentication-Less Portal Desktop

The authentication-less (authless anonymous) Portal Desktop supports read-only communication channels.

Read-Only Communication Channels Facts and Considerations

You can configure read-only access to Address Book, Calendar, and Mail channels for the authless anonymous Portal Desktop. End users can access the information in a read-only communication channel by simply accessing the Portal Desktop; therefore, by entering the following URL in an Internet browser:

`http://`*`hostname.domain`*`:port/portal/dt, for example`
`http://psserver.company22.example.com:80/portal/dt`

Without logging in, end users have access to any read-only communication channels that administrators have configured. However, end users are usually prevented from editing these channels. For more information about the authentication-less Portal Desktop, including enabling anonymous log in, see the *Sun ONE Portal Server 6.2 Desktop Customization Guide.*

The calendar channel is the channel most commonly shared by multiple users; therefore, the following steps are for configuring a read-only calendar channel. In this example, the calendar being shared belongs to user *library*. The public read-only calendar is titled *Library Schedule.* The following calendar set up demonstrates one possible approach. For more information about setting up users for the Sun ONE Calendar Server, see the `create` *userid* option of the `csuser` command in the *Sun ONE Calendar Server Administrator's Guide*

To Set Up a Calendar User

1. Create a calendar user by issuing a command such as the following:

   ```
   csuser -g Library -s Admin -y libadmin -l en -m
   libadmin@library.com -c librarySchedule create libadmin
   ```

   Where user **`libadmin`** has a given name of **`Library`**, surname of **`Admin`**, password of **`libadmin`**, preferred language of **`en`** (English), email address of **`libadmin@library.com`**, and calendar ID of **`librarySchedule`**.

2. Set the access permissions to world readable for:

   `libadmin:librarySchedule`

   You can set the access permissions using the `cscal` utility or the end user can do this using Calendar Express.

To Configure a Read-Only Communication Channel

1. Configure the settings for the end user—which in this case is authless anonymous—and create a calendar SSO adapter configuration.

   a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

   b. Click the Identity Management tab to display the View drop down list in the navigation pane.

   c. Click Users in the View drop down list.

   d. Scroll down as needed to the authless anonymous user and click the accompanying arrow to bring up the authlessanonymous page in the data pane.

      Now you can add the SSO Adapter service to the authless anonymous user.

   e. Click Services in the View drop down list within the authlessanonymous page to display the available services.

   f. Click Add.

   g. Click the checkbox for SSO Adapter

   h. Click Save.

2. Create a calendar SSO Adapter configuration for the authless anonymous user.

   a. If not already logged in, log into the Sun ONE Identity Server admin console.

   b. Click the Identity Management tab to display the View drop down list in the navigation pane.

   c. Select Services in the View drop down list to display the list of configurable services.

d. Scroll down the navigation pane to the Single Sign-on Adapter Configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

e. In the blank configuration description field, type in a group-oriented SSO Adapter configuration string (with a User ID and password). A typical configuration has been provided subsequently for your reference. The attributes available in this string can vary depending upon how you configured the Sun ONE Portal Server SSO Adapter template. By default the SSO Adapter template expects the user to specify the following information:

- host
- port
- client port
- uid
- password

If the configuration description field is not blank when you get to it, select all the text in the field and delete it before entering a string in the following format:

`default|undef://?`*`uid:password@host:port`*`/?`
`configName=`*`configuration-name`*
`&configDesc=`*`configuration-description`*

For example:

```
default|undef://?libadmin:libadmin@example.com:3080/?
configName=sunOneCalendar_librarySchedule
&configDesc=SUN-ONE-CALENDAR
```

f. Click Add.

g. Click Save.

3. Create a new calendar channel for the authless anonymous user that is based on the newly created SSO Adapter configuration.

a. If not already logged in, log into the Sun ONE Identity Server admin console.

b. Click the Identity Management tab to display the View drop down list in the navigation pane.

c. Click Users in the View drop down list.

d. Scroll down as needed to the authless anonymous user and click the accompanying arrow to bring up the authlessanonymous page in the data pane.

Now you can create a new calendar channel for the authless anonymous user.

e. Click Portal Desktop in the View drop down list within the authlessanonymous page to display the Edit link.

f. Click the Edit link.

g. Click the Channel and Container Management link.

h. Scroll down to the Channels section and click New.

i. Enter a name in the Channel Name field. For example:

`LibraryScheduleChannel`

j. Choose the correct provider from the provider drop down list. For this example the correct provider is Calendar Provider.

k. Click OK, which returns you to the Channel and Container Management page.

Now you can edit the channel properties.

l. Scroll down to the Channels section and click Edit Properties next to your newly created channel. For example:

`LibraryScheduleChannel`

m. Edit fields as appropriate. For example:

- title: Library Schedule
- description: Library Schedule
- ssoAdapter: sunOneCalendar_librarySchedule
- loadSubscribedCalendars: false (no checkmark)
- is editable: false (no checkmark)

n. Scroll as needed and click Save.

Now you can add the new calendar channel to Portal Desktop of the Authless Anonymous user.

o. Near the top of the page, click Top, which returns you to the Channel and Container Management page.

p. Scroll down the Container Channels section and click the link for the container that you want to add the new channel to. For example, `MyFrontPageTabPanelContainer`. Do not click the accompanying Edit Properties link.

q. Under the Channel Management heading, click the name of the channel you just created. For example, `LibraryScheduleChannel`, which is in the Existing Channels list.

r. Click the Add button that is next to the Available and Visible list. This makes the channel available to users and visible without any further configuration.

s. Scroll back up the page to click Save under the Channel Management heading.

t. Restart the web container.

Configuring Microsoft Exchange Server or IBM Lotus Notes

Besides supporting Sun ONE Messaging Server and Sun ONE Calendar Server for the communication channels, Sun ONE Portal Server 6.2 also supports Microsoft Exchange Server and IBM Lotus Notes server.

You can configure Microsoft Exchange Server to work with Sun ONE Portal Server, giving end users access to the Microsoft Outlook Web Access solution. End users gain this access after clicking Launch Address Book, Launch Calendar, or Launch Mail in the respective channel on Portal Desktop.

Similarly, you can configure IBM Lotus Notes server to work with Sun ONE Portal Server, giving end users access to the IBM Lotus Domino Webmail solution through the Address Book, Calendar, and Mail channels.

NOTE Microsoft Exchange Server and IBM Lotus Notes server do not support administrator proxy authentication or single sign on. Because of the single sign on limitation, when end users launch a channel connected to one of these servers, they will need to reenter their credentials before being connected.

To Configure Microsoft Exchange Server for Address Book, Calendar, and Mail

1. Log into your Primary Domain Controller (PDC) as an administrator of the domain.
2. Select Start, Programs, Administrative Tools, User Manager for Domains and create an account with user name MAXHost.
3. Select Groups and add MAXHost to the groups, Administrators, and Domain Admins.
4. Ensure that MAXHost can log on locally to the MAIL_HOST, Domain Controllers, and MAX_HOST.
5. Set the password.
6. Log in to your Exchange 5.5 (MAIL_HOST) as MAXHost.
7. Go to Start, Programs, Microsoft Exchange, Microsoft Exchange Administrator.
8. For each end user, set permissions to the mailbox.
9. To enable the permissions tab, go to Tools, Options, Permissions, and enable Show Permissions Page for All Objects.
10. Double-click on the user name.
11. Select the permissions tab and select Add from the permissions page to add MAXHost and leave role as User.

    Repeat steps 9 through 11 for each user who will be accessing the communication channels.
12. Unzip the `ocxhost.zip` file located in the following directory:

    */Portal-server-install-dir/*SUNWps/export.

    When unzipping the file, you will see the following file format:

```
Archive: ocxhost.zip
creating: ocxhost
creating: ocxhost/international
inflating:ocxhost/international/ocxhostEnglishResourceDll.dll
inflating:ocxhost/ocxhost.exe
```

13. Register ocxhost as follows:

a. Locate the `ocxhost.exe`.

b. Select Start and Run.

c. Type the following in the Run window:

```
ocxhost.exe /multipleuse
```

14. To set the properties of ocxhost utility:

a. Configure the necessary DCOM settings for the ocxhost utility using the `dcomcnfg` utility. That is:

I. Select Start and Run.

II. Type `dcomcnfg` and select OK.

b. In the Distributed COM Configuration Properties dialog box:

I. Select Default Properties tab:

- Check the Enable Distributed COM on the computer check box.
- Set the default Authentication Level to Connect.
- Set the default Impersonation Level to Identify.

II. Select the Applications tab.

III. Double-click the ocxhost utility in the Properties dialog.

The ocxhost properties window is displayed.

IV. Check Run Application on this Computer under the Location tab.

V. Set Use custom access permissions, Use custom launch permissions, and Use custom configuration permissions under the Security tab.

VI. Select Edit for the Access, Launch, and Configuration settings and ensure that the following users are included in the Access Control List (ACL):

- Interactive
- Everyone
- System

VII. Select a User under the Identity tab in the ocxhost properties window.

VIII. Select Browse and locate the MAXHost.

IX. Enter the password and confirm the password.

c. Select OK.

The ocxhost DCOM component is now configured and ready to communicate with the Exchange Servers.

To Configure Lotus Domino Server for Address Book, Calendar, and Mail

1. Open the Lotus Administrator by selecting Start, Programs, Lotus Applications, and Lotus Administrator.
2. Go to Administration, Configuration, Server, Current Server Documents.
3. In the Security tab, set the following settings:
   a. Under Java / COM Restrictions, set Run restricted Java / Javascript / COM and Run unrestricted Java / Javascript / COM to *.
   b. Under Security Settings, set:
      - Compare Notes Public keys against those stored in Directory to No.
      - Allow anonymous Notes connections to No.
      - Check Passwords on Notes IDs to Disabled.
   c. Under Server Access, set Only allow server access to users listed in this Directory to No.
   d. Under Web Server Access, set Web Server Authentication to More Name Variations with lower security.
4. In the Ports tab:
   a. Select the Notes Network Ports tab and ensure that TCPIP is ENABLED.
   b. Select Internet Ports tab and the Web tab.
      I. Ensure that TCP / IP port status is Enabled.
      II. Under Authentication options, ensure that Name and password and Anonymous are Yes.
   c. Select the Directory tab and ensure that:
      - TCP / IP port status is Enabled.
      - Authentication options items Name and Password and Anonymous are Yes.
      - SSL port status is Disabled.

d. Select the Mail tab and ensure that:

- TCP/IP port status is Enabled.
- Authentication options Name and Password and Anonymous are set as follows:

	Mail (IMAP)	Mail (POP)	Mail (SMTP Inbound)	SMTP (Outbound)
Name and Password	Yes	Yes	No	N/A
Anonymous	N/A	N/A	Yes	N/A

e. Select the IIOP tab and ensure that:

- TCP/IP port status is Enabled.
- Authentication options items Name and Password and Anonymous are Yes.
- TCP/IP port number is not set to `0`. It should be `63148`.
- SSL port status is Disabled.

5. Select the Internet Protocols tab and the IIOP sub-tabs. Ensure that the Number of threads is at least 10.

6. Save and close.

7. Restart the server by typing the following in the Domino server console:

```
restart server
```

Restarting the server enables the settings to take effect.

8. Enable DIIOP server by typing the following command in the console:

```
load diiop
```

9. Check to see if `diiop_ior.txt` has been generated at location:

```
C:\Lotus\Domino\Data\domino\html\diiop_ior.txt
```

10. Enable HTTP service by typing the following command in the console:

```
load http
```

- If there is another service using port 80, the HTTP service will not start. Stop the service running on port 80 and retype the following in the console:
  `load http`

  Or

- Use the existing service. To do this, copy the `diiop_ior.txt` file into the root or home directory of the web server running on port 80. You can include both the HTTP service and the DIIOP service in the `notes.ini` file to ensure that both services start when you start the server.

Configuration for Lotus Notes

To access a Lotus Notes system using the Sun ONE Portal Server Mail and Calendar channels, you need to add another file to the Sun ONE Portal Server. This file is called `NCSO.jar`. It must be obtained from the Lotus Notes product CD or the IBM web site.

It is available with the Domino Designer and Domino Server products from IBM in the `domino\java` subdirectory. It is also available in a Web download from the following Web site:

`http://www-10.lotus.com/ldd/toolkits`

Go to the Lotus Domino Toolkit link and then to the Java/Corba R5.0.8 update link.

NOTE The download file is a `.exe` file, which performs the extraction of this file and other files.

Place the `NCSO.jar` file in the global class path of the web container (web server or application server) as described in the subsequent sections about each of the four possible web containers. For three of the four web containers, the `NCSO.jar` file is placed in `/usr/share/lib`. The following table summarizes the steps that follow. The table outlines the process of placing the JAR file in the global class path by indicating where the `NCSO.jar` file can be placed: in the System Classpath or in the Portal WAR. The table also indicates if special instructions are needed. If so, they are included later in this section.

Web Container	System Classpath	Portal WAR	Special Instructions
Sun ONE Web Server	Yes	Yes	N/A

Web Container	System Classpath	Portal WAR	Special Instructions
Sun ONE Application Server	Yes	Yes	N/A
BEA WebLogic Server	Yes	No	How to update system classpath
IBM WebSphere Application Server	No	Yes	How to prune JAR file

For the following steps, you need administrative rights to the web container. Also, you should have access to the web container documentation in order to reference detailed information on various web container processes and commands. For more information concerning the Sun ONE web containers, see *Sun ONE Application Server Administrator's Guide* or *Sun ONE Web Server, Enterprise Edition Administrator's Guide*.

Sun ONE Web Server

Sun ONE Application Server

BEA WebLogic Server

IBM WebSphere Application Server

Sun ONE Web Server

1. Place the `NCSO.jar` in the following Sun ONE Portal Server directory:

   `/usr/share/lib`

2. Update the web container class path to include:

   `/usr/share/lib/NCSO.jar`

   a. Launch the Sun ONE Web Server admin console.

   b. Select the Sun ONE Web Server instance.

   c. Click Manage.

   d. Select the Java tab.

   e. Select the JVM Path Settings.

   f. Add `/usr/share/lib/NCSO.jar` to the classpath suffix.

   g. Select ok

h. Select Apply

3. Restart the Sun ONE Web Server; though often not mandatory, this is a good practice.

Optional Placement of the NCSO.jar File

1. Place the `NCSO.jar` file in the following directory:

   Portal-server-install-dir`/SUNWps/web-src/WEB-INF/lib`

2. Redeploy the web application with the following command:

   ***/Portal-server-install-dir*/SUNWps/bin/deploy redeploy**

3. Restart the web container.

Sun ONE Application Server

1. Place the `NCSO.jar` in the following Sun ONE Portal Server directory:

   `/usr/share/lib`

2. Update the web container class path to include `/usr/share/lib/NCSO.jar` using the Sun ONE Application Server admin console.

   a. Launch the Sun ONE Application Server admin console.

   b. Select the domain.

   c. Select the server instance.

   d. Select the JVM Settings tab in the server instance view.

   e. Select Path Settings under the JVM Settings tab.

   f. Add `/usr/share/lib/NCSO.jar` in the Classpath Suffix list.

   g. Select Save.

   h. Select Apply Changes under the General tab of the instance.

   i. Select Restart.

Optional Placement of the NCSO.jar File

1. Place the `NCSO.jar` file in the following directory:

   Portal-server-install-dir`/SUNWps/web-src/WEB-INF/lib`

2. Redeploy the web application with the following command:

   ***/Portal-server-install-dir*/SUNWps/bin/deploy redeploy**

   Where ***Portal-server-install-dir*** represents the directory in which the Portal Server was originally installed.

3. Restart the web container.

BEA WebLogic Server

1. Place the `NCSO.jar` in the following Sun ONE Portal Server directory:

   `/usr/share/lib`

2. Update the web container class path to include `/usr/share/lib/NCSO.jar` using the command line.

   a. Change directories to the web container install directory:

      web-container-install-dir`/bea/wlserver6.1/config`

      Where *web-container-install-dir* represents the directory in which the web container was originally installed.

   b. Change directories to the directory that contains the domain instance:

      `mydomain`

   c. Edit the `startWebLogic.sh` file using the editor of your choice.

   d. Add `/usr/share/lib/NCSO.jar` to the end of the CLASSPATH.

NOTE The `startWebLogic.sh` file may contain multiple CLASSPATH definitions. Locate the last definition of the variable and add the following string to the very end of the CLASSPATH:

`/usr/share/lib/NCSO.jar`

---

   e. Restart the web container.

IBM WebSphere Application Server

1. Prune the classes under `org/w3c/dom/` and `org/xml/sax/` from the `NCSO.jar` file and rejar.

   The classes should include the following:

   - `org/w3c/dom/Document.class`

- `org/w3c/dom/Node.class`
- `org/xml/sax/InputSource.class`
- `org/xml/sax/SAXException.class`

There are many ways to perform this task. Two examples are provided for you here. Follow the method that suits you best:

- The following method requires you to manually unjar and rejar the file:

a. Download and place the file in the following directory:

`/tmp/ncsoprune/work`

b. Unjar the file while it is in that directory.

c. Remove the preceding four classes.

d. Rejar the file.

- The following method requires you to run a script that automates the jar and unjar logic.

a. Download and place the file in the following directory:

`/tmp/ncsoprune/work`

b. Run the following script:

```
#!/bin/ksh
JAR=/usr/j2se/bin/jar
JAR_FILE=NCSO.jar
RM=/usr/bin/rm
BASE_DIR=/tmp/ncsoprune
WORK_DIR=${BASE_DIR}/work
# cd to director of jar file
cd $WORK_DIR
# unjar
$JAR xvf $JAR_FILE
# prune classes
$RM $WORK_DIR/org/w3c/dom/Document.class
$RM $WORK_DIR/org/w3c/dom/Node.class
$RM $WORK_DIR/org/xml/sax/InputSource.class
$RM $WORK_DIR/org/xml/sax/SAXException.class
# jar
$JAR cvf $BASE_DIR/$JAR_FILE META-INF com lotus org
```

2. Place the re-jarred `NCSO.jar` file in the following directory:

Portal-server-install-dir`/SUNWps/web-src/WEB-INF/lib`

3. Redeploy the web application with the following command:

   ***/Portal-server-install-dir*/SUNWps/bin/deploy redeploy**

   Where ***Portal-server-install-dir*** represents the directory in which the Portal Server was originally installed.

4. Restart the web container.

Creating a New User Under the Default Organization

1. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Identity Management tab to display the View drop down list in the navigation pane.
3. Select Users in the View drop down list to display the User page.
4. Click New to display the New User page in the data pane.
5. Select the services to be assigned to the user.

   Select at a minimum Portal Desktop and SSO Adapter.

6. Enter the user information.
7. Click Create.

   The new user's name appears in the Users list in the navigation pane.

Configuring the Mail Provider to Work with an HTTPS Enabled Messaging Server

The Mail channel automatically supports the HTTP protocol; it does not automatically support the more secure HTTPS protocol. However, if your Sun ONE Messaging Server is enabled for HTTPS, you can follow the steps in this section to configure the Mail provider to work properly with the Sun ONE Messaging Server. These steps do not apply to Microsoft Exchange Server and IBM Lotus Notes server.

Web Container Facts and Considerations

In terms of configuring the mail provider for HTTPS for Sun ONE Messaging Server, the steps regarding the web container differ depending upon which web container you are using: Sun ONE Web Server, Sun ONE Application Server, BEA WebLogic Server, or IBM WebSphere Application Server. You need administrative rights to the web container regardless of which one you use. Also, you should have access to the web container documentation in order to reference detailed information on initializing a trust database, adding certificates, and restarting the web container. For more information on these tasks and other security-related issues concerning the Sun ONE web containers, see *Sun ONE Application Server Administrator's Guide to Security* or *Sun ONE Web Server, Enterprise Edition Administrator's Guide*.

To Configure the Mail Provider to Work with an HTTPS Enabled Messaging Server

1. Initialize the trust database for the web container running Sun ONE Portal Server. For more information, refer to the proper documentation as discussed in the preceding paragraph.
2. Install the SSL certificate for the Trusted Certificate Authority (TCA) if it is not already installed.
3. Restart the web container; though often not mandatory, this is a good practice.
4. Add a new SSO Adapter template specifically for HTTPS. The name of the template used in this example is `SUN-ONE-MAIL-SSL`, which is descriptive since the security protocol, SSL, is included in the name

NOTE You can configure an SSO Adapter template and related SSO Adapter configurations in many ways. The steps presented to you subsequently explain a typical configuration. These steps describe how to create a new template and a new configuration since this is a safer practice than simply editing existing templates and configurations.

If you feel comfortable with the editing option, then proceed in that manner. However, if you change the name of the SSO Adapter template and SSO Adapter configuration as part of the edits you make, you will also need to change the SSO Adapter name by editing the properties of the Mail channel.

The two items you would need to edit in the SSO Adapter template or SSO Adapter configuration are:

- `clientProtocol`
- `clientPort`

In creating a new SSO Adapter Template for this example, the `clientProtocol` attribute is set as a `default` attribute. Therefore, it appears in an SSO Adapter template not in an SSO Adapter configuration. The `clientProtocol` attribute must be changed from `http` to `https`. The edited template fragment for this attribute appears as follows:

```
clientProtocol=https
```

For this example, the `clientPort` attribute is set as a `merge` attribute. Therefore, it appears in an SSO Adapter configuration (see Step 5 on page 362). If the `clientPort` attribute were set as a `default` attribute, it would appear in an SSO Adapter template. The client port should be changed to a port reserved exclusively for HTTPS. Here port `443` is used since the HTTPS protocol uses this port number as the default. The edited template fragment for this attribute appears as follows:

```
&clientPort=443
```

a. From an Internet browser, log into the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

b. Click the Service Configuration tab to display the list of configurable services in the navigation pane.

c. Scroll down the navigation pane to the Single Sign-on Adapter Configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

d. Click in the blank configuration description field—which is just above the Add and Remove buttons— it is in the box labeled SSO Adapter Templates under the heading Global as opposed to Dynamic.

e. In the blank configuration description field, type in an entire SSO Adapter Template string (or copy, paste, and edit another string into the field as needed). Code Example 12-8 is a typical configuration which has been provided for your reference. The template you enter will probably have different information. For example, you will probably enter a different value for the `configName` property type unless you want to use the name `SUN-ONE-MAIL-SSL`. Furthermore, the attributes you set as `default` and `merge` will probably differ from this example, depending upon the needs of your site.

If the configuration description field is not blank when you get to it, select all the text in the field and delete it.

Code Example 12-8 Mail SSO Adapter Template for an HTTPS Messaging Server

```
default|imap:///?configName=SUN-ONE-MAIL-SSL
    &encoded=password
    &default=protocol
    &default=clientProtocol
    &default=type
    &default=subType
    &default=enableProxyAuth
    &default=proxyAdminUid
    &default=proxyAdminPassword
    &default=ssoClassName
    &merge=host
    &merge=port
    &merge=uid
    &merge=password
    &merge=smtpServer
    &merge=clientPort
    &clientProtocol=https
    &enableProxyAuth=false
    &proxyAdminUid=[PROXY-ADMIN-UID]
    &proxyAdminPassword=[PROXY-ADMIN_PASSWORD]
    &type=MAIL-TYPE
    &subType=sun-one
    &ssoClassName=com.sun.ssoadapter.impl.JavaMailSSOAdapter
    &default=enablePerRequestConnection
    &enablePerRequestConnection=false
```

f. Click Add.

g. Click Save.

At this point, there may be more than one string that begins with the IMAP protocol. This is acceptable.

5. Add a new SSO Adapter configuration specifically for HTTPS. The name of the configuration used in this example is `sunOneMailSSl` because it is similar to the name used for the respective SSO Adapter template.

NOTE See the Note from the preceding step, Step 4 on page 359.

a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

b. Click the Identity Management tab to display the View drop down list in the navigation pane.

c. Click Services in the View drop down list.

d. Scroll down the navigation pane to the Single Sign-on Adapter configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

e. Click in the blank configuration description field—which is just above the Add and Remove buttons.

f. In the blank configuration description field, type in an entire SSO Adapter configuration string (or copy, paste, and edit another string into the field as needed). A typical configuration has been provided subsequently for your reference. However, the configuration you enter will probably have different information. For example, you will probably enter a different value for the `configName` property type unless you want to use the name `sunOneMailSSL`. For the `ConfigDesc` property type, you must use the same name you used for the `configName` property type in the respective

SSO Adapter template as demonstrated in Code Example 12-8 on page 361. You could also use different port numbers than those supplied here, if desired. Furthermore, the attributes available in this string can vary depending upon how you configured the SSO Adapter template.

If the configuration description field is not blank when you get to it, select all the text in the field and delete it.

```
default|imap:///?configName=sunOneMailSSL&configDesc=SUN-ONE-
MAIL-SSL&port=143&smtpPort=25&clientPort=443
```

g. Click Add.

h. Click Save.

6. Add a new Mail channel to Portal Desktop.

Step 4 and Step 5 explained how to create a new SSO Adapter template and SSO Adapter configuration; those are the steps for creating a new channel. In this step you will make the channel available to end users.

The criteria for choosing a name for the new channel is simply one that is descriptive; therefore the example name chosen here is `SunOneMailSSLChannel`.

a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

b. Click the Identity Management tab to display the View drop down list in the navigation pane.

c. Select Services in the View drop down list to display the list of configurable services.

d. Under the Portal Server Configuration heading, click the arrow next to Portal Desktop to bring up the Portal Desktop page in the data pane

e. Scroll as needed and click the Channel and Container Management link.

f. Scroll down to the Channels heading and click New.

g. In the Channel Name field, type your site's name for the new channel. For example, `SunOneMailSSLChannel`.

h. In the Provider drop down menu, select MailProvider.

i. Click OK, which returns you to the Channel and Container Management Web page where the channel you just created now exists.

j. Scroll down to the Channels heading and click Edit Properties next to the name of the channel you just created, which for this example is `SunOneMailSSLChannel`.

k. Scroll down to the title field, select and delete any words that currently exist, for example `mail`, and type a provider title. A possible name is `SSL Mail Account`.

l. In the description field, select and delete any words that currently exist, for example `mail`, and type a provider description. The same example is used here for description as for the title in the preceding substep: `SSL Mail Account`.

m. Scroll down the page; select and delete any words that currently exist in the SSO Adapter field, for example `sunOneMail`; and type the same SSO Adapter configuration name used in Step 5 on page 362, which for this example is `sunOneMailSSL`.

n. Scroll down and click Save.

o. Scroll back up the page to click the word `top`, which is the first item following the words `Container Path`.

p. Scroll down to the Container Channels heading and click the link for the container that you want to add the new channel to. For example, `MyFrontPageTabPanelContainer`. Do not click the accompanying Edit Properties link.

q. Scroll down to the Channel Management heading, scroll as needed in the Existing Channels frame, and click the name of your newly created channel to select it.

Remember, for this example the channel name is `SunOneMailSSLChannel`.

r. Click the Add button that is next to the Available and Visible list.

This makes the channel available to users and visible without any further configuration.

s. Scroll back up the page and click Save under the Channel Management heading.

You should now be able to log in and use an HTTPS enabled messaging server.

Chapter 13

Managing the Sun ONE Portal Server System

This chapter describes the various administrative tasks associated with maintaining the Sun™ ONE Portal Server system.

This chapter contains these sections:

- Configuring Secure Sockets Layer (SSL)
- Backing Up and Restoring Sun ONE Portal Server Configuration
- Managing a Multiple UI Node Installation
- Configuring a Sun ONE Portal Server Instance to Use an HTTP Proxy
- Managing Sun ONE Portal Server Logs
- Debugging Sun ONE Portal Server

Configuring Secure Sockets Layer (SSL)

You can configure Secure Sockets Layer (SSL) with Sun ONE Portal Server and associated components in the following ways:

- Sun ONE Portal Server—If you configure SSL for just the Sun ONE Portal Server system and not a gateway, then your intranet is “open.”

  You can use SSL between the Sun ONE Portal Server user interface node (where the iPlanet™ Directory Server Access Management Edition administration console, Desktop, servlets, and so on run) and gateway node; and between the Sun ONE Portal Server user interface node and end user computers.

- Sun™ ONE Directory Server—You can configure SSL for the Sun ONE Directory Server and use a secure connection between Sun ONE Identity Server and the Sun ONE Portal Server. See Chapter 6, “Basic Configurations” in the *iPlanet Directory Server Access Management Edition Installation and Configuration Guide* at the following URL for information on enabling SSL on the directory server:

  `http://docs.sun.com/source/816-5626-10/contents.html`

NOTE If you have configured SSL on a directory server, you must disable SSL before uninstalling the directory server with the Sun ONE Portal Server installation script. In addition, to use the `dpadmin` command at the command line, you must also disable SSL.

- Sun™ONE Portal Server: Secure Remote Access —When you configure SSL for the gateway, your intranet is “secure.” See the *Sun ONE Portal Server: Secure Remote Access 6.0 Administrator’s Guide* for the steps to configure SSL on the gateway.

To Configure SSL with Sun ONE Portal Server

Use this procedure if you answered **y** when asked “Do you want to run SSL on *hostname*?” during the Sun ONE Portal Server installation. See the *Sun ONE Portal Server 6.1 Installation Guide* for more information.

1. Create a trust database for the web server on which you installed Sun ONE Portal Server.

   See Chapter 5, “Creating a Trust Database” in the *Sun ONE Web Server 6.0, Enterprise Edition Administrator’s Guide* at the following URL for more information:

   `http://docs.sun.com/source/816-5682-10/index.htm`

2. Request a certificate for the web server on which you installed Sun ONE Portal Server software and install the certificate on the web server instance.

   See Chapter 5, “Requesting and Installing a VeriSign Certificate” or “Requesting and Installing Other Server Certificates” in the *Sun ONE Web Server 6.0, Enterprise Edition Administrator’s Guide* for more information.

3. Turn on encryption for the Sun ONE Portal Server web server instance.

   In the web server administration console, select the Preferences tab, select Add Listen Socket, then select Edit Listen Socket and turn on security.

   See Chapter 5, "Turning Security On," in the *Sun ONE Web Server 6.0, Enterprise Edition Administrator's Guide* for more information,

4. Click Apply and Apply Changes in the web server administration console.

5. Restart Sun ONE Portal Server.

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

6. The system prompts you for the password to get to the certificate database.

   This step occurs each time you restart the web server (executing `/etc/init.d/amserver start`).

NOTE To avoid having to type the passphrase on each reboot, create a file named `.wtpass` that contains the web server passphrase and place it in the *DSAME-BASEDIR*`/SUNWam/config` directory. If you reboot the system with a secure web server without having this file, you must type in the passphrase at the system console.

7. Verify that you can now log on to the Sun ONE Portal Server portal using SSL:

   - To log on to the Sun ONE Identity Server administration console, type:
     `https://`*server*`:`*port*`/amconsole`
   - To log on as a user to the Desktop, type:
     `https://`*server*`:`*port*`/`*deploy_uri*

     for example,
     `https://sesta:80/portal/dt`

To Modify an Existing Sun ONE Portal Server Installation to Use SSL

Use this procedure if you answered **n** when asked "Do you want to run SSL on *hostname*?" during the Sun ONE Portal Server installation. See the *Sun ONE Portal Server 6.1 Installation Guide* for more information.

1. Log in to the Sun ONE Identity Server admin console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

3. Click the properties arrow next to Platform.

   The Platform attributes appear in the data pane.

4. In the server list, change `http` to `https`.

5. Click Save to save your changes.

6. Install the certificate on the web server.

   See Step 1 through Step 4 in "To Configure SSL with Sun ONE Portal Server" on page 366 for details.

7. Copy the `server.xml` and `magnus.conf` files from */BaseDir*`/SUNWam/servers/https-`*hostname-domain*`/conf_bk` directory to the */BaseDir*`/SUNWam/servers/https-`*hostname-domain*`/config` directory. *BaseDir* is the Sun ONE Identity Server base directory.

8. Add the following line to the */BaseDir*`/SUNWam/lib/AMConfig.properties` file if the root CA is not installed for your certificate.

   ```
   com.sun.am.jssproxy.trustAllServerCerts=true
   ```

   This option tells JSS to trust the certificate.

9. In the */BaseDir*`/SUNWam/lib/AMConfig.properties` file, change `http` to `https` for the following:

   ```
   com.sun.am.server.protocol
   com.sun.am.naming.url
   com.sun.am.notification.url
   com.sun.am.session.server.protocol
   com.sun.services.cdsso.CDCURL
   com.sun.services.cdc.authLoginUrl
   ```

10. Restart Sun ONE Portal Server.

    a. To restart a single Sun ONE Portal Server instance, type:

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

b. To restart multiple Sun ONE Portal Server instances, type:

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

11. The system prompts you for the password to get to the certificate database.

See Chapter 11, "Managing SSL" in the *Sun ONE Directory Server 5.1 Administrator's Guide* for more information.

To Configure a Sun ONE Portal Server Instance to Use SSL

1. Log in to the Sun ONE Identity Server admin console as administrator.

By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

3. Click the properties arrow next to Platform.

The Platform attributes appear in the data pane.

4. In the server list, change `http` to `https`.

5. Click Save to save your changes.

6. Install the certificate on the web server.

See Step 1 through Step 4 in "To Configure SSL with Sun ONE Portal Server" on page 366 for details.

7. If this server is part of a multi-instance installation, copy the `server.xml` and `magnus.conf` files from `/BaseDir/SUNWam/servers/https-`*instance_nickname*`/conf_bk` directory to the `/BaseDir/SUNWam/servers/https-`*instance_nickname*`/config` directory.

8. Add the following line to the `/`*BaseDir*`/SUNWam/lib/AMConfig-`*instance_nickname*`.properties` file if the root CA is not installed for your certificate.

```
com.sun.am.jssproxy.trustAllServerCerts=true
```

This option tells JSS to trust the certificate.

9. In the */BaseDir*/SUNWam/lib/AMConfig-*instance_nickname*.properties file, change http to https for the following:

   com.sun.am.server.protocol

   com.sun.am.naming.url

   com.sun.am.notification.url

   com.sun.am.session.server.protocol

   com.sun.services.cdsso.CDCURL

   com.sun.services.cdc.authLoginUrl

10. Restart Sun ONE Portal Server.

    a. To restart a single Sun ONE Portal Server instance, type:

       /etc/init.d/amserver start

    b. To restart multiple Sun ONE Portal Server instances, type:

       /etc/init.d/amserver startall

11. The system prompts you for the password to get to the certificate database.

See Chapter 11, "Managing SSL" in the *Sun ONE Directory Server 5.1 Administrator's Guide* for more information.

Backing Up and Restoring Sun ONE Portal Server Configuration

The Sun ONE Portal Server user and service configuration is stored on the directory server in an LDAP Directory Information Tree (DIT). This allows you to back up and restore configuration information via a Lightweight Directory Interchange Format (LDIF) file.

To Back Up a Sun ONE Portal Server Configuration

To back up Sun ONE Portal Server configuration information use the db2ldif command. This command is available in the slapd-*hostname* directory within the base directory of the directory server. For example, if the directory server was installed to the default install directory (/usr/ldap) on the server sesta, the base directory would be /usr/ldap/slapd-sesta.

1. Change directories to the directory server base directory containing the `db2ldif` command.

   `cd` *`DS_BASEDIR/slapd-HOSTNAME`*

2. Save the configuration to an LDIF file using the `db2ldif` command with the -s option specifying the top level of the DIT for Sun ONE Portal Server. For example, to save a configuration in which the top level of the DIT is `isp`, type the following:

   `./db2ldif -s "o=isp"`

   The data are saved to an LDIF file. The command saves the file to a the current directory. The following format is used to name the file:

   `YYYY_MM_DD_HHMMSS``.ldif`

   After the file is saved, the following example output displays:

```
[16/May/2002:14:11:25 -0700] - Backend Instance: userRoot
ldiffile: /usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif
[16/May/2002:14:11:28 -0700] - export userRoot: Processed 178 entries (100%).
```

To Restore a Sun ONE Portal Server Configuration

You can restore the Sun ONE Portal Server configuration information you have backed up via the `db2ldif` command using the `ldif2db` command. This command is available in the `slapd-`*`hostname`* directory within the base directory of the directory server. For example, if the directory server was installed to the default install directory (`/usr/ldap`) on the server `sesta`, the base directory would be `/usr/ldap/slapd-sesta`.

1. Change directories to the directory server base directory containing the `ldif2db` command by entering:

   `cd` *DS_BASEDIR/slapd-HOSTNAME*

2. Stop the directory server by entering:

   `./stop-slapd`

3. Restore the configuration from the LDIF file to the directory server using the `ldif2db` command with the `-s` option specifying the top level of the DIT for Sun ONE Portal Server and the `-i` option specifying the file name. For example, to restore the LDIF file saved in the previous procedure to the top level of the DIT of `isp`, type the following:

   `./ldif2db -s "o=isp" -i /usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif`

   After the configuration is restored, the following example output displays:

   ```
   importing data ...

   [16/May/2002:16:37:02 -0700] - Backend Instance: userRoot

   [16/May/2002:16:37:03 -0700] - import userRoot: Index buffering
   enabled with bucket size 13

   [16/May/2002:16:37:03 -0700] - import userRoot: Beginning import
   job...

   [16/May/2002:16:37:03 -0700] - import userRoot: Processing file
   "/usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif"

   [16/May/2002:16:37:04 -0700] - import userRoot: Finished scanning
   file "/usr/ldap/slapd-sesta/ldif/2002_05_16_141122.ldif" (178
   entries)

   [16/May/2002:16:37:05 -0700] - import userRoot: Workers finished;
   cleaning up...

   [16/May/2002:16:37:08 -0700] - import userRoot: Workers cleaned
   up.

   [16/May/2002:16:37:08 -0700] - import userRoot: Cleaning up
   producer thread...

   [16/May/2002:16:37:08 -0700] - import userRoot: Indexing
   complete.  Post-processing...

   [16/May/2002:16:37:08 -0700] - import userRoot: Flushing
   caches...

   [16/May/2002:16:37:08 -0700] - import userRoot: Closing files...

   [16/May/2002:16:37:09 -0700] - import userRoot: Import complete.
   Processed 178 entries in 6 seconds. (29.67 entries/sec)
   ```

4. Restart the directory server by entering:

   `./start-slapd`

Changing Sun ONE Portal Server Network Settings

To physically move a server running Sun ONE Portal Server software from one network to another, you need only change the fully qualified domain name mapping the IP address in the `/etc/hosts` file. There are no other hardcoded addresses that need to be changed.

Managing a Multiple UI Node Installation

When you install Sun ONE Portal Server software onto multiple UI nodes, you need to make a configuration change to the Platform attributes in the Sun ONE Identity Server administration console. You edit the Server List attribute to include the URLs for each UI node.

The Sun ONE Identity Server naming service reads the Server List attribute at initialization time. This list contains the Sun ONE Identity Server session servers in a single Sun ONE Identity Server configuration. For example, if two Sun ONE Identity Server servers are installed and should work as one, they must both be included in this list. If the host specified in a request for a service URL is not in this list, the naming service will reject the request. The first value in the list specifies the host name and port of the server specified during installation. Additional servers can be added using the format *protocol*`://`*server*`:`*port*.

To Add Additional Portal Servers to the Server List

1. Log in to the Sun ONE Identity Server admin console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

   The global services appear in the navigation pane.

3. Click the properties arrow next to Platform.

   The Platform attributes appear in the data pane.

4. Edit the Server List attribute.

   For each server functioning as a UI node, type the server URL, for example, `http://host1.sesta.com:80` and then click the Add button. The URL then appears in the Server List.

5. Click Save.

6. Restart Sun ONE Portal Server.

   `/etc/init.d/amserver start`

Configuring a Sun ONE Portal Server Instance to Use an HTTP Proxy

If the Sun ONE Portal Server software is installed on a host that cannot directly access certain portions of the Internet or your intranet, you might want to configure the instance to use an HTTP proxy

1. Change directories to the directory server base directory containing the configuration for the instance by entering:

   `cd /`*BaseDir*`/SUNWam/servers/https-`***hostname-domain***`/config`**

2. Edit the `jvm12.conf` within this directory and add the following lines:

   `http.proxyHost=`*proxy_host*
   `http.proxyPort=`*proxy_port*

   where *proxy_host* is the fully-qualified domain name of the proxy host and *proxy_port* is the port on which the proxy is run.

NOTE If the `jvm12.conf` file has a proxy set up (using the `http.proxyHost=` and `http.proxyPort=` options) you may want to add the `http.nonProxyHosts=`*proxy_host* option. It is possible that the portal server may not be accessible through the proxy server, unless the portal server is added to the proxy server access list.

Managing Sun ONE Portal Server Logs

You can configure Sun ONE Portal Server logging to log information to a flat file or to a database. When logging to a database, the JDBC protocol is used.

To Configure Logging to a File

1. Log in to the Sun ONE Identity Server admin console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

   The global services appear in the navigation pane.

3. Click the properties arrow next to Logging.

   The Logging attributes appear in the data pane.

4. Select File as the Logging Type attribute.

5. Specify the directory path for the log files in the Log Location attribute.

6. Specify the maximum file size in bytes for the log file in the Max Log Size attribute.

7. Specify the number of backup logs in the Number of History Files attribute.

8. Click Save.

To Configure Logging to a Database

1. Log in to the Sun ONE Identity Server admin console as administrator.

   By default, Identity Management is selected in the location pane and All created organizations are displayed in the navigation pane.

2. Choose Service Configuration in the location pane.

   The global services appear in the navigation pane.

3. Click the properties arrow next to the Logging service in the navigation pane.

   The Logging attributes appear in the data pane.

4. Select JDBC as the Logging Type attribute.

5. Specify a user name and password with which to connect to the database in the Database User Name and Database User Password attributes.
6. Specify the driver to use for logging in the Database Driver Name attribute.
7. Click Save.

Debugging Sun ONE Portal Server

This section describes how to set the debug level to help you troubleshoot various Sun ONE Portal Server components.

To Set the Debug Level for Sun ONE Identity Server

The debug level allows you to define the types of messages sent to the debug log. The following levels are supported:

- off—No messages are sent to the debug log.
- error—Error messages are sent to the debug log.
- warning—Warning and error messages are sent to the debug log.
- message—Status, warning, and error messages are sent to the debug log.

By default, debug messages are sent to log files in the `/var/opt/SUNWam/debug` directory.

To set the debug level:

1. Define the debug level in the following line of the `/etc/opt/SUNWps/desktopconfig.properties` file:

   `debugLevel=`*value*

2. Restart Sun ONE Portal Server:

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

3. Examine the various log files under `/var/opt/SUNWam/debug` as well as the Sun ONE Web Server log file.

Chapter 14

Command-Line Utilities

The Sun ONE Portal Server provides a set of command-line utilities in addition to its graphical user interface.

The command-line utilities discussed in this chapter are listed in Table 14-1 and pertain to administrative tasks only. This two column table lists the commands in the first column and a brief description in the second column. They are grouped by function: Desktop, Rewriter and Search.

Table 14-1 Sun ONE Portal Server Command-Line Utilities

Command	Description
`deploy`	Deploys a web application in the web container.
`pdeploy`	Deploys a portlet web application in the web container.
`dpadmin`	Enables display profile objects to be retrieved, added, modified, and removed from a display profile document.
`par`	Performs functions involving the `.par` file for transport of channels and / or providers.
`rwadmin`	Enables the administrator to manage the Rewriter data.
`rdmgr`	Performs all the functions needed by the Search server to work with resource descriptions and the search database.
`sendrdm`	Provides a mechanism for CGI or command-line based search.
`StartRobot`	Starts the Robot searching (crawling) the web.

Also available is the command-line interface `amadmin` for Sun ONE dientity Server administration. The primary purpose of the `amadmin` tool is to help an administrator perform batch administrative tasks on the Identity Server, for example, create, register, and activate new services; and create, delete, and read (get) organizations, people containers, groups, roles, and users. For more information, see the *Sun ONE Identity Server 6.1 Programmer's Guide*.

deploy

Description

The `deploy` command packages the source files for the Sun ONE Portal Server web application files and deploys the package to the web container that hosts the portal server software.

The source files for the Sun ONE Portal Server are stored in the `/opt/SUNWps/web-src` directory. The `WEB-INF/xml` subdirectory contains `web.xml` fragment files that are combined by the `deploy` command to form the `web.xml` file for the Sun ONE Portal Server web application. The corresponding sections of the `web.xml` fragment files are combined based on the alphabetic order of the `web.xml` fragment files. Once the final web.xml file is formed, the files in the `opt/SUNWps/web-src` directory are put into a web application archive (WAR) file using the `jar` command. This WAR file is deployed to a web container using the `deploy` command.

Syntax

```
/opt/SUNWps/bin/deploy [redeploy]
```

Subcommands

The `deploy` command takes the redeploy subcommand. If the deploy command is invoked without the redeploy option, it prompts for configuration information from standard input.

redeploy

Description

The `redeploy` subcommand specifies that the deploy command reuse the Uniform Resource Indicator (URI) and other information assoicated with the Sun ONE Portal Server web application from the current deployment.

Syntax

```
deploy redeploy
```

pdeploy

Description

The `pdeploy` is a command line tool that can be used to deploy and undeploy the portlet web application into the portal server.

The `pdeploy` command requires:

- A subcommand
- A user distinguished name and password to access the directory server
- A distinguished name to identify the LDAP node or the global option for the global level (the node where the portlets need to be added)
- The admin password for the web container
- Portlet War File.

Some of the default settings used by the `pdeploy` command to deploy portlet applications are available in the `PDConfig.properties` file. This file is installed into the `/etc/opt/SUNWps/portlet` directory.

When the pdeploy command deploys the portlet application, it refers to the following parameters in the PDConfigure.properties file:

`logger.log.level=SEVERE`	By default, the log level is set to SEVERE. Valid values are ALL, OFF, INFO, WARNING, SEVERE.

`logger.file.dir=/var/opt/SUNWam/debug`	This parameter specifies the directory where the log file for the deployed portlet application is stored.
validate_schema=true	This parameter specifies whether schema validation should be performed during deployment.

Syntax

This section describes the `pdeploy` command syntax.

Short-Named Format

`pdeploy deploy -u` *uid* `-w` *password* `{-g|-d` *dn*`} -p` *webcontainerpassword* `-V -r rolesfile -f` *userinfofile* `-v -l` *warfile*

`pdeploy undeploy -u` *uid* `-w` *password* `{-g|-d` *dn*`} -p` *webcontainerpassword* `-V -v -l` *warfile*

Long-Named Format

`pdeploy deploy --runasdn` *uid* `--password` *password* `{--global|--dn` *dn*`} --wc_password` *webcontainerpassword* `--rolesfile` *rolesfile* `--userinfofile` *userinfofile* `--verbose --locale` *warfile*

`pdeploy deploy --help`

`pdeploy deploy --version`

`pdeploy undeploy --runasdn` *uid* `--password` *password* `{--global|--dn` *dn*`} --wc_password` *password* `--verbose --locale` *portletwebapp*

`pdeploy undeploy --help`

`pdeploy undeploy --version`

Subcommands

The `pdeploy` command takes these subcommands:

- deploy - deploys the portlet application
- undeploy - removes the portlet application

deploy

Description

If the subcommand is `deploy`, the `pdeploy` command deploys the portlet web application into the portal server. After this command completes, you can create channels based on the portlets defined in the deployed portlet web application.

Syntax

`pdeploy deploy -u` *uid* `-w` *password* `{-g|-d` *dn*`} -p` *webcontainerpassword warfile*

`pdeploy deploy -h|--help`

Options

Table 12-2, which describes what options are supported, contains two columns: the first column lists the possible options, arguments or operands for the deploy subcommand; the second column gives a brief description.

Table 14-2 `deploy` Subcommand Options

Option	Description
`-v` or `--verbose`	Generates debug messages.
`-d` or `--dn`	Specifies the distinguished name in the LDAP node to access the display profile document. The -d or -g option is required.
`-f` or `--userinfofile`	Specifies the file containing the yser information mapping information.
`-g` or `--global`	Specifies the global level node in LDAP to access the display profile document. The -d or -g option is required.
`--help`	Prints help message to stdout.
`-l` or `--locale`	Prints the locale information.
`-p` or `--wc_password`	Specifies the web container password This option is required.
`-r` or `--rolesfile`	Specifies the file containing the Sun ONE Identity Server software and portlet application role mapping information.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This option is required.
`-V` or `--version`	Generates version information.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.

Table 14-3 `deploy` Subcommand Operands

Operand	Description
warfile	Specifies the path to the war file.

Examples

Example 1 In the following example, the `pdeploy` command deploys the `/tmp/SamplePortletApp.war` into the portal server.

```
pdeploy deploy -u “uid=amAdmin,ou=people,o=sesta.com,o=isp” -w admin
-p sunone -g /tmp/SamplePortletApp.war
```

Example 2 Sometimes the portlet application defines logical roles in the `portlet.xml` file. During deployment, the logical roles need to be mapped to the actual roles defined in the system. To accomplish this, supply a role mapping file.

The role mapping file is expected to contain `ActualRole=LogicalRole` entries. The file supplied must follow Java™ porperty file format. For example:

```
cn\=HRManager,dc\=iplnaet,dc\=com=Manager
```

```
cn\=Emp,dc\=iplnaet,dc\=com=Employee
```

The following `pdeploy` command will provide the role mapping file for deploying the `SamplePortletApp.war` file into the portlet application.

```
pdeploy deploy -u “uid=amAdmin,ou=People,o=sesta.com,o=isp” -w admin
-p sunone -r /tmp/RoleMaps -g /tmp/SamplePortletApp.war
```

Example 3 Sometimes the portlet application will need access to the information associated with each user. During deployment, logical user information entry name *must* be mapped to the actual user information entry name defined in the system. To accomplish this during deployment, a user information entry map can be supplied.

The user information file is expected to contain `ActualEntryName=LogicalEntryName` entries. For example:

```
lastname=lname
```

```
firstname=fname
```

The following `deploy` command will provide the user information file for deploying the `SamplePortletApp.war` file into the portlet application.

```
pdeploy deploy -u “uid=amAdmin,ou=People,o=sesta.com,o=isp” -w admin
-p sunone -f /tmp/UserInfoMaps -g /tmp/SamplePortletApp.war
```

undeploy

Description

The `undeploy` subcommand removes the portlet application from the portal server. However, it does not remove all the channel definitions already created for portlets defined in the portlet web application. All channels associated with the portlet web application (being removed) *must* be manually removed.

Syntax

`pdeploy undeploy -u uid -w` *password* `{-g|-d dn} -p` *webcontainerpassword* `-v` *portletwebapp*

`pdeploy undeploy -h|--help`

Options

Table 14-4 and Table 14-5 describes what options are supported and contains two columns: the first column lists the possible options, arguments, or operands for the `undeploy` subcommand; the second column gives a brief description.

Table 14-4 `undeploy` Subcommand Options

Option	Description
`-v` or `--verbose`	Generates debug messages.
`-d` or `--dn`	Specifies the distinguished name in the LDAP node to access the display profile document. The -d or -g option is required.
`-g` or `--global`	Specifies the global level node in LDAP to access the display profile document. The -d or -g option is required.
`--help`	Prints the help message to stdout.
`-l` or `--locale`	Provides locale information.
`-p` or `--wc_password`	Specifies the web container password This option is required.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This option is required.
`-V` or `--version`	Generates version information.

Table 14-4 `undeploy` Subcommand Options *(Continued) (Continued)*

Option	Description
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.

Table 14-5 `undeploy` Subcommand Operands

Operand	Description
portletwebapp	Specifies the name of the deployed portlet web application. Typically, it is the same name as the war file name without the `.war` extension.

Example The following `pdeploy` command undeploys the portlet web application named `SamplePortletApp` from the portal server.

```
pdeploy undeploy -u “uid=amAdmin,ou=People,o=sesta.com,o=isp” -w
admin -g
```

dpadmin

Description

The `dpadmin` command enables display profile objects to be retrieved, added, modified, and removed from a display profile document by using the subcommands. All interactions with display profile objects must be in their native XML format. The `dpadmin` command can operate on a single display profile document only.

The `dpadmin` command requires:

- A subcommand (see Subcommands)
- A user distinguished name and password to access the directory server.
- A target display profile document. A distinguished name to identify the LDAP node or `--global` (`-g`) option for the global level display profile document.

NOTE A display profile document is uniquely identified by the `-d` or `-g` option:

global: `-g`

organization: `-d "dc=`*org*`,dc=com"`

sub-organization: `-d "o=`*sub-org*`,dc=`*org*`,dc=com"`

role: `-d "cn=`*rolename*`,dc=`*org*`,dc=com"`

user: `-d "uid=`*username*`,ou=people,dc=`*org*`,dc=com"`

Syntax

This section describes the `dpadmin` command syntax. You cannot mix long-named options with short ones in one command line.

Short-Named Format

```
$ dpadmin list|merge|modify|add|remove [command-specific options] -u uid
-w password {-g|-d dn} [-l locale] [-r] [-b] [-V] [-h] [file]
```

```
$ dpadmin batch [-c] -f batch-script-filename [-l locale] [-b] [-h]
```

Long-Named Format

```
$ dpadmin list|merge|modify|add|remove [command-specific options]
--runasdn uid --password password {--global|--dn dn} [--locale locale]
[--dryrun] [--verbose] [--version] [--help] [file]
```

```
$ dpadmin --version
```

```
$ dpadmin batch [--continue] --file batch-script-filename [--locale locale]
[--verbose] [--help]
```

Subcommands

The `dpadmin` command takes these subcommands:

- `list`
- `merge`
- `modify`

- `add`
- `remove`
- `batch`

list

Description

This subcommand retrieves the specified display profile node object from the specified display profile document. If no display profile node object is specified, the entire display profile document is retrieved. The display profile object is displayed in its native XML format on standard out.

The `list` subcommand takes these options:

- Administrator's distinguished name and password for accessing the LDAP database by using the `-u or --runasdn` and `-w or --password` options respectively. These options are required.
- Name of the display profile node object to display by using the `-n or --name` option.
- Display profile node object to display defined by the `-g or --global` option for the global level node, or `-d or --dn` option with a specific non-global level node specified. The `-g` or `-d` option is required. In the absence of the command-specific `-n or --name` option, it displays the entire display profile document. The `-g or --global` option displays the entire root display profile document.

Syntax

```
$ dpadmin list -u|--runasdn uid -w|--password password
{(-g|--global)|(-d|--dn dn)} [-n|--name name]

$ dpadmin list -h|--help
```

Options

Table 14-6 contains two columns: the first column lists the possible options, arguments or operands for the `list` subcommand; the second column gives a brief description. The following options are supported:

Table 14-6 `list` Subcommand Options

Argument/Operand	Description
`-d` or `--dn`	Specifies the distinguished name in the LDAP node to access the display profile document. The `-d` or `-g` option is required.
`-g` or `--global`	Specifies the global level node in LDAP to access the display profile document. The `-d` or `-g` option is required.
`-n` or `--name`	Specifies the fully qualified name of the display profile container, channel, or provider object to display. This option is not required.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This option is required.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.

Examples

Example 1

```
$ dpadmin list -n TemplateTableContainer -u
"uid=amAdmin,ou=people,dc=org,dc=com" -w joshua -d "dc=org,dc=com"
```

This example gets the named `TemplateTableContainer` from the `dc=org,dc=com` organization node and prints it to standard out.

Example 2

```
$ dpadmin list -n mailcheck -u "uid=amAdmin,ou=people,dc=org,dc=com"
-w joshua -g
```

This example goes to the global level only to get `mailcheck` and if found, prints it to standard out.

Example 3

```
$ dpadmin list -n TemplateTableContainer/Bookmark2 -u
"uid=amAdmin,ou=people,dc=org,dc=com" -w joshua -d "dc=org,dc=com"
```

This example gets the channel named `Bookmark2` located in the container `TemplateTableContainer` and prints it to standard out.

merge

Description

This subcommand retrieves and displays the merged result of the specified DP node objects. Objects are displayed in their native XML format. The object to be displayed is sent to standard out. If you do not use the `-n` or `--name` option, then an error is reported.

The `merge` command accepts the following arguments:

`--name` or `-n`

The `name` argument specifies the fully-qualified name of the DP container, channel, or provider object to display. If the `name` argument is absent, then the entire DP document is displayed. If the `name` argument does not identify a DP node object, then an error is reported.

NOTE The `merge` subcommand merely displays the merged view of the object and does not persist the result. The underlying data does not get affected by running this subcommand.

Examples

```
$ dpadmin list -n “Bookmark” \
 -u “uid=amAdmin,ou=People,dc=iplanet,dc=com” -w joshua \
 -d “dc=iplanet,dc=com”
<Channel name=”Bookmark” provider=”BookmarkProvider”>
        <Properties merge=”fuse” lock=”false” name=”_properties”>
 <String name=”title” value=”My Bookmarks” merge=”replace”
lock=”false”/>
 <String name=”refreshTime” value=”600” merge=”replace”
lock=”false”/>
 <Collection name=”targets” merge=”fuse” lock=”false”>
 <String value=”Sun home page|http://www.sun.com” merge=”replace”
lock=”false”/>
 <String value=”Everything you want to know about Sun ONE
...|http://www.sun.com/software/products/portal_srvr/home_portal.ht
ml” merge=”replace” lock=”false”/>
 <String value=”Sun ONE home page|http://www.sun.com/software”
advanced=”false” merge=”replace” lock=”false”/>
```

```
 </Collection>
        </Properties>
</Channel>
$ dpadmin list -n “Bookmark” \
        -u “uid=amAdmin,ou=People,dc=iplanet,dc=com” -w joshua \
        -d “cn=HR Role,dc=iplanet,dc=com”
<Channel name=”Bookmark” provider=”BookmarkProvider”>
        <Properties merge=”fuse” lock=”false” name=”_properties”>
 <String name=”title” value=”HR Admin Bookmarks” merge=”replace”
lock=”false”/>
 <Collection name=”targets” merge=”fuse” lock=”false”>
 <String value=”HR Admin home page|http://hr.acme.com”
merge=”replace” lock=”false”/>
 </Collection>
        </Properties>
</Channel>
$ dpadminmerge -n “Bookmark” \
        -u “uid=amAdmin,ou=People,dc=iplanet,dc=com” -w joshua \
        -d “uid=hradmin,ou=people,dc=iplanet,dc=com”
<Channel name=”Bookmark” provider=”BookmarkProvider”>
        <Properties merge=”fuse” lock=”false” name=”_properties”>
 <String name=”title” value=”HR Admin Bookmarks” merge=”replace”
lock=”false”/>
 <Collection name=”targets” merge=”fuse” lock=”false”>
 <String value=”Sun home page|http://www.sun.com” merge=”replace”
lock=”false”/>
 <String value=”Everything you want to know about Sun ONE
...|http://www.sun.com/software/products/portal_srvr/home_portal.ht
ml” merge=”replace” lock=”false”/>
 <String value=”Sun ONE home page|http://www.sun.com/software”
advanced=”false” merge=”replace” lock=”false”/>
 <String value=”HR Admin home page|http://hr.acme.com”
merge=”replace” lock=”false”/>
 </Collection>
```

```
                <Collection name="GlobalThemes" merge="fuse"
lock="false">
                        <Collection name="theme1" merge="fuse"
lock="false">
                                <String name="description" value="Sun
ONE" merge="replace" lock="false"/>
                                                        ...
                        </Collection>
                </Collection>
               <Collection name="locales" merge="fuse" lock="false"
propagate="true" advanced="false">
                        <String name="en_US" value="English (United
States)" merge="replace" lock="false"/>
                </Collection>
              <String name="docroot" value="/docs/" merge="replace"
lock="false"/>
                <String name="helpURL" value="desktop/usedesk.htm"
merge="replace" lock="false"/>
        </Properties>
</Channel>
```

This is the merged result of the Bookmark channel for the user`hradmin` who is assigned to the `HR Role`.

NOTE The output from the `merge` subcommand is comprised of an aggregated result, meaning that all DP objects that are availble will be listed. For instance, properties such as `GlobalThemes` and `locales` are not specifically defined in Bookmark definition yet they show up in the output because these were merged in from one or more parent of the Bookmark channel.

---

modify

Description

This subcommand changes the value of an existing display profile object. The data that is supplied to the `dpadmin modify` command is either from one or more input files or from standard input (an XML fragment that follows the command).

This XML data always requires a proper XML header as well as a nams that uniquely defines which display profile object is to be modified. An example of proper XML header is:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
```

The semantics of the `modify` subcommand vary depending on the type of the display profile object being modified. When the `combine` option is specified, the new elements (like properties) in the display profile object are combined with the existing ones rather than replacing them. The variations of the `modify` subcommand are as follows:

- Display Profile — An entire display profile document can be changed to the new object value specified by using a file. When the `combine` option is specified, every display profile object in the display profile document is recursively combined. See the information below for how `combine` works for each display profile object.
- Channel or Container — A channel or container can be changed to the new object value. When modifying a channel or container, if the parent option is:
  - Specified, the specified parent container is searched for a channel or container that matches the name of the new display profile object. If it is found, it is replaced by the new display profile object.
  - Absent, the root display profile object is assumed to be the parent container. So the root is searched for a channel or container that matches the name of the new display profile object. If it is found, it is replaced with the new display profile object.

  When the `combine` option is specified, the existing Properties, Available, and Selected objects are combined with the new display profile objects.
- Properties — A display profile object's properties can be changed to the new value. The `parent` option is required to modify a display profile object's properties. A display profile node (either channel or container) or display profile provider object that matches the specified name is searched for under the specified parent. If found, the object's properties object is replaced by the new display profile object. When the `combine` option is specified, the existing properties are combined with the new display profile object.

- Available or Selected — The Available or Selected list in a container can be replaced with the new display profile object. The `parent` option is required to modify display profile objects of this type. A display profile container that matches the parent name is searched for. Then the Selected or Available object is replaced by the new display profile object. When the `combine` option is specified, the existing Selected or Available object is combined with the new display profile object.
- String, Boolean, Integer, Collection, or Locale — A String, Boolean, Integer, Collection, or Locale property in a display profile object can be replaced by new display profile object property.

  If the `parent` option is specified, the display profile node (either a channel or container) or display profile provider (in that order) that matches the specified name is searched for. If found, a property that matches the name of the new property is searched for. If found, the property in the display profile object is replaced by new display profile object property.

  If the `parent` option is absent, then the display profile root node is used and the property is replaced at the root node.

  When the `combine` option is specified, the existing Collection or Locale object is combined with the new display profile object. The `combine` option is not supported for atomic display profile properties such as String, Boolean, and Integer.

  The atomic display profile properties such as String, Boolean, and Integer need not be named. If unnamed, the name is assumed to be equal to the string representation of the value. For example, the following two display profile integer objects are equal:

  ```
  <Integer value="3"/>
  ```

  ```
  <Integer name="3" value="4"/>
  ```

- Provider — An existing display profile Provider object can be replaced with the display profile provider of the same name. A display profile provider object that matches the name of the new display profile provider object is searched for under the root display profile node. If present, the new display profile provider object is inserted under the root display profile object, replacing the existing display profile provider of the same name. Since providers can only exist under the root node (the root node is an implicit container), the `parent` option must not be specified.

The `modify` subcommand takes these options:

- Administrator's distinguished name and password for accessing the LDAP database by using the `-u or --runasdn` and `-w or --password` options respectively. These options are required.
- Display profile node object to modify defined by the `-g or --global` option for the global level node, or `-d or --dn` option with a specific non-global level node specified. The `-g` or `-d` option is required.
- Name of the file where the XML input is included by using the *file* argument. This argument is optional. When not used, XML input is expected from standard input.
- Fully qualified name of the parent of the display profile object to be modified by using the `-p or --parent` option.
- The `-m or --combine` option performs a merge of display profile objects.

Syntax

```
$ dpadmin modify -u|--runasdn uid -w|--password password
{(-g|--global)|(-d|--dn dn)} [-p|--parent parent] [-m|--combine]
file|<<EOF

$ dpadmin modify -h|--help
```

Options

Table 14-7 contains two columns: the first column lists the possible options, arguments or operands for the `modify` subcommand; the second column gives a brief description. The following options are supported:

Table 14-7 `modify` Subcommand Options

Argument/Operand	Description
`-d` or `--dn`	Specifies the distinguished name in the LDAP node to access the display profile document. The `-d` or `-g` option is required.
`-g` or `--global`	Specifies the global level node in LDAP to access the display profile document. The `-d` or `-g` option is required.
file	If present, the *file* argument must be the last argument on the command line. It specifies a path to an XML file that contains an XML fragment that conforms to the display profile DTD. If the *file* argument is absent from the `modify` subcommand, then input must be redirected into `dpadmin` from standard input.

Table 14-7 `modify` Subcommand Options

Argument/Operand	Description
`-m` or `--combine`	Combines the specified display profile objects with the new display profile objects. The `combine` option can only be used with these display profile objects: Display Profile root, Channel, Container, Properties, Available, Selected, Collection, and Locale. This is an optional option.
`-p` or `--parent`	Specifies the fully qualified name of the parent of the display profile object to be modified. This is an optional option.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This option is required.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.

Examples

Example 1

```
$ dpadmin modify -p TemplateTableContainer -u
"uid=amAdmin,ou=people,dc=org,dc=com" -w joshua -d "dc=org,dc=com"
<<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>

<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">

<Channel name="NewNews" provider="newsprovider">

   <Properties>

      <String name="title" value="News Channel"/>

      <String name="description" value="This channel is all about
news"/>

   </Properties>

</Channel>

EOF
```

In this example, modify (replace) the channel named `NewNews` in the container `TemplateTableContainer` with value specified as XML text on standard input.

Example 2

```
$ dpadmin modify -p TemplateTableContainer/NewNews -u
"uid=amAdmin,ou=people,dc=org,dc=com" -w joshua -d "dc=org,dc=com"
farble.xml
```

In this example, in the channel NewNews, replace the property named in the file farble.xml with the new object in farble.xml file, where farble.xml contains:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>

<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">

<String name="welcome" value="Hi, welcome to farble land!!!!"/>
```

Example 3

```
$ dpadmin list -n TemplateTableContainer -u
"uid=amAdmin,ou=people,dc=org,dc=comp" -w joshua -d "dc=org,dc=com"

...

<Collection name="news">

   <Collection name="bar">

      <String name="msg" value="hi"/>

   </Collection>

</Collection>

...

$ dpadmin modify -p TemplateTableContainer -u
"uid=amAdmin,dc=org,dc=com" -w joshua -d "dc=org,dc=com" -m <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>

<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">

<Collection name="news">

   <Collection name="bar">

      <String name="msg2" value="woo hoo"/>

   </Collection>

</Collection>

EOF

$ dpadmin list -n TemplateTableContainer -u
"uid=amAdmin,dc=org,dc=com" -w joshua -d "dc=org,dc=com"

...

<Collection name="news">

   <Collection name="bar">

      <String name="msg" value="hi"/>

      <String name="msg2" value="woo hoo"/>
```

```
    </Collection>
</Collection>
...
```

In this example, using the combine option, a new property named "msg2" is added to the collection named "bar". Note that the existing property, "msg" still remains in the result.

Example 4

```
$ dpadmin list -n test -u "uid=amAdmin,ou=people,dc=org,dc=com" -w
joshua -d "dc=org,dc=com"
<Container name="test" provider="testprovider">
    <Properties>
        <String name="title" value="test"/>
    </Properties>
    <Available/>
    <Selected/>
    <Channels>
        <Channel name="test1" provider="test1provider">
            <Properties>
                <Collection name="news">
                    <String name="msg1" value="blah"/>
                    <Collection name="bar">
                        <String name="msg2" value="hi"/>
                    </Collection>
                </Collection>
            </Properties>
        </Channel>
    </Channels>
</Container>
$ dpadmin modify -u "uid=amAdmin,ou=people,dc=org,dc=com" -w joshua
-d "dc=org,dc=com" -m <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
```

```
<Container name="test" provider="testprovider">
   <Properties>
      <String name="title" value="Test Container"/>
   </Properties>
   <Available>
      <Reference value="test1"/>
   </Available>
   <Selected>
      <Reference value="test1"/>
   </Selected>
   <Channels>
      <Channel name="test1" provider="test1provider">
         <Properties>
            <Collection name="news">
               <String name="msg1" value="123"/>
               <Collection name="bar">
                  <String name="msg3" value="123/>
               </Collection>
            </Collection>
         </Properties>
      </Channel>
   </Channels>
</Container>
EOF
$ dpadmin list -n test -u "uid=amAdmin,ou=people,dc=org,dc=com" -w
joshua -d "dc=org,dc=com"
<Container name="test" provider="testprovider">
   <Properties>
      <String name="title" value="Test Container"/>
   </Properties>
   <Available>
```

```
        <Reference value="test1"/>
    </Available>
    <Selected>
        <Reference value="test1"/>
    </Selected>
    <Channels>
        <Channel name="test1" provider="test1provider">
            <Properties>
                <Collection name="news">
                    <String name="msg1" value="123"/>
                    <Collection name="bar">
                        <String name="msg2" value="hi"/>
                        <String name="msg3" value="123"/>
                    </Collection>
                </Collection>
            </Properties>
        </Channel>
    </Channels>
</Container>
```

In this example, the value of "`title`" and "`msg1`" are replaced with the new values. `Available` and `Selected` have both had a `Reference value` added. The "`news`" collection has added "`msg3`". This example shows that the `-m` or `-combine` option with the `modify` subcommand can be used to combine and replace as necessary.

Example 5

```
$ dpadmin list -n test \
    -u "uid=amAdmin,ou=People,dc=iplanet,dc=com"  -w joshua \
    -d "dc=iplanet,dc=com"
<Channel name="test" provider="testprovider">
    <Properties>
       <Collection  name="foo">
            <String name="foo1" value="bar"/>
```

```
        </Collection>
    </Properties>
</Channel>

$ dpadmin modify -p test \
     -u "uid=amAdmin,ou=People,dc=iplanet,dc=com"  -w joshua \
     -d "dc=iplanet,dc=com"  -m <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<ConditionalProperties condition="client"  value="nokia">
   <Collection name="foo">
        <String name="foo1"  value="nokia bar"/>
   </Collection>
</ConditionalProperties>
EOF

$ dpadmin list -n test \
     -u "uid=amAdmin,ou=People,dc=iplanet,dc=com"  -w joshua \
     -d "dc=iplanet,dc=com"
<Channel name="test" provider="testprovider">
     <Properties>
        <Collection  name="foo">
             <String name="foo1" value="bar"/>
        </Collection>
        <ConditionalProperties  condition="client" value="nokia">
           <Collection  name="foo">
               <String name="foo1" value="nokia bar"/>
           </Collection>
        </ConditionalProperties>
   </Properties>
</Channel>
```

```
$ dpadmin modify -p test \
    -u "uid=amAdmin,ou=People,dc=iplanet,dc=com"  -w joshua \
    -d "dc=iplanet,dc=com"  -m <<EOF
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
<ConditionalProperties condition="client"  value="nokia">
    <ConditionalProperties  condition="locale" value="en">
        <String name="abc" value="nokia  en abc"/>
    </ConditionalProperties>
</ConditionalProperties>
EOF

$ dpadmin list -n test \
    -u "uid=amAdmin,ou=People,dc=iplanet,dc=com"  -w joshua \
    -d "dc=iplanet,dc=com"
<Channel name="test" provider="testprovider">
    <Properties>
       <Collection  name="foo">
            <String name="foo1" value="bar"/>
       </Collection>
       <ConditionalProperties  condition="client" value="nokia">
          <Collection  name="foo">
              <String name="foo1" value="nokia bar"/>
          </Collection>
           <ConditionalProperties condition="locale" value="en">
              <String  name="abc" value="nokia en abc"/>
           </ConditionalProperties>
       </ConditionalProperties>
   </Properties>
</Channel>
```

In this example, the `Combine` option is used to add conditional properties.

add

Description

This subcommand adds a new display profile object to the display profile. This subcommand requires that the object to be added does not exist in the display profile. The `add` subcommand reads data for the new object from standard input or from one or more files specified as an argument to the command. Data for the new object must be XML and conform the Sun ONE Portal Server display profile DTD.

This XML data always requires a proper XML header as well as a name that uniquely defines which display profile object is to be modified. An example of proper XML header is:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
```

NOTE Appendix B contains the display profile DTD.

The semantics of the `add` subcommand vary depending on the type of the display profile object being added. That is:

- Display profile — An entire display profile document can be added to the specified LDAP node. If the document already exists at the node, then an error is reported. The `parent` option must not be specified when adding a new display profile document.
- Channel or container — A channel or container can be added. If the `parent` option is present, the parent display profile object is located for the given name and under that parent container, the specified Channel or Container is added. If the `parent` option is absent, the parent display profile object is assumed to be the root display profile object, so under root the specified Channel or Container object is added.
- Properties — Because a properties bundle is required for all display profile nodes and display profile provider objects, they already exist and cannot be added. Use the `modify` subcommand.
- Available or selected — Because Available and selected objects are required for a display profile container, they already exist and cannot be added. Use the `modify` subcommand.

- String, Boolean, Integer, Collection, or Locale — The display profile object String, Boolean, Integer, Collection, or Locale properties can be added. The `parent` option must be specified to add display profile object properties of this type. Under the specified parent, a display profile node (either a channel or container) or display profile provider is searched for (in that order) that matches the name. If found, the given display profile property is added to the display profile node or display profile provider.

  The atomic display profile properties such as String, Boolean, and Integer need not be named. If unnamed, the name is assumed to be equal to the string representation of the value.

- Provider — A display profile provider is inserted under the root node. Because providers can only exist under the root node, do not use the `parent` option. If an object of the same name already exists, then an error is reported.

The `add` subcommand takes these options:

- Administrator's distinguished name and password for accessing the LDAP database (the `-u or --runasdn` and `-w or --password` options respectively). These options are required.
- Display profile document to add or the display profile document where the object must be added (the `-d or --dn` option). The display profile object to add defined by the `-g or --global` option for the global level node. The `-g` or `-d` option is required.
- Name of the file where the XML input is included (the *file* argument).
- Fully qualified name of the parent where the display profile node object is to be added to (the `-p or --parent` option).

Syntax

```
$ dpadmin add -u|--runasdn uid -w|--password password
{(-g|--global)|(-d|--dn dn)} [-p|--parent parent] file|<<EOF

$ dpadmin add -h|--help
```

Options

Table 14-8 contains two columns: the first column lists the possible options, arguments or operands for the `add` subcommand; the second column gives a brief description. The following options are supported:

Table 14-8 add Subcommand Options

Argument/Operand	Description
`-d` or `--dn`	Specifies the distinguished name in the LDAP node to access the display profile document. The `-d` or `-g` option is required.
`-g` or `--global`	Specifies the global level node in LDAP to access the display profile document. The `-d` or `-g` option is required.
file	If present, the *file* argument must also be the last argument on the command line. It specifies a path to an XML file that contains an XML fragment that conforms to the display profile DTD. If the *file* argument is absent for the `add` subcommand, then input must be redirected into `dpadmin` from standard input.
`-p` or `--parent`	Specifies the fully qualified name of the parent of the display profile object to add.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This option is required.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.

Example

```
$ dpadmin add -p SampleTabPanelContainer/Postal -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w joshua -d
"o=sesta.com,o=isp" <<EOF

<?xml version="1.0" encoding="utf-8" standalone="no"?>

<!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">

<Collection name="zipCodes">

   <Integer value="98012"/>

   <Integer value="98036"/>

   <Integer value="94025"/>

   <Integer value="95112"/>

</Collection>

EOF
```

In this example, the command adds the collection property named "zipCodes" specified on standard input to the channel named Postal in the container named SampleTabPanelContainer.

remove

Description

This subcommand removes an existing display profile object from the display profile. If the object to be removed does not exist in the specified display profile document, an error is reported. This subcommand takes type, parent, and name options.

The type option specifies the type of display profile object to remove. The parent option specifies the fully qualified name of the parent display profile object to remove the display profile object from. The parent display profile object type varies depending on the type of display profile object being removed. The name option specifies the name of the object to remove.

The semantics of the parent and name options vary depending on the type of display profile object being removed. Table 14-9 contains two columns: the first column lists the possible values for the type options; the second column gives a brief description of exactly what is removed.

Table 14-9 dpadmin remove parent and name semantics

Value for type Option	Semantics for parent and name Options
root	Removes the entire display profile document from the LDAP node as specified by the *distinguishedname* option or global level display profile if -g (--global) option is supplied. The name option is not needed when type=root.
channel	The name option is required. If the parent option is absent, then the parent container is assumed to be the root display profile node. Otherwise, the parent option is assumed to be the parent container name of the channel to remove. The name option specifies the name of the channel or container to remove.
provider	The parent option must not be specified since providers reside under the root display profile node. The name option is required and specifies the provider to remove.

Table 14-9 `dpadmin remove` parent and name semantics

Value for type Option	Semantics for parent and name Options
`property`	The `parent` option specifies the fully-qualified name of the parent container, channel, or provider object to remove the property from. If the `parent` option is absent, then the root display profile node is assumed to be the parent object. The `name` option specifies the name of the property to remove. If the `name` option is absent, an error is reported. For unnamed display profile properties, the name is equal to the string representation of the value.
`available` or `selected`	Both `parent` and `name` options are required. The `parent` option is assumed to name the parent container or channel object to remove the available (selected) reference from. The `name` option gives the value of the reference to be removed. If the `name` option is absent, then an error is reported.

The `remove` subcommand takes these options:

- Administrator's distinguished name and password for accessing the LDAP database by using the `-u or --runasdn` and `-w or --password` options respectively. These options are required.
- Name of the display profile node object to remove by using the `-n or --name` option. This option is required except when `type=root`.
- Display profile document node in the LDAP database where the display profile document containing the object to be removed exists by using the `-d` (`--dn`) or `-g` (`--global`) option. Either the `-d` (`--dn`) or `-g` (`--global`) option is required.
- Type of display profile node object to remove by using the `-t or --type` option. This option is required.
- Fully qualified name of the parent of the display profile node object to remove by using the `-p or --parent` option.

Syntax

```
$ dpadmin remove -u|--runasdn uid -w|--password password
{(-g|--global)|(-d|--dn dn)} [-n|--name name] [-p|--parent parent]
-t|--type type

$ dpadmin remove -h|--help
```

Options

Table 14-10 contains two columns: the first column lists the possible options, arguments or operands for the `remove` subcommand; the second column gives a brief description. The following options are supported:

Table 14-10 `remove` Subcommand Options

Argument/Operand	Description
`-d` or `--dn`	Specifies the distinguished name in the LDAP node to access the display profile document. The `-d` or `-g` option is required.
`-g` or `--global`	Specifies the global level node in LDAP to access the display profile document. The `-d` or `-g` option is required.
`-n` or `--name`	Specifies the display profile container, channel, or provider object to remove. This option is required except when `type=root`.
`-p` or `--parent`	Specifies the fully qualified name of the parent of the display profile object to remove.
`-t` or `--type`	Specifies the type of display profile object being removed. This option is required.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This option is required.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.

Examples

Example 1

```
$ dpadmin remove -t property -p Bookmarks -n locations -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w joshua -d
"o=sesta.com,o=isp"
```

In this example, the command removes the property named `locations` from the channel or container named `Bookmarks`.

Example 2

```
$ dpadmin remove -t provider -n "pctest" -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w joshua -g
```

Remove the provider `pctest` from the global display profile

Example 3

```
$ dpadmin remove --type channel --parent TemplateTableContainer
--name "Test" --runasdn "uid=amAdmin,ou=people,o=sesta.com,o=isp"
--password joshua --dn "o=sesta.com,o=isp"
```

In this example, the command removes the channel named `Test` that is in the parent container named `TemplateTableContainer`.

Example 4

```
$ dpadmin list -n X -u "uid=amAdmin,ou=people,o=sesta.com,o=isp" -w
joshua -d "o=sesta.com,o=isp"


<Container name=”X” ...>
   <Channels>
      <Container name=”Y” ...>
         <Channels>
            <Channel name=”z” .../>
         </Channels>
      </Container>
   </Channels>
</Container>
```

To remove channel z, you can execute any of the following commands:

```
$ dpadmin remove -t channel -p X -n Y/z -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w joshua -d
"o=sesta.com,o=isp"

$ dpadmin remove -t channel -p X/Y -n z -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w joshua -d
"o=sesta.com,o=isp"

$ dpadmin remove -t channel -n X/Y/z -u
"uid=amAdmin,ou=people,o=sesta.com,o=isp" -w joshua -d
"o=sesta.com,o=isp"
```

batch

Description

The batch subcommand enables the processing of multiple display profile commands in an optimized fashion. The subcommands are listed in a batch script file (required) and executed consecutively. If an error occurs, the default is to report the error and exit. The `-c or --continue` option denotes continuous processing mode. In this mode, if an error occurs, it is reported and dpadmin continues with the next subcommand.

The command batch scripts must be text (ASCII) documents and can contain any number of subcommands for input into dpadmin, except a batch subcommand. A subcommand must be entered on a single line (the new line character indicates the end of the command). For each subcommand, the administrator's distinguished name and password must be specified in the command line. The syntax for the subcommand is exactly the same as if the subcommand was entered directly into a shell (without the dpadmin part). The script cannot contain XML, so subcommands that require XML input must have it in a file. If the distinguished name (or DN) contains space(s), use double-quotes around it.

Example batch-script file (each command is on a single line):

```
add -p PostalMailer -u uid=amAdmin,ou=People,dc=iplanet,dc=com -w
joshua -d dc=iplanet,dc=com zipCodes.xml

add -p PostalStamps -u uid=amAdmin,ou=People,dc=iplanet,dc=com -w
joshua -d dc=iplanet,dc=com zipCodes.xml stampRates.xml

add -p PostalRates -d "cn=hr role,dc=iplanet,dc=com" zipCodes.xml
stampRates.xml
```

The batch subcommand takes the `-c or --continue` option and requires the name of the batch script file specified by using `-f or --file`.

Syntax

`$ dpadmin batch [-c|--continue] -u|--runasdn` *uid* `-w|--password` *password* `-f|--file` *batch-script-file*

```
$ dpadmin batch -h|--help
```

Options

Table 14-11 contains two columns: the first column lists the possible options, arguments or operands for the batch subcommand; the second column gives a brief description. The following options are supported:

Table 14-11 `batch` Subcommand Options

Argument/Operand	Description
`-c` or `--continue`	Indicates a continuous operation mode. Errors are reported, but `dpadmin` continues with the next subcommand if this option is specified. By default, `dpadmin` exits after reporting an error.
`-f` or `--file`	Specifies the batch script file. This argument is required.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This is used with the `list`, `modify`, `add`, and `remove` subcommands only. This option is optional. If specified, the distinguished name is used to authenticate throughout the batch process. In addition, each subcommand in the batch script can also have its own authentication which will override this distinguished name.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is optional. If specified, the distinguished name is used to authenticate throughout the batch process. In addition, each subcommand in the batch script can also have its own authentication which will override this password.

Options

Table 14-12 is a summary of the `dpadmin` command. The table is organized by subcommands listed in subheads. It contains two columns: the first column lists the possible options, arguments or operands; the second column gives a brief description. The following options are supported:

Table 14-12 `dpadmin` Command Options

Argument/Operand	Description
`-V` or `--version`	Specify this option to `dpadmin` to print descriptive information about the utility, such as its version, legal notices, and other similar information to standard output. Any subcommand and all other options are ignored when this option is present.
Options Common to all Subcommands	
`-b` or `--verbose`	Specify this option to produce debugging messages.

Table 14-12 `dpadmin` Command Options

Argument/Operand	Description
`-h` or `--help`	Specify this option to `dpadmin` to print out a brief help page to standard output. If no subcommand is present, a generic help page for `dpadmin` is printed. If one of the `dpadmin` subcommands is present, then a brief help page that is specific to the subcommand is printed.
`-l` or `--locale`	Use this option to have all debug / error messages localized in the specified locale. If not specified, it defaults to system locale.
The `list`, `add`, `modify`, and `remove` Subcommands Options	
`-d` or `--dn`	Specifies the distinguished name in the LDAP node to access the display profile document. The `-d` or `-g` option is required.
`-g` or `--global`	Specifies the global level node in LDAP to access the display profile document. The `-d` or `-g` option is required.
`-r` or `--dryrun`	Reports error or success of subcommand to sysout. Does not put the resulting changes of the subcommand in LDAP.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This is used with the `list`, `modify`, `add`, and `remove` subcommands only. This option is required.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.
The `list and remove` Subcommand Options	
`-n` or `--name`	Specifies the fully qualified name of the display profile container, channel, or provider object to display or remove. This option is not required.
The `add`, `modify`, and `remove` Subcommands Options	
`-p` or `--parent`	Specifies the fully qualified name of the parent of the display profile object to add, to modify, or to remove.
The `add` and `modify` Subcommands Options	
file	If present, the *file* argument must also be the last argument on the command line. It specifies a path to an XML file that contains XML fragment that conforms to the display profile DTD and contains a proper XML header. Subcommands that require XML input include `modify` and `add`. If the *file* argument is absent for these subcommands, then input must be redirected into `dpadmin` from standard input.

Table 14-12 `dpadmin` Command Options

Argument/Operand	Description
The `modify` Subcommand Options	
`-m` or `--combine`	Combines the specified display profile objects with the new display profile objects. This can be used only with the `modify` subcommand. The `combine` option can only be used with these display profile objects: Display Profile root, Channel, Container, Properties, Available, Selected, Collection, and Locale.
The `remove` Subcommand Options	
`-t` or `--type`	Specifies the type of display profile object to be removed. Types can be: `root`, `channel`, `provider`, `property`, `available` or `selected`.
The `batch` Subcommand Options	
`-c` or `--continue`	Indicates a continuous operation mode. This is used with the `batch` subcommand only. Errors are reported, but `dpadmin` continues with the next subcommand if this option is specified. By default, `dpadmin` exits after reporting an error.
`-f` or `--file`	Specifies the batch script file. This ASCII file is used only with the `batch` subcommand.
`-u` or `--runasdn`	Specifies the distinguished name of the user to use to bind to the Directory Server. This is used with the `list`, `modify`, `add`, and `remove` subcommands only. This option is required.
`-w` or `--password`	Specifies the password of the distinguished name of the user used to bind to the Directory Server. This option is required.

par

Description

The `par` command performs functions involving the specified `.par` file. It can be used for exporting and importing channels or providers to and from the Sun ONE Portal Server.

Syntax

The par command syntax is described in this section. Mixing long-named options with short ones in one command line is not recommended.

Short-Named Format

`par containers -r` *uid* `-p` *password* `[-d]` *dn*`|global`

`par describe [-d]` *parfile*

`par export -r` *uid* `-p` *password* `[-m] [-d] -s` *staticdir* `[-v]` *parfile* *dn*`|global {`*exportfile*`|from=}...`

`par import -r` *uid* `-p` *password* `[-o] [-d] -s` *staticdir* `[-v]` *parfile* `[`*dn*`|global [`*op*`...]]`

`par import -r` *uid* `-p` *password* `-a [-d] -s` *staticdir* `[-v]` *parfile* `[`*dn*`|global]`

Long-Named Format

`par containers --runasdn` *uid* `--password` *password* `[--debug]` *dn*`|global`

`par describe [--debug]` *parfile*

`par export --runasdn` *uid* `--password` *password* `[--modify] [--debug] --staticdir` *staticdir* `[--verbose]` *parfile* *dn*`|global {`*exportfile*`|from=}...`

`par import --runasdn` *uid* `--password` *password* `[--overwrite] [--debug] --staticdir` *staticdir* `[--verbose]` *parfile* `[`*dn*`|global [`*op*`...]]`

`par import --runasdn` *uid* `--password` *password* `--auto [--debug] --staticdir` *staticdir* `[--verbose]` *parfile* `[`*dn*`|global]`

Subcommands

The following subcommands are supported:

- `containers`
- `describe`
- `export`
- `import`

containers

Description

Lists all available containers and channels in a particular display profile document, indicated by the specified directory server name (or `global`). This can be used as an aid to formulating other commands.

Syntax

`par containers -r|--runasdn` *uid* `-p|--password` *password* `[-d|--debug] [-v|--verbose]` *dn*`|global`

Example

```
par containers -r "uid=amAdmin,ou=people,o=sesta.com,o=isp" -p
joshua -d "o=sesta.com,o=isp"
```

In this example, the command lists all available containers in the display profile document residing in the LDAP node "`o=sesta.com,o=isp`".

describe

Description

Describes the contents of the specified `.par` file, including the entries, and any built-in autoextract operations defined for the entries.

Syntax

`par describe` *parfile*

Example

```
par describe myfile.par
```

In this example, the command output or description of the `myfile.par` may be something like the following:

```
Class Root: /
Property Based File Root: /pbfiles
Display Profile Root: /dp
Static Content Root: /static
Entry: mychannel
AutoExtract:dpnode=o%3Dsesta.com%2Co%3Disp,channel,entry=mychann
el
DP Document: this my JSP based channel.
Channel: SampleJSP.a
Includes: Property Based File, root templateBaseDir, path
default/mychannel/samplecontent.jsp (channel)
```

```
Includes: Property Based File, root templateBaseDir, path
default/mychannel/sampledoedit.jsp (channel)
Includes: Property Based File, root templateBaseDir, path
default/mychannel/sampleedit.jsp (channel)
Includes: Property Based File, root templateBaseDir, path
default_en_US/mychannel/samplecontent.jsp
(channel)
Includes: Property Based File, root templateBaseDir, path
default_en_US/mychannel/sampledoedit.jsp
(channel)
Includes: Property Based File, root templateBaseDir, path
default_en_US/mychannel/sampleedit.jsp
(channel)
```

export

Description

Populates the specified `.par` file by exporting the provider or channel information from the portal server. The command takes a `.par` file, a directory server name argument (or keyword `global`) corresponding to the desired display profile document to update, and any number of (requires at least one) *exportfile* or `from` specifications. The `from` specifications contain exactly the same information as an export file; the only difference is that the "lines" are separated by semicolons.

The `par export` command without the `-m` option creates a `.par` file. The `par export` command with the `-m` option is used to update and / or add to an already existing `.par` file that defines a provider, channel or container.

Syntax

`par export -r|--runasdn` *uid* `-p|--password` *password* `[-d|--debug] -s|--staticdir` *staticdir* `[-v|--verbose]` *parfile dn*`|global {`*exportfile*`|from=}...`

`par export -r|--runasdn` *uid* `-p|--password` *password* `[-d|--debug] -s|--staticdir` *staticdir* `[-v|--verbose] -m|--modify` *parfile dn*`|global {`*exportfile*`|from=}...`

Example

```
par export -r "uid=amAdmin,ou=people,dc=sesta,dc=com" -p joshua
mychannel.par "o=sesta.com,o=isp" myexport.txt
```

Here `myexport.txt` contains:

```
from: channel mychannel
directory: templateBaseDir . mychannel
description: this is my JSP based channel
```

In this example, the command exports the channel definition and template files for `mychannel` into `mychannel.par` from the "`dc=sesta,dc=isp`" *dn*. Also, if it were a JSPProvider channel, the `directory` line transfers all of the `.jsp` files, including locale-specific versions.

import

Description

Imports objects from the specified `.par` file into the portal server. The command takes a `.par` file, and optional arguments for the display profile document to import the objects into the indicated display node in the directory server (or the root display profile indicated by the keyword `global`), and operations to be performed. If these things are not specified, they are taken from the .par file. The *auto* option can be used to indicate that you wish to simply perform the autoextract operations already contained in the `.par` file.

If you want to add a new channel, you can use the `par import` command with or without the `-o` option. If a channel already exists, you must use the `-o` option with the `par import` command to completely replace (overwrite) the old channel. You can use this subcommand to import providers as well as channels.

Syntax

`par import -r|--runasdn` *uid* `-p|--password` *password* `[-o] [-d|--debug] -s|--staticdir` *staticdir* `[-v|--verbose]` *parfile* `[`*dn*`|global [`*op*`...]]`

`par import -r|--runasdn` *uid* `-p|--password` *password* `-a|--auto [-d|--debug] -s|--staticdir` *staticdir* `[-v|--verbose]` *parfile* `[`*dn*`|global]`

Examples

Example 1

```
par import -r "uid=amAdmin,ou=people,o=sesta.com,o=isp" -p joshua
--auto myfile.par "o=sesta.com,o=isp"
```

In this example, the command extracts the channel from `myfile.par` file, if that is the automatic operation defined in the `myfile.par` parfile.

Example 2

```
par import -r "uid=amAdmin,ou=people,o=sesta.com,o=isp" -p joshua
myfile.par "o=sesta.com,o=isp"
"entry=mychannel,channel=anothername,avail=topcontainer"
```

In this example, the command extracts the channel explicitly, installing it with a different name in the target *dn*, and making it available in container `topcontainer`.

Options

Table 14-13 contains two columns: the first column lists the possible options for the `par` command; the second column gives a brief description. This command supports the following options (listed in alphabetical order):

Table 14-13 `par` Command Options

Options	Description
`-a` or `--auto`	Use with the `import` command to apply the autoextract operations from the `.par` file. There should be no operations specified on the command line in this case. The *dn* argument may still be specified; if specified, it replaces the *dn* in the autoextract operations. If operations are specified on the command line they are ignored.
`-d or --debug`	Specify this to produce extra debugging information on error messages.
`-m` or `--modify`	Use with the `export` command to update an existing `.par` file rather than replace it. Any new files added for an entry supplement or replace the old ones. Also, use this command to add new files to an existing provider or channel by using a `.par` file.
`-o` or `--overwrite`	Use with the `import` command to replace existing channels.
`-p` or `--password`	Specifies the password for authentication. Required for all of the subcommands except `describe`. If not specified, the `par` utility prompts for it.
`-r` or `--runasdn`	Specifies the distinguished name of the user for authentication. Required for all of the commands except `describe`. If not provided, the `par` utility prompts for it. Use the format uid=*userName*,*ou=people*,`o=`*organizationName*,`o=`*organizational Unit*

Table 14-13 `par` Command Options

Options	Description
`-s` or `--staticdir`	Defines the host-specific directory of the static content directory to be used for import or export.
`-v` or `--verbose`	Describes operations as they are executed. Use with the `import` and `export` commands.
`-V` or `--version`	Specify this option to `par` to print descriptive information about the utility, such as its version, legal notices, and other similar information to standard output. Any subcommand and all other options are ignored when this option is present.
`-?`	Obtain help for any subcommand.

Arguments

Table 14-14 contains two columns: the first column lists the possible arguments for the `par` command; the second column gives a brief description. This command takes the following arguments:

Table 14-14 `par` Command Arguments

Argument	Description
dn	Specifies the distinguished node in the directory server to access. Use the format "`o=`*organizationName*`,o=`*organizationalUnit*"
`global`	Specifies the global level node in LDAP to access the display profile document.
exportfile	These files each correspond to an entry (provider, channel, or provider/channel combination) in the `.par` file, and simply specify the data to be inserted into the specified `.par` file. It can be a small file if the information is too large to list on the command line. See Export Files for more information.
`from`	Specified on the command line, this is taken as equivalent to an export file containing the "from" line, followed by an equal to ("=") sign, and any other lines separated by a semicolon (";"). See from in Table 14-15 for more information on line properties.
op	Specifies the operation to perform. See Operations for more detail.

Table 14-14 par Command Arguments

Argument	Description
parfile	Specifies the par file to operate upon; that is, indicates the par file to import, export, or describe.

Export Files

These files simply specify data to be inserted into a `.par` file. The file consists of lines containing a keyword, followed by a colon and white space delimited fields. The line "from:" is required and it must be the first line of the file. Lines beginning with "#" are treated as comments.

Table 14-15 contains two columns: the first column lists the possible line keywords; the second column gives a brief description.

Table 14-15 Export File Line Properties

Line	Description
from: *types name*	“from” indicates what entity is being exported. *types* can be “channel”, “provider”, or “channel,provider”, and “channel+provider”. The *name* indicates the channel name, or a provider name if a provider is being exported. The *name* must be URL encoded if the name contains white space (+), commas (%2C), colons (%3A), semicolons (%3B), plus signs (%2B), or percent signs (%25).
auto: none auto: *op*	"auto" specifies the autoextract operation for the entry. It takes the *op* argument followed by the operation. "none" can also be entered, suppressing the autoextract. If no "auto:" line is specified, a default autoextract is produced. The default operation is to extract the channel and / or provider with its original names.
file: *root* | . *path* [*types*]	"file" indicates that a file, based on a property setting, is to be included. The property can come from either the "desktop properties" file, located by default in `/etc/opt/SUNWps/desktop/desktopconfig.properties` file or from the display profile visible to a `getProperty()` call for the item being exported or imported. *root* specifies the root of the file location and *path* specifies the path to the rest of the file. `root` is a property name that corresponds to a directory (like). If *root* is given as ".", the file is assumed to be static content located at the web server’s doc root. You can also specify the types of operation the file is to be associated with, defaulting to "channel". *types* can be “channel”, “provider”, or “channel,provider”, and “channel+provider”.

Table 14-15 Export File Line Properties

Line	Description
class: *class* [*types*]	"class" indicates that a class file is to be packaged with the entry, and you may optionally specify the types of operations that the class file are associated with. If not specified, "provider" is assumed. *types* can be "channel", "provider", or "channel,provider", and "channel+provider"; also, when specifying both, you can use a space.
directory: *root* | . *dir* + | . | *filter* [*types*]	"directory" implies an entire directory search with all non-directory files to be included as if entered as "file" lines. It includes the capability of specifying a "filter", that is a directory component which must be present in recursive directory searches. *root* specifies the root of the directory, or "." to indicate static content. *dir* is the directory underneath the root to search from, which can be given as "." to start at the root itself. *filter* specifies the filter component which must be in the directory, which implies a recursive descent. It can be given as "+" for a recursive descent with no filter, or "." for no recursive descent (just the contents of the actual directory). You can also specify the types of operation, which default to "channel". *types* can be “channel”, “provider”, or “channel,provider”, and “channel+provider”.
entry: *name*	"entry" specifies the entry name used in the `.par` file. If not specified, it defaults to the name from the "from: " line.
desc: *text*	Any number of "desc" lines may appear, and are concatenated together as a user-visible description packaged with the entry.

Operations

Each operation (*op*), in the export file or on the command line, must be specified as a comma separated list of keywords that can have values, most of which are optional. The operations are in a blank or space separated list. Each operation is in the following format.

`dpnode=`*dn*`,entry=`*name*`,provider[=`*name*`],channel[=`*name*`],container=`*name*`[,avail=`*name*`,selected]`

dpnode

This specifies the distinguished name in the directory server (or the keyword `global`) for the display profile document that this operation is targeted at. May not apply if the context it is being specified in has already provided this. For example, if the `import` subcommand defines the distinguished name, the distinguished name in the file is ignored.

entry

This specifies the entry name in the `.par` file. This is not needed if the:

- `.par` file only contains one entry, as the operation defaults to that entry
- Operation is already associated with an entry such as the autoextract option for an entry.

The `par` utility defaults to the first entry in the file if an entry is not specified.

provider
This indicates that a provider extraction is to take place. If the name is missing, it simply uses the name packaged with the provider in the `.par` file.

channel
This indicates that a channel extraction is to take place. If the name is missing, it simply uses the name provided with the channel in the `.par` file.

container
This applies only to channel extractions and indicates which container the channel is to be inserted into. If omitted, the channel is inserted into the "channels" element at the display profile document root.

avail
This applies only to channel extractions and indicates a container whose "avail" (or available) list is to receive a reference to the new channel. If omitted, no new channel reference is created.

selected
This applies only if "avail" was used. It indicates that the container whose "avail" list received a reference, also has a reference placed in its "selected" list.

If the *op* information is in both the `par import` command and in the `.par` file, the command information takes precedence.

Par Files

This section contains supplemental information on the par file format. You do not require this information to run the `par` command.

The par file is a jar file with manifest entries for transporting channels, providers, and their associated files. It is intended allow flexibly when installing providers, channels, or both. The `.par` file contains 4 major types of files:

1. XML documents containing the provider and / or channel information for the display profile. This document is a "parEntry", as described in the display profile dtd. This parEntry contains a channel, a provider, or a channel / provider combination.
2. Class files associated with the provider and / or channel.
3. Property based files. These are general files associated with the channel, portlet, or provider (usually the channel), which have to be deployed underneath some configurable root on the portal server.
4. Static content files. These are files deployed as documents on the web server.

Par File Contents

Table 14-16 contains two columns: the first column lists the required global headers; the second column gives a brief description. The `.par` file must contain the following headers:

Table 14-16 Global Headers

Header	Description
PS-Version	Specifies a portal server specific version number of the .par file. Also verifies that this is a .par file.
PS-DefaultEntry	Names the entry used for operations involving an unnamed entry.
PS-DPRoot PS-ClassRoot PS-PBFileRoot PS-StaticRoot	Indicates the root directories in the archive for parEntry documents, classes, property based files, and static content, respectively. If unspecified, the corresponding files are rooted at the top of the archive.

In the `.par` file, there must be a named entry for each parEntry XML file. Table 14-17 contains two columns: the first column lists the possible headers; the second column gives a brief description. The section for each named entry may contain the following headers:

Table 14-17 Named Entry Headers

Header	Description
PS-EntryName	Specifies the command visible name of the entry.

Table 14-17 Named Entry Headers

Header	Description
PS-AutoExtract	Specifies the autoextract operation for the entry, if one exists.
PS-Include	Contains a comma separated list of archived files specified by their actual archive path. The path implies what type of file they are according to the "root" specifications. The files are appended with a parenthesized number which corresponds to the types of operations the file applies to (a mask with 1 for provider, 2 for channel). This can be ignored if there are no files other than the XML document associated with the entry.

If the `.par` file contains only one entry, entries need not be named in manipulating the file since the default entry is the entry used if none is named.

rwadmin

Description

The `rwadmin` command enables the administrator to manage the Rewriter data available in the iPlanet Directory Server Access Management Edition Rewriter service.

Syntax

The `rwadmin` command syntax is described in this section.

Short Named Format

`rwadmin list -u` *uid* `-w` *password* `[-l` *locale*`] [-b] [-h]`

`rwadmin store -u` *uid* `-w` *password* `[-l` *locale*`] [-b] [-h]` *filename*

`rwadmin get -r` *rulesetname* `-u` *uid* `-w` *password* `[-l` *locale*`] [-b] [-h] [`*filename*`]`

`rwadmin remove -r` *rulesetname* `-u` *uid* `-w` *password* `[-l` *locale*`] [-b] [-h]`

Long Named Format

`rwadmin list --runasdn` *uid* `--password` *password* `[--locale` *locale*`] [--verbose] [--version] [--help]`

`rwadmin store --runasdn` *uid* `--password` *password* `[--locale` *locale*`] [--verbose] [--version] [--help]` *filename*

`rwadmin get --rulesetid` *rulesetname* `--runasdn` *uid* `--password` *password* `[--locale` *locale*`] [--verbose] [--version] [--help] [`*filename*`]`

`rwadmin remove --rulesetid` *rulesetname* `--runasdn` *uid* `--password` *password* `[--locale` *locale*`] [--verbose] [--version] [--help]`

Subcommands

These subcommands are supported:

- list
- store
- get
- remove

list

Description

This command lists all the available ruleset names.

Syntax

`rwadmin list -u|--runasdn` *uid* `-w|--password` *password*

Example

```
rwadmin list -u “uid=amAdmin,ou=people,o=sesta.com,o=isp” -w joshua
```

In this example, the command displays names of all available rulesets.

store

Description

This command stores the Rules available in the local file system into iPlanet Directory Server Access Management Edition. If you want to store the DefaultRuleSet, use the following command:

```
rwadmin store -u uid -w password /resources/DefaultRuleSet.xml
```

where `/resources/DefaultRuleSet.xml` is the location of RuleSet stored in `rewriter.jar` file. Note that when this command is executed, if a ruleset with the same ID already exists, no new data is stored. Delete the existing ID and try again.

Syntax

```
rwadmin store -u|--runasdn uid -w|--password password filename
```

Example

```
rwadmin store -u “uid=amAdmin,ou=people,o=sesta.com,o=isp” -w joshua
/opt/data/ExampleRuleSet.xml
```

In this example, the command stores the Rules available at `/opt/data/ExampleRuleSet.xml` into the iPlanet Directory Server Access Management Edition.

get

Description

This command gets the RuleSet from iPlanet Directory Server Access Management Edition. If the *filename* is provided, the retrieved RuleSet is stored in the specified file, or else it is displayed on `stdout` (or the console).

Syntax

```
rwadmin get -r|--rulesetid ruleset -u|--runasdn uid -w|--password
password [filename]
```

Examples

Example 1

```
rwadmin get -r “ExampleRuleSet” -u
“uid=amAdmin,ou=people,o=sesta.com,o=isp” -w joshua
```

In this example, the command retrieves the RuleSet named ExampleRuleSet from iPlanet Directory Server Access Management Edition and displays it on the console.

Example 2

```
rwadmin get -r “ExampleRuleSet” -u
“uid=amAdmin,ou=people,o=sesta.com,o=isp” -w joshua /tmp/abc.xml
```

In this example, the command retrieves the RuleSet named ExampleRuleSet from iPlanet Directory Server Access Management Edition and saves it in the file `abc.xml` in the `/tmp` directory.

remove

Description

This command deletes the RuleSet from iPlanet Directory Server Access Management Edition. This command deletes the RuleSet without any warning.

Syntax

`rwadmin remove -r|--rulesetid` *ruleset* `-u|--runasdn` *uid* `-w|password` *password*

Example

```
rwadmin remove -r “ExampleRuleSet” -u
“uid=amAdmin,ou=people,o=sesta.com,o=isp” -w joshua
```

In this example, the command deletes the RuleSet whose name is ExampleRuleSet from iPlanet Directory Server Access Management Edition.

Options

Table 14-18 is a summary of the `rwadmin` command. It contains two columns: the first column lists the possible options; the second column gives a brief description. The command supports the following options (listed in alphabetical order):

Table 14-18 `rwadmin` Command Options

Option	Description
`-b` or `--verbose`	Specify this argument to `rwadmin` to provide detailed information what is happening when the command is executed
filename	Specify this option with the `store` subcommand to indicate the file to get the RuleSet data from when importing into the iPlanet Directory Server Access Management Edition. Specify this with the `get` subcommand to indicate the file in which the retrieved RuleSet data should be stored.
`-h` or `--help`	Specify this option to `rwadmin` to print out a brief help page to standard output. If no subcommand is present, a generic help page for `rwadmin` is printed. If one of the `rwadmin` subcommands is present, then a brief help page that is specific to the subcommand is printed.
`-l` or `--locale`	Use this option to have all output messages localized in the specified locale. If not specified, it defaults to system locale.

Table 14-18 `rwadmin` Command Options

Option	Description
`-r` or `--rulesetid`	Use this option to specify the name of the RuleSet to operate upon
`-u` or `--runasdn`	Specify this option with the distinguished name of the user to use to bind to the Directory Server.
`--version`	Specify this option to `rwadmin` to print descriptive information about the utility, such as its version, legal notices, and other similar information to standard output. Any subcommand and all other arguments are ignored when this option is present.
`-w` or `--password`	Specify this option with the password of the distinguished name of the user used to bind to the Directory Server.

rdmgr

Description

The `rdmgr` command is the main command used to work with the Search service. It gives the administrator two types of subcommands: ones that are used to work with resource descriptions (RDs); and ones that are used for database maintenance. The `rdmgr` command is normally run in a search-enabled Portal Server instance directory, which is the `/server-instance-directory/deployment_uri` directory. This is the deployment uri path you selected at install time. If you chose the default Portal Server install, this is the `/var/opt/SUNWps/https-`*servername*`/portal` directory. Where the value of the *servername* is the default Portal Server instance name—the fully qualified name of your Portal Server.

Syntax

The general syntax of the `rdmgr` command is:

```
# rdmgr   [subcommand]  [options]  [input]
```

The RD subcommands more specifically follow this syntax:

```
# rdmgr   [-umgdnUL]  [-ACSTNPq]  [-a att,att,...] [-b number]
```

```
[-c search.conf] [-i charset] [-o charset] [-j number] [-l number]
[-p progress] [-r number] [-s schema] [-y dbname] [filename|-Q query]
```

The database maintenance subcommands more specifically follow this syntax:

```
# rdmgr   [-OXIERGBL]  [-ASTDVNP]  [-a att,att,...] [-b number]
[-c search.conf] [-j number] [-l number] [-p progress] [-r number]
[-s schema] [-y dbname]
```

You can use `-l number` to set the log level number for any RD or database subcommand. A setting of 1 (default) logs all the `rdmgr` commands. The higher the number the more detail the log file contains. The possible levels are 1- 100. If this option is not specified, this command assumes the setting defined by the `debug-loglevel` in the `search.conf` file. The log file name is defined by the `rdmgr-logfile` in the `search.conf` file.

Where the `-c search.conf` option gives the location of the `search.conf` file. If you do not use this option, the default value is `config/search.conf` in the current directory. The `search.conf` file lists all the specific search values you have set.

You can use `-p progress` to show the progress of any RD or database subcommand. If you only enter `-p`, the progress is displayed on `stdout`.

Subcommands

The following subcommands are supported:

- Resource Description Subcommands
- Database Maintenance Subcommands
- Usage Message and Version Subcommands

Resource Description Subcommands

Description

The RD subcommands allow an administrator a batch process to insert or replace RDs, merge RDs filtered by a view, retrieve RDs filtered by a view, delete RDs and count RDs. Table 14-19 is a two-column table that lists the subcommand in the first column with a brief description in the second column.

Table 14-19 `rdmgr` RD Subcommands

Subcommand	Description
-u	Insert or replace RDs. This subcommand is the default subcommand if none is stated.
-m	Merge RDs filtered by a view.
-g	Retrieve RDs filtered by a view.
-d	Delete RDs.
-n	Count the RDs
-U	Dump database in SOIF to `stdout`.
-L	Lists selected fields from the database to `stdout`. Requires the `-a` *att* option.

Syntax

`# rdmgr [-umgdnUL] [-ACSTNPq] [-a` *att*`,`*att*`,...] [-b` *number*`]`

`[-c` *search.conf*`] [-i` *charset*`] [-o` *charset*`] [-j` *number*`] [-l` *number*`]`

`[-p` *progress*`] [-r` *number*`] [-s` *schema*`] [-y` *dbname*`] [`*filename*`|-Q` *query*`]`

Options

Table 14-20 is a two-column table that lists the options or arguments in the first column with a brief description in the second column. The following options are supported:

Table 14-20 Options for `rdmgr` RD Subcommands

Argument / Operand	Description
-A	Do not use schema aliases in `config/schema.rdm` file in the default search directory. Use with the `u` and `m` subcommands.
-C	Do not create database if database is missing. Use with the `u` and `m` subcommands.
-S	Disable schema checking. Use with the `u` and `m` subcommands.

Table 14-20 Options for `rdmgr` RD Subcommands *(Continued)*

Argument / Operand	**Description**
-T	Operate on the taxonomy. The taxonomy is used for browsing and classifying the database contents and is in the `config/taxonomy.rdm` file in the default search directory. Use with any resource description command.
-N	The function you specified in the command works only on the non-persistent data in the resource description. RDs in the database are a merge of persistent and non-persistent data.
-P	The function you specified in the command works only on the persistent data in the resource description. RDs in the database are a merge of persistent and non-persistent data.
-q	Delete SOIF input file on exit. Use with the `u`, `m`, `g` and `d` subcommands.
-a *att,att...*	Specifies attribute view list. The *att* names are not case sensitive and can be any attribute whether or not they are defined in the schema; for example, author or title. If you have a multi-valued *att* like `class-1`, `class-2`, and `class-3`, only enter `class` as the *att* name.
-b *number*	Set the indexing batch size to this number of RDs. Use with the `u` and `m` subcommands.
`-c` *search.conf*	Specify where the *search.conf* file is. If you do not use this option, the default is the `config/search.conf` file in the default search directory. Other wise, you have to give the full path to the file.
-i *charset* | -o *charset*	The `-i` option specifies the character set of the input SOIF stream. The -o option specifies the character set of the output SOIF stream. For example, ISO8859-1, UTF-8, UTF-16. Character sets ISO8859-1 through ISO8859-15 are supported. Use `-i` with the `u`, `m` and `d` subcommands. Use `-o` with the `g`, `U` and `L` subcommands.
-j *number*	Limits the number of retrieved results. Use with the `u` subcommand. If not stated, the default value is unlimited except with the `Q` option (default 20).
-l *number*	Set log level to number. A setting of 1 (default) logs all the `rdmgr` commands. The higher the number the more detail the log file contains. The possible levels are 1- 100. This works with all subcommands.

Table 14-20 Options for `rdmgr` RD Subcommands *(Continued)*

Argument / Operand	Description
-p {stdout | stderr | *filename*}	Prints or displays progress to `stdout`, `stderr` or the *filename* file. This works with all subcommands. Timing information is reported in seconds.
-r *number*	Use with the progress option. A report is generated every *number* of RDs. The default is 500. Use with the `u`, `m`, `g`, `d` and `U` subcommands.
-s *schema*	Specifies the schema definition file. If you do not use this option, the default is the `config/schema.rdm` file in the search server instance directory.
`-y` *dbname*	Specifies the search database name. If you are running this command on any database other than the default one, you need to use this option. The default database is the database defined in the `config/search.conf` file labeled `datbase-name=`*logicaldbname*.
filename | `-Q` *query*	This input option is used with the `u`, `m`, `g` and `d` subcommands. The *filename* is a file of RDs using the default schema (use `-s` option for any other schema) in SOIF format: The *query* is any regular search query.

NOTE If you enter `rdmgr` with no subcommand, the command assumes the `-u` subcommand. If you enter `rdmgr` with no subcommand and a query (`-Q`), the command assumes the `-g` subcommand.

Examples

Example 1

Set environment variable `LD_LIBRARY_PATH` to `/opt/SUNWps/lib`.

In the `/var/opt/SUNWps/https-sesta.com/portal` directory, type

```
# /opt/SUNWps/bin/rdmgr -U
```

In this example, the entire default database of resource descriptions is printed out to `stdout` in UTF-8 SOIF format.

Example 2

In the default search directory of `/var/opt/SUNWps/https-sesta.com/portal`,

```
# /opt/SUNWps/bin/rdmgr -d -Q java
```

In this example, all the resource descriptions that have java any where in them are deleted.

Database Maintenance Subcommands

Description

The database subcommands allow an administrator to optimize a search database, to truncate or empty a database, to reindex a database, to delete expired RDs from a database, and recover a database. Table 14-21 is a two-column table that lists the subcommand in the first column with a brief description in the second column.

Table 14-21 `rdmgr` Database Maintainance Subcommands

Subcommands	**Description**
-O	Optimize the database. If you are running this subcommand on any database other than the default one, you need to use the `-y` option. The default database is the database defined in the `config/search.conf` file labeled `datbase-name=`*logicaldbname*. For example, the default value is `datbase-name=default` and the default database directory is `db/default`. Databases do not normally need to be optimized.
-X	Truncate or empty a database. If you are running this subcommand on any database other than the default one, you need to use the `-y` option. Disk space used for indexes is recovered, but disk space used by the main database is not recovered, instead, it is reused as new data is added to the database.
-I	Reindex a database. If you are running this subcommand on any database other than the default one, you need to use the `-y` option.
-E	Delete expired RDs from a database. If you are running this subcommand on any database other than the default one, you need to use the `-y` option.

Table 14-21 `rdmgr` Database Maintainance Subcommands *(Continued)*

Subcommands	Description
`-R`	Recover all databases. This is a global command and takes no options. All database processes, including other `rdmgr` instances and the main search server must be stopped before running this command.
-G	Repartition the database. This command takes no options. The partitions are defined in the `config/search.conf` file labeled `datbase-partitions=`*p1,p2,p3,....* where `p1`, `p2`, and `p3` are the filenames of the partitions. The server needs to be restarted after running this command.
-B	Completely deletes the database. Recovers all the disk space. There should be no indexing happening and the Portal server has to be off when you run this subcommand.
`-L`	Lists selected fields from the database to `stdout`. Requires the `-a` *att* option. If you are running this subcommand on any database other than the default one, you need to use the `-y` option.

Syntax

`# rdmgr [-OXIERGBL] [-ASTDVNP] [-a` *att*`,`*att*`,...] [-b` *number*`]`

`[-c` *search.conf*`] [-j` *number*`] [-l` *number*`] [-p` *progress*`] [-r` *number*`]`

`[-s` *schema*`] [-y` *dbname*`]`

Options

Table 14-22 is a two-column table that lists the options or arguments in the first column with a brief description in the second column. The following options are supported:

Table 14-22 Options for `rdmgr` Database Maintenance Subcommands

Argument / Operand	Description
-A	Do not use schema aliases in `config/schema.rdm` file in the default search directory. Use with the `I` subcommand.
-S	Disable schema checking. Use with the `I` subcommand.

Table 14-22 Options for `rdmgr` Database Maintenance Subcommands *(Continued)*

Argument / Operand	**Description**
-T	Operate on the taxonomy. The taxonomy is used for browsing and classifying the database contents and is in `config/taxonomy.rdm` file in the default search directory. Use with `O`, `X`, `I`, `E`, `B`, `U`, and `L` subcommands.
-D	Update the database only; do not update the index. Use with `E` and `X` commands.
-V	Update the index only; do not update the database. Use with `E` and `X` commands.
-N	The function you specified in the command works only on the non-persistent data in the resource description. RDs in the database are a merge of persistent and non-persistent data. Use with `I`, `E`, `U`, and `L` commands.
-P	The function you specified in the command works only on the persistent data in the resource description. RDs in the database are a merge of persistent and non-persistent data. Use with `I`, `E`, `U`, and `L` commands.
-a *att,att...*	Specifies attribute view list. The *att* names are not case sensitive and can be any attribute whether or not they are defined in the schema; for example, author or title. If you have a multi-valued *att* like `class-1`, `class-2`, and `class-3`, only enter `class` as the *att* name.
-b *number*	Set the indexing batch size to this number of RDs. Use with the `I` command.
-c *search.conf*	Specify where the *search.conf* file is. If you do not use this option, the default is the `config/search.conf` file in the default search directory. Other wise, you have to give the full path to the file.
-j *number*	Limits the number of retrieved results. Use with the `E` subcommand. If not stated, the default value is unlimited.
-l *number*	Set log level to number. A setting of 1 (default) logs all the `rdmgr` commands. The higher the number the more detail the logfile contains. The possible levels are 1- 100. This works with all subcommands.
-p {stdout | stderr | *filename*}	Prints or displays progress to `stdout`, `stderr` or *filename*. This works with all subcommands.
-r *number*	Use with the progress option. A report is generated every *number* of RDs. The default is 500. Use with the `u`, `m`, `g`, `d` and `U` subcommands.

Table 14-22 Options for `rdmgr` Database Maintenance Subcommands *(Continued)*

Argument / Operand	Description
-s *schema*	Specifies the schema definition file. The default is the `config/schema.rdm` file in the default search directory.
`-y` *dbname*	Specifies the search database name. If you are running this command on any database other than the default one, you need to use this option. You do not need to use this option for the default database. The default database is the database defined in the `config/search.conf` file labeled `datbase-name=`*filename*.

Examples

Example 1

In the default search directory,

```
# /opt/SUNWps/bin/rdmgr -E -j 13 -p stdout -r 5
```

In this example, up to 13 RDs are removed from the database if they are expired. The progress report to `stdout`, prints the elapsed time in seconds and the number of RDs processed so far after every five resource descriptions.

Example 2

The Search engine is ‘hung’ or not responding. In the default search directory,

```
# /opt/SUNWps/bin/rdmgr -R
```

This recovers all the search databases and makes the Search engine available again. Use this command to release stale locks in the database and to roll back incomplete data transactions. Stale locks and incomplete transactions can result from a database process being abnormally terminated.

Usage Message and Version Subcommands

Table 14-23 lists the subcommands to display the usage message or to view the version information in the first column and a brief description in the second column.

Table 14-23 rdmgr Subcommands for Usage Message and Version

Argument / Operand	Description
`-h` or `-?`	Show the usage message.

Table 14-23 rdmgr Subcommands for Usage Message and Version

Argument / Operand	Description
`-v`	Show version information

Return Codes

The rdmgr command returns return codes to the shell.

0 - Success

1 - Failure

sendrdm

Description

The `sendrdm` command provides a mechanism for a CGI or command-line based search. An rdm (resource description manager) request is sent in SOIF format to the Search server. This command is normally run in a search-enabled Portal Server instance directory, which is the `/server-instance-directory/deployment_uri` directory. This is the deployment uri path you selected at install time. If you chose the default Portal Server install, this is the `/var/opt/SUNWps/https-`*servername*`/portal` directory. Where the value of the *servername* is the default Portal Server instance name—the fully qualified name of your Portal Server.

NOTE For the default installation, set the environment variable `LD_LIBRARY_PATH` to `/opt/SUNWps/lib`.

Syntax

The syntax of the `sendrdm` command is:

```
# sendrdm [-dv] [-t n] [-u uri] RDM-in [RDM-out]
```

Options

Table 14-24 is a summary of the `sendrdm` command. It contains two columns: the first column lists the possible options; the second column gives a brief description. The command supports the following options (listed in alphabetical order):

Table 14-24 `sendrdm` Command Options

Argument/Operand	Description
`-d`	Debug mode. The default is off. This option turns it on.
`-t` *n*	Specifies time in seconds. Command times out after *n* seconds. Default is 300 seconds.
`-u` *uri*	Designates the uri directory (enter full path) for the server you are importing from.
`-v`	Version.
RDM-in	The RDM request file name. This is a required argument.
RDM-out	The RDM result file name. The default is standard out.

Example

In the `/var/opt/SUNWps/https-`*servername*`/portal` directory as root:

```
# /opt/SUNWps/lib/sendrdm -t 3600 -u /rdm/incoming rdmquery.soif
result.soif
```

This example imports from a Compass Server 3.01x using `/rdm/incoming` as the uri with the timeout set to one hour. The content of `rdmquery.soif` is:

Code Example 0-1

```
@RDMHEADER { -
catalog-service-id{48}: x-catalog://frankie.sesta.com:89/Compass-2

rdm-type{10}:   rd-request
rdm-version{3}: 1.0
rdm-query-language{8}:  gatherer
}
@RDMQUERY { -
scope{3}:       all
}
```

StartRobot

The `StartRobot` script can be used by an administrator to start the robot manually. Usually this script is used by the scheduler to start the robot at set times (cron job). The `StartRobot` command is in the `/var/opt/SUNWps/https-`*servername*`/portal` directory.

Syntax

```
#  StartRobot
```

Options

There are no options.

StopRobot

The `StopRobot` script can be used by an administrator to stop the robot manually. Usually this script is used by the scheduler to stop the robot at set times (cron job). The `StopRobot` command is in the `/var/opt/SUNWps/https-`*servername*`/portal` directory.

Syntax

```
#  StopRobot
```

Options

There are no options.

Appendix A

Configuration Files

This appendix describes the `desktopconfig.properties` and `search.conf` configuration files.

This appendix contains these sections:

- Desktop Configuration Properties
- Search Configuration Properties

Overview of Sun ONE Portal Server Configuration Files

Sun ONE Portal Server uses certain files to manage the configuration of the Desktop and Search services. The Desktop configuration file, `desktopconfig.properties`, defines server-specific parameters, and is discussed in this appendix.

There is the Portal Server as a service of Sun ONE Identity Server (see Appendix B).

The Search service uses specific configuration files. This appendix covers configuration considerations for the `search.conf` file.

At installation time, you are given the option of defining values or using the default values for the Base Directory (/opt), the Deployment URI (/portal) and the Deploy Instance (cate.sesta.com).

Desktop Configuration Properties

The `desktopconfig.properties` file defines server-specific parameters that the Desktop reads during initialization. Any changes to this file require a server restart in order to go into effect. By default, this file is in the `/etc/opt/SUNWps/desktop` directory.

Code Example A-1 `desktopconfig.properties` File

```
#
# Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
# PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.
#

#########################
# Desktop Configuration #
#########################

#
# Debug level
#
# Possible values for the debugLevel are: off | error | warning | message.
#
# The debug output will be logged in a file, called ‘desktop.debug’ located
# under ‘/var/opt/SUNWam/debug’ by default.
#
debugLevel=error

#
# Perf (log) level
#
# Possible values for the perfLevel are: off | error | warning | message.
#
# The performance output will be logged in a file, called ‘desktop.perf’
located
# under ‘/var/opt/SUNWam/debug’ by default.
#
perfLevel=off

#
# ServiceAppContext Class Name
#
serviceAppContextClassName=com.sun.portal.desktop.context.DSAMEServiceAppConte
xt

#
# Template Base Directory
#
templateBaseDir=/etc/opt/SUNWps/desktop/

#
# Provider Class Base Directory
```

Code Example A-1 `desktopconfig.properties` File *(Continued)*

```
#
providerClassBaseDir=/etc/opt/SUNWps/desktop/classes

#
# JSP Compiler WAR Classpath
#
#jspCompilerWARClassPath=/export/home/ias60sp3/ias/APPS/modules/ps/WEB-INF/lib
jspCompilerWARClassPath=<Used only on application server>

#
# Desktop type
#
defaultDesktopType=default

#
# Provider getter pool settings (initializing channels)
#
getterPoolMinSize=0
getterPoolMaxSize=0
getterPoolPartitionSize=0

#
# Provider caller pool settings (fetching channel content)
#
callerPoolMinSize=0
callerPoolMaxSize=0
callerPoolPartitionSize=0

#
# prefix used for all desktop cookies
#
cookiePrefix=desktop.

#
# template file rescan time in seconds
#
templateScanInterval=30
```

Parameters marked as Internal are not customizable. So you can only configure the debug level and the base directory for additional classes. Table A-1 is a two column table; the first column lists the parameter with its default value and the second column gives a description of its function and possible values.

Table A-1 `desktopconfig.properties` Parameters

Parameter/ Default Value	Description
`debugLevel=error`	Level of debugging messages to be produced by Desktop. Debug output is stored in `/var/opt/SUNWam/debug/desktop.debug`. Use caution when increasing the value of logLevel for excessive logging causes intensive IO operations leading to performance degradation. Possible values are: (in the ascending order of less to more logging) `off`, `error`, `warning`, `message`, or `on` • `off`: No logging • `error`: Log errors only • `warning`: Log errors and warning • `message`: Log everything Default value: `error`
`perfLevel=off`	[Internal] Level of performance metrics to be logged by Desktop. Output is stored in `/var/opt/SUNWam/debug/desktop.perf`. Under production environment, this parameter should always be `off`. Possible values are: `off` or `message` • `off`: No performance metrics logged • `message`: Log all performance metrics Default value: `off`
`serviceAppContextClassName=com.sun.portal.desktop.context.DSAMEServiceAppContext`	[Internal] Default value: `com.sun.portal.desktop.context.DSAMEServiceAppContext`
`templateBaseDir=/etc/opt/SUNWps/desktop/`	[Internal] Root directory under which all template files are located. Default value: `/etc/opt/SUNWps/desktop/`
`providerClassBaseDir=/etc/opt/SUNWps/desktop/classes`	Root directory under which the customer is allowed to place provider classes, whether those are overriding the out-of-the-box providers, or their own new providers (usually the case). They must be placed in this directory, either in a jar at the top level, or in a com (or whatever) package directory. Default value: `/etc/opt/SUNWps/desktop/classes`

Table A-1 `desktopconfig.properties` Parameters *(Continued)*

Parameter/ Default Value	Description
`jjspCompilerWARClassPath=<Used only on application server>` `jjspCompilerWARClassPath=/export/home/ias60sp3/ias/APPS/modules/ps/WEB-INF/lib`	[Internal] Used only on application server. Default value:
`defaultDesktopType=default`	[Internal] Default desktop type used by the ErrorProvider when DesktopAppContext is available but DesktopContext is not available. Default value: `default`
`getterPoolMinSize=0`	[Internal] Default value: `0`
`getterPoolMaxSize=0`	[Internal] Default value: `0`
`getterPoolPartitionSize=0`	[Internal] Default value: `0`
`callerPoolMinSize=0`	[Internal] Default value: `0`
`callerPoolMaxSize=0`	[Internal] Default value: `0`
`callerPoolPartitionSize=0`	[Internal] Default value: `0`
`cookiePrefix=desktop.`	[Internal] Prefix used for all desktop cookies. Default value: `desktop.`
`templateScanInterval`	Defines number of seconds between scans (checking for changes) of the template files in the `/etc/opt/SUNWps` directory. This interval can improve the performance and scalability because the server uses the cached information between scans. The default value is 30 seconds

Search Configuration Properties

In the default installation, the `search.conf` file is in the `/var/opt/SUNWps/https-`*instancename*`/portal/config` directory. The `search.conf` file lists all the specific search values you have set. The `/opt/SUNWps/samples/config` directory contains a sample `search.conf` file.

Code Example A-2 search.conf File

```
#
#  search.conf - Search configuration
#

csid=x-catalog://cate.sesta.com:80/cate.sesta.com
bindir=/opt/SUNWps/bin
database-directory=/var/opt/SUNWps/https-cate.sesta.com/portal/db
database-root=/var/opt/SUNWps/https-cate.sesta.com/portal/db
database-max-concurrent=8
database-name=default
database-logdir=db
security-mode=OFF
security-manager=com.sun.portal.search.rdmserver.DSameSecurityManager
debug-logfile=/var/opt/SUNWps/https-cate.sesta.com/portal/logs/rdmserver.log
debug-loglevel=1
filters-check-dns=on
filters-check-redirect=on
filters-check-virtual=on
import-config=/var/opt/SUNWps/https-cate.sesta.com/portal/config/import.conf
libdir=/opt/SUNWps/lib
logfile=/var/opt/SUNWps/https-cate.sesta.com/portal/logs/rdm.log
disable-rdm-log=false
multiple-classifications=3
classification-stats-during-browse=true
browse-root-classification=false
search-logfile=/var/opt/SUNWps/https-cate.sesta.com/portal/logs/searchengine.l
og
search-max-index-batch=2000
search-query-threads=6
search-index-threads=1
search-index-type=AWord
search-index-partition-size=32
search-dictionary-type=partial
search-lookup-limit=-1
search-highlights=true
search-max-passages=3
search-passage-context=6
#search-field-multipliers=”title 1.0”
reports-exclude-gv-queries=false
reports-exclude-browse=false
rdmgr-logfile=/var/opt/SUNWps/https-cate.sesta.com/portal/logs/rdmgr.log
# comment rdmgr-pidfile to prevent rdmgr daemonization
# rdmgr-pidfile=/var/opt/SUNWps/https-cate.sesta.com/portal/logs/rdmgr.pid
```

Code Example A-2 search.conf File *(Continued)*

```
schema-description=/var/opt/SUNWps/https-cate.sesta.com/portal/config/schema.r
dm
server-description=/var/opt/SUNWps/https-cate.sesta.com/portal/config/server.r
dm
server-root=/var/opt/SUNWps/https-cate.sesta.com/portal
taxonomy-database-name=taxonomy
taxonomy-description-refresh-rate=60
taxonomy-description=/var/opt/SUNWps/https-cate.sesta.com/portal/config/taxono
my.rdm
tmpdir=/var/opt/SUNWps/https-cate.sesta.com/portal/tmp
rlog-max-logs=10
robot-refresh=30000
admin-category_editor_nodes_per_page=25,50,100,250,500,-1
admin-category_editor_max_combo_element=10
```

The default install assigns `$CSROOT` to `/var/opt/SUNWps/http-`*instancename*`/portal`, `$CSBIN` to `/opt/SUNWps/bin`, and `$CSLIB` to `/opt/SUNWps/lib`. Most of these parameters can be changed in the Sun ONE Application Server Enterprise Edition administration console under Search Server Settings or Search Server Advanced Settings.

Table A-2 is a three column table. Column one gives the possible parameters you can change, column two lists the default value of the parameter, and column three provides a brief description.

Table A-2 `search.conf` Parameters

Parameter	Default	Description
`csid`	`x-catalog://$HOST:$PORT/$NICK`	Defined at installation. Server identifier string, mainly for backward compatibility with Compass Server.
`bindir`	`$CSBIN`	Defined at installation. Location of binaries.
`database-directory`	`$CSROOT/db`	Defined at installation. Location of database (used by server).
`database-root`	`$CSROOT/db`	Defined at installation. Location of database (used by indexer).
`database-max-concurrent`	`8`	Limits the number of server threads that can access the database at any one time. You can change this value for performance reasons, but it should be set to about 1.25 times the number of index threads for best performance.

Table A-2 `search.conf` Parameters *(Continued)*

Parameter	Default	Description
`database-name`	`default`	The logical database name. You can change this value to another database including an external one.
`database-logdir`	`db`	Directory where database transaction logs are kept.
`security-mode`	`OFF`	Enables or disables document level security. Can be reset in the administration console under Server Settings.
`security-manager`	`com.sun.portal.search.rdmserver.DSameSecurityManager`	Security manager class name. Do not edit.
`security-dsame-group`	`OFF`	Whether to use group in addition to user role for security control.
`debug-logfile`	`$CSROOT/logs/rdmserver.log`	Logs internal server activity. Defined at installation. Can be reset in the administration console under Server Advanced Settings.
`debug-loglevel`	`1`	Sets the default log level. Can be reset in the administration console under Server Advanced Settings.
`filters-check-dns`	`on`	Checks for number of servers aliased to the same address. Can be reset in the administration console under Robot Simulator.
`filters-check-redirect`	`on`	Checks for any server redirects. Can be reset in the administration console under Robot Simulator.
`import-config`	`$CSROOT/config/import.conf`	Defined at installation. Contents generated by the Search server when you define an import agent in the administration console under Database Import.
`libdir`	`$CSLIB`	Defined at installation.
`logfile`	`$CSROOT/logs/rdm.log`	Log of RDM server requests. Defined at installation. Can be reset in the administration console under Server Advanced Settings.
`disable-rdm-log`	`false`	Disables RDM request logging. Can be reset in the administration console under Server Advanced Settings.
`classification-stats-during-browse`	`true`	If true, server records how many documents are found in each browse category.

Table A-2 `search.conf` Parameters *(Continued)*

Parameter	Default	Description
`browse-root-classification`	`false`	Whether to browse for documents at the root of the category tree.
`search-logfile`	`$CSROOT/logs/searchengine.log`	Search engine logfile. Defined at installation. Can be reset in the administration console under Server Advanced Settings.
`search-max-index-batch`	`2000`	Maximum number of documents in each index batch.
`search-query-threads`	`6`	Number of search query threads. Should be set to 3-6 threads per cpu that you wish to utilize.
`search-index-threads`	`1`	Number of search index threads. Usually left at 1.
`search-index-type`	`AWord`	The format of the search engine index. Do not edit.
`search-index-partition-size`	`32`	The blocking factor used during index merges. Do not edit.
`search-dictionary-type`	`partial`	Format of the search dictionary. Do not edit.
`search-lookup-limit`	`-1`	Controls the timeout (milliseconds) of slow wildcard searches. -1 means unlimited.
`search-highlights`	`true`	Enable search result highlighting.
`search-max-passages`	`3`	Maximum number of dynamic summary passages to generate.
`search-passage-context`	`6`	Size of context (in words) around each highlight passage.
`#search-field-multipliers`	`"title 1.0"`	Search weights assigned to different document fields. Can be a comma separated list.
`rdmgr-logfile`	`$CSROOT/logs/rdmgr.log`	Log file for the indexer process. Defined at installation. Can be reset in the administration console under Server Advanced Settings.
`schema-description`	`$CSROOT/config/schema.rdm`	The default Search Engine Schema. Defined at installation.
`server-description`	`$CSROOT/config/server.rdm`	The RDM server description returned by server description requests. Defined at installation.
`server-root`	`$CSROOT`	Server instance root directory. Defined at installation. Can be reset in the administration console under Server Settings.
`taxonomy-database-name`	`taxonomy`	The logical name of the taxonomy index database.

Table A-2 `search.conf` Parameters *(Continued)*

Parameter	Default	Description
`taxonomy-description-refresh-rate`	`3600 -> 60`	Polling interval for automatic taxonomy reloads.
`taxonomy-description`	`$CSROOT/config/taxonomy.rdm`	The RDM Taxonomy definition. Edit using the Category Editor under Categories. Defined at installation.
`tmpdir`	`$CSROOT/tmp`	Defined at installation. Can be reset in the administration console under Robot Crawling.
`robot-refresh`	`30000`	Number of milliseconds between refreshes of the Robot Control page of the administration console.
`admin-category_editor_nodes_per_page`	`25,50,100,250,500,-1`	List of available choices, defining the maximum number of categories displayed per page. -1 = display all tree.
`admin-category_editor_max_combo_element`	`10`	Maximum number of elements in the category editor drop down select list of target categories.

NOTE The following parameters are not used, so are not included in Table A-2: `filters-check-virtual`, `multiple-classifications`, `reports-exclude-gv-queries`, `reports-exclude-browse`, `rdmgr-pidfile`, and `rlog-max-logs`.

Appendix B

XML Reference

As an iPlanet™ Directory Server Access Management Edition application, the Sun™ ONE Portal Server registers its services into the Sun ONE Identity Server Service Management Services (SMS) framework. This occurs during the pre-installation of the Sun ONE Portal Server and post-installation for Sun ONE Identity Server..

NOTE In general, any service-related data that is not server-specific is stored in the Sun ONE Identity Server directory. Server-specific data can be stored in properties files that are local to the specific server.

SMS provides a mechanism for services to define and manage their configuration data by using an Extensible Markup Language (XML) file that adheres to the SMS Document Type Definition (DTD). The definition of the configuration parameters through the XML file is called the schema for the service. Each Sun ONE Portal Server service (Desktop, Netmail, Rewriter, and Search) has its own XML and properties files for presenting and modifying service specific data.

Within the Sun ONE Identity Server framework, Sun ONE Portal Server defines services related to the following functional areas:

- Desktop—The `SunPortalDesktopService` includes data associated with the Desktop component, including the display profile and other configuration parameters associated with the Desktop.
- Search Engine—The `SunPortalSearchService` defines the data associated with the Search component, such as the search person and search instances. One or more instances of Search service instances can be defined.
- NetMail—The `SunPortalNetMailService` includes data associated with the NetMail application primarily consisting of the user's preferences.

- Rewriter—The `SunPortalRewriterService` includes data associated with the Rewriter component, including the named rule sets that control the rewriting operation. The Rewriter API makes reference to the named rule sets that are stored in the directory.

In addtion, the Sun ONE Portal Server also uses other DTDs to define LDAP attribute values for the display profile and the Rewriter ruleset.

The Display Profile Document Type Definition (DTD) defines how the Display Profile is structured. The underlying data format for a display profile document is XML. It is intended to define the display configuration for the Desktop. It does that by defining provider, portlet, and channel objects, and their properties. See Chapter 5, "Administering the Display Profile"" for more information about the display profile.

The Rewriter ruleset DTD defines the structure of the ruleset. The Rewriter includes a default ruleset. Chapter 7, "Administering the Rewriter Service"" gives more details about the Rewriter and how it uses a ruleset.

This appendix provides a file listing for various XML files used to define the services of theSun ONE Portal Server. It contains the following sections:

Sun ONE Portal Server Desktop Service Definition

With the default installation, the Service Management Services Document Type Definition is found in the `/opt/SUNWam/dtd/sms.dtd` file. The portal server Desktop service definition is in the `/opt/SUNWps/export/psDesktop.xml` file.

Code Example B-1 Desktop Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--
    Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
    PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.

    Sun ONE Portal Server (iPS) Desktop Service Definition
-->

<!DOCTYPE ServicesConfiguration
    PUBLIC "=//Sun ONE//Service Management Services (SMS) 1.0 DTD//EN"
    "jar://com/iplanet/sm/sms.dtd">

<ServicesConfiguration>
    <Service name="SunPortalDesktopService" version="1.0">
            <Schema
                i18nFileName="psDesktop"
                serviceHierarchy="/ps.configuration/SunPortalDesktopService"
        propertiesViewBeanURL="/portal/dtadmin/DesktopAdminService"
                i18nKey="sunPortalDesktopServiceDescription">
                <Global>
                    <AttributeSchema name="serviceObjectClasses"
                        type="list"
                        syntax="string"
            i18nKey="">
                        <DefaultValues>
                                <Value>sunPortalDesktopPerson</Value>
                        </DefaultValues>
    </AttributeSchema>
<!--
    desktop session reaping has been disabled per
    bug #4720290. see this bug report for details.

    when this bug is resolved, the entries must be added
    back into psDesktop.ldif, as they have been removed
    completely for lack of a commenting feature


                    <AttributeSchema name="sunPortalDesktopSessionReapInterval"
                        type="single"
                        syntax="string"
                        any="display"
                        i18nKey="g2">
                        <DefaultValues>
                                <Value>1800</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopSessionInactiveMaximum"
                        type="single"
                        syntax="string"
                        any="display"
                        i18nKey="g3">
```

Code Example B-1 Desktop Service Definition *(Continued)*

```
                        <DefaultValues>
                                <Value>3600</Value>
                        </DefaultValues>
            </AttributeSchema>
-->
                    <AttributeSchema name="sunPortalDesktopDpIsValidating"
                        type="single"
                        syntax="boolean"
                        any="display"
                        i18nKey="g5">
                        <DefaultValues>
                                <Value>true</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpNamespaceURI"
                        type="single"
                        syntax="string"
                        any="display"
                        i18nKey="g6">
                        <DefaultValues>
                                <Value>http://www.iplanet.com</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpDocument"
                        type="single"
                        syntax="xml"
                        any="display"
                        i18nKey="g7">
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpLastModified"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>
                                <Value>-1</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopSessionReturnURLParamName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>
                                <Value>goto</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DSAMEDPContext</Value>
                        </DefaultValues>
```

Code Example B-1 Desktop Service Definition *(Continued)*

```
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopDpUserContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DSAMEDPUserContext</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopDebugContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DSAMEDebugContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopServiceContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DSAMEServiceContext</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopSessionAppContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DSAMESessionAppContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopSessionContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DSAMESessionContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
```

Code Example B-1 Desktop Service Definition *(Continued)*

```
                <AttributeSchema
name="sunPortalDesktopAuthlessSessionContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.CookieSessionContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopDesktopContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.PSDesktopContext</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopContainerProviderContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.PSContainerProviderContext</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopProviderManagerContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.PSProviderContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopPropertiesContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DPPropertiesContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopTemplateContextClassName"
```

Code Example B-1 Desktop Service Definition *(Continued)*

```
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.FileTemplateContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopClientContextClassName"
                        type="single"
                        syntax="string"
                        any="display">
                        <DefaultValues>

<Value>com.sun.portal.desktop.context.DSAMEClientContext</Value>
                        </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalDesktopAuthorizedAuthlessUIDs"
                         type="list"
                         syntax="string"
                 any="display"
                         i18nKey="g8">
                        <DefaultValues>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDefaultAuthlessUID"
                        type="single"
                        syntax="string"
                        any="display"
            i18nKey="g9">
                        <DefaultValues>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopChannelImportModules"
                        type="list"
                        syntax="string"
                        any="display">
                        <DefaultValues>
                        </DefaultValues>
                    </AttributeSchema>
                </Global>

                <Dynamic>
                    <AttributeSchema name="sunPortalDesktopDefaultChannelName"
                        type="single"
                        syntax="string"
                        cosQualifier="default"
                        any="display"
                        i18nKey="d1">
                        <DefaultValues>
                                <Value>JSPTabContainer</Value>
                        </DefaultValues>
```

Code Example B-1 Desktop Service Definition *(Continued)*

```
            </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopEditProviderContainerName"
                        type="single"
                        syntax="string"
                        cosQualifier="default"
                        any="display"
            i18nKey="d2">
                        <DefaultValues>
                                <Value>JSPEditContainer</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopType"
                        type="single"
                        syntax="string"
                        cosQualifier="default"
                        any="display"
                        i18nKey="d3">
                        <DefaultValues>
                                <Value>default</Value>
                        </DefaultValues>
                    </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpDocument"
                        type="single"
                        syntax="xml"
                        cosQualifier="default"
                        any="display"
                        i18nKey="d4">
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpLastModified"
                        type="single"
                        syntax="string"
                        cosQualifier="default"
                        any="display"
                        i18nKey="">
                        <DefaultValues>
                                <Value>-1</Value>
                        </DefaultValues>
            </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpCanView"
                        type="single"
                        syntax="boolean"
                        cosQualifier="default"
                        any="display"
                        i18nKey="d5">
                        <DefaultValues>
                                <Value>true</Value>
                        </DefaultValues>
                    </AttributeSchema>
                </Dynamic>

                <Policy>
                    <ActionSchema name="sunPortalDesktopExecutable"
                        type="single"
                        syntax="boolean"
```

Code Example B-1 Desktop Service Definition *(Continued)*

```
                        any="display"
                        cosQualifier="default"
            i18nKey="p1">
                        <DefaultValues>
                                <Value>true</Value>
                        </DefaultValues>
            </ActionSchema>
                </Policy>

                <User>
                    <AttributeSchema name="sunPortalDesktopDefaultChannelName"
                        type="single"
                        syntax="string"
                        cosQualifier="default"
                        any="display"
                        i18nKey="d1">
                    </AttributeSchema>
                    <AttributeSchema
name="sunPortalDesktopEditProviderContainerName"
                        type="single"
                        syntax="string"
                        cosQualifier="default"
                        any="display"
                        i18nKey="d2">
                    </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopType"
                        type="single"
                        syntax="string"
                        cosQualifier="default"
                        any="display"
                        i18nKey="d3">
                    </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpDocumentUser"
                        type="single"
                        syntax="xml"
                        any="display"
                        i18nKey="u1">
                    </AttributeSchema>
                    <AttributeSchema name="sunPortalDesktopDpLastModifiedUser"
                        type="single"
                        syntax="string"
                        any="display"
                        i18nKey="">
                        <DefaultValues>
                                <Value>-1</Value>
                        </DefaultValues>
                    </AttributeSchema>
                </User>

            </Schema>
    </Service>
</ServicesConfiguration>
```

Sun ONE Portal Server NetMail Service Definition

With the default installation, the Service Management Services Document Type Definition is found in the `/opt/SUNWam/dtd/sms.dtd` file. The portal server NetMail service definition is in the `/opt/SUNWps/export/psNetMail.xml` file

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--
  Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
  PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.

  Sun ONE Portal Server (iPS) NetMail Service Definition
-->

<!--
    FIXME(P@): replace below DTD path with a token that can
                be substituted by postinstall script
-->

<!DOCTYPE ServicesConfiguration
    PUBLIC "=//Sun ONE//Service Management Services (SMS) 1.0 DTD//EN"
    "file:/opt/SUNWam/dtd/sms.dtd">

<ServicesConfiguration>
    <Service name="SunPortalNetMailService" version="1.0">
        <Schema
            i18nFileName="psNetMail"
            serviceHierarchy="/ps.configuration/SunPortalNetMailService"
            i18nKey="sunPortalNetmailServiceDescription">
            <Global>
                <AttributeSchema name="serviceObjectClasses"
                    type="list"
                    syntax="string"
                            i18nKey="">
                    <DefaultValues>
                        <Value>sunPortalNetmailPerson</Value>
                    </DefaultValues>
                        </AttributeSchema>
            </Global>
            <Dynamic>
                <AttributeSchema name="sunPortalNetmailIMAPServerName"
                    type="single"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a01">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailSMTPServerName"
                    type="single"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a02">
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                </AttributeSchema>
               <AttributeSchema name="sunPortalNetmailDefaultMailDomain"
                    type="single"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a03">
                    </AttributeSchema>
               <AttributeSchema name="sunPortalNetmailRootFolder"
                    type="single"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a04">
                    <DefaultValues>
                        <Value>Mail</Value>
                    </DefaultValues>
                    </AttributeSchema>
               <AttributeSchema name="sunPortalNetmailSentMessagesFolder"
                    type="single"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a15">
                    <DefaultValues>
                        <Value>Mail/Sent</Value>
                    </DefaultValues>
                    </AttributeSchema>
               <AttributeSchema name="sunPortalNetmailReplyWithAuthor"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a20">
                    <DefaultValues>
                        <Value>false</Value>
                    </DefaultValues>
                    </AttributeSchema>
               <AttributeSchema name="sunPortalNetmailReplyWithDate"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a21">
                    <DefaultValues>
                        <Value>false</Value>
                    </DefaultValues>
                    </AttributeSchema>
               <AttributeSchema name="sunPortalNetmailReplyWithBody"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a22">
                    <DefaultValues>
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                        <Value>true</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailIndentPrefix"
                    type="single"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a18">
                    <DefaultValues>
                        <Value>></Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAddSignature"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a19">
                    <DefaultValues>
                        <Value>false</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailInitialHeaders"
                    type="single"
                    syntax="numeric"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a07">
                    <DefaultValues>
                        <Value>10</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailInactivityInterval"
                    type="single"
                    syntax="numeric"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a09">
                    <DefaultValues>
                        <Value>5</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailMaxAttachLen"
                    type="single"
                    syntax="numeric"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a10">
                    <DefaultValues>
                        <Value>0</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutoload"
                    type="single_choice"
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                    syntax="numeric"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a11">
                    <ChoiceValues>
                        <ChoiceValue i18nKey="autoload.All">0</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.None">1</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.New">2</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.Unread">3</ChoiceValue>
                        <ChoiceValue
i18nKey="autoload.New_and_Unread">4</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.Found">5</ChoiceValue>
                    </ChoiceValues>
                    <DefaultValues>
                        <Value>0</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutosave"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a17">
                    <DefaultValues>
                        <Value>true</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutopurge"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a14">
                    <DefaultValues>
                        <Value>false</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutoFolderLoad"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a05">
                    <DefaultValues>
                        <Value>false</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailMultipleReadWindows"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a13">
                    <DefaultValues>
                        <Value>false</Value>
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailHeadersPerPage"
                    type="single"
                    syntax="numeric"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a08">
                    <DefaultValues>
                        <Value>10</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailNewestFirst"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a12">
                    <DefaultValues>
                        <Value>true</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailNoPrefsList"
                    type="multiple_choice"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a23">
                    <ChoiceValues>
                        <ChoiceValue>IMAPServerName</ChoiceValue>
                        <ChoiceValue>IMAPUserName</ChoiceValue>
                        <ChoiceValue>IMAPPassword</ChoiceValue>
                        <ChoiceValue>SMTPMailServer</ChoiceValue>
                        <ChoiceValue>rootFolder</ChoiceValue>
                        <ChoiceValue>inactivityInterval</ChoiceValue>
                        <ChoiceValue>initialHeaders</ChoiceValue>
                        <ChoiceValue>multipleReadWindows</ChoiceValue>
                        <ChoiceValue>resetSize</ChoiceValue>
                        <ChoiceValue>autopurge</ChoiceValue>
                        <ChoiceValue>replyToAddress</ChoiceValue>
                        <ChoiceValue>indentPrefix</ChoiceValue>
                        <ChoiceValue>replyFields</ChoiceValue>
                        <ChoiceValue>saveSentMessages</ChoiceValue>
                        <ChoiceValue>sentMessagesFolder</ChoiceValue>
                        <ChoiceValue>signature</ChoiceValue>
                        <ChoiceValue>autosave</ChoiceValue>
                        <ChoiceValue>autoFolderLoad</ChoiceValue>
                        <ChoiceValue>autoload</ChoiceValue>
                        <ChoiceValue>maxAttachLen</ChoiceValue>
                        <ChoiceValue>textStyle</ChoiceValue>
                        <ChoiceValue>textSize</ChoiceValue>
                        <ChoiceValue>textColor</ChoiceValue>
                        <ChoiceValue>backgroundColor</ChoiceValue>
                    </ChoiceValues>
                    </AttributeSchema>
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                <AttributeSchema name="sunPortalNetmailLDAPServers"
                    type="list"
                    syntax="string"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a06">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailLogMessages"
                    type="single"
                    syntax="boolean"
                    cosQualifier="default"
                    any="display"
                    i18nKey="a16">
                    <DefaultValues>
                        <Value>false</Value>
                    </DefaultValues>
                    </AttributeSchema>
            </Dynamic>

            <Policy>
                <ActionSchema name="sunPortalNetmailExecutable"
                    type="single"
                    syntax="boolean"
                    any="display"
                    cosQualifier="default"
                            i18nKey="p1">
                    <DefaultValues>
                        <Value>true</Value>
                    </DefaultValues>
                    </ActionSchema>
            </Policy>

            <User>
                <AttributeSchema name="sunPortalNetmailIMAPUserid"
                        type="single"
                        syntax="string"
                            any="display"
                        i18nKey="u1">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailIMAPPassword"
                        type="single"
                        syntax="encrypted_password"
                            any="display"
                        i18nKey="u2">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailReplyToAddress"
                        type="single"
                        syntax="string"
                            any="display"
                        i18nKey="u3">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailSignature"
                        type="single"
                        syntax="paragraph"
                            any="display"
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                    i18nKey="u4">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailFavoriteFolders"
                    type="list"
                    syntax="string"
                        any="display"
                    i18nKey="">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailPersonalAddressBook"
                    type="list"
                    syntax="string"
                        any="display"
                    i18nKey="">
                </AttributeSchema>
                <!-- attributes duplicated from the dynamic section -->
            <AttributeSchema name="sunPortalNetmailIMAPServerName"
                type="single"
                syntax="string"
                any="display"
                i18nKey="a01">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailSMTPServerName"
                type="single"
                syntax="string"
                any="display"
                i18nKey="a02">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailDefaultMailDomain"
                type="single"
                syntax="string"
                any="display"
                i18nKey="a03">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailRootFolder"
                type="single"
                syntax="string"
                any="display"
                i18nKey="a04">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailSentMessagesFolder"
                type="single"
                syntax="string"
                any="display"
                i18nKey="a15">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailReplyWithAuthor"
                type="single"
                syntax="boolean"
                any="display"
                i18nKey="a20">
                </AttributeSchema>
            <AttributeSchema name="sunPortalNetmailReplyWithDate"
                type="single"
                syntax="boolean"
                any="display"
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                    i18nKey="a21">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailReplyWithBody"
                    type="single"
                    syntax="boolean"
                    any="display"
                    i18nKey="a22">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailIndentPrefix"
                    type="single"
                    syntax="string"
                    any="display"
                    i18nKey="a18">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAddSignature"
                    type="single"
                    syntax="boolean"
                    any="display"
                    i18nKey="a19">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailInitialHeaders"
                    type="single"
                    syntax="numeric"
                    any="display"
                    i18nKey="a07">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailInactivityInterval"
                    type="single"
                    syntax="numeric"
                    any="display"
                    i18nKey="a09">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailMaxAttachLen"
                    type="single"
                    syntax="numeric"
                    any="display"
                    i18nKey="a10">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutoload"
                    type="single_choice"
                    syntax="numeric"
                    any="display"
                    i18nKey="a11">
                    <ChoiceValues>
                        <ChoiceValue i18nKey="autoload.All">0</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.None">1</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.New">2</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.Unread">3</ChoiceValue>
                        <ChoiceValue
i18nKey="autoload.New_and_Unread">4</ChoiceValue>
                        <ChoiceValue i18nKey="autoload.Found">5</ChoiceValue>
                    </ChoiceValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutosave"
                    type="single"
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                    syntax="boolean"
                    any="display"
                    i18nKey="a17">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutopurge"
                    type="single"
                    syntax="boolean"
                    any="display"
                    i18nKey="a14">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailAutoFolderLoad"
                    type="single"
                    syntax="boolean"
                    any="display"
                    i18nKey="a05">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailMultipleReadWindows"
                    type="single"
                    syntax="boolean"
                    any="display"
                    i18nKey="a13">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailSortKey"
                    type="single_choice"
                    syntax="numeric"
                    any="display"
                    i18nKey="">
                    <ChoiceValues>
                        <ChoiceValue
i18nKey="sort-key.IMAP_Number">0</ChoiceValue>
                        <ChoiceValue i18nKey="sort-key.Status">1</ChoiceValue>
                        <ChoiceValue i18nKey="sort-key.Cached">2</ChoiceValue>
                        <ChoiceValue i18nKey="sort-key.From">3</ChoiceValue>
                        <ChoiceValue i18nKey="sort-key.Size">4</ChoiceValue>
                        <ChoiceValue i18nKey="sort-key.Date">5</ChoiceValue>
                       <ChoiceValue i18nKey="sort-key.Subject">6</ChoiceValue>
                    </ChoiceValues>
                    <DefaultValues>
                        <Value>0</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailViewKey"
                    type="single_choice"
                    syntax="numeric"
                    any="display"
                    i18nKey="">
                    <ChoiceValues>
                        <ChoiceValue i18nKey="view-key.All">0</ChoiceValue>
                        <ChoiceValue i18nKey="view-key.New">1</ChoiceValue>
                        <ChoiceValue
i18nKey="view-key.Non-deleted">2</ChoiceValue>
                        <ChoiceValue i18nKey="view-key.Cached">3</ChoiceValue>
                        <ChoiceValue
i18nKey="view-key.Non-cached">4</ChoiceValue>
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                        <ChoiceValue i18nKey="view-key.Found">5</ChoiceValue>
                        <ChoiceValue i18nKey="view-key.Unread">6</ChoiceValue>
                    </ChoiceValues>
                    <DefaultValues>
                        <Value>0</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailComposeWinBounds"
                    type="single"
                    syntax="string"
                    any="display"
                    i18nKey="">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailFolderWinBounds"
                    type="single"
                    syntax="string"
                    any="display"
                    i18nKey="">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailReadWinBounds"
                    type="single"
                    syntax="string"
                    any="display"
                    i18nKey="">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailGridHeight"
                    type="single"
                    syntax="numeric"
                    any="display"
                    i18nKey="">
                    <DefaultValues>
                        <Value>0</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailGridColWidths"
                    type="single"
                    syntax="string"
                    any="display"
                    i18nKey="">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailTextColor"
                    type="single_choice"
                    syntax="number"
                    any="display"
                    i18nKey="">
                    <ChoiceValues>
                        <ChoiceValue i18nKey="white">-1</ChoiceValue>
                        <ChoiceValue i18nKey="pink">-20561</ChoiceValue>
                        <ChoiceValue i18nKey="red">-65536</ChoiceValue>
                        <ChoiceValue i18nKey="orange">-14336</ChoiceValue>
                        <ChoiceValue i18nKey="yellow">-256</ChoiceValue>
                        <ChoiceValue i18nKey="green">-16711936</ChoiceValue>
                        <ChoiceValue i18nKey="cyan">-16711681</ChoiceValue>
                        <ChoiceValue i18nKey="blue">-16776961</ChoiceValue>
                        <ChoiceValue i18nKey="magenta">-65281</ChoiceValue>
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                        <ChoiceValue i18nKey="lightGray">-4144960</ChoiceValue>
                        <ChoiceValue i18nKey="darkGray">-12566464</ChoiceValue>
                         <ChoiceValue i18nKey="black">-16777216</ChoiceValue>
                    </ChoiceValues>
                    <DefaultValues>
                        <Value>-16777216</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailBackgroundColor"
                    type="single_choice"
                    syntax="number"
                    any="display"
                    i18nKey="">
                    <ChoiceValues>
                        <ChoiceValue i18nKey="white">-1</ChoiceValue>
                        <ChoiceValue i18nKey="pink">-20561</ChoiceValue>
                        <ChoiceValue i18nKey="red">-65536</ChoiceValue>
                        <ChoiceValue i18nKey="orange">-14336</ChoiceValue>
                        <ChoiceValue i18nKey="yellow">-256</ChoiceValue>
                        <ChoiceValue i18nKey="green">-16711936</ChoiceValue>
                        <ChoiceValue i18nKey="cyan">-16711681</ChoiceValue>
                        <ChoiceValue i18nKey="blue">-16776961</ChoiceValue>
                        <ChoiceValue i18nKey="magenta">-65281</ChoiceValue>
                       <ChoiceValue i18nKey="lightGray">-4144960</ChoiceValue>
                       <ChoiceValue i18nKey="darkGray">-12566464</ChoiceValue>
                        <ChoiceValue i18nKey="black">-16777216</ChoiceValue>
                    </ChoiceValues>
                    <DefaultValues>
                        <Value>-1</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailTextSize"
                    type="single_choice"
                    syntax="numeric"
                    any="display"
                    i18nKey="">
                    <ChoiceValues>
                        <ChoiceValue>8</ChoiceValue>
                        <ChoiceValue>10</ChoiceValue>
                        <ChoiceValue>12</ChoiceValue>
                        <ChoiceValue>14</ChoiceValue>
                        <ChoiceValue>16</ChoiceValue>
                        <ChoiceValue>18</ChoiceValue>
                        <ChoiceValue>20</ChoiceValue>
                        <ChoiceValue>24</ChoiceValue>
                        <ChoiceValue>28</ChoiceValue>
                    </ChoiceValues>
                    <DefaultValues>
                        <Value>12</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailTextStyle"
                    type="single_choice"
                    syntax="numeric"
                    any="display"
```

Code Example B-2 NetMail Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
                    i18nKey="">
                    <ChoiceValues>
                        <ChoiceValue i18nKey="plain">0</ChoiceValue>
                        <ChoiceValue i18nKey="bold">1</ChoiceValue>
                        <ChoiceValue i18nKey="italic">2</ChoiceValue>
                        <ChoiceValue i18nKey="bold_italic">3</ChoiceValue>
                    </ChoiceValues>
                    <DefaultValues>
                        <Value>0</Value>
                    </DefaultValues>
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailHeadersPerPage"
                    type="single"
                    syntax="numeric"
                    any="display"
                    i18nKey="a08">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailNewestFirst"
                    type="single"
                    syntax="boolean"
                    any="display"
                    i18nKey="a12">
                    </AttributeSchema>
                <AttributeSchema name="sunPortalNetmailLogMessages"
                    type="single"
                    syntax="boolean"
                    any="display"
                    i18nKey="a16">
                    </AttributeSchema>
            </User>

        </Schema>
    </Service>
</ServicesConfiguration>
```

Sun ONE Portal Server Rewriter Service Definition

With the default installation, the Service Management Services Document Type Definition is found in the `/opt/SUNWam/dtd/sms.dtd` file. The portal server Rewriter service definition is in the `/opt/SUNWps/export/psRewriter.xml` file.

Code Example B-3 Rewriter Service Definition

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
  Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
  PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.
-->
```

Code Example B-3 Rewriter Service Definition

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE ServicesConfiguration
    PUBLIC "=//Sun ONE//Service Management Services (SMS) 1.0 DTD//EN"
    "jar://com/iplanet/sm/sms.dtd">

<ServicesConfiguration>
        <Service name="SunPortalRewriterService" version="1.0">

                <Schema
serviceHierarchy="/ps.configuration/SunPortalRewriterService"
                          propertiesViewBeanURL="/portal/rwadmin/SelectRule"
                          i18nFileName="psRewriter"
                          i18nKey="sunPortalRewriterServiceDescription">
                        <Global>
                                <SubSchema name="SunPortalRewriterGlobal">
                                        <SubSchema
name="SunPortalRewriterRuleSets" inheritance="multiple">
                                                <AttributeSchema
name="sunPortalRewriterRuleset" syntax="xml" />
                                        </SubSchema>
                                </SubSchema>
                        </Global>
                </Schema>

                <Configuration>
                        <GlobalConfiguration>
                             <SubConfiguration name="SunPortalRewriterGlobal">
                                </SubConfiguration>
                        </GlobalConfiguration>
                </Configuration>

        </Service>
</ServicesConfiguration>
```

Sun ONE Portal Server Search Service Definition

With the default installation, the Service Management Services Document Type Definition is found in the `/opt/SUNWam/dtd/sms.dtd` file. The portal server Rewriter service definition is in the `/opt/SUNWps/export/psSearch.xml` file.

Code Example B-4 Search Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
  Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
  PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.

  Sun ONE Portal Server (iPS) Search Service Definition
-->

<!DOCTYPE ServicesConfiguration
```

Code Example B-4 Search Service Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
    PUBLIC "=//Sun ONE//Service Management Services (SMS) 1.0 DTD//EN"
    "file:/opt/SUNWam/dtd/sms.dtd">

<ServicesConfiguration>
    <Service name="SunPortalSearchService" version="1.0">
        <Schema
            serviceHierarchy="/ps.configuration/SunPortalSearchService"
            propertiesViewBeanURL="/portal/searchadmin/"
            i18nFileName="psSearch"
            i18nKey="sunPortalSearchServiceDescription">
            <Global>
                <AttributeSchema name="serviceObjectClasses"
                    type="list"
                    syntax="string"
                    i18nKey="">
                    <DefaultValues>
                        <Value>sunPortalSearchPerson</Value>
                    </DefaultValues>
                </AttributeSchema>
                <AttributeSchema name="sunPortalSearchInstances"
                    type="list"
                    syntax="string"
                    i18nKey="cs_instances">
                    <DefaultValues>
                    </DefaultValues>
                </AttributeSchema>
            </Global>

        </Schema>

    </Service>
</ServicesConfiguration>
```

Display Profile DTD

In the default installation, the display profile DTD is in the `/opt/SUNWps/dtd/psdp.dtd` file.

Code Example B-5 Display Profile DTD

```
<!ELEMENT DisplayProfile
    (
    Properties,
    Channels,
    Providers
    )
>
<!ATTLIST DisplayProfile
    xmlns:DisplayProfile CDATA #FIXED 'http://www.iplanet.com'
```

Code Example B-5 Display Profile DTD

```
    name CDATA #FIXED "_root"
    version CDATA #REQUIRED
    merge (replace|fuse) "fuse"
    lock (true|false) "false"
    advanced (true|false) "false"
    priority CDATA #REQUIRED
>

<!ELEMENT Channels
    (
    (Container|Channel)*
    )
>
<!ATTLIST Channels
>

<!ELEMENT Providers
    (
    (Provider)*
    )
>
<!ATTLIST Providers
>

<!ELEMENT Provider
    (
    Properties
    )
>
<!ATTLIST Provider
    name CDATA #REQUIRED
    class CDATA #REQUIRED
    merge (replace|remove|fuse) "fuse"
    lock (true|false) "false"
    advanced (true|false) "false"
>

<!ELEMENT Channel
    (
    Properties
    )
>
<!ATTLIST Channel
    name CDATA #REQUIRED
    provider CDATA #REQUIRED
    merge (replace|remove|fuse) "fuse"
    lock (true|false) "false"
    advanced (true|false) "false"
>

<!ELEMENT Container
    (
    Properties,
    Available,
    Selected,
```

Code Example B-5 Display Profile DTD

```
    Channels
    )
>
<!ATTLIST Container
    name CDATA #REQUIRED
    provider CDATA #REQUIRED
    merge (replace|remove|fuse) "fuse"
    lock (true|false) "false"
    advanced (true|false) "false"
>

<!ELEMENT Available
    (Reference*)
>
<!ATTLIST Available
    merge (replace|fuse) "fuse"
    lock (true|false) "false"
    advanced (true|false) "false"
>

<!ELEMENT Selected
    (Reference*)
>
<!ATTLIST Selected
    merge (replace|fuse) "fuse"
    lock (true|false) "false"
    advanced (true|false) "false"
>

<!ELEMENT Properties
    (
    Collection|
    Integer|
    String|
    Boolean|
    Locale
    )*
>
<!ATTLIST Properties
    name CDATA #FIXED "_properties"
    merge (replace|fuse) "fuse"
    lock (true|false) "false"
    propagate (true|false) "true"
    advanced (true|false) "false"
>

<!ELEMENT Locale
    (
    Collection|
    Integer|
    String|
    Boolean
    )*
>
<!ATTLIST Locale
```

Code Example B-5 Display Profile DTD

```
    language CDATA #IMPLIED
    country CDATA #IMPLIED
    variant CDATA #IMPLIED
    merge (replace|remove|fuse) "fuse"
    lock (true|false) "false"
    propagate (true|false) "true"
    advanced (true|false) "false"
>

<!ELEMENT Collection
    (
    Collection|
    Integer|
    String|
    Boolean
    )*
>
<!ATTLIST Collection
    name CDATA #REQUIRED
    merge (replace|remove|fuse) "fuse"
    lock (true|false) "false"
    propagate (true|false) "true"
    advanced (true|false) "false"
>

<!ELEMENT Integer EMPTY>
<!ATTLIST Integer
    name CDATA #IMPLIED
    value CDATA #REQUIRED
    merge (replace|remove) "replace"
    lock (true|false) "false"
    propagate (true|false) "true"
    advanced (true|false) "false"
>

<!ELEMENT String (#PCDATA)>
<!ATTLIST String
    name CDATA #IMPLIED
    value CDATA #IMPLIED
    merge (replace|remove) "replace"
    lock (true|false) "false"
    propagate (true|false) "true"
    advanced (true|false) "false"
>

<!ELEMENT Reference EMPTY>
<!ATTLIST Reference
    value CDATA #REQUIRED
    merge (replace|remove) "replace"
    lock (true|false) "false"
    propagate (true|false) "true"
    advanced (true|false) "false"
>

<!ELEMENT Boolean EMPTY>
```

Code Example B-5 Display Profile DTD

```
<!ATTLIST Boolean
    name CDATA #IMPLIED
    value (true|false) #REQUIRED
    merge (replace|remove) "replace"
    lock (true|false) "false"
    propagate (true|false) "true"
    advanced (true|false) "false"
>

<!ELEMENT ParEntry
    (
           Description?,
    Provider?,
    Channel?
    )
>
<!ATTLIST ParEntry
    xmlns:ParEntry CDATA #FIXED 'http://www.iplanet.com'
    name CDATA #REQUIRED
    version CDATA #REQUIRED
    date CDATA #REQUIRED
    author CDATA #REQUIRED
    requiredClass CDATA #REQUIRED
>

<!ELEMENT Description (#PCDATA) >
```

Rewriter Ruleset DTD

In the default installation, the Rewriter Ruleset DTD is in the `/opt/SUNWps/web-src/WEB-INF/lib/rewriter.jar` jar file under `resources/RuleSet.dtd`.

Code Example B-6 Rewriter Ruleset DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ENTITY % gtype 'GROUPED'>
<!ENTITY % stype 'SCATTERED'>
<!ENTITY % jURL 'URL'>
<!ENTITY % jEXPRESSION 'EXPRESSION'>
<!ENTITY % jDHTML 'DHTML'>
<!ENTITY % jDJS 'DJS'>
<!ENTITY % jSYSTEM 'SYSTEM'>
<!ENTITY % ruleSetElements '(HTMLRules | JSRules | XMLRules)? '>
<!ENTITY % htmlElements '(Form | Applet | Attribute | JSToken)*'>
<!ENTITY % jsElements '(Variable | Function)*'>
<!ENTITY % xmlElements '(Attribute | TagText)*'>
```

Code Example B-6 Rewriter Ruleset DTD

```
<!ELEMENT RuleSet (%ruleSetElements;,%ruleSetElements;,%ruleSetElements;)>
<!ATTLIST RuleSet
    type (%gtype; | %stype;) "GROUPED"
    id ID #REQUIRED
>

<!ELEMENT HTMLRules (%htmlElements;)>
<!ATTLIST HTMLRules
    type (%gtype; | %stype;) "GROUPED"
    id CDATA "html_rules"
>
<!ELEMENT Form EMPTY>
<!ATTLIST Form
    source CDATA #REQUIRED
    name CDATA #REQUIRED
    field CDATA #REQUIRED
    valuePatterns CDATA ""
>
<!ELEMENT JSToken (#PCDATA)>
<!ELEMENT Applet EMPTY>
<!ATTLIST Applet
    source CDATA #REQUIRED
    code CDATA #REQUIRED
    param CDATA "*"
    valuePatterns CDATA ""
>
<!ELEMENT JSRules (%jsElements;)>
<!ATTLIST JSRules
    type (%gtype; | %stype;) "GROUPED"
    id CDATA "js_rules"
>
<!ELEMENT Variable (#PCDATA)>
<!ATTLIST Variable
    type (%jURL; | %jEXPRESSION; | %jDHTML; | %jDJS; | %jSYSTEM;) "URL"
>
<!ELEMENT Function EMPTY>
<!ATTLIST Function
    type (%jURL; | %jEXPRESSION; | %jDHTML; | %jDJS;) "URL"
    name CDATA #REQUIRED
    paramPatterns CDATA #REQUIRED
>
<!ELEMENT XMLRules (%xmlElements;)>
<!ATTLIST XMLRules
    type (%gtype; | %stype;) "GROUPED"
    id CDATA "xml_rules"
>
<!ELEMENT TagText EMPTY>
<!ATTLIST TagText
    tag CDATA #REQUIRED
    attributePatterns CDATA ""
>
<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute
    name CDATA #REQUIRED
```

Code Example B-6 Rewriter Ruleset DTD

```
    tag CDATA "*"
    valuePatterns CDATA ""
>
```

Default Ruleset

In the default installation, the default ruleset is in the `/opt/SUNWps/web-src/WEB-INF/lib/rewriter.jar` jar file under `resources/DefaultRuleSet.xml`. This file is also in the `/opt/SUNWps/export` directory.

Code Example B-7 Default Ruleset

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE RuleSet SYSTEM "jar://rewriter.jar/resources/RuleSet.dtd">

<RuleSet id="default_ruleset">

<!-- Rules for Rewriting HTML Source -->
<HTMLRules>

<!-- Rules for Rewriting Form Input/Option Values List -->


<!-- Rules for Rewriting Applet/Object Parameter Values List -->


<!-- Rules for Rewriting HTML Attributes -->
    <Attribute name="action" />
    <Attribute name="background" />
    <Attribute name="codebase" />
    <Attribute name="code" />
    <Attribute name="href" />
    <Attribute name="src" />
    <Attribute name="value" />
    <Attribute name="imagePath" />
    <Attribute name="lowsrc" />
    <Attribute name="archive"
valuePatterns="***;**,**,**,**,**,**,**,**,**,**,**,**"/>
    <Attribute name="style" />
    <Attribute name="content" tag="meta" />

<!-- Rules for Rewriting HTML Attributes containing Java Script -->
    <JSToken>onAbort</JSToken>
    <JSToken>onBlur</JSToken>
    <JSToken>onChange</JSToken>
    <JSToken>onClick</JSToken>
```

Code Example B-7 Default Ruleset

```
    <JSToken>onDblClick</JSToken>
    <JSToken>onError</JSToken>
    <JSToken>onFocus</JSToken>
    <JSToken>onKeyDown</JSToken>
    <JSToken>onKeyPress</JSToken>
    <JSToken>onKeyUp</JSToken>
    <JSToken>onLoad</JSToken>
    <JSToken>onMouseDown</JSToken>
    <JSToken>onMouseMove</JSToken>
    <JSToken>onMouseOut</JSToken>
    <JSToken>onMouseOver</JSToken>
    <JSToken>onMouseUp</JSToken>
    <JSToken>onReset</JSToken>
    <JSToken>onSelect</JSToken>
    <JSToken>onSubmit</JSToken>
    <JSToken>onUnload</JSToken>
</HTMLRules>

<!-- Rules for Rewriting JavaScript Source -->
<JSRules>

<!-- Rules for Rewriting JavaScript variables in URLs -->
    <Variable type="URL"> imgsrc </Variable>
    <Variable type="URL"> location.href </Variable>
    <Variable type="URL"> _fr.location </Variable>
    <Variable type="URL"> mf.location </Variable>
    <Variable type="URL"> parent.location </Variable>
    <Variable type="URL"> self.location </Variable>
    <Variable type="EXPRESSION"> location </Variable>
    <Variable type="SYSTEM"> window.location.pathname </Variable>

<!-- Rules for Rewriting JavaScript Function Parameters -->
    <Function type="URL" name="openURL" paramPatterns="y"/>
    <Function type="URL" name="openAppURL" paramPatterns="y"/>
    <Function type="URL" name="openNewWindow" paramPatterns="y"/>
    <Function type="URL" name="parent.openNewWindow" paramPatterns="y"/>
    <Function type="URL" name="window.open" paramPatterns="y"/>
    <Function type="DHTML" name="document.write" paramPatterns="y"/>
    <Function type="DHTML" name="document.writeln" paramPatterns="y"/>

</JSRules>

<!-- Rules for Rewriting XML Source -->
<XMLRules>

<!-- Rules for Rewriting Attributes -->
    <Attribute name="xmlns"/>
    <Attribute name="href" tag="a"/>

<!-- Rules for Rewriting TagText -->
    <TagText tag="baseroot" />
    <TagText tag="img" />
</XMLRules>
```

Code Example B-7 Default Ruleset

```
</RuleSet>
```

Appendix C

Portal Desktop Attributes

The Desktop Service consists of global and dynamic attributes. The values applied to the global attributes are applied across the Sun™ ONE Identity Server configuration and are inherited by every configured organization. They cannot be applied directly to roles or organizations as the goal of global attributes is to customize the Sun ONE Identity Server application. Values applied to the dynamic attributes are assigned to a role or organization. When the role is assigned to a user or a user is created in an organization, the dynamic attribute then becomes a characteristic of the user.

The Desktop attributes are divided into:

- Desktop Global Attributes
- Desktop Dynamic Attributes

Desktop Global Attributes

Table C-1 describes the global attributes for the Desktop Service.

The table contains three columns: the first column identifies the attribute, the second column provides the default value for the attribute, and the third column describes the attribute.

Table C-1 Desktop Service - Global Attributes

Attribute	Default Value	Description
Enable XML Parsing Validation (Use with Caution)	True (checked)	Specifies whether to enforce validation while parsing the display profile XML document. Unchecking this attribute can improve system performance. However, this can potentially introduce corruption in the display profile document because the resulting XML document might include some fragments that do not conform to the DTD.
Namespace URI	`http://www.iplanet.com`	Specifies the unique identifier for the XML namespaces or Uniform Resource Indicator (URI) in the form of a URL. This guarantees that XML tags will be unique.
Enable Federation	False (unchecked)	Enables Identity Federation so that a user can associate, connect or bind multiple internet service providers' local identities, enabling them to have one network identity.
Hosted Provider ID	None	Specifies the unique identifier of the host that provides the network identity of a user.
Client Session Reap Interval (seconds)	1800	Defines in seconds the time interval between checks for removing inactive client sessions.
Client Session Maximum Inactive Session Time Before Reap (seconds)	3600	Specifies the maximum number of seconds a client session can be idle before it is considered inactive. If a session is idle for more than this value, it is made a candidate for session reaping and can be removed the next time the client session times out.

Table C-1 Desktop Service - Global Attributes

Attribute	Default Value	Description
Display Profile	The default value depends on the type of installation performed. If the sample portal is installed, the Display Profile contains the definitions for the built-in providers (the basic providers of Sun™ ONE Portal Server), such as bookmark and notes. If the sample portal was not installed, the global Display Profile is blank.	Displays several controls for manipulating the global display profile, an XML document that defines the container management, channel attributes, and provider definitions for the organization. The controls include: • Disable Authnetciation-less Access for Federated Users—Prevents a user with a federated network identity to access the portal without entering a user name and password.. • Upload XML—Allows you to upload an XML file containing display profile information to the Portal Server. • Download XML—·Allows you to download the display profile to your local drive. • Channel and Container Management—Provides a graphical user interface to manage container channels and channels without the need to edit the XML file. These links are not attributes. Selecting these links allows you to manipulate the display profile. Display profile elements defined in the global display profile are inherited by all users on the system, regardless of the organization or role to which they belong.

Table C-1 Desktop Service - Global Attributes

Attribute	Default Value	Description
Authentication-less Portal Desktop Configuration	Enable	Displays several controls for configuring authentication-less configuration of the portal desktop. . The controls are: • Disable Authentication-less Access for Federated Users—Allows you to pevent users with a network identity on a hosted provided to access the portal desktop with providing a user name and password.. • DefaultAuthentication-less User ID—Defines the User IDs that are authorized to access the Desktop without authenticating. • Authorized Authentication-less User ID—Defines the User IDs that are authorized to access the Desktop without authenticating.

Desktop Dynamic Attributes

Table C-2 describes the dynamic attributes for the Desktop Service.

The table contains three columns: the first column identifies the attribute, the second column provides the default value for the attribute, and the third column describes the attribute.

Table C-2 Desktop Service - Dynamic Attributes

Attribute	Default Value	Description
Conflict Resolution Level	Highest	Sets the conflict resolution level for the Desktop service template used to resolve conflicts when multiple Desktop templates are merged. There are seven conflict resolution settings available ranging from Highest to Lowest. Do not confuse this setting with the display profile document priority. The display profile document priority is a numeric value that is set in the XML file with the `priority=` *syntax* tag. When a merge occurs, it starts with the lowest display profile priority document (lowest number) and proceeds in increasing priority number, until it arrives at the user (base), the highest priority display profile. When an attribute conflict occurs, the attribute on the template set with the highest conflict resolution level is returned.
Default Channel Name	`JSPTabContainer`	Identifies which default channel is rendered when the Desktop is called with an unspecified provider.

Table C-2 Desktop Service - Dynamic Attributes

Attribute	Default Value	Description
Default Edit Channel Name	JSPEditChannel	Specifies which default edit channel to use to wrap the content when one is not specified in the URL. When a channel is edited, an “Edit” request URL is sent to the Desktop Servlet. The URL generated for the "Edit” of each of the channels inside a container depends on the property "editContainerName” defined in the display profile. If you have migrated containers from iPlanet™ Portal Server 3.0, you must specify the default edit channel with which to wrap the content using this attribute because the URL format has changed.

Table C-2 Desktop Service - Dynamic Attributes

Attribute	Default Value	Description
Desktop Type	`default`	Retrieves template files for the specified Desktop type when different Desktop configurations are needed and when different sets of templates and JSPs are required for those configurations. IThe Desktop type attribute of the Desktop service is a comma-separated string. type, that the Portal Desktop uses as an ordered list. The list is used by the Desktop lookup operation when searching for templates and JSPs. The lookup starts at the first element in the list and each element represents a sub directory under the Desktop template base directory. If a template is not found in the first directory, then it proceeds to the next one in the list. This continues until the item is found (or not), for all Desktop type elements in the list. If the default directory is not included in the list, it will be added at the end of the list implicitly. For example, if the Desktop type is sampleportal, the target template will be searched in the sampleportal sub directory, then the default sub directory. By default, if the sample portal is installed, then the Desktop type attribute, `sunPortalDesktopType`, is set to `sampleportal`. If the sample portal is not installed, then the Desktop type attribute value is set to `default`. Most sites will not use the `default` Desktop type, as they will have different channels, different logo, different look and feel, and the like.

Table C-2 Desktop Service - Dynamic Attributes

Attribute	Default Value	Description
Display Profile	The default value depends on the type of installation performed. If the sample portal was installed, a sample display profile document is installed at the organization level that contains channels that display the built-in providers defined in the global display profile.	Displays several links for manipulating the display profile, an XML document that defines the container management, channel attributes, and provider definitions for this specific node (role, organization, suborganization). Links are: • Edit XML—Allows you to edit the entire display profile XML file. • Upload XML—Allows you to upload an XML file containing display profile information to the Portal Server. • Download XML—Allows you to download the display profile to your local drive. These links are not attributes. Selecting these links allows you to manipulate the display profile.
Show Desktop Service Attributes	True (checked)	Specifies whether the Desktop Service attributes are displayed to the users associated with the role. This dynamic attribute is mainly used for role-based delegated administration, Values applied to this attribute are only in effect for a role. When the role is assigned to a user and the value of this attribute is false, users (usually delegated administrators) cannot see any Desktop Service attributes except the Channel and Container Management link when they navigate into all the roles within the organization.

Appendix D

NetMail Attributes

The NetMail Service consists solely of dynamic attributes. Values applied to the dynamic attributes are assigned to a role or organization. When the role is assigned to a user or a user is created in an organization, the dynamic attribute then becomes a characteristic of the user.

NetMail Dynamic Attributes

Table D-1 describes the dynamic attributes for the NetMail Service.

The table contains three columns: the first column identifies the attribute, the second column provides the default value for the attribute, and the third column describes the attribute.

Table D-1 NetMail Service - Dynamic Attributes

Attribute	Default Value	Description
Incoming mail (IMAP) server	(Set by the administrator)	Specifies the host name of the IMAP server to which NetMail should connect.
Outgoing mail (SMTP) server	(Set by the administrator)	Specifies the server that NetMail uses to send outgoing messages through SMTP.
Default mail domain	(Set by the administrator)	Specifies the name of the default mail domain.
IMAP top-level folder	Mail	Specifies the folder in which user mail folders reside on the IMAP mail server.

Table D-1 NetMail Service - Dynamic Attributes *(Continued)*

Attribute	Default Value	Description
Cache folder list	False (unchecked)	Specifies whether to load list of folders into user memory caches automatically when they disconnect from the mail server. By caching the folder list, users can move and copy messages when they use NetMail in disconnected mode. False (unchecked) activates loading. True (checked) turns loading off.

Table D-1 NetMail Service - Dynamic Attributes *(Continued)*

Attribute	Default Value	Description
LDAP server details to use in address book search	(Set by the administrator)	Specifies what LDAP server information to use when performing an address book search. The information is used only in the NetMail applet. Each entry is a comma separated list of name / value pairs in the following format: *name*="*value*." Quotation marks are not allowed within any value. Users cannot modify this value. The valid names and corresponding preferences are: • `name`—The name of the LDAP server that is visible to users. • `server`—The LDAP server and port number, in the form *ldapserver*[: *ldapportnumber*]. • `base`—The string used as base to search for the users and groups, for example: `c=US,o=sesta`. • `searchin`—A comma separated list, for example: `cn,gn,sn`. • `result`—LDAP attribute name, for example, `mail`. • `filter`—LDAP search filters; see RFC2254. • `referral`— LDAP referals, follow or ignore (does not apply). Example: `name="Sesta LDAP",server="ldap-server.sesta.com",base="dc=sesta, dc=com"`
Initial Headers downloaded from the IMAP server	`10`	Controls the number of initial headers that are downloaded from the IMAP server when a user opens a folder. Applies only to NetMail Java.

Table D-1 NetMail Service - Dynamic Attributes *(Continued)*

Attribute	Default Value	Description
Message headers displayed per page	`10`	Specifies the number of initial headers that are downloaded from the IMAP server when users open folders. Applies only to NetMail Lite.
Check new mail every (mins)	`5`	Determines the frequency, in minutes, that NetMail checks for new messages in the currently selected folder. If set to 0, NetMail does not check for new messages.
Do not load attachments larger than (KB)	`0`	Specifies the maximum size of attachments, in kilobytes, that NetMail loads automatically into the memory cache when users disconnect. Specify 0 to load all attachments. Applies only to NetMail Java.
Types of messages to load on disconnect	`All`	Specifies the type of messages that are automatically loaded into the memory cache when users disconnect. Valid values are: All, None, New, Unread, New and Unread, and Found. For example, if Unread is selected, all unread messages in the inbox are loaded to memory cache Users can then read these messages in disconnected mode. Applies only to NetMail Java.
Sort by newest messages first	True (checked)	Identifies what messages are to appear first in the selected folder. True (checked) specifies that newest messages are to appear first. False (unchecked) specifies oldest messages first. Applies only to NetMail Lite.

Table D-1 NetMail Service - Dynamic Attributes *(Continued)*

Attribute	Default Value	Description
Open messages in separate windows	False (unchecked)	Specifies whether to open a new read window for each new message that users view. True (checked) specifies that a new read window should be opened for each message. False (unchecked) specifies that new messages be viewed in the current window. Applies only to NetMail Java.
Purge deleted messages on exit	False (unchecked)	Specifies whether to remove messages from the inbox that are flagged as deleted when users exit or disconnect. True (checked) specifies that flagged messages be deleted upon disconnect. False (unchecked) specifies that messages not be deleted.
Sent messages folder (on server)	`Mail/Sent`	Specifies the folder in which outgoing messages are logged.
Save sent messages in the Sent folder (on server)	False (unchecked)	Specifies to save user messages in the `Sent` folder stored on the IMAP server. True (checked) specifies to save messages in the Sent folder on the server. False (unchecked) specifies not to save messages in the Sent folder.
Keep copy of sent messages in local cache	True (checked)	Specifies whether to save the cache to disk at exit automatically. If not enabled, users are prompted to Save the cache to disk before exiting. True (checked) specifies to save the cache to disk automatically. False (unchecked) specifies to prompt the user whether to cache. Applies to NetMail locally installed applet only.
Quote prefix for replies	>	Specifies which character string precedes each text line of a reply message.

Table D-1 NetMail Service - Dynamic Attributes *(Continued)*

Attribute	Default Value	Description
Append signature to outgoing mail	False (unchecked)	Specifies whether to append user signatures to outgoing messages. True (checked) specifies to append the signature. False (unchecked) specifies not to append the signature.
Include the author in reply	False (unchecked)	Specifies whether to include the author of the original message with a reply message. True (checked) specifies to include the author of the original message. False (unchecked) specifies not to include the author of the original message. Applies only to NetMail Java.
Include the Date of the original message in reply	False (unchecked)	Specifies whether to include the date of the original message with a reply message. True (checked) specifies to include the date of the original message. False (unchecked) specifies not to include the date of the original message. Applies only to NetMail Java.
Include the Body of the original message in reply	True (checked)	Specifies whether to include the body of the original message with a reply message. True (checked) specifies to include the body of the original message. False (unchecked) specifies not to include the body of the original message. Applies only to NetMail Java.

Table D-1 NetMail Service - Dynamic Attributes *(Continued)*

Attribute	Default Value	Description
Preferences the user cannot change	(Set by the administrator)	Specifies the NetMail preferences attributes that the end user cannot change. The valid values are: `IMAPPassword`, `IMAPServerName`, `IMAPUserName`, `SMTPMailServer`, `autoFolderLoad`, `autoload`, `autopurge`, `autosave`, `backgroundColor`, `inactivityInterval`, `indentPrefix`, `initialHeaders`, `saveSentMessage`, `maxAttachLen`, `multipleReadWindows`, `replyFields`, `replyToAddress`, `resetSize`, `rootFolder`, `sentMessagesFolder`, `signature`, `textColor`, `textSize`, and `textStyle`.

Appendix E

Rewriter Attributes

The Rewriter provides a Java™ class library for rewriting URL references in various web languages such as HTML, JavaScript, and WML, and in HTTP Location headers (redirections). The Rewriter Service does not consist of any attributes.

To implement the service, you create Rewriter rules that define how rewriting is to be done and the data to be rewritten. You can create and edit Rewriter rules through the administration console. For information on creating Rewriter rules, refer to Chapter 7, “Administering the Rewriter Service”.

Appendix F

Search Attributes

This appendix describes attributes that you can configure for the search engine through the Sun ONE Identity Server administration console.

When you select Search Properties from the Service Management View, a two-toned tabbed menu bar is displayed. This appendix is organized according to the topics or tabs on the upper portion of the menu bar.

- Server
- Robot
- Database
- Categories
- Reports

When one of these tabs is selected, the menu bar below lists the related subtopics for the topic. The default Search page selects Server / Settings. Each subtopic uses one or more tables to explain the attributes for that subtopic. The tables are divided into three columns: Attribute, Default Value and Description. The Attribute gives the descriptive text found on the page; the Default Value provides the default value for the Attribute; and the Description explains the Attribute and its format.

Every Search Properties page gives you the Select Server attribute as described in the Table F-1.

Table F-1 Search Select Server Attribute

Attribute	Default Value	Description
Select Server	`http://`*servername*`:80/portal`	Fully qualified server name of your Search server.

Server

The Server section is where you configure the preferences for your server. You select what directory to use for temporary files, what information to log and how much detail should be in the logs. The Server attributes are displayed on two pages:

- Settings
- Robot

Settings

This page contains the basic settings for the administration and operation of the search server.

Table F-2 Server Settings Attributes

Attribute	Default Value	Description
Server Root	`/var/opt/SUNWps/https-`*servernamefull*`/portal`	Houses the configuration, log, database, and robot information files. Also it is the root directory for all of the search files that are generated and updated when conducting a search. This is not configurable.
Temporary Files	`/var/opt/SUNWps/https-`*servernamefull*`/portal/tmp`	Contains all temporary files used to manage a search during the search. It includes newly generated resource descriptions that have not yet been added to the main database. These are removed when the search is completed.

Table F-2 Server Settings Attributes *(Continued)*

Attribute	Default Value	Description
Document level security	`Off`	Controls who can access documents. When this setting is changed, the server must be restarted. `Values:` • `Off` (default) means all users have access to the RDs. • `On` means that the `ReadACL` field in an RD is checked to see if the user asking for the RD has permission because the user is in an acceptable organization or role, or is an acceptable individual user. The `ReadACL` field is set in the Database, Resource Descriptor page.

Robot

This page contains the advanced settings for the administration and operation of the search server. Here is where you configure the log files for user queries, index maintenance, resource description management, and debugging.

Table F-3 Server Advanced Settings Attributes

Attribute	Default Value	Description
Search (rdm)	`/var/opt/SUNWps/https-`*servername*`/portal/logs/rdm.log`	Logs the queries end users make of the database. You can check the Disable Search Log checkbox to suppress this logging. If you do, you cannot view the User Queries (rdm) report.
Disable Search Log	False (unchecked) - enabled	Controls use of query log. In the report section, you can generate a report the lists the most popular queries based on this log. Values: • Checked—disabled • Unchecked—enabled. Every user query is entered in this log.

Table F-3 Server Advanced Settings Attributes

Attribute	Default Value	Description
Index Maintenance	`/var/opt/SUNWps/https-`*servername*`/portal/logs/searchengine.log`	Logs the transactions involving the search engine, except for not registration of resource descriptions.
RD Manager	`/var/opt/SUNWps/https-`*servername*`/portal/logs/rdmgr.log`	Logs the registration of resource descriptions from the robot or import agents into the database. You can view this log as a RD Manager (rdmgr) report.
RDM Server	`/var/opt/SUNWps/https-`*servername*`/portal/logs/rdmserver.log`	Logs debugging information on RDM transactions. The level of detail is controlled by the Log Level. You can view this log as a RDM Server (rdmsvr) report.
Log Level	`1`	Controls the amount of detail the RDM Server log file contains. The possible levels are 2, 10, 20, 50, 100, and 999. A setting of 1 (default) logs only severe errors. The higher the number, the more detail the RDM Server log file contains.

Robot

The properties for the robot are quite complex. You can select the sites to be searched or crawled, check to see if a site is valid, define what types of documents should be picked up, and schedule when the searches take place.

This section is organized as follows:

- Overview
- Sites
- Filters
- Crawling
- Indexing
- Simulator
- Site Probe
- Schedule

Overview

The Robot Overview panel is where you can see what the robot is doing: if it is Off, Idle, Running, or Paused; and if it is Running, what progress it is making in the search since the panel is refreshed about every 30 seconds. The refresh rate is defined using the `robot-refresh` parameter in the `search.conf` file.

The two buttons on the top right are appropriate for its state. If the robot is Off, the buttons are Start and Remove Status. If it is Running or Idle, the two buttons are Stop and Pause. If it is Paused, the two buttons are Stop and Resume. By selecting on any of the Attributes, you go to the Reports section where you can get a detailed up-to-the-minute report of that Attribute.

Table F-4 Robot Overview Attributes

Attribute	Default Value	Description
The Robot is	Current activity	The Robot's state. Value can be `Idle`, `Running`, `Paused`, or `Off`
Updated at date	Date and time last refreshed.	This page is refreshed to keep you aware of what progress the robot is making.
Starting Points	Number defined	Number of sites you have selected to be searched. A site is disabled (not included in a search) on the Robot, Site page.
URL Pool	Number URLs waiting	Number of URLs yet to be investigated. When you begin a search, the starting point URLs are entered into the URL pool. As the search progresses, the robot discovers links to other URLs. These URLs get added to the pool. After all the URLs in the pool have been processed, the URL pool is empty and the robot is idle.
Extracting	Number connections per second	Number of resources looked at in a second. Extracting is the process of discovering or locating resources, documents or hyperlinks to be included in the database and filtering out unwanted items.
Filtering	Number URLs rejected	Total number of URLs that are excluded.

Table F-4 Robot Overview Attributes *(Continued)*

Attribute	Default Value	Description
Indexing	Number URLs per second	Number of resources or documents turned into a resource description in a second. Indexing is the phase when all the information that has been gathered on a document is turned into a resource description for inclusion in the search database.
Excluded URLs	Number URLs excluded by filters	Number of URLs that did not meet the filtering criteria.
	Number URLs excluded by errors	Number of URLS where the robot encountered errors as file not found.
Resource Descriptions	Number RDs contributed	Number of resource descriptions added to the database.
	Number Bytes of RDs contributed	Number of bytes added to the database.
General Stats	Number URLs retrieved	Number of URLs retrieved during run.
	Number Bytes average size of RDs	Average number of bytes per resource description.
	Time in days, hours, minutes, and seconds running	The amount of time the robot has been running.

Sites

The initial page in this section shows what sites are available to be searched.

A site can be enabled (On) and disabled (Off) by using the radio buttons. A disabled site is not searched when the robot is run. The Edit link displays a page where you can change how a search site is defined.

To delete a site, check the checkbox and select Delete.

To add a new site, select New. Add a URL or Domain in the text box and select a depth for the search. Select Create to use the default values. Otherwise, select Create and Edit to select non-default values and go to the Edit page to define the search site.

Table F-5 Robot Manage Sites Attributes

Attribute	Default Value	Description
Lock or cluster graphic	Status of site	Lock open means that the URL is accessible. The closed lock means that the site is a secure web server and uses SSL. The cluster means that the site is a domain.
On / Off	`On`	Choose to search this site or not when the robot is run.

The New Site page allows you to set up an entire site for indexing.

Table F-6 Robot New Site Attributes

Attribute	Default Value	Description
New site	URL	URL - format: `http://www.sesta.com` Domain - format: `*.sesta.com`
Depth	`10`	You have a choice of 1 for this URL only, 2 for this URL and first links, 100 for the robot, , 3 - 10 or unlimited. The default value is set on the Robot, Crawling page.

The edit page is where you can define the search site more completely. You can specify what type of server it is, redefine the depth of the search, and select what type of files to add to the search database. The attributes for URL and Domain sites are mostly the same. The additional column in this table shows which attributes are shared and which are unique.

A number of actions are performed on this page. You can verify the server name for the search site you entered. You can add more servers to the server group by selecting Add in the Server Group section. You can add more starting points by selecting Add in the Starting Points section. In the Filter Definition section, you can add or delete, exclude or include certain types of files as well as change the order the filters for these files are applied.

Table F-7 Robot Sites Edit Attributes

Attribute	URL/ Domain	Default Value	Description
Site Nickname	URL / D	Site entered - `www.sesta.com`	Name that is displayed on the initial page. The default is the URL or domain you entered. You can change this name here.
Checkbox to select site for deletion or verification	URL / D	Unchecked	Unchecked—not selected Checked—selected
Server Group - Name	URL	URL - `www.sesta.com`	Is either a single server or a part of a single server. The entry must include the full host name. If you specify just a host name, the site is limited to that host. If you provide directory information in addition to the host name, the site is defined as only that directory and any of its subdirectories.
Domain Suffix	D	Domain entered - `*.sesta.com`	Includes all servers within a domain, such as `*.sesta.com`.
Port	URL / D	`80` for URL; blank for Domain	If the site you are searching uses a different port, enter it here.
Type	URL	`Web Server`	Web Server, File Server, FTP Server, Secure Web Server
Allowed Protocols	D	Checkboxes all checked	Checkboxes for http, file, ftp, https
Starting Points- Checkbox to select site for deletion	URL / D	Unchecked	Unchecked—not selected Checked—selected
Starting Points- URL	URL / D	`http:// URL:80`	URL or domain
Starting Points - Depth	URL / D	`10`	1 - this URL only 2 - this URL and first links 3-10 unlimited
Filter Definition - Checkbox to select file type for deletion	URL / D	Unchecked	Unchecked - not selected Checked - selected

Table F-7 Robot Sites Edit Attributes *(Continued)*

Attribute	URL/ Domain	Default Value	Description
Filter Definitions	URL / D	In this order, the defaults are Archive Files; Audio Files; Backup Files; Binary Files; CGI Files; Image Files; Java, Javascript, Style Sheet Files; Log Files; Revision Control Files; Source Code Files; Temporary Files; Video Files.	The possible choices are Archive Files; Audio Files; Backup Files; Binary Files; CGI Files; Image Files; Java, Javascript, Style Sheet Files; Log Files; Power Point Files; Revision Control Files; Source Code Files; Temporary Files; Video Files; Spreadsheet Files; Plug-in Files; Lotus Domino Documents; Lotus Domino OpenViews; System Directories (UNIX); System Directories (NT).
Comment	URL / D	Blank	Text field that describes the site to you. It is not used by the robot.
DNS Translation	URL	Blank	The DNS translation modifies the URL and the way it is crawled by replacing a domain name or alias with a cname. Format: alias1->cname1,alias2->cname1

Filters

The initial page in this section shows all the defined filter rules and the site definitions that use them. Each filter name is proceeded by a checkbox to select that document type and two radio buttons to turn the Filter Rule On and Off. If a checkbox is checked, the filter is selected and can be deleted. You can add a new filter by selecting New. The new filter page is an abbreviated Edit page, requiring only a Nick Name and one rule. Another option is selecting the Edit link, which takes you to a page where you define the rules for that file type or what that filter does. Each rule is made up of a drop down list of Filter Sources, a drop down list to Filter By, and a text box to enter the filter string specifics in.

Table F-8 Robot Filter Edit Attributes

Attribute	Default Value	Description
Filter Name	Prompts for new name. File name of the file type you choose to edit.	A descriptive name that reflects the type of file the filter applies to.

Table F-8 Robot Filter Edit Attributes

Attribute	Default Value	Description
Drop down list of Filter Sources	URL for new filter. Displays previously chosen information for that particular file type.	URL, protocol, host, path, MIME type
Drop down list of positions	`is` for new filter. Displays previously chosen information for that particular file type. For example, Binary Files ends with `exe`.	is, contains, begins with, ends with, regular expression
Text box for type (directory, protocol, file extensions) specifics	Blank for new filter. Displays previously entered information for that particular file type. For example, Temporary Files contains `/tmp/`.	In this text box, list what you want to match. What would match in this example - http:/ / docs.sesta.com / manual.html protocol is http; host contains sesta; file ends with `html`.
Description	Prompts for new description. Displays previously entered description for that particular file type.	Describe the filter rule for yourself. The robot does not use it.
New Site	True (checked) for new filter. Displays previously chosen value for that particular file type.	Use this as one of the default filters when creating new sites. If you do not check this, you can still add this filter to a new site by editing the site on the Robot, Sites page.
By Default	Nothing selected for a new filter. Default selected previously for a defined file type.	Exclude documents matching this filter. Include documents matching this filter. Selection for a new filter does not affect existing site definitions. To use your new filter on an existing site, you must add it by editing the site on the Robot, Sites page.
Deployment	Lists the sites that use this filter.	

Crawling

The settings on this page control the robot's operational parameters and defaults. It is divided into these sections: Speed, Completion Actions, Logfile Settings, Standards Compliance, Authentication Parameters, Proxying, Advanced Settings and Link Extraction.

Table F-9 Robot Crawling Attributes

Attribute	Default Value	Description
Server Delay	`No Delay`	No Delay (default), 1 second, 2 seconds, 5 seconds, 10 seconds, 30 seconds, 1 minute, 5 minutes.
Maximum Connections - Max concurrent retrieval URLs	`8`	1, 2, 4, 8 (default), 10, 12, 16, 20.
Maximum Connections per Site	`2`	(no limit), 1, 2, 4, 8, 10, 12, 16, 20.
Send RDs to Indexing every	`30 minutes`	3 minutes, 5 minutes, 10 minutes, 15 minutes, 30 minutes (default), 1 hour, 2 hours, 4 hours, 8 hours.
Script to Launch	`nothing (default)`	nothing (default). For sample files, see the `cmdHook` files in the `/opt/SUNWps/samples/robot` directory (for the default installation).
After Processing all URLs	`go idle (default)`	go idle (default), shut down, start over.
Contact Email	`user@domain`	Enter your own.
Log Level	`1 - Generation`	0 Errors only; 1 Generation (default); 2 Enumeration, Conversion; 3 Filtering; 4 Spawning; 5 Retrieval
User Agent	`SunONERobot/6.0`	Version of the search server.
Ignore robots.txt protocol	False (unchecked)	Some servers have a `robot.txt` file that says robots do not come here. If your search robot encounters this file on a site and this attribute is false, it does not search the site. If this attribute is true, the robot ignores the file and searches the site.
Perform Authentication	`Yes`	Yes
		No
Robot Username	`anonymous`	Robot uses the anonymous user name to gain access to a site.
Password	user@domain	Frequently a site that allows anonymous users requires a email address as a password. This address is in plain text.

Table F-9 Robot Crawling Attributes *(Continued)*

Attribute	Default Value	Description
Proxy Username	anonymous	Robot uses the anonymous user name to gain access to a site.
Password	user@domain	Frequently a site that allows anonymous users requires an email address as a password. This address is in plain text.
Proxy Connection Type	Direct Internet Connection	Direct Internet Connection, Proxy—Auto Configuration, Proxy—Manual Configuration
Auto Proxy Configuration Type	`Local Proxy File`	Local Proxy File, Remote Proxy File
Auto Proxy Configuration Location	Blank	The auto proxy has a file that lists all the proxy information needed. An example of a local proxy file is robot.pac. An example of a emote proxy file is `http://proxy.sesta.com:8080/proxy.pac`
Manual Configuration HTTP Proxy	Blank	Format: `server1.sesta.com:8080` These three manual configuration values are put in the `robot.pac` file in the `/var/opt/SUNWps/https-`*servername*`/portal/config` directory.
Manual Configuration HTTPS Proxy	Blank	This manual configuration value is put in the `robot.pac` file. Format: `server1.sesta.com:8080`
Manual Configuration FTP Proxy	Blank	This manual configuration value is put in the `robot.pac` file. Format: `server1.sesta.com:8080`
Follow links in HTML	True (checked)	Extract hyperlinks from HTML
maximum links	`1024`	Limits the number of links the robot can extract from any one HTML resource. As the robot searches sites and discovers links to other resources, it could conceivably end up following huge numbers of links a great distance from its original starting point.
Follow links in plain text	False (unchecked)	Extract hyperlinks from plain text.

Table F-9 Robot Crawling Attributes *(Continued)*

Attribute	Default Value	Description
maximum links	`1024`	Limits the number of links the robot can extract from any one text resource.
Use Cookies	False (unchecked)	If checked, the robot uses cookies when it crawls. Some sites require the use of cookies in order for them to be navigated correctly. The robot keeps its cookies in a file called `cookies.txt` in the robot state directory. The format of `cookies.txt` is the same format as used by the Netscape™ Communicator browser.
Use IP as Source	True (checked)	In most cases, the robot operates only on the domain name of a resource. In some cases, you might want to be able to filter or classify resources based on subnets by Internet Protocol (IP) address. In that case, you must explicitly allow the robot to retrieve the IP address in addition to the domain name. Retrieving IP addresses requires an extra DNS lookup, which can slow the operation of the robot. If you do not need this option, you can turn it off to improve performance.
Smart Host Heuristics	False (unchecked)	If checked, the robot converts common alternate host names used by a server to a single name. This is most useful in cases where a site has a number of servers all aliased to the same address, such as `www.sesta.com`, which often have names such as `www1.sesta.com`, `www2.sesta.com`, and so on. When you select this option, the robot will internally translate all host names starting with `wwwn` to `www`, where `n` is any integer. This attribute only operates on host names starting with `wwwn`. This attribute cannot be used when CNAME resolution is `OFF` (false).

Table F-9 Robot Crawling Attributes *(Continued)*

Attribute	Default Value	Description
Resolve hostnames to CNAMEs	False (unchecked)	If checked, the robot validates and resolves any host name it encounters into a canonical host name. This allows the robot to accurately track unique RDs. If unchecked, the robot validates host names without converting them to the canonical form. So you may get duplicate RDs listed with the different host names found by the robot. For example, `devedge.sesta.com` is an alias for `developer.sesta.com`. With CNAME resolution on, a URL referenced as `devedge.sesta.com` is listed as being found on `developer.sesta.com`. With CNAME resolution off, the RD retains the original reference to `devedge.sesta.com`. Smart Host Heuristics cannot be enabled when CNAME resolution is `OFF` (false).
Accepts commands from ANY host	False (unchecked)	Most robot control functions operate through a TCP/IP port. This attribute controls whether commands to the robot must come from the local host system (false), or whether they can come from anywhere on the network (true). It is recommended that you restrict direct robot control to the local host (false). You can still administer the robot remotely through the Administration Console.
Default Starting Point Depth	`10`	1- starting points only, 2- bookmark style, 3-10, unlimited. Default value for the levels of hyperlinks the robot traverses from any starting point. You can set the depth for any given starting point by editing the site on the Robot, Sites page.
Work Directory	`/var/opt/SUNWps/https-`*servernamefull*`/portal/tmp`	Full pathname of a temporary working directory the robot can use to store data. The robot retrieves the entire contents of documents into this directory, often many at a time, so this space should be large enough to handle all of those at once.

Table F-9 Robot Crawling Attributes *(Continued)*

Attribute	Default Value	Description
State Directory	`/var/opt/SUNWps/https-s`*`ervernamefull`*`/portal/robot`	Full pathname of a temporary directory the robot uses to store its state information, including the list of URLs it has visited, the URL pool, and so on. This database can be quite large, so you might want to locate it in a separate partition from the Work Directory.

Indexing

The robot searches sites and collects documents based on the filters you have selected. The documents collected are in many different formats. To make them uniform and easily readable they need to be in one format, which is HTML. This page controls some of the parts that go into each resource description.

Table F-10 Robot Index Attributes

Attribute	Default Value	Description
Full Text or Partial Text	`Partial Text`	Full uses the complete document in the resource description. Partial text only uses the specified number of bytes in the resource description.
extract first # bytes	`4096`	Enter the number of bytes.
Extract Table Of Contents	True (checked)	True includes the Table of Contents in the resource description.
Extract data in META tags	True (checked)	True includes the META tags in the resource description.

Table F-10 Robot Index Attributes *(Continued)*

Attribute	Default Value	Description
Document Converters	All checked (true); if false, that type of document cannot be indexed.	Adobe PDF
		Corel Presentations
		Corel Quattro Pro
		FrameMaker
		Lotus Ami Pro
		Lotus Freelance
		Lotus Word Pro
		Lotus 1-2-3
		Microsoft Excel
		Microsoft Powerpoint
		Microsoft RTF
		Microsoft Word
		Microsoft Works
		Microsoft Write
		WordPerfect
		StarOffice™ Calc
		StarOffice™ Impress
		StarOffice™ Writer
		XyWrite
Converter Timeout	`600`	Time in seconds allowed for one document to be converted to HTML. If this time is exceeded, the URL is excluded.

Simulator

This page is a debugging tool that performs a partial simulation of robot filtering on a URL. You can type in a new URL to check. It checks the URL, DNS translations (including Smart Host Heuristics), and site redirections. It does not check the contents of the document specified by the URL, so it does not detect duplications, MIME types, network errors, permissions, and the like. The simulator indicates whether the listed sites would be accepted by the robot (ACCEPTED) or not (WARNING).

Table F-11 Robot Simulator Properties

Attribute	Default Value	Description
URL	URLs you have already defined and one blank text box.	You can check access to a new site by typing its URL in the blank text box. This checks to see if the new site accepts crawling. Format `http://www.sesta.com:80/`
Check for DNS aliases	True (checked)	True (checked) checks for number of servers aliased to the same address.
Check for Server Redirects (302)	True (checked)	True (checked) checks for any server redirects.

Site Probe

This page is a debugging tool that checks for DNS aliases, server redirects, and virtual servers. This tool returns information about site but does not test its acceptance of crawling.

Table F-12 Robot Site Probe Attributes

Attribute	Default Value	Description
Site	Blank	Type in URL in format `http://www.sesta.com:80`
Show advanced DNS information	False (unchecked)	True (checked) displays more information about the site including IP addresses.

Schedule

This page is where you set up the automatic search schedule for the robot.

Table F-13 Robot Schedule Attributes

Attribute	Default Value	Description
Start Robot Time in hours and minutes	`00:00`	This is the time that the robot starts to search.
Days	none selected	Sun, Mon, Tue, Wed, Thu, Fri, or Sat Check at least one day.

Table F-13 Robot Schedule Attributes

Attribute	Default Value	Description
Stop Robot Time in hours and minutes	`00:00`	If you plan to run the robot continuously, it is recommended that you stop and restart it at least once per day. This gives the robot a chance to release resources and reinitialize itself.
Days	none selected	Sun, Mon, Tue, Wed, Thu, Fri, or Sat

Database

The Database attributes are divided as follows:

- Management
- Import Agents
- Resource Descriptions
- Schema
- Analysis
- Schedule

NOTE To partition the database, you must use the command line function because stopping the search server is required.

Management

The initial Management page lists the available databases. You can create a new one, reindex, purge, or expire an existing one. Use the checkbox to select a database on which to perform an action. Use the small icons above the checkbox to select or deselect all the databases. When you select Reindex, Purge or Expire, a prompt confirming that you want to perform the action with a list of database names displays. To perform the action, select OK.

You should reindex the database if you have edited the schema to add or remove an indexed field (as author), or if a disk error has corrupted the index. You need to restart the server after you change the schema.

Because the time required to reindex the database is proportional to the number of RDs in the database, a large database should be reindexed when the server is not in high demand.

When you purge the contents of the database, disk space used for indexes will be recovered, but disk space used by the main database will not be recovered; instead, it is reused as new data is added to the database.

Expiring a database deletes all RDs that are deemed out-of-date. It does not decrease the size of the database. By default, an RD is scheduled to expire in 90 days from the time of creation.

You can also edit the database by selecting the Edit link which takes you to a page where you define the database attributes.

Table F-14 Database Management Attributes

Attribute	Default Value	Description
Name	Default	Name for the database used by Search.
Title	Blank	A title for the database.
Description	Blank	Describe the database for yourself.

Import Agents

Import agents are the processes that bring resource descriptions from other servers or databases and merge them into your search database.

The initial Import page lists the available import agents. You can create a new one, or run, edit or delete an existing one. Use the checkbox to select an agent to delete. Use the small icons above the checkbox to select or deselect all import agents. Use the radio buttons to turn an Agent Action On or Off. To schedule the import agents, select Schedule on the lower menu bar.

If you choose to edit or modify an existing import agent or create a new one, the following attributes are displayed.

Table F-15 Database Import Agent Attributes

Attribute	Default Value	Description
Charset	Blank for new	Specifies the character set of the input SOIF stream. For example, ISO8859-1, UTF-8, UTF-16. Character sets ISO8859-1 through ISO8859-15 are supported.
Import From	Local File	Select either Local File or Search Server (if one is enabled).

Table F-15 Database Import Agent Attributes *(Continued)*

Attribute	Default Value	Description
Local File Path	Blank for new	Gives the full path name of local file that contains valid resource descriptions in SOIF (*Summary Object Interchange Format).* This can be a file on another server, as long as the path is addressable as if it were locally mounted.
Database Name	Default	Name of the destination database.
Remote Server	Blank for new	Gives the URL of the search server to retrieve resource descriptions from; format `http://www.sesta.com:80`
Instance Name	Blank for new	Server instance name used by the search server. You can find this instance name in the Server Preferences for the server you are importing from. Value must be 3.01C or 3.01C SP1.
Search URI	blank for new	Enter full path and file names. Use `/portal/search`.
Is Compass Server 3.01X?	False (unchecked)	Is the server you are importing from a Compass Server 3.01X?
Enable SSL	False (unchecked)	If this is a server-to-server transaction, select if the servers should use the SSL (Secure Sockets Layer) protocol.
Authentication	`None (default)`	None (default) or Use User / Password This specifies how the import agent should identify itself to the system it imports from. By default, no authentication is used. If the server you want to import from requires authentication, you can specify a user name and password for the import agent to use. Importing from 3.01C does not require authentication. Importing data from 3.01C SP1 requires authentication.
User	Blank for new or none	If you selected Use User / Password, enter a user.
Password	Blank for new or none	If you selected Use User / Password, enter a password (shown as *).

Table F-15 Database Import Agent Attributes *(Continued)*

Attribute	Default Value	Description
Content Transfer	`Use Incremental Gathering of Full Contents (default)`	Choice of Use Incremental Gathering of Full Contents (default) or Use Search Query These specify which resource descriptions to import from the source. By default, an import agent asks for all resource descriptions added or changed since its last import from the same source. The search query specifies that the import agent should request only certain resource descriptions from the source. This is much the same way that users request listings of resources from the search database. Use Scope, View-Attributes and View-Hits fields to specify the query.
Scope	Blank for new	The text of the query. The query syntax is identical to that used for end-user queries from the server.
View-Attributes	Blank for new	Lists which fields (not case sensitive) you want to import in each resource description. For example, title and author. The default is all.
View-Hits	Blank for new	The maximum number of matching resource descriptions to import. If no hits are specified, it defaults to 20.
Agent Description	Blank for new	Appears in the list of available import agents on the initial Import page. It is ignored by the program. If this field is blank, the Resource Description Source file name or server name is used to identify the import agent. Note here if user name and password are needed.
Newest Resource Description	Blank for new	The date of the creation of the newest resource description previously imported by this import agent. This date is used by the Use Incremental Gathering of Full Contents option to determine which resources are new and should be imported.

Table F-15 Database Import Agent Attributes *(Continued)*

Attribute	Default Value	Description
Network Timeout in seconds	Blank for new	Specifies the number of seconds the import agent will allow before timing out the connection over the network. You can adjust this to allow for varying network traffic and quality.

Resource Descriptions

The initial Resource Descriptions page allows you to search the Resource Descriptions in the database. For example, you can correct a typographical error in an RD or manually assign RDs discovered by the robot to categories.

t

Table F-16 Resource Descriptions Attributes

Attribute	Default Value	Description
Search For	`All RDs`	All RDs, Uncategorized RDs, Categorized RDs, RDs by category, Specific RD by URL, RDs that contain
Text box	Blank	Enter a unique text string to identify the RDs searched for. Use with the RDs by category, Specific RD by URL, and RDs that contain attribute values.
Database	Default	Name of the database to search.
Select Category		Browse and select a category from the category tree.
Delete		Delete one or more selected RDs that are returned from an RD search.
Next		Display the next set of RDs returned from an RD search
Previous		Display the previous set of RDs returned from an RD search
Edit Selected		Edit the attributes of one or more RDs that are returned from an RD search.
Edit All		Edit the attributes of the current set of RDs that are returned from an RD search.

To limit the search by category, select Select Category. A Category Editor page displays allowing you to specify the category from the taxonomy for the search. You can specify the category in the Selected Category text box or browse the taxonomy to select it. After specifying the category, select OK to return to the RD search page.

Table F-17 Category Editor Attributes

Attribute	Default Value	Description
Selected Categories	Blank	Text field that displays the selected categories
Expand All		Expands the taxonomy so that all entries in the hierarchy display for browsing.
Collapse All	Blank	Collapses the taxonomy so that only categories within the first two levels of the hierarchy display for browsing.
Categories per page	25	Drop down list of the number of categories to display per page. Values are 25, 50, 100, 250, 500, and all.

A successful search displays the Number of RDs found and a list box with the RDs found. After clicking on the Edit link of an RD, the following attributes, which you can edit, and partial text of the RD are displayed. All these attributes except Classification are set to editable in the Database / Schema page.

Table F-18 Database RD Editable Attributes

Attribute	Default Value	Description
Author	`Blank`	Author(s) of the document.
Author e-mail	`Blank`	Email address to contact the Author(s) of the document.
Classification	`Category name of selected RD.`	Category name if classified; No Classification if not classified.
ReadACL	Blank	Related to document level security.
Content-Charset		Content-Charset information from HTTP Server.
Content-Encoding	Blank	Content-Encoding information from HTTP Server.

Table F-18 Database RD Editable Attributes *(Continued)*

Attribute	Default Value	Description
Content-Language	Blank	Content-Language information from HTTP Server.
Content-Length	Blank	Content-Length information from HTTP Server.
Content-Type	Blank	Content-Type information from HTTP Server.
Description	Description from the selected RD.	Description from RD.
Expires	Valid date.	Date on which resource description is no longer valid.
Full-Text	Blank	Entire contents of the document.
Keywords	Keywords, if any, from the selected RD.	Keywords taken from meta- tags.
Last-Modified	Last modification date	Date when the document was last modified.
Partial-text	Partial text of the document	Partial selection of text from the document
Phone	Blank	Phone number for Author contact
Title	Title of the selected RD.	Title of RD
URL	Blank	Uniform Resource Locator for the document

Schema

The schema determines what information is in a resource description and what form that information is in. You can add new attributes or fields to an RD and set which ones can be edited and which ones can be indexed. When importing new RDs, you can convert schemas embedded in new RDs into your own schema.

Table F-19 Database Schema Edit Attributes

Attribute	Description
Author	Author(s) of the document.
Author-EMail	Email address to contact the Author(s) of the document.
Content-Charset	Content-Charset information from HTTP Server.
Content-Encoding	Content-Encoding information from HTTP Server.
Content-Language	Content-Language information from HTTP Server.

Table F-19 Database Schema Edit Attributes *(Continued)*

Attribute	Description
Content-Length	Content-Length information from HTTP Server.
Content-Type	Content-Type information from HTTP Server.
Description	Brief one-line description for document.
Expires	Date on which resource description is no longer valid.
Full-Text	Entire contents of the document.
Keywords	Keywords that best describe the document.
Last-modifie	Date when the document was last modified.
Partial-Text	Partial selection of text from the document.
Phone	Phone number for Author contact.
ReadACL	Used by Search servers to enforce security.
Title	Title of the document.
URL	Uniform Resource Locator for the document
Aliases Name Description	When you import new RDs, you can convert schemas embedded in new RDs into your own schema. You would use this conversion when there are discrepancies between the names used for fields in the import database schema and the schema used for RDs in your database. An example would be if you imported RDs that used Writer as a field for the author and you used Author in your RDs as the field for the author. The conversion would be Writer to Author, so you would enter Writer in this text box.
Data Type	Defines the data type.
Editable	If true (checked), the selected attribute (field) appears in the Database RD Editor, so you can change its values. Description, Keywords, Title and ReadACL are editable.
Indexable	If true (checked), the selected attribute (field) can be used as a basis for indexing. Author, Title and URL appear in the menu in the Advanced Search screen for the end user. This allows end users to search for values in those particular fields. Author, Expires, Keywords, Last Modified, Title, URL and ReadACL can be used as the basis for indexing.
Score Multiplier	A weighting field for scoring a particular element. Any positive value is valid.

Analysis

The Analysis page shows a sorted list of all sites and the number of resources from that site currently in the search database. Select Update Analysis to update the analysis on file.

Table F-20 Database Analysis Attributes

Attribute	Default Value	Description
Total number of RDs	Current number of RDs in database.	Lists current total number of resource descriptions in the database.
Number of servers	Current number of servers that the database is partitioned across.	The database can be partitioned and placed on a number of servers.
Site	URL or domain that the robot has successfully searched.	A URL or domain that has added resource descriptions to the database.
Number of RDs	Current number of RDs from that site.	Lists current number of RDs from that site.
Type	Type of RD	Resource descriptions can be of many different types, for example, http.
Percentage	Type of RD / Total number of RDs	Percentage of this type of document compared to the total number of resource descriptions.

Schedule

This page is where you set up the schedule for running the import agents.

Table F-21 Database Import Schedule Attributes

Attribute	Default Value	Description
Start Import Time in hours and minutes	`00:00`	Time that the import agent starts to import.
Days	none selected	Sun -Sat Check at least one day.

Categories

End users interact with the search database in two distinct ways: They can type direct queries to search the database, or they can browse through the database contents using a set of categories you design. You assign resources in a search database to categories to clarify complexity. If a large number of items are in the database, it is helpful to group related items together. Your primary concern in setting up your categories should be usability so that end users can more quickly locate specific kinds of items.

The search server uses a hierarchy of categories called a *taxonomy*. The term taxonomy in general describes any system of categories. In the context of a networked resource database such as the search server database, it describes any method you choose of categorizing network resources to facilitate retrieval.

The Categories topic is divided into the following subtopics:

- Category Editor
- Classification Rules Editor

Category Editor

The Category Editor page displays a listing the categories in the taxonomy allowing you to browse the categories. After browsing to the category, you may select the category link to bring up the Classification Rules Editor to set up the Robot collections under specific categories.

Table F-22 Category Editor Attributes

Attribute	Default Value	Description
Expand All		Expands the taxonomy so that all entries in the hierarchy display for browsing.
Collapse All		Collapses the taxonomy so that only categories within the first two levels of the hierarchy display for browsing.
Reindex		Reindexes the database. If you have just created your taxonomy, you need to index the database to make category search available to your end users. If you have changed your categories, you need to reindex the database to make it up-to-date. Save the categories tree before you reindex the database. Load the new taxonomy.

Table F-22 Category Editor Attributes

Attribute	Default Value	Description
Categories per page	25	Drop down list of the number of categories to display per page. Values are 25, 50, 100, 250, 500, and all.
Name	Selected category	Displays the name of the selected category to edit
Description	Blank	Displays the description of the selected category.
Matching Rule	Blank	Displays the the matching rule to use with the selected category.
Update		Updates the category definition.
Add as a child		Adds the category as a child.
Add as a sibling		Adds the category as a sibling.

Classification Rules Editor

After you set up the categories for your database, Click New to set or change the rules the robot for selected categories to assign resources to categories.

Table F-23 Categories Classification Rules Editor Attributes

Attribute	Default Value	Description
Source	Author	The valid attributes include: • Author • Author-EMail • Content-Charset • Content-Encoding • Content-Language • Content-Length • Content-Type • Description • Expires • Full-Text • Keywords • Last-modified • Partial-Text • Phone • ReadACL • Title • URL • Host • Protocol • IP • Path • Type
Method	is	is, contains, begins with, ends with, regular expression
Criteria	Blank	Specifies the criteria for the rule.

Table F-23 Categories Classification Rules Editor Attributes

Attribute	Default Value	Description
Classification	.Blank	Category to in which to classify the RD if the rule conditions are met. Type the category or use the Select Category Edit page to browse to it.

Reports

The Reports section allows you to monitor your search server. You can see a summary of its activity: what sites were searched, what URLs were excluded and why, detailed information about URLs visited by the robot, and what your end users are interested in.

The Reports topic is divided into the following subtopics:

- Starting Points
- Excluded URLs
- Robot Advanced Reports
- Log Files
- Popular Searches

Starting Points

The robot will visit all the enabled sites every time it starts.

Table F-24 Reports Starting Points Attributes

Attribute	Default Value	Description
Enabled	Current value of site.	Yes or No. This is set on the Robot / Sites page.
Starting Point	Chosen `URL:80`	Link brings up chosen URL.
in site definition	Chosen `URL`	Links to Robot / Sites edit page.
Depth	Lists selected level of search.	1-n Set on the Robot / Sites edit page.

Excluded URLs

This page shows a list of robot runs. To display a list of reasons URLs were excluded, select a robot run to examine, select View Selected, then select one of the Reasons for Exclusion. Displayed is a list of the excluded URLs for that reason. Duplicate and warning exclusions have been removed.

Table F-25 Reports Excluded URLs Attributes

Attribute	Default Value	Description
Log	Lists log from most recent run.	Lists all run logs available.
Count	Numbers	List of numbers with reasons for exclusion.
Reason for Exclusion	List of reasons sites have not been allowed. Each reason is linked to a list of all the URLs that were excluded for that reason.	Filter rules, file not found, site not allowed, protocol not allowed, errors, duplication are some of the reasons URLs were excluded.

Robot Advanced Reports

This page gives you access to a number of different reports from the robot. Select from a drop down list to get information for chosen report to show up. The Refresh button gets the current information.

Table F-26 Reports Robot Advance Reports Attribute

Attribute	Default Value	Description
Advanced Robot Reports	`Version`	Version, DNS Cache Dump, Performance, Servers Found-All, Server Found-RDM, Status-Current Configuration, Status -Database (internal), Status-Libnet, Status -Modules, Status-Overview, URLs-ready for extraction, URLs-ready for indexing, URLs-waiting for filtering (URL pool), URLs-waiting for indexing, all reports.

Log Files

This page allows you to view the entries or specific lines from a log file. Drop down list of log files. Enter the number of lines you want to be displayed when you select View button.

Table F-27 Reports View Log Files Attributes

Attribute	Default Value	Description
View this logfile	`Excluded URLs (filter)`	Excluded URLs (filter), RD Manager (rdmgr), RDM Server (rdmsvr), Robot Activities (robot), Search Engine (searchengine), User Queries (rdm).
Number of lines	25	A number you can enter to display the most current entries in the log file.

Popular Searches

This page allows you to see what users are searching for. The most frequent searches appear first in the report.

Table F-28 Reports Popular Searches Attribute

Attribute	Default Value	Description
Exclude Browsing	False (unchecked)	False (unchecked) includes what categories users browse in. True (checked) excludes browsing statistics.

Appendix G

Subscriptions Attributes

The Subscriptions Service consists of dynamic and user attributes. Values applied to the dynamic attributes are applied across the Sun ONE Identity Server configuration and are inherited by every configured organization. They cannot be applied directly to roles or organizations as the goal of global attributes is customize the Sun ONE Identity Server application. Values applied to the user attributes are assigned per user. When the role is assigned to a user or a user is created in an organization, the dynamic attribute then becomes a characteristic of the user.

The Subscriptions attributes are divided into:

Subscriptions Dynamic Attributes

Subscriptions User Attributes

Subscriptions Dynamic Attributes

Table G-1 describes the dynamic attributes for the Subscriptions Service.

The table contains three columns: the first column identifies the attribute, the second column provides the default value for the attribute, and the third column describes the attribute.

Table G-1 Subscriptions Service - Dynamic Attributes

Attribute	Default Value	Description
Maximum number of Categories subscriptions	5	Specifies the maximum number of subscriptions on categories that can be defined and stored in the Identity Server.

Table G-1 Subscriptions Service - Dynamic Attributes

Attribute	Default Value	Description
Maximum number of Discussion subscriptions	5	Specifies the maximum number of subscriptions on discussions that can be defined and stored in the Identity Server.
Maximum number of Saved searches	5	Specifies the maximum number of saved search subscriptions that can be defined and stored in the Identity Server.
Conflict Resolution Level	Highest	Sets the conflict resolution level for the Subscriptions service template used to resolve conflicts when multiple Subscriptions templates are merged. There are seven conflict resolution settings available ranging from Highest to Lowest. Do not confuse this setting with the display profile document priority. The display profile document priority is a numeric value that is set in the XML file with the `priority=` syntax tag. When a merge occurs, it starts with lowest display profile priority document (lowest number) and proceeds by increasing priority number until it arrives at the user (base), the highest priority display profile. When an attribute conflict occurs, the attribute on the template set with the highest conflict resolution level is returned.

Subscriptions User Attributes

Table G-2 describes the user attributes for the Subscriptions Service.

The table contains three columns: the first column identifies the attribute, the second column provides the default value for the attribute, and the third column describes the attribute.

Table G-2 Subscriptions Service - User Attributes

Attribute	Default Value	Description
Category Subscriptions	No default subscriptions	This field defines the subscriptions details. The format is *label* | *target category* | *scope* | *lapsed time* where: • *label* is a free-form string • *target category* is the colon-separated string representation of the category ID. • *scope* is the search query criteria. • *lapsed time* is how far back in time content is the object of the subscription from the time of subscription is evaluated. Subscriptions evaluation is done when users access the Subscriptions channel.

Table G-2 Subscriptions Service - User Attributes

Attribute	Default Value	Description
Category Subscriptions	No default subscriptions	This field defines the subscriptions details. The format is *label* | *target discussion RD's URL* | *scope* | *lapsed time* | *minimum rating* where: • *label* is a free-form string • *target discussion RD's URL* is the space-separated string representation of the discussion ID. • *scope* is the search query criteria. • *lapsed time* is how far back in time content is the object of the subscription from the time of subscription is evaluated. Subscriptions evaluation is done when users access the Subscriptions channel. • *minimum rating* is the threshold rating above which subscription yields content. This field is a numeric value (-1,0,1,2,3) that corresponds to end user rating choices (Irrelevant, Routine, Interesting, Important, Must read).
Saved Search Subscriptions	No default subscriptions	This field defines the subscriptions details. The format is *label* | *scope* | *lapsed time* where: • *label* is a free-form string • *scope* is the search query criteria. • *lapsed time* is how far back in time content is the object of the subscription from the time of subscription is evaluated. Subscriptions evaluation is done when users access the Subscriptions channel.

Appendix H

SSO Adapter Templates and Configurations

This appendix describes how to configure the single sign-on (SSO) adapter in order to adjust options available to end users.

This appendix contains the following sections:

Overview of the Single Sign-On Adapter

The single sign-on adapter service allows end users to use applications, such as a portal server provider or any other web application, to gain authenticated access to various resource servers after signing in once. The resource servers that can be accessed depend on the implementations of the SSO Adapter interface that are available in the system. Currently, Sun™ ONE Portal Server provides SSO Adapters for the following resource servers: Address Book, Calendar, and Mail. Single Sign-On for the Instant Messaging channel is not achieved through SSO Adapter but through the use of the Sun ONE Identity Server authentication method. For information on this method, see the `authMethod` property in Table 12-1 on page 318. The Address Book, Calendar, and Mail services are available through the products:

- Sun™ ONE Calendar Server 5.1.1, 6.0
- Sun™ ONE Messaging Server 5.2, 6.0

Resource servers are typically accessed by an application using a standard application programming interface (API), such as JavaMail for accessing a mail server. To create an authenticated connection using the API, the API must be provided the configuration data for the connection. The purpose of the SSO Adapter is to provide this configuration data, and the SSO Adapter service is used to store that data.

The SSO Adapter service defines two levels of data, templates and configurations. An SSO Adapter template defines a class of connections that are going to be made available to users. A single template is used by many users. It defines data values that are the same for all users that use the template including default values and identification of what values can be edited by a user. Therefore, SSO Adapter templates are defined at a global service level.

An SSO Adapter configuration builds upon a template by providing data values that are specific to an organization, role, or user. A configuration references a template, and takes data values from the template for those properties that are not editable by the user. When an end user changes the user-editable properties of an SSO Adapter configuration, that configuration would then apply only to that one user.

A Sun ONE Portal Server communication channel that uses the SSO Adapter service references either a template or a configuration to get data values needed to obtain a connection to a resource server. If the channel references a template, and the user saves configuration information, the reference is changed to refer to a configuration instead. The configuration then references the template.

SSO Adapter Template Format: Global

Global Attributes for the SSO Adapter

Table 14-25 on page 539 describes the global attributes—which is actually just one attribute—for the SSO Adapter. The table contains three columns: the first column identifies the attribute, the second column provides the default value for the attribute, and the third column describes the attribute.

Table 14-25 SSO Adapter - Global Attributes

Attribute	Default Value	Description
SSOAdapterTemplates	The default value depends on the services that were configured during installation	The SSOAdapterTemplates attribute is a list of strings where each string is in the format of a URL. This string effectively defines a set of name / value pairs. This attribute defines all of the SSO adapters that are available in the system. It also defines all of the ways in which an SSO adapter can be configured. The default values for an SSO adapter are defined in the SSOAdapterTemplates attribute, and organization, role, or user-specific instances are stored in the SSOAdapterConfigurations attribute.

Accessing SSO Adapter Templates

To access the SSO Adapter Template from the Sun™ ONE Identity Server admin console:

1. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Service Configuration tab to display the list of configurable services in the navigation pane (the lower left frame).
3. Scroll down the navigation pane to the Single Sign-on Adapter Configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane (lower right frame).

About SSO Adapter Templates

SSO Adapter templates are created in order to handle server settings.The templates are represented as uniform resource locators (URLs) described in RFC 1738 published by the World Wide Web Consortium (W3C).

The template string contains various properties that—when configured—provide required information to back-end systems.

Template strings are editable in order to allow administrators to assign values to properties within the strings and to apply certain rules of use to those properties.

Template strings start with the word "`default`" followed by the pipe symbol, "`|`." Therefore, any template string entered by an administrator is required to start with the "`default|`" combination. Each template string contains a protocol that follows the pipe symbol. Strings that contain the IMAP and POP protocols apply to Mail SSO Adapter implementations; strings that begin with the HTTP protocol are used by Calendar SSO Adapter implementations; and strings that begin with the LDAP protocol are used by Address Book SSO Adapter implementations.

Code Example 14-1 is an Address Book SSO Adapter template. This example uses LDAP port `489` instead of the LDAP default port, `389`. Using a non-default LDAP port in this example demonstrates the use of two fragments of code that are not necessary when the default LDAP port is used: a colon paired with the LDAP port number—`:489`–and the following substring—`&default=port`.

Template strings appear in the field as one long string; however, for readability purposes, the following string has been divided here into separate lines where line breaks have been added preceding each ampersand (&).

Code Example 14-1 Address Book SSO Adapter Template

```
default|ldap://company22.example.com:489/?configName=SUN-ONE-ADDRESS-BOOK
    &pabSearchBase=o=pab
    &userSearchBase=o=example.com
    &aid=uid=msg-admin,ou=People,o=company22.example.com,o=example.com
    &adminPassword=admin
    &imapHost=imserver.company22.example.com
    &imapPort=143
    &clientPort=1080
    &enableProxyAuth=false
    &proxyAdminUid=[PROXY-ADMIN-UID]
    &proxyAdminPassword=[PROXY-ADMIN-PASSWORD]
    &userAttribute=uid
    &type=AB-TYPE
    &subType=sun-one
    &ssoClassName=com.sun.ssoadapter.impl.LDAPABSSOAdapter
    &encoded=password
    &default=ssoClassName
    &default=host
    &default=port
    &default=pabSearchBase
    &default=userSearchBase
    &default=aid
    &default=adminPassword
    &default=imapHost
    &default=imapPort
    &default=clientPort
    &default=type
    &default=subType
```

Code Example 14-1 Address Book SSO Adapter Template

```
&default=enableProxyAuth
&default=proxyAdminUid
&default=proxyAdminPassword
&default=userAttribute
&merge=uid
&merge=password
&default=enablePerRequestConnection
&enablePerRequestConnection=true
```

The following line of code is an example of the possible properties in the front portion of an SSO Adapter template string or an SSO Adapter configuration string. This portion when compared to Code Example 14-1 demonstrates how properties are assigned values. Table 14-26 clarifies each property and Table 14-27 explains the property types.

protocol`://`*uid*`:`*password*`@`*host*`:`*port*`/?configName=`*configuration-name*`&ssoClassName=`*sso-adapter-class*`&...`

The preceding portion of an SSO Adapter template string is the proper format for both templates—which apply to all users of that Sun ONE Portal Server instance—and configurations—which apply to specific organizations, roles, and users. However, certain fragments of the preceding portion often do not appear within a template or configuration string. For example, the fragment "*uid:password@*" is not commonly used within templates because it is generally a value that is specific to a particular user.

Using the aforementioned fragment within an SSO Adapter template sets the same user ID and password for all users. This type of configuration is plausible in some situations. For example, a site might want to create a read-only calendar that lists site-wide events. All users would get the Calendar channel on their Desktops using the same user ID and password and they would see the same calendar.

Table 14-26 Some of the Properties in an SSO Adapter Template String

Property Name	Description	Necessity
`protocol`	The protocol used to talk to the server	Optional
`uid`	The User ID of the user who is on the server that is referenced by host	Optional
`password`	The password—which is encoded—of the user on the server referenced by host	Optional
`host`	The server host name	Optional

Table 14-26 Some of the Properties in an SSO Adapter Template String

Property Name	Description	Necessity
`port`	The server port number	Optional
`configName`	The name of the SSO Adapter Template	Mandatory
`ssoClassName`	The fully qualified class name for the SSO Adapter	Mandatory
`type`	The type of service to which an SSOAdapter template or configuration applies. This property is useful for putting a collection of SSO Adapter templates or configurations into type-related groups; for example, to help select a default configuration when the selection is not explicit. Currently, the value can be one of the following: • AB-TYPE • MAIL-TYPE • CALENDAR-TYPE	Optional
`subType`	The vendor or product specific platform to which the SSO Adapter template or configuration applies. This property is useful when you want to support features for a specific product or platform. Currently, the value can be one of the following: • sun-one • notes • exchange	Optional
`enablePerRequestConnection`	A performance tuning option, this boolean property has a default setting of true. If `enablePerRequestConnection=true` then every request to the portal desktop opens a new connection to the back-end store and closes the connection. If `enablePerRequestConnection=false` then the Portal Desktop opens a new connection to the back-end store at portal log in and closes the connection when the user's session is terminated.	Optional

SSO Adapter templates recognize the following property types:

Table 14-27 Property Types in an SSO Adapter Template String

Property Type	Description
merge	Denotes that this value is user editable. In Code Example 14-1 on page 540, notice that only two values can be edited by the user: `uid` and `password`.

default	Denotes that an attribute is set to a default, which is actually a two step process. In Code Example 14-1 on page 540, notice that `imapPort` is set to a specific port at one point in the string, `&imapPort=143,` and set as the default later in the string, `&default=imapPort`.
	
	The following example—which includes fragments of an SSO Adapter template and an SSO Adapter configuration—demonstrates how the `default` property works:
	
	• SSO Adapter template:
	
	`configName=t1&ex1=ex2&exa=exb&default=exa`
	
	• SSO Adapter configuration:
	
	`configName=c1&configDesc=t1`
	
	Then the resulting list of properties that is seen by the SSOAdapter implementation is just:
	
	`exa=exb`
	
	The `ex1=ex2` value in the template is ignored because it is not listed as a default attribute.

Table 14-27 Property Types in an SSO Adapter Template String

Property Type	Description

encoded	Denotes that the attribute is not passed in clear text, but instead is obfuscated. In Code Example 14-1 on page 540, notice that only one value `password`, is encoded.
	
	The value `adminPassword` is not encoded. To encode a property such as `adminPassword`:
	
	1. Enter the encrypted value into the SSO Adapter template string. For this example, the encrypted value follows the equal sign of the following substring:
	
	`&adminPassword=`
	
	• To encrypt a plain text value, use the Sun ONE Identity Server Software Development Kit (SDK) class that follows, where *plain-text-value* is the value of a property before it is encrypted:
	
	`AMPaswordUtil.encrypt` *plain-text-value*
	
	• Use the encrypt method with the preceding class to obtain the encrypted value.
	
	2. Add a substring that assigns `encoded` to the property you want to encrypt. For this example the property is `adminPassword` and it is added to the end of the string (a convenient location):
	
	`&encoded=adminPassword`
	
	This action changes the SSO Adapter template to end as follows:
	
	`...merge=uid&merge=password&encoded =adminPassword`

SSO Adapter Configuration Format: Dynamic

Dynamic Attributes for the SSO Adapter

Table 14-28 describes the dynamic attributes for the SSO Adapter. The table contains three columns: the first column identifies the attribute, the second column provides the default value for the attribute, and the third column describes the attribute.

Table 14-28 SSO Adapter - Dynamic Attributes

Attribute	Default Value	Description
Conflict ResolutionLevel	Highest	Sets the conflict resolution level for the SSO adapter template used to resolve conflicts when multiple templates are merged. There are seven conflict resolution settings available ranging from Highest to Lowest. When an attribute conflict occurs, the attribute on the template set with the highest conflict resolution level is returned.
SSOAdapterConfigurations	The default value depends on the services that were configured during installation	The SSOAdapterConfigurations attribute is a list of strings where each string is in the format of a URL. This string defines specific instances of the SSO adapters that are defined in the SSOAdapterTemplates attribute. The default values for an SSO adapter are defined in the SSOAdapterTemplates attribute, and organization, role, or user-specific instances are stored in the SSOAdapterConfigurations attribute.

Accessing SSO Adapter Configurations

NOTE To edit the SSO Adapter configurations, follow the steps as shown subsequently—which access the configurations by selecting the Identity Management tab, as described in step 2. Do not access the configurations in the Service Configuration tab as described in "Accessing SSO Adapter Templates."

To access the SSO Adapter configurations from the Sun ONE Identity Server admin console:

1. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
2. Click the Identity Management tab to display the View drop down list in the navigation pane.
3. Select Services in the View drop down list to display the list of configurable services.
4. Scroll down the navigation pane to the Single Sign-on Adapter Configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

About SSO Adapter Configurations

Code Example 14-2 is a Mail SSO Adapter configuration.

Code Example 14-2 Mail SSO Adapter Configuration

```
default|imap:///?configName=sunOneMail&configDesc=SUN-ONE-MAIL
```

As mentioned previously, Dynamic SSO Adapter configurations have the same format as the Global SSO Adapter Service templates:

protocol`://`*uid*`:`*password@host*`:`*port*`/?configName=`*configuration-name*
`configDesc=`*sso-adapter-template*`&....`

For SSO Adapter templates, certain fragments of the preceding portion, such as "*uid:password@host:port*," tend not to appear; however, for SSO Adapter configuration strings, while that type of fragment would tend not to appear at the organization or role level, it would tend to appear at the user level.

The properties recognized at the dynamic level are:

Table 14-29 Properties in an SSO Adapter Configuration String

Properties	Description	Necessity
`configName`	This is the unique identifier of the SSO Adapter template or configuration definition.	Mandatory
`configDesc`	This is an SSO Adapter Template value. The value for the `configDesc` property from a Dynamic SSO Adapter configuration string is the same as the value for the `configName` property from a Global SSO Adapter string (assuming the two strings begin with the same protocol).	Mandatory

SSO Adapter Template and Configuration Examples

Two examples follow of how to create and share portal channel configurations. For both examples the data that is distributed between the SSO Adapter template and configuration is almost exactly the same. However, the first example demonstrates how to share the properties globally while the second example demonstrates how to share the properties within a single organization. For both of these examples, users are limited (in the editing they need to do) to entering user ID and password information, which then enables them to launch that channel from the desktop.

Server Is Defined within the SSO Adapter Template

Server Is Defined at the Organization Level

Some Users Won't See Configuration Changes

When you make changes to the SSO Adapter templates and configurations, which are described in the next two examples, not all users will see the changes on their desktops. Users who have already edited their channel preferences by editing a channel from the desktop will not see future changes made by administrators to any channels, existing or new. The steps for implementing administrators' configurations to these users are described in "Some Users Won't See Configuration Changes."

Server Is Defined within the SSO Adapter Template

This section describes configuring an SSO Adapter template on a server that is shared globally. Therefore, all subdivisions of the global level—from organizations to roles—share the same configuration. For information on configuring a server at the organizational level see "Server Is Defined at the Organization Level" on page 554.

For this configuration, the outcome is that users will have a Mail channel on their Desktop where the Mail channel is editable and where the user only needs to enter their credentials—user ID (uid) and password— to complete the configuration.

The following example creates a new SSO Adapter template, SSO Adapter configuration, and Mail channel.

1. Add a new SSO Adapter template, which for this example is named `credentialMailTemplate`.

    a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*`hostname:port`*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

    b. Click the Service Configuration tab to display the list of configurable services in the navigation pane.

    c. Scroll down the navigation pane to Single Sign-on Adapter Configuration and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

    d. Click in the blank configuration description field—which is just above the Add and Remove buttons— it is in the box labeled SSO Adapter Templates under the heading Global as opposed to Dynamic.

e. In the blank configuration description field, type in the entire SSO Adapter Template string as shown subsequently in Code Example 14-3; replace the variable information with the specific information for your site, unless a particular example also fits the information for your site (therefore, replace some, if not all, the following values: `credentialMailTemplate`, `company22.example.com:143`, `company22.example.com`, `true`, and `1080`).

If the field is not blank when you get to it, select all the text in the field and delete it.

Code Example 14-3 Mail SSO Adapter Template for Sharing Globally

```
default|imap://company22.example.com:143/?configName=credentialMailTemplate
    &encoded=password
    &default=protocol
    &default=clientProtocol
    &default=type
    &default=subType
    &default=enableProxyAuth
    &default=proxyAdminUid
    &default=proxyAdminPassword
    &default=ssoClassName
    &default=enablePerRequestConnection
    &default=userAttribute
    &default=host
    &default=port
    &default=smtpServer
    &default=clientPort
    &default=smtpPort
    &enableProxyAuth=false
    &proxyAdminUid=[PROXY-ADMIN-UID]
    &proxyAdminPassword=[PROXY-ADMIN_PASSWORD]
    &type=MAIL-TYPE
    &subType=sun-one
    &ssoClassName=com.sun.ssoadapter.impl.JavaMailSSOAdapter
    &enablePerRequestConnection=true
    &userAttribute=uid
    &clientProtocol=http
    &smtpServer=company22.example.com
    &sentFolderCopy=true
    &clientPort=1080
    &smtpPort=25
    &merge=uid
    &merge=password
```

f. Click Add.

g. Click Save.

At this point, there may be more than one string that begins with the IMAP protocol. This is acceptable.

2. Add a new SSO Adapter Configuration, which for this example is named `credentialMail`. Chose your own template name for your site.

a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

b. Click the Identity Management tab to display the View drop down list in the navigation pane.

c. Select Services in the View drop down list to display the list of configurable services.

d. Scroll down the navigation pane to the Single Sign-on Adapter configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

e. Click in the blank configuration description field—which is just above the Add and Remove buttons.

f. In the blank configuration description field, type in the following line of code, where for this example the configuration name is `credentialMail` and the configuration description is `credentialMailTemplate` (replace the names used for configuration name and configuration description with the specific information for your site):

```
default|imap:///?configName=credentialMail
&configDesc=credentialMailTemplate
```

If the field is not blank when you get to it, select all the text in the field and delete it.

g. Click Add.

h. Click Save

3. Add a new Mail Channel to the Desktop. For this example the name of the new channel is `CredentialMailChannel`.

a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

b. Click the Identity Management tab to display the View drop down list in the navigation pane.

c. Select Services in the View drop down list to display the list of configurable services.

d. Scroll down the navigation pane to the Portal Server Configuration heading, click the arrow next to Portal Desktop to bring up the Portal Desktop page in the data pane.

e. Click the Channel and Container Management link.

f. Under the Channels heading, click New.

g. In the Channel Name field, type the name for the new channel, which for this example is `CredentialMailChannel`.

h. In the Provider drop down menu, select MailProvider.

i. Click OK, which returns you to the Channel and Container Management Web page where the channel you just created now exists.

j. Under the Channels heading, click Edit Properties next to the name of the channel you just created, which for this example is `CredentialMailChannel`.

k. In the "title" field, select and delete any words that currently exist, for example `mail`, and type a provider title, which for this example is `Credential Only Mail Account`.

l. In the description field, select and delete any words that currently exist, for example `mail`, and type a provider description, which for this example is again `Credential Only Mail Account`.

m. Scroll down the page (still in the data pane); select and delete any words that currently exist in the "ssoAdapter" field, for example `SunOneMail`; and type the same SSO Adapter configuration name used in Step 2, which for this example is `credentialMail`.

n. Scroll as needed and click Save.

o. Scroll back up the page to click the word `top`, which is the first item following the words `Container Path`.

p. Scroll down to the Container Channels heading and click the link for the container that you want to add the new channel to. For example, `MyFrontPageTabPanelContainer`. Do not click the accompanying Edit Properties link.

q. Under the Channel Management heading, click the name of the channel you just created. For example, `CredentialMailChannel`, which is in the Existing Channels list.

r. Click the Add button that is next to the Available and Visible list. This makes the channel available to users and visible without any further configuration.

s. Scroll back up the page to click Save under the Channel Management heading.

You have finished adding a new Mail channel to the Desktop. Now, limit the fields that end users will see (and be able to edit) when they click the edit button in the Mail channel. You will only keep the User ID and password fields.

4. Use the Identity Server admin console to retrieve the organization's display profile document from the directory server. See *"To Download and Upload a Display Profile" on page 165* and follow the steps for downloading and saving (locally) the display profile document.

5. Use the editor of your choice to open the display profile document and to locate the name of the channel you created in Step 3 on page 550, for example, `CredentialMailChannel`. The section of text you need to locate will look similar to the following:

```
<Channel name="CredentialMailChannel" provider="MailProvider" merge="replace">
        <Properties>
            <String name="title" value="Credential Only Mail Account"/>
            <String name="description" value="Credential Only Mail Account"/>
            <String name="ssoAdapter" value="credentialMail"/>
        </Properties>
</Channel>
```

6. Add an `SSOEditAttributes` collection that only contains a `uid` and `password`. Such a collection looks similar to the following:

```
<Collection name="ssoEditAttributes">
        <String name="uid" value="string|User Name:"/>
        <String name="password" value="password|User Password:"/>
</Collection>
```

After adding this type of collection, the channel definition will look similar to the following:

```
<Channel name="CredentialMailChannel" provider="MailProvider" merge="replace">
         <Properties>
             <String name="title" value="Credential Only Mail Account"/>
             <String name="description" value="Credential Only Mail Account"/>
             <String name="ssoAdapter" value="credentialMail"/>

             <Collection name="ssoEditAttributes">
                   <String name="uid" value="string|User Name:"/>
                   <String name="password" value="password|User Password:"/>
             </Collection>

         </Properties>
</Channel>
```

7. Use the Identity Server admin console to upload the newly edited display profile document. Again, see *"To Download and Upload a Display Profile" on page 165* . This time, follow the steps for uploading a display profile.

8. Create a new portal end user and authenticate to the desktop (optional).

   If you create new users, they will see the configuration changes, you just made, on their portal desktops. Existing users who have not previously configured any of the channels from their desktops will also see the changes you just made. However, existing users who have configured a channel from their desktops won't see the changes you just made. To allow them to see those changes, refer to "Some Users Won't See Configuration Changes" on page 557

   a. Click the Identity Management tab—if it is not already selected—to display the View drop down list in the navigation pane.

   b. Select Users in the View drop down list.

   c. Click New to display the New User page in the data pane.

   d. Click in the checkboxes next to the services to be assigned to the user.

      At a minimum, select Portal Desktop and SSO Adapter.

   e. Enter the user information in the appropriate text fields, scrolling as needed.

   f. Scroll as needed and click Create.

      The new user's name then appears in the navigation pane.

Server Is Defined at the Organization Level

This section describes configuring an SSO Adapter template at the organizational level. The data used in the SSO Adapter template and SSO Adapter configuration in this example is almost exactly the same as the data used in the example in "Server Is Defined within the SSO Adapter Template" on page 548. However, in the following example, more of the properties appear within the SSO Adapter configuration and fewer appear within the SSO Adapter template. Putting the properties in the SSO Adapter configuration allows you to share those properties within an organization rather than sharing the properties globally.

The following example creates a new SSO Adapter configuration and Mail channel. The default SSO Adapter template is used in this example. You do not need to create another template:

Code Example 14-4 Mail SSO Adapter Template for Sharing within an Organization.

```
default|imap:///?configName=SUN-ONE-MAIL
    &encoded=password
    &default=ssoClassName
    &default=protocol
    &default=clientProtocol
    &merge=host
    &merge=port
    &merge=uid
    &merge=password
    &merge=smtpServer
    &merge=smtpPort
    &merge=sentFolderCopy
    &merge=clientPort
    &clientProtocol=http
    &ssoClassName=com.sun.ssoadapter.impl.JavaMailSSOAdapter
```

1. Add a new SSO Adapter Configuration, which for this example is named `orgCredentialMail`. Chose your own name for your site.
   a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
   b. Click the Identity Management tab to display the View drop down list in the navigation pane.
   c. Select Services in the View drop down list to display the list of configurable services.

d. Scroll down the navigation pane to the Single Sign-on Adapter configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

e. Click in the blank configuration description field—which is just above the Add and Remove buttons.

f. In the blank configuration description field, type in the following line of code, where for this example the configuration name is `orgcredentialMail` and the configuration description is `SUN-ONE-MAIL`; replace the variable information with the specific information for your site, unless a variable example used here also fits the information for your site (therefore, replace some, if not all, the following values: `company22.example.com:143`, `orgcredentialMail`, `ccompany22.example.com`, `true`, `1080`, and `25`).

```
default|imap://company22.example.com:143/?configName=orgCrede
ntialMail
&configDesc=SUN-ONE-MAIL
&smtpServer=company22.example.com
&sentFolderCopy=true
&clientPort=1080
&smtpPort=25
```

If the field is not blank when you get to it, select all the text in the field and delete it.

g. Click Add.

h. Click Save.

At this point, there may be more than one string that begins with the IMAP protocol. This is perfectly acceptable.

2. Add a new Mail Channel to the My Front Page tab; for this example, the name of the new channel is `OrgCredentialMailChannel`.

a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*hostname:port*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`

b. Click the Identity Management tab to display the View drop down list in the navigation pane.

c. Select Services in the View drop down list to display the list of configurable services.

d. Scroll down the navigation pane to the Portal Server Configuration heading and click the arrow next to Portal Desktop to bring up the Portal Desktop page in the data pane

e. Click the Channel and Container Management link.

f. Under the Channels heading, click New.

g. In the Channel Name field, type the name for the new channel, which for this example is `OrgCredentialMailChannel`.

h. In the Provider drop down menu, select MailProvider.

i. Click Create, which returns you to the Channel and Container Management Web page where the channel you just created now exists.

j. Under the Channels heading, click Edit Properties next to the name of the channel you just created, which for this example is `OrgCredentialMailChannel`.

k. In the title field, select and delete any words that currently exist, for example `mail`, and type a provider title, which for this example is `Organization Defined Credential Only Mail Account`.

l. In the description field, select and delete any words that currently exist, for example `mail`, and type a provider description, which for this example is again `Credential Only Mail Account`.

m. Scroll down the page (still in the data pane); select and delete any words that currently exist in the ssoAdapter field, for example `sunOneMail`; and type the same SSO Adapter configuration name used in Step 1, which for this example is `orgCredentialMail`.

n. Scroll as needed and click Save.

o. Scroll back up the page to click the word `top`, which is the first item following the words `Container Path`.

p. Scroll down to the Container Channels heading and click the link for the container that you want to add the new channel to. For example, `MyFrontPageTabPanelContainer`. Do not click the accompanying Edit Properties link

q. Under the Channel Management heading, click the name of the channel you just created. For example, OrgCredentialMailChannel, which is in the Existing Channels list.

r. Click the Add button that is next to the Available and Visible list. This makes the channel available to users and visible without any further configuration.

s. Scroll back up the page to click Save under the Channel Management heading.

You have finished adding a new Mail channel to the Desktop. For this type of configuration, when end users click the edit button for this channel, all the editable fields will be populated except for User ID and password, which they will need to fill in. If you want to remove all the fields except for the User ID and password fields, follow the steps from the *Server Is Defined within the SSO Adapter Template* section starting with Step 4 on page 552

3. Create a new portal user and authenticate to the desktop (optional).

If you create new users, they will see the configuration changes, you just made, on their portal desktops. Existing users who have not previously configured any of the channels from their desktops will also see the changes you just made. However, existing users who have configured a channel from their desktops won't see the changes you just made. To allow them to see those changes, refer to "Some Users Won't See Configuration Changes" on page 557.

a. Click the Identity Management tab—if it is not already selected—to display the View drop down list in the navigation pane.

b. Select Users in the View drop down list.

c. Click New to display the New User page in the data pane.

d. Click in the checkboxes next to the services to be assigned to the user.

At a minimum, select Portal Desktop and SSO Adapter.

e. Enter the user information in the appropriate text fields, scrolling as needed.

f. Scroll as needed and click Create.

The new user's name then appears in the navigation pane.

Some Users Won't See Configuration Changes

Administrators make channel configuration changes—including the adding of new channels— by editing or creating SSO Adapter templates and configurations. These changes do not affect all users. The users they affect are:

- users who have never configured their channels
- users who will be added after the configuration changes are made

However, for users who have previously changed one or more of their channel configurations—which they do by editing a channel from their desktops—administrators need to make configuration changes directly at the user level before the changes take affect. For example, when administrators add a channel at the global, organization, or role level, the channel does not appear on these users' desktops.

This situation occurs because of the way Class of Service functions in the directory server. Users who configure one or more of their channels overwrite the SSO Adapter templates and configurations. Thereafter—for these users—values added by administrators at the global, organization, or role level are no longer inherited at the user level.

Therefore, configure changes directly at the user level for every user who has previously changed one or more of their channel configurations. It is usually more convenient to configure changes for other users first then to copy those template and configuration strings and paste them directly at the user level for the users that need them; the following instructions assume that you will configure changes in this manner. However, you can key in configuration strings directly at the user level without copying and pasting, if you wish.

After making configuration changes for others, make configuration changes directly at the user level—for those who need it—with one of the two following methods:

- User-Level Configuration Changes for One to a Few Users
- User-Level Configuration Changes for Many Users (Using a Script)

User-Level Configuration Changes for One to a Few Users

Copy and paste the SSO Adapter template string or the SSO Adapter configuration string that you just edited or created for other users to the users who need the changes made directly at the user level as follows:

1. Copy a string from an SSO Adapter configuration:
    a. From an Internet browser, log on to the Sun ONE Identity Server admin console at `http://`*`hostname:port`*`/amconsole`, for example `http://psserver.company22.example.com:80/amconsole`
    b. Click the Identity Management tab to display the View drop down list in the navigation pane.

c. Select Services in the View drop down list to display the list of configurable services.

d. Scroll down the navigation pane to the Single Sign-on Adapter Configuration heading and click the arrow next to SSO Adapter to bring up the SSO Adapter page in the data pane.

e. Click the string that you want to copy, such as "`default|http:///...`","`default|imap:///...`", etc.

f. With the string you just selected showing in the configuration description field—which is just above Add and Remove buttons—select and copy the entire contents of the field and go on to Step 2.

2. In the Sun ONE Identity Server admin console, click the Identity Management tab to display the View drop down list in the navigation pane.

3. Select Users in the View drop down list to display the list of Sun ONE Portal Server users.

4. Click the arrow next to the user whose user level configuration you want to edit to display another View drop down list, but in the data pane.

5. Click SSO Adapter in the View drop down list within the data pane.

6. Click in the configuration description field—which is just above Add and Remove buttons.

7. Paste the SSO Adapter configuration string that you copied in Step 1 on page 558 here in the configuration description field.

8. Scroll to the far right, past the SSO Adapter configurations box.

9. Select Customize from the drop down list, if it is not already selected (possible selections are Customize, Inherit, and Ignore).

10. Scroll to the left as needed and click Save.

User-Level Configuration Changes for Many Users (Using a Script)

To apply configuration changes to many users directly at the user level you will create a file made up of simple scripts that—among other things—identify specific users and the specific SSO Adapter template(s) and configuration(s) you want to connect each of these users to.

You will then issue an `ldapmodify` command that references the file; the file then modifies the directory server by implementing the user configuration changes in the scripts.

1. Create a file similar to that in Code Example 14-5 using information specific to your site.

   - Name the file using the `.ldif` suffix.
   - In the file, create a separate entry for every user who needs the SSO Adapter configured for them directly at the user level.
   - Include the four lines of code for each user.

   The following example file includes only two example entries. This file is named `attr.ldif`; the two users are named `user1` and `user2`; the organization name is `example`; the configuration name—which is the name of the SSO Adapter configuration that both users happen to be referencing—is `group1imapmail`; the configuration description—which identifies which SSO Adapter template the `group1imapmail` configuration is referencing—is `everyoneimap`. For this example, it is the same for both users.

Code Example 14-5 A file named `attr.ldif` with Scripts for the Directory Server

```
dn: uid=user1,ou=People,o=example.com,o=isp
changetype: modify
add:sunSSOAdapterConfigurations
sunSSOAdapterConfigurations:
    default|imap:///?configName=group1imapmail&configDesc=everyoneimap

dn: uid=user2,ou=People,o=example.com,o=isp
changetype: modify
add:sunSSOAdapterConfigurations
sunSSOAdapterConfigurations:
    imap:///?configName=group1imapmail&configDesc=everyoneimap
```

2. Use an `ldapmodify` command similar to that used in Code Example 14-6 to send the file you created in step 1 to the directory server. When entering the code for these commands, use information specific to your site

   The following example lists commands needed to send a file to the directory server to be read. The following information is specific to this example site: the password is `mypassword`, the host name is `localhost`; the port number is the default, `389`; and the file being sent is named `attr.ldif`.

Code Example 14-6 Sending a File Named `attr.ldif` to the Directory Server

```
setenv LD_LIBRARY_PATH Directory-server-install-dir/lib

Directory-server-install-dir/shared/bin/ldapmodify -D "cn=Directory Manager
    " -w mypassword -h localhost -p 389 -f attr.ldif
```

The preceding code first sets the path `LD_LIBRARY_PATH` and indicates the location of the `ldapmodify` command. Then, `ldapmodify` is issued. A summary of each option used with this command follows:

- `-D` specifies the distinguished name, in this case "`cn=Directory Manager,`" to bind to the directory
- `-w` specifies the password, in this case `mypassword`, for authenticating to the directory
- `-h` specifies the host, in this case `localhost`, on which the directory server is running
- `-p` specifies the port, in this case the default port `389`, through which the directory server is listening
- `-f` specifies a file, in this case `attr.ldif`, to be read by the directory server

For more information about the `ldapmodify` command see *Sun ONE Directory Server Administration Guide*.

Appendix I

Schema Reference

This appendix provides a reference for the Sun™ ONE Portal Server LDAP schema definitions.

This appendix contains these sections:

- Sun ONE Portal Server Desktop Schema
- Sun ONE Portal Server NetMail Schema
- Sun ONE Portal Server Search Schema

Sun ONE Portal Server Desktop Schema

The `psDesktop.ldif` file on a default installation is in the `/opt/SUNWps/export` directory.

Code Example I-1 Desktop Schema

```
#
# Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
# PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.
#
#
# Sun ONE Portal Server (iPS) Desktop Service Schema
# Last Modified October 2001
#

dn: cn=schema
changetype:modify
add:attributeTypes
```

Code Example I-1 Desktop Schema *(Continued)*

```
attributeTypes: ( sunPortalDesktopSessionReturnURLParamName-oid NAME
‘sunPortalDesktopSessionReturnURLParamName’ DESC ‘iPS Desktop Attribute’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDpIsValidating-oid NAME
‘sunPortalDesktopDpIsValidating’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDpNamespaceURL-oid NAME
‘sunPortalDesktopDpNamespaceURL’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopEditProviderContainerName-oid NAME
‘sunPortalDesktopEditProviderContainerName’ DESC ‘iPS Desktop Attribute’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDpContextClassName-oid NAME
‘sunPortalDesktopDpContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDpUserContextClassName-oid NAME
‘sunPortalDesktopDpUserContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopContainerProviderContextClassName-oid NAME
‘sunPortalDesktopContainerProviderContextClassName’ DESC ‘iPS Desktop
Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access
Management’ )

attributeTypes: ( sunPortalDesktopDebugContextClassName-oid NAME
‘sunPortalDesktopDebugContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopServiceContextClassName-oid NAME
‘sunPortalDesktopServiceContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopSessionAppContextClassName-oid NAME
‘sunPortalDesktopSessionAppContextClassName’ DESC ‘iPS Desktop Attribute’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopSessionContextClassName-oid NAME
‘sunPortalDesktopSessionContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )
```

Code Example I-1 Desktop Schema *(Continued)*

```
attributeTypes: ( sunPortalDesktopAuthlessSessionContextClassName-oid NAME
‘sunPortalDesktopAuthlessSessionContextClassName’ DESC ‘iPS Desktop Attribute’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDesktopContextClassName-oid NAME
‘sunPortalDesktopDesktopContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopTemplateContextClassName-oid NAME
‘sunPortalDesktopTemplateContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopClientContextClassName-oid NAME
‘sunPortalDesktopClientContextClassName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopProviderManagerContextClassName-oid NAME
‘sunPortalDesktopProviderManagerContextClassName’ DESC ‘iPS Desktop Attribute’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopPropertiesContextClassName-oid NAME
‘sunPortalDesktopPropertiesContextClassName’ DESC ‘iPS Desktop Attribute’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopAuthorizedAuthlessUIDs-oid NAME
‘sunPortalDesktopAuthorizedAuthlessUIDs’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDefaultAuthlessUID-oid NAME
‘sunPortalDesktopDefaultAuthlessUID’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDefaultChannelName-oid NAME
‘sunPortalDesktopDefaultChannelName’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopType-oid NAME ‘sunPortalDesktopType’ DESC
‘iPS Desktop Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE
Access Management’ )

attributeTypes: ( sunPortalDesktopDpDocument-oid NAME
‘sunPortalDesktopDpDocument’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )
```

Code Example I-1 Desktop Schema *(Continued)*

```
attributeTypes: ( sunPortalDesktopDpLastModified-oid NAME
‘sunPortalDesktopDpLastModified’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopExecutable-oid NAME
‘sunPortalDesktopExecutable’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDpDocumentUser-oid NAME
‘sunPortalDesktopDpDocumentUser’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalDesktopDpLastModifiedUser-oid NAME
‘sunPortalDesktopDpLastModifiedUser’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

dn: cn=schema
changetype:modify
add:objectClasses
objectClasses: ( sunPortalDesktopPerson-oid NAME ‘sunPortalDesktopPerson’ DESC
‘Sun ONE Desktop Service’ SUP top AUXILIARY MAY (
sunPortalDesktopSessionReturnURLParamName $ sunPortalDesktopDpIsValidating $
sunPortalDesktopDpNamespaceURL $ sunPortalDesktopEditProviderContainerName $
sunPortalDesktopDpContextClassName $ sunPortalDesktopDpUserContextClassName $
sunPortalDesktopContainerProviderContextClassName $
sunPortalDesktopDebugContextClassName $
sunPortalDesktopServiceContextClassName $
sunPortalDesktopSessionAppContextClassName $
sunPortalDesktopSessionContextClassName $
sunPortalDesktopAuthlessSessionContextClassName $
sunPortalDesktopDesktopContextClassName $
sunPortalDesktopTemplateContextClassName $
sunPortalDesktopClientContextClassName $
sunPortalDesktopProviderManagerContextClassName $
sunPortalDesktopPropertiesContextClassName $
sunPortalDesktopAuthorizedAuthlessUIDs $ sunPortalDesktopDefaultAuthlessUID $
sunPortalDesktopDpDocument $ sunPortalDesktopDpLastModified $
sunPortalDesktopDefaultChannelName $ sunPortalDesktopType $
sunPortalDesktopExecutable $ sunPortalDesktopDpDocumentUser $
sunPortalDesktopDpLastModifiedUser) X-ORIGIN ‘Sun ONE Access Management’ )
```

Sun ONE Portal Server NetMail Schema

The `psNetMail.ldif` file on a default installation is in the `/opt/SUNWps/export` directory.

Code Example I-2 NetMail Schema

```
#
# Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
# PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.
#
#
# Sun ONE Portal Server (iPS) Netmail Service Schema
# Last Modified October 2001
#

dn: cn=schema
changetype:modify
add:attributeTypes

attributeTypes: ( sunPortalNetmailIMAPServerName-oid NAME
‘sunPortalNetmailIMAPServerName’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailSMTPServerName-oid NAME
‘sunPortalNetmailSMTPServerName’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailDefaultMailDomain-oid NAME
‘sunPortalNetmailDefaultMailDomain’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailRootFolder-oid NAME
‘sunPortalNetmailRootFolder’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailSentMessagesFolder-oid NAME
‘sunPortalNetmailSentMessagesFolder’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailReplyWithAuthor-oid NAME
‘sunPortalNetmailReplyWithAuthor’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailReplyWithDate-oid NAME
‘sunPortalNetmailReplyWithDate’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )
```

Code Example I-2 NetMail Schema *(Continued)*

```
attributeTypes: ( sunPortalNetmailReplyWithBody-oid NAME
‘sunPortalNetmailReplyWithBody’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailIndentPrefix-oid NAME
‘sunPortalNetmailIndentPrefix’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailAddSignature-oid NAME
‘sunPortalNetmailAddSignature’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailInitialHeaders-oid NAME
‘sunPortalNetmailInitialHeaders’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailInactivityInterval-oid NAME
‘sunPortalNetmailInactivityInterval’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailMaxAttachLen-oid NAME
‘sunPortalNetmailMaxAttachLen’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailAutoload-oid NAME ‘sunPortalNetmailAutoload’
DESC ‘iPS NetMail Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailAutosave-oid NAME ‘sunPortalNetmailAutosave’
DESC ‘iPS NetMail Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailAutopurge-oid NAME
‘sunPortalNetmailAutopurge’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailAutoFolderLoad-oid NAME
‘sunPortalNetmailAutoFolderLoad’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailMultipleReadWindows-oid NAME
‘sunPortalNetmailMultipleReadWindows’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )
```

Code Example I-2 NetMail Schema *(Continued)*

```
attributeTypes: ( sunPortalNetmailSortKey-oid NAME ‘sunPortalNetmailSortKey’
DESC ‘iPS NetMail Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailViewKey-oid NAME ‘sunPortalNetmailViewKey’
DESC ‘iPS NetMail Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailComposeWinBounds-oid NAME
‘sunPortalNetmailComposeWinBounds’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailFolderWinBounds-oid NAME
‘sunPortalNetmailFolderWinBounds’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailReadWinBounds-oid NAME
‘sunPortalNetmailReadWinBounds’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailGridHeight-oid NAME
‘sunPortalNetmailGridHeight’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailGridColWidths-oid NAME
‘sunPortalNetmailGridColWidths’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailTextColor-oid NAME
‘sunPortalNetmailTextColor’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailBackgroundColor-oid NAME
‘sunPortalNetmailBackgroundColor’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailTextSize-oid NAME ‘sunPortalNetmailTextSize’
DESC ‘iPS NetMail Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailTextStyle-oid NAME
‘sunPortalNetmailTextStyle’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )
```

Code Example I-2 NetMail Schema *(Continued)*

```
attributeTypes: ( sunPortalNetmailHeadersPerPage-oid NAME
‘sunPortalNetmailHeadersPerPage’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailNewestFirst-oid NAME
‘sunPortalNetmailNewestFirst’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailNoPrefsList-oid NAME
‘sunPortalNetmailNoPrefsList’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailLDAPServers-oid NAME
‘sunPortalNetmailLDAPServers’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailIMAPUserid-oid NAME
‘sunPortalNetmailIMAPUserid’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailIMAPPassword-oid NAME
‘sunPortalNetmailIMAPPassword’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailReplyToAddress-oid NAME
‘sunPortalNetmailReplyToAddress’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailSignature-oid NAME
‘sunPortalNetmailSignature’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailFavoriteFolders-oid NAME
‘sunPortalNetmailFavoriteFolders’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailPersonalAddressBook-oid NAME
‘sunPortalNetmailPersonalAddressBook’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalNetmailExecutable-oid NAME
‘sunPortalNetmailExecutable’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )
```

Code Example I-2 NetMail Schema *(Continued)*

```
attributeTypes: ( sunPortalNetmailLogMessages-oid NAME
‘sunPortalNetmailLogMessages’ DESC ‘iPS NetMail Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

dn: cn=schema
changetype:modify
add:objectClasses
objectClasses: ( sunPortalNetmailPerson-oid NAME ‘sunPortalNetmailPerson’ DESC
‘Sun ONE NetMail Service’ SUP top AUXILIARY MAY (
sunPortalNetmailIMAPServerName $ sunPortalNetmailSMTPServerName $
sunPortalNetmailDefaultMailDomain $ sunPortalNetmailRootFolder $
sunPortalNetmailSentMessagesFolder $ sunPortalNetmailReplyWithAuthor $
sunPortalNetmailReplyWithDate $ sunPortalNetmailReplyWithBody $
sunPortalNetmailIndentPrefix $ sunPortalNetmailAddSignature $
sunPortalNetmailInitialHeaders $ sunPortalNetmailInactivityInterval $
sunPortalNetmailMaxAttachLen $ sunPortalNetmailAutoload $
sunPortalNetmailAutosave $ sunPortalNetmailAutopurge $
sunPortalNetmailAutoFolderLoad $ sunPortalNetmailMultipleReadWindows $
sunPortalNetmailSortKey $ sunPortalNetmailViewKey $
sunPortalNetmailComposeWinBounds $ sunPortalNetmailFolderWinBounds $
sunPortalNetmailReadWinBounds $ sunPortalNetmailGridHeight $
sunPortalNetmailGridColWidths $ sunPortalNetmailTextColor $
sunPortalNetmailBackgroundColor $ sunPortalNetmailTextSize $
sunPortalNetmailTextStyle $ sunPortalNetmailHeadersPerPage $
sunPortalNetmailNewestFirst $ sunPortalNetmailNoPrefsList $
sunPortalNetmailLDAPServers $ sunPortalNetmailIMAPUserid $
sunPortalNetmailIMAPPassword $ sunPortalNetmailReplyToAddress $
sunPortalNetmailSignature $ sunPortalNetmailFavoriteFolders $
sunPortalNetmailPersonalAddressBook $ sunPortalNetmailExecutable $
sunPortalNetmailLogMessages) X-ORIGIN ‘Sun ONE Access Management’ )
```

Sun ONE Portal Server Search Schema

The `psSearch.ldif` file on a default installation is in the `/opt/SUNWps/export` directory.

Code Example I-3 Search Schema

```
#
# Copyright 2001 Sun Microsystems, Inc.  All rights reserved.
# PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license terms.
#
#
```

Code Example I-3 Search Schema *(Continued)*

```
# Sun ONE Portal Server (iPS) Search Service Schema
# Last Modified April 2002
#

dn: cn=schema
changetype:modify
add:attributeTypes

attributeTypes: ( sunPortalSearchInstances-oid NAME ‘sunPortalSearchInstances’
DESC ‘iPS Search Attribute’ SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun
ONE Access Management’ )

attributeTypes: ( sunPortalSearchSelectedInstances-oid NAME
‘sunPortalSearchSelectedInstances’ DESC ‘iPS Desktop Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalSearchExecutable-oid NAME
‘sunPortalSearchExecutable’ DESC ‘iPS Search Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

attributeTypes: ( sunPortalSearchAdminExecutable-oid NAME
‘sunPortalSearchAdminExecutable’ DESC ‘iPS Search Attribute’ SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN ‘Sun ONE Access Management’ )

dn: cn=schema
changetype:modify
add:objectClasses
objectClasses: ( sunPortalSearchPerson-oid NAME ‘sunPortalSearchPerson’ DESC
‘Sun ONE Search Service’ SUP top AUXILIARY MAY ( sunPortalSearchInstances $
sunPortalSearchExecutable $ sunPortalSearchAdminExecutable $
sunPortalSearchSelectedInstances) X-ORIGIN ‘Sun ONE Access Management’ )
```

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X