

Administration Guide

Sun™ ONE Identity Server

Version 6.0

December 2002
816-6686-10

Copyright © 2002 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, and the Sun logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Federal Acquisitions: Commercial Software -- Government Users Subject to Standard License Terms and Conditions. The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of the Sun Microsystems, Inc. and its licensors, if any.

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

Copyright © 2002 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, le Sun logo, et iPlanet sont des marques déposées ou des marques déposées enregistrées de Sun Microsystems, Inc. aux États-Unis et d'autres pays. Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite préalable de Sun Microsystems, Inc., le cas échéant, de ses bailleurs de licence.

CETTE DOCUMENTATION EST FOURNIE "EN L'ÉTAT", ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Some preexisting portions Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

About This Guide	17
About Identity Server 6.0	17
What You Are Expected to Know	17
Sun ONE Identity Server Documentation Set	18
Documentation Conventions Used in This Guide	19
Typographic Conventions	19
Terminology	19
Related Information	20
Part 1 Identity Server Console Guide	23
Chapter 1 Product Overview	25
Sun ONE Identity Server	25
Features of Identity Server	26
Service Configuration	26
Policy Management	26
SAML	26
Federation Management	26
Authentication	27
Single Sign-On	27
URL Policy Agents	27
Identity Management	27
Identity Server Console	28
Installing Identity Server	28
The Identity Server Console	29
Location Pane	29

Navigation Pane	30
Data Pane	30
Chapter 2 Identity Management	31
The Identity Management Interface	31
Identity Management View	31
User Profile View	32
Managing Identity Server Objects	33
Properties Function	33
Organizations	34
Create an Organization	34
Delete an Organization	35
Add an Organization to a Policy	35
Groups	36
Create a Managed Group	36
Delete a Managed Group	37
Add a Group to a Policy	37
Users	37
Create a User	37
Add a User to Services, Roles and Groups	38
Delete a User	38
Add a User to a Policy	38
Services	39
Register a Service	39
Create a Template for a Service	39
Unregister a Service	40
Roles	40
Create a Role	41
Delete a Role	42
Add Users to a Role	42
Remove Users from a Role	43
Add a Role to a Policy	43
Role Properties View	43
Customize Service Access	44
Customize Attribute Access	45
Policies	46
Containers	46
Create a Container	47
Delete a Container	47
People Containers	47
Create a People Container	48
Delete a People Container	48
Group Containers	48

Create a Group Container	49
Delete a Group Container	49
Chapter 3 Service Configuration	51
Definition of a Service	51
Identity Server Services Defined	52
Administration	52
Authentication	52
Anonymous	52
Certificate-based	52
Core	53
LDAP	53
Membership (Self-Registration)	53
NT	53
RADIUS	53
SafeWord	53
Unix	53
Authentication Configuration	54
Client Detection	54
Logging	54
Naming	54
Platform	54
Policy Configuration	54
SAML	55
Session	55
User	55
Identity Server Security Service	55
Attribute Types	56
Dynamic Attributes	56
User Attributes	56
Organization Attributes	57
Global Attributes	57
Policy Attributes	57
Service Configuration Interface	57
Chapter 4 Current Sessions	59
The Current Sessions Interface	59
Session Management Pane	60
Session Information Window	60
Terminating a Session	61

Chapter 5 Federation Management	63
Liberty Alliance and Federated Identity	63
Federation Management Concepts	64
The Liberty Alliance Project	66
Federation Management Process	67
Managing Authentication Domains and Providers	69
Authentication Domains	69
Creating An Authentication Domain	69
Modifying An Authentication Domain	70
Deleting An Authentication Domain	70
Providers	71
Creating Remote Providers	71
Modifying Remote Providers	73
Creating Hosted Providers	74
Modifying Hosted Providers	76
Deleting Providers	81
 Chapter 6 Policy Management	 83
What is a Policy?	83
Policy Types	84
Normal Policy	84
Referral Policy	85
Policy Management	85
Registering Policy Configuration Services	86
Creating Policies	87
Modifying Policies	88
Modify a Normal Policy	88
Modify a Referral Policy	93
Creating Policies for Peer and Suborganizations	95
 Chapter 7 Authentication Options	 97
Core Authentication	98
To Register and Enable the Core Service	98
Anonymous Authentication	99
To Register and Enable Anonymous Authentication	99
Logging In Using Anonymous Authentication	100
Certificate-based Authentication	100
To Register and Enable Certificate-based Authentication	101
Logging In Using Certificate-based Authentication	102
LDAP Directory Authentication	102
To Register and Enable LDAP Authentication	102
Logging In Using LDAP Authentication	103
Enabling LDAP Authentication Failover	104

Membership Authentication	104
To Register and Enable Membership Authentication	104
Logging In Using Membership Authentication	105
NT Authentication	105
To Register and Enable NT Authentication	106
Logging In Using NT Authentication	107
RADIUS Server Authentication	107
To Register and Enable RADIUS Authentication	107
Logging In Using RADIUS Authentication	108
SafeWord Authentication	109
To Register and Enable SafeWord Authentication	109
Logging In Using SafeWord Authentication	110
Unix Authentication	110
To Register and Enable Unix Authentication	110
Logging In Using Unix Authentication	112
Authentication Configuration	112
Authentication Configuration User Interface	112
Authentication Configuration for Organizations	116
Authentication Configuration for Roles	117
Authentication Configuration for Services	118
Authentication Configuration for Users	118
Authentication By Authentication Level	119
Authentication By Module	120

Part 2 Command Line Reference Guide 121

Chapter 8 The amadmin Command Line Tool	123
The amadmin Command Line Executable	123
The amadmin Syntax	124
amadmin Options	125
Creating Policies with amadmin	127
Chapter 9 The amserver Command Line Tool	129
The amserver Command Line Executable	129
amserver Syntax	129
amserver Commands for Solaris	129
amserver Commands for Windows 2000	130
Using amserver for Multi-Server Installer Administration	130
Chapter 10 The ampassword Command Line Tool	135
The ampassword Command Line Executable	135

The ampassword Syntax	135
ampassword Options	136
Running ampassword on SSL	136

Chapter 11 The am2bak Command Line Tool	139
The am2bak Command Line Executable	139
The am2bak Syntax	139
am2bak Options	140
Backup Procedure	141

Chapter 12 The bak2am Command Line Tool	143
The bak2am Command Line Executable	143
The bak2am Syntax	143
bak2am Options	144

Chapter 13 The VerifyArchive Command Line Tool	145
The VerifyArchive Command Line Executable	145
VerifyArchive Syntax	145
VerifyArchive Options	146

Part 3 Attribute Reference Guide	147
---	------------

Chapter 14 Administration Attributes	149
Global Attributes	149
Enable Federation Management	150
Enable User Management	150
Show People Containers	150
Display Containers In Menu	150
Show Group Containers	151
Managed Group Type	151
Default Role Permissions (ACIs)	152
No Permissions	152
Organization Admin	152
Organization Help Desk Admin	152
Organization Policy Admin	152
Domain Component Tree Enabled	153
Admin Groups Enabled	154
Compliance User Deletion Enabled	154
Dynamic Admin Roles ACIs	155
Container Help Desk Admin	155

Organization Help Desk Admin	155
Container Admin	155
Organization Policy Admin	155
People Container Admin	156
Group Admin	156
Top-level Admin	156
Organization Admin	156
User Profile Service Classes	157
Organization Attributes	157
Groups Default People Container	158
Groups People Container List	158
User Profile Display Class	158
Display User's Roles	159
Display User's Groups	159
User Group Self Subscription	159
User Profile Display Options	159
User Creation Default Roles	160
View Menu Entries	160
Maximum Results Returned From Search	160
Timeout For Search (sec.)	160
JSP Directory Name	160
Online Help Documents	161
Required Services	161
User Search Key	161
User Search Return Attribute	161
User Creation Notification List	161
User Deletion Notification List	162
User Modification Notification List	162
Maximum Entries Per Page	163
Chapter 15 Anonymous Authentication Attributes	165
Authentication Level	165
Valid Anonymous User List	166
Default Anonymous User Name	166
Chapter 16 Certificate Authentication Attributes	167
Match Certificate in LDAP	168
Attribute In Subject DN To Use To Search LDAP	168
Match Certificate to CRL	168
Attribute In Issuer DN To Use To Search CRL	168
Enable OCSP Validation	169
LDAP Server and Port	169

LDAP Start Search DN	169
LDAP Server Principal User	170
LDAP Server Principal Password	170
LDAP Attribute for Profile ID	170
SSL On For LDAP Access	170
Field in Cert to Use to Access User Profile	170
Other Field In Cert To Use To Access User Profile	171
Authentication Level	171
 Chapter 17 Core Authentication Attributes	 173
Global Attributes	173
Pluggable Auth Module Classes	174
Supported Auth Modules for Clients	174
LDAP Connection Pool Size	174
LDAP Connection Default Pool Size	174
Organization Attributes	175
Organization Authentication Modules	176
User Profile	176
Admin Authenticator	177
User Profile Dynamic Creation Default Roles	177
Persistent Cookie Mode	177
Persistent Cookie Max Time (seconds)	177
People Container For All Users	178
Alias Search Attribute Name	178
Default Auth Level	179
User Naming Attribute	179
Default Auth Locale	179
Organization Authentication Configuration	181
Login Failure Lockout Mode	181
Login Failure Lockout Count	181
Login Failure Lockout Interval (minutes)	182
Email Address to Send Lockout Notification	182
Warn User After N Failure	182
Login Failure Lockout Duration (minutes)	182
Lockout Attribute Name	182
Lockout Attribute Value	183
Default Success Login URL	183
Default Failure Login URL	183
Authentication PostProcessing Class	183
User Name Generator Mode	183
Pluggable User Name Generator Class	183

Chapter 18 LDAP Authentication Attributes	185
Primary LDAP Server and Port	186
Secondary LDAP Server and Port	186
DN to Start User Search	186
DN for Root User bind	186
Password for Root User Bind	187
Password For Root User Bind (Confirm)	187
User Naming Attribute	187
User Entry Search Attributes	187
User Search Filter	187
Search Scope	188
Enable SSL to LDAP Server	188
Return User DN To Auth	188
Authentication Level	188
 Chapter 19 Membership Authentication Attributes	 191
Minimum Password Length	192
Default User Roles	192
User Status After Registration	192
Primary LDAP Server and Port	192
Secondary LDAP Server and Port	193
DN to Start User Search	193
DN for Root User bind	193
Password for Root User Bind	193
Password for Root User Bind (Confirm)	193
User Naming Attribute	194
User Entry Search Attributes	194
User Search Filter	194
Search Scope	194
Enable SSL to LDAP Server	194
Return User DN To Auth	195
Authentication Level	195
 Chapter 20 NT Authentication Attributes	 197
NT Authentication Domain	197
NT Authentication Host	197
NT Module Authentication Level	198
 Chapter 21 RADIUS Authentication Attributes	 199
RADIUS Server 1	199
RADIUS Server 2	200
RADIUS Shared Secret	200

RADIUS Shared Secret (Confirm)	200
RADIUS Server's Port	200
Authentication Level	200
Timeout (Seconds)	201
Chapter 22 SafeWord Authentication Attributes	203
SafeWord Server Specification	203
SafeWord System Name	204
SafeWord Server Verification Files Path	204
SafeWord Logging Level	204
SafeWord Log Path	204
SafeWord Module Authentication Level	205
Chapter 23 Unix Authentication Attributes	207
Global Attributes	207
Unix Helper Configuration Port	208
Unix Helper Authentication Port	208
Unix Helper Timeout (Minutes)	208
Unix Helper Threads	208
Organization Attribute	208
Unix Module Authentication Level	209
Chapter 24 Authentication Configuration Attributes	211
Authentication Configuration	211
Login Success URL	213
Login Failure URL	213
Authentication Post Processing Class	213
Conflict Resolution Level	213
Chapter 25 Client Detection Attributes	215
Client Types	215
Default Client Type	215
Client Detection Class	215
Client Detection Enabled	216
Chapter 26 Logging Attributes	217
Max Log Size	218
Number of History Files	218
Log Location	218
Logging Type	218
Database User Name	218

Database User Password	219
Database User Password (Confirm)	219
Database Driver Name	219
Configurable Log Fields	219
Log Verification Time	219
Log Signature Time	219
Secure Logging	220
Maximum Number of Records	220
Remote Buffer Size	220
Number Of Files Per Archive	220
Chapter 27 Naming Attributes	221
Profile Service URL	221
Session Service URL	222
Logging Service URL	222
Policy Service URL	222
Auth Service URL	222
SAML Web Profile/Artifact Service URL	223
SAML SOAP Service URL	223
SAML Web Profile/POST Service URL	223
SAML Assertion Manager Service URL	223
Chapter 28 Platform Attributes	225
Server List	225
Platform Locale	226
Cookie Domains	226
Login Service URL	226
Logout Service URL	227
Available Locales	227
Client Char Sets	227
Chapter 29 Policy Configuration Attributes	229
Global Attribute	229
Resource Comparator	230
Organization Attributes	230
LDAP Server and Port	231
LDAP Base DN	231
LDAP Bind DN	231
LDAP Bind Password	232
LDAP Bind Password (Confirm)	232
LDAP Org Search Filter	232
LDAP Org Search Scope	232

LDAP Groups Search Filter	232
LDAP Groups Search Scope	232
LDAP Users Search Filter	233
LDAP Users Search Scope	233
LDAP Roles Search Filter	233
LDAP Roles Search Scope	233
LDAP Organization Search Attribute	233
LDAP Groups Search Attribute	234
LDAP Users Search Attribute	234
LDAP Roles Search Attribute	234
Maximum Results Returned From Search	234
Timeout For Search (seconds)	234
LDAP SSL Enabled	234
LDAP Connection Pool Minimal Size	235
LDAP Connection Pool Maximum Size	235
Selected Policy Subjects	235
Selected Policy Conditions	235
Selected Policy Referrals	235
Subjects Result Time To Live	235
 Chapter 30 SAML Attributes	 237
Site ID And Site Issuer Name	238
Sign Request	238
Sign Response	238
Sign Assertion	238
Artifact Name	238
Target Specifier	238
Artifact Timeout (seconds)	239
Assertion Skew Factor For notBefore Time	239
Assertion Timeout (seconds)	239
Trusted Partner Sites	239
POST To Target URLs	243
 Chapter 31 Session Attributes	 245
Max Session Time (Minutes)	245
Max Idle Time (Minutes)	246
Max Caching Time (Minutes)	246
 Chapter 32 User Attributes	 247
User Attributes	247
User Preferred Language	248
User Preferred Timezone	248

Inherited Locale	248
Admin DN Starting View	248
Default User Status	249
User Profile Attributes	249
First Name	249
Last Name	250
Full Name	250
Password	250
Password (Confirm)	250
Email Address	250
Employee Number	250
Telephone Number	250
Home Address	250
User Status	251
Account Expiration Date	251
User Authentication Configuration	251
User Alias List	252
Preferred Locale	252
Success URL	252
Failure URL	252
Unique User IDs	252
 Chapter 33 Identity Server Security Service Attributes	255
Enrollment URL	255
Inherited Country	256
 Appendix A Configuring Identity Server in SSL Mode	257
 Index	261

About This Guide

The *Sun One™ Identity Server Administration Guide* offers information on how to manage the Sun™ One Identity Server through the User and Command Line Interface. This preface contains the following sections:

- About Identity Server 6.0
- What You Are Expected to Know
- Sun ONE Identity Server Documentation Set
- Documentation Conventions Used in This Guide
- Related Information

About Identity Server 6.0

Sun ONE Identity Server, prior to the 6.0 release, was known as iPlanet Directory Server Access Management Edition (DSAME). The product was renamed shortly before the launch of version 5.1.

Identity Server is designed to help organizations manage identities and enforce secure access to their network services and web-based resources. It contains a number of services towards this end as well as the Sun ONE Directory Server as a data store. For the latest information about new features and enhancements in this release of Identity Server, please see the online release notes at

<http://www.sun.com/software/> or the *Sun ONE Identity Server Product Brief*.

What You Are Expected to Know

This Administration Guide is intended for use by IT administrators and custom software developers who manage identities and access to their web resources using Sun ONE servers and software. It is recommended that administrators understand directory server technologies, including Lightweight Directory Access Protocol

(LDAP), and have some experience with Java™, Java Server Pages, HyperText Markup Language (HTML) and eXtensible Markup Language (XML). Particularly, they should also be familiar with Sun ONE Directory Server and the documentation provided with that product.

Sun ONE Identity Server Documentation Set

The Sun ONE Identity Server documentation set contains the following titles:

- *Product Brief* provides an overview of the Sun ONE Identity Server application and its features and functions.
- *Installation Guide* provides details on how to install and deploy the Identity Server on Solaris™, Linux and Windows® 2000 systems.
- *Administration Guide* (this guide) describes how to use the Identity Server console as well as manage user and service data via the command line.
- *Programmer's Guide* documents how to customize an Identity Server system specific to your organization. It also includes instructions on how to augment the application with new services using the public APIs.
- *Policy Agent Guide* documents how to install and configure an Identity Server policy agent on a remote server. It also includes troubleshooting and information specific to each agent.
- *Getting Started Guide* documents how to use various features of the Identity Server product to set up a simple organization with identities, policies and roles.
- The *Release Notes* file gathers an assortment of last-minute information, including a description of what is new in this release, known problems and limitations, installation notes, and how to report problems.

NOTE

Be sure to check the Identity Server documentation web site at <http://docs.sun.com/db/prod/slidsrv#hic> for updates to the release notes and for revisions to the guides. Updated documents will be marked with a revision date.

Documentation Conventions Used in This Guide

In the Identity Server documentation, certain typographic conventions and terminology are used. These conventions are described in the following sections.

Typographic Conventions

This book uses the following typographic conventions:

- *Italic type* is used within text for book titles, new terminology, emphasis, and to format text that can be substituted by actual values, such as a placeholder in a path name.
- `Monospace font` is used for sample code and code listings, APIs and programming language elements (such as function names and class names). It is also used for filenames, pathnames, directory names, HTML tags, URLs, and any text that must be typed on the screen.
- `<sample text>` is used to represent a variable placeholder. When this convention is used in a directory path or URL, the text and surrounding carats should be replaced with deployment-specific information.

NOTE	Notes, Cautions and Tips highlight important conditions or limitations. Be sure to read this information.
-------------	---

Terminology

Below is a list of general terms used in the Sun One™ Identity Server documentation set:

- `<identity_server_root>` is a variable placeholder for the path to the home directory where Identity Server is installed.
- `<directory_server_root>` is a variable placeholder for the path to the home directory where Sun ONE Directory Server is installed.

Related Information

In addition to the documentation provided with Identity Server, there are several other sets of documentation that might be helpful. This section lists these and additional sources of information.

iPlanet Directory Server Documentation

iPlanet Directory Server 5.1 documentation can be found at

http://docs.sun.com/db/coll/S1_ipDirectoryServer_51.

iPlanet/Sun ONE Web Server Documentation

iPlanet/Sun ONE Web Server documentation can be found at

http://docs.sun.com/db/coll/S1_ipwebsrvree60_en.

Sun ONE Certificate Server Documentation

Sun ONE Certificate Server documentation can be found at

http://docs.sun.com/db/coll/S1_slCertificateServer_47.

iPlanet Proxy Server Documentation

iPlanet Proxy Server documentation can be found at

http://docs.sun.com/db/coll/S1_ipwebproxysrvr36.

Other iPlanet Product Documentation

Documentation for all other Sun ONE servers and technologies can be found at

<http://docs.sun.com/db/prod/sunone>.

Download Center

Links to download any of Sun's Sun ONE/iPlanet software are at

<http://www.sun.com/software/download/>.

Sun ONE Technical Support

Technical Support can be contacted through

<http://www.sun.com/service/support/software/iplanet/index.html>.

Professional Services Information

Professional Service can be contacted through

<http://www.sun.com/service/sunps/iplanet/>.

Sun Enterprise Services for Solaris Patches And Support

Solaris patches and support can be obtained through

<http://www.sun.com/service/>

Developer Information

Information on Sun One™ Identity Server, LDAP, the Sun ONE Directory Server, and associated technologies can also be found at

<http://developer.iplanet.com/tech/directory/>

Related Information

Identity Server Console Guide

This is part one of the *Sun™ One Identity Server Administration Guide*. It discusses the Identity Server graphical user interface and how to navigate through it. This section contains the following chapters:

- Product Overview
- Identity Management
- Service Configuration
- Current Sessions
- Federation Management
- Policy Management
- Authentication Options

Chapter 1

Product Overview

This chapter provides an overview of the features of Sun ONE Identity Server. It contains the following sections:

- Sun ONE Identity Server
- Features of Identity Server
- Installing Identity Server
- The Identity Server Console

Sun ONE Identity Server

Sun ONE Identity Server technology is part of the Sun Open Net Environment (Sun ONE) Platform for Network Identity. Identity Server is a set of tools used to leverage the management and security potential of Sun ONE Directory Server, the Lightweight Directory Access Protocol-based (LDAP) data store. Identity Server integrates Directory Server with a user authentication and single sign-on function which increases data security. It also allows administrators to initiate user entry management based on *roles*, an entry grouping mechanism which appears as an attribute in a user entry. Lastly, developers can define and manage the configuration parameters of a multitude of default and custom-made services. All three of these functions are accessed through a customizable graphical user interface, the web-based Identity Server console.

Features of Identity Server

Identity Server is built on top of an installation of Directory Server. The concept is to give directory administrators a more consistent and intuitive interface to work from as well as features used to extend the capabilities of Directory Server.

Service Configuration

Configuration parameters for default and custom-made business services can be specified with Identity Server service management component. Using XML and the DTD defined within the Identity Server framework, service developers can define the parameters of a corporate service (such as a mail service, a billing service or a logging service) and manage the service's parameters or *attributes*. In addition, Identity Server allows service administrators to define the value of these attributes.

Policy Management

Identity Server also provides a method to define, modify or remove the rules that control access to business resources. Collectively, these rules are referred to as *policy*.

SAML

Identity Server uses the Security Assertion Markup Language (SAML) for exchanging security information. SAML defines an eXtensible Markup Language (XML) framework to achieve inter-operability across different vendor platforms that provide this type of information. The SAML framework is described in the *Sun One Identity Server Programmer's Guide*.

Federation Management

Identity Server has integrated a Federation Management module to make use of the open standards for federated network identity being developed by the Liberty Alliance Project.

Authentication

Identity Server 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 Identity Server enterprise. Before being allowed access to a Identity Server session, a user must pass through authentication successfully.

Single Sign-On

Once the user is authenticated, Identity Server's 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.

URL Policy Agents

The URL Policy Agents are installed onto a Web Server. It is a specific instance of the Identity Server policy component. This agent serves as an additional authentication step when a user sends a request for a web resource that lives on the protected web server. This authentication is in addition to any user authentication check which the resource must do. The agent protects the web server; the resource is protected by the authentication plug-in.

Identity Management

The Identity Management component allows for the creation and management of identity-related objects. User, role, group, policies, organization, suborganization and container objects can be defined, modified or deleted using either the Identity Server console or the command line interface. The console has default administrators with varying degrees of privileges used to create and manage the organizations, groups, containers, users, services, and policies. (Additional administrators can be created based on roles.) The administrators are defined within the Directory Server when installed with Identity Server. These administrators are:

- Top-level Administrator with read and write access to all entries within the Identity Server enterprise.

- top-level Help Desk Administrator with read access to all entries within the Identity Server enterprise.
- Organization Administrator with read and write access to all entries within its organization.
- Organization Help Desk Administrator with read access to all entries within its organization.
- Container Administrator with read and write access to all Group Administrator with read and write access to all members of its group.

Identity Server Console

This HTML-based console provides a graphical user interface for businesses to manage the Identity Server enterprise.

Installing Identity Server

The goal of Identity Server is to provide an interface for managing user objects, policies and services for organizations using Directory Server. When the Identity Server installer is run, an instance of Directory Server may be installed, but Identity Server typically runs with a remote instance of Directory Server. This instance serves as the data store for Identity Server. In addition, three modules are integrated into the Directory Server: the Policy module, the Management module, and the URL Policy Agent module.

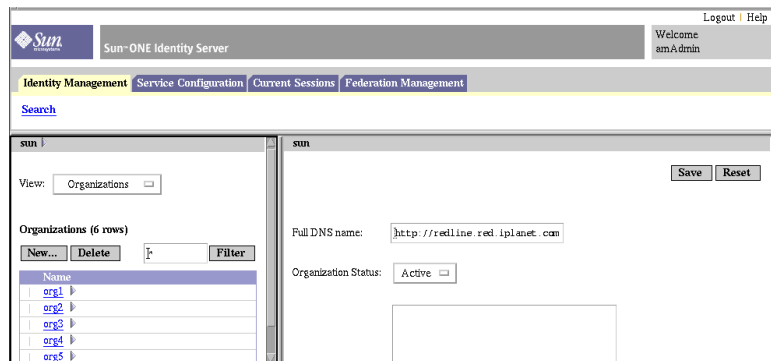
The policy service consists of the Authentication, Naming, Session, Policy, and Logging services. The Management module provides policy, user and service management functions through either the Identity Server console or the command line interface. The Policy Agent validates a user's SSO and web resource access. All of these functions can be accessed through a web browser using the Identity Server console.

NOTE The Identity Server installer can install the three Identity Server modules to expand upon the capabilities of Directory Server. For information on how this is done, please see the *SunTM One Identity Server Installation and Configuration Guide*.

The Identity Server Console

The Identity Server console is divided into three sections: the location pane, the navigation pane and the data pane. By using all three panes, the administrator is able to navigate the directory, perform user and service configurations and create policies.

Figure 1-1 The Identity Server Console



Location Pane

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

- Identity Management module - allows for the creation and management of identity-related objects.
- Service Configuration module - allows for the configuration of Identity Server's default services.
- Current Sessions module - allows administrators to view current session information, as well as terminating any session.
- Federation Management module - allows for the utilization of the open standards for federated network identity being developed by the Liberty Alliance Project.

The *Location* field provides a trail to the administrator's position in the directory tree. This path is used for navigational purposes.

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

The *Search* link displays an interface that allows the user to search for entries of a specific Identity Server object type. Use the pull-down menu to select the object type and enter the search string. The Results are returned in the search table. Wildcards are accepted.

The *Help* link opens a browser window containing information on Identity Management, Current Sessions, Federation Management and Part 3 of this documentation, the Attribute Reference Guide.

The *Logout* link allows the user to log out of the Identity Server.

Navigation Pane

The Navigation pane is the left portion of the Identity Server console. The *Directory Object* portion (within the grey box) displays the name of the directory object that is currently open and its *Properties* link. (Most objects displayed in the Navigation pane will have a corresponding *Properties* link. Selecting this link will render the object's attributes in the Data frame to the right.) The View 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. This is where all object attributes and their values are displayed and configured and where entries are selected for their respective group, role or organization.

TIP You can select or deselect all of the items in a list by clicking the Select All, or Deselect All icons.



Chapter 2

Identity Management

This chapter describes the user management features of Sun ONE Identity Server. The Identity Management module interface provides a way to view, manage and configure all Identity Server objects and identities. This chapter contains the following sections:

- The Identity Management Interface
- Managing Identity Server Objects

The Identity Management Interface

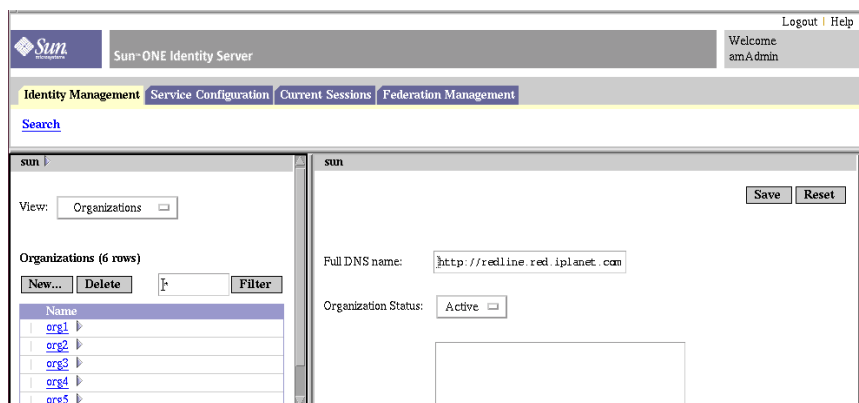
There are two basic views of the Identity Server graphical user interface. Depending on the roles of the user logging in, they might gain access to the Identity Management View or the User Profile View.

Identity Management View

When a user with an administrative role authenticates to the Identity Server, their default view is the Identity Management view. In this view the administrator can perform administrative tasks. Depending on the role of the administrator, this can include, but is not limited to, creating objects (users, organizations, policies, and so forth), and configuring services.

NOTE In order to view the Identity Management objects, User Management must be enabled (it is enabled by default). If user Management is not enabled, you can enable it by selecting the Service Configuration module, clicking the Administration service, and selecting the Enable User Management attribute. Click Save to save the changes.

Figure 2-1 Identity Management View with Organization Properties Displayed



User Profile View

When a user who has not been assigned an administrative role authenticates to the Identity Server, the default view is their own User Profile. In this view the user can modify the values of the attributes particular to their personal profile. This can include, but is not limited to, name, home address and password. The attributes displayed in the User Profile View can be extended. For more information on adding customized attributes for objects and identities, see the *Sun One Identity Server Programmer's Guide*.

Figure 2-2 User Profile View

The screenshot displays the 'User Profile View' in the Sun ONE Identity Server interface. The top navigation bar includes the Sun logo, the text 'Sun ONE Identity Server', and links for 'Logout' and 'Help'. A secondary bar shows a 'Welcome' message to 'User One'. The main content area features a form with the following fields and values:

- First Name:** User
- Last Name:** One
- Full Name:** user1
- Password:** (masked with asterisks)
- Password (confirm):** (masked with asterisks)
- Email Address:** user1@user1.com
- Employee Number:** 1
- Telephone Number:** 555-555-1111

At the top of the form, there are 'Save' and 'Reset' buttons. The interface is presented in a standard web browser window.

Managing Identity Server Objects

The User Management interface contains all the components needed to view and manage the Identity Server objects (organizations, groups, users, services, roles and policies). This section explains the object types and details on how to configure them.

Properties Function

To view or modify an entry's properties, click the Properties arrow next to the object's name. Its attributes and corresponding values are displayed in the data pane. Different objects display different properties.

- Organizations properties allow status modification between active and inactive, full DNS name, DNS alias name, and a list of unique attributes.
- The Groups properties contains two views, the General view and the User view. The General view allows or disallows users to self-subscribe to the group. In the User view, if the group is static, the administrator can add and remove users. If the group is dynamic, the administrator can only modify the filter to control which users are members of the group. For more information on static and dynamic groups, see "Create a Managed Group," on page 36.

- User properties include, but are not limited to, basic user information such as first name, last name, home address, telephone number and password.
- Service properties include any of the attributes listed in Part 3, “Attribute Reference Guide” depending on the service.
- Role properties include role and permission descriptions and the services registered to the role.
- Policy properties allow you to modify all aspects of the created policies, including rules, referrals, subjects, and so forth. See “Policy Management,” on page 83 for more information.

See the *Sun One Identity Server Programmer’s Guide* for information on how to extend an entry’s properties.

Organizations

This object represents the top-level of a hierarchical structure used by an enterprise to manage its departments and resources. Upon installation, Identity Server dynamically creates a top-level organization (defined during installation) to manage the Identity Server enterprise configurations. Additional organizations can be created after installation to manage separate enterprises. All created organizations fall beneath the top-level organization.

Create an Organization

1. Choose Organizations from the View menu in the Identity Management module.
2. Click New in the navigation pane.

The New Organization template displays in the data pane.

3. Enter a value for the name of the Organization in the New Organization template.
4. Choose a status of `active` or `inactive`.

The default is `active`. This can be changed at any time during the life of the organization by selecting the Properties icon. Choosing `inactive` disables log in to the organization.

5. Enter the values, if desired, for the optional fields. The optional fields are:
 - Full DNS Name - Enter the full Domain Name System (DNS) name for the organization, if it has one.

- DNS Alias Name - Allows you to add alias names for the DNS name for the organization.
 - Unique Attribute List - Allows you to add a list of unique attributes for users in the organization. For example, if you add a unique attribute specifying an email address, you would not be able to create two users with the same email address.
6. Enter the DNS alias name, and click Add to add it to the List of DNS Alias Names.
 7. Enter a value for the unique attribute and click Add to add it to the Unique Attribute list.

This field contains a list of attributes defined in the Sun ONE Directory Server schema.
 8. Click Create.

The new organization displays in the navigation pane.

Delete an Organization

1. Choose Organizations from the View menu in Identity Management.

All created organizations are displayed. To display specific organizations, enter a search string and click Filter.
2. Select the checkbox next to the name of the Organization to be deleted.
3. Click Delete.

NOTE There is no warning message when performing a delete. All entries within the organization will be deleted.

Add an Organization to a Policy

Identity Server objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Modifying Policies," on page 88.

Groups

A group represents a collection of users with a common function, feature or interest. Typically, this grouping has no privileges associated with it. Groups can exist at two levels, within an organization and within other managed groups as a sub group. Users can be added to Managed Groups either statically or dynamically (filtered).

Membership By Subscription. When you specify group membership by subscription, a static group is created based on the Managed Group Type you specify. If the Managed Group Type value is `static`, group members are added to a group entry using the `groupOfNames` or `groupOfUniqueNames` object class. If the Managed Group Type value is `dynamic`, a specific LDAP filter is used to search and return only user entries that contain the `memberof` attribute. For more information, see “Managed Group Type” on page 151.

Membership By Filter. A filtered group is a dynamic group that is created through the use of an LDAP filter. All entries are funneled through the filter and dynamically assigned to the group. The filter would look for any attribute in an entry and return those that contain the attribute. For example, if you were to create a group based on a building number, you can use the filter to return a list all users containing the building number attribute.

Create a Managed Group

1. Navigate to the organization (or group) where the group will be created.
2. Choose Groups from the View menu.
3. Click New.
4. Select the group type from within the data pane.
 - If a static subscription group is to be created, select Membership By Subscription.
 - a. Enter a name for the group in the Name field. Click Next.
 - b. Select the Users Can Subscribe to this Group attribute to allow users to subscribe to the group themselves.
 - c. Add users to the group by selecting Add from the Member List.
 - d. Enter the search criteria and click Filter. When the user list is returned, select the users you wish to add and click Submit. Adding users to the group is optional. They can be added after the group is created.
 - e. Click Create.

- If a dynamic (LDAP filtered) group is to be created, select **Membership By Filter** and click **Save**.
 - a. Enter a name for the group in the **Name** field. Click **Next**.
 - b. Construct the LDAP search filter.
 - c. The fields used to construct the filter use either an **OR** or **AND** operator. All the fields listed in the UI are used. If a field is left blank it will match all possible entries for that particular attribute.
 - d. Click **Create**.

Delete a Managed Group

1. Navigate to the organization where the group exists.
2. Choose **Groups** from the **View** menu.
3. Select the checkbox next to the name of the group to be deleted.
4. Click **Delete**.

Add a Group to a Policy

Identity Server objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's **Subject** page. Once the subject is defined, the policy will be applied to the object. For more information, see "Modifying Policies," on page 88.

Users

Users represent the identity of a person. Users can be created and deleted, and added or removed from services, roles or groups through the Identity Management module.

Create a User

1. Navigate to the organization where the user should be created.
2. Choose **Users** from the **View** menu.
3. Click **New**.

This displays the **New User** page in the **Data** pane.

4. Enter values for the required attributes and any optional fields.

Information on the user profile attributes can be found in “User Attributes,” on page 247.

5. Click Create.

Add a User to Services, Roles and Groups

1. Navigate to the Organization where the user should be modified.
2. Choose Users from the View menu.
3. In the Navigation pane, select the user you wish to modify and click the Properties arrow.
4. From the View menu in the Data pane, select Services, Roles or Groups.

The User view allows you to modify any attributes defined the User service.

5. Select the service, role, or group that to which you wish to add the user, and click Save.

Delete a User

1. Navigate to the Organization where the user exists.
2. Choose Users from the View menu.
3. Select the checkbox next to the name of the user to be deleted.
4. Click Delete.

Add a User to a Policy

Identity Server objects are added to a policy through the policy’s subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy’s Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see “Modifying Policies,” on page 88.

Services

Activating a service for an organization is a two step process. In the first step you need to register the service with the organization. After a service is registered, a template configured specifically for that organization must be created. For additional information, see Chapter 3, “Service Configuration.” Only top-level administrators can register, unregister and assign services to their own profiles.

NOTE A new service must first be imported into the Identity Server through the command line’s `amadmin`. Information on importing a service’s XML schema can be found in the *Sun One Identity Server Programmer’s Guide*.

Register a Service

1. Navigate to the Organization where you will add services.
Choose Organizations from the View menu in the Identity Management module and select the organization from the navigation pane. The Location path displays the default top-level organization and chosen organization.
2. Choose Services from the View menu.
3. Click Register.
The data pane will display a list of services available to register to this organization.
4. Select the checkbox next to the services to be added.
5. Click Register.

Create a Template for a Service

1. Navigate to the organization or role where the registered service exists.
Choose Organizations from the View menu in the Identity Management Management module and select the organization from the navigation pane.
2. Choose Services from the View menu.
3. Click the properties icon next to the name of the service to be activated.
The data pane displays the message *No Template Available For This Service. Do you want to create it?*

4. Click Create.

A template is created for this service for the parent organization or role. The data pane displays the default attributes and values for this service. Descriptions for the attributes for the default services are described in the “Attribute Reference Guide,” on page 147.

5. Accept or modify the default values and click Save.

Unregister a Service

1. Navigate to the organization where you will remove services.

Choose Organizations from the View menu in Identity Management module and select the organization from the navigation pane.

2. Choose Services from the View menu.

3. Select the checkboxes for the services to remove.

4. Click Unregister.

Roles

Roles are a Directory Server entry mechanism similar to the concept of a *group*. A group has members; a role has members. A role’s members are LDAP entries that are said to *possess* the role. The criteria of the role itself is defined as an LDAP entry with attributes, identified by the Distinguished Name (DN) attribute of the entry. Directory Server has a number of different types of roles but Identity Server can manage only one of them: the managed role.

NOTE

The other Directory Server role types can still be used in a directory deployment; they just can not be managed by Identity Server.

Users can possess one or more roles. For example, a contractor role which has attributes from the Session Service and the URL Policy Agent Service might be created. Thus, when new contractors start, the administrator can assign them this role rather than setting separate attributes in the contractor entry. If the contractor were then to become a full-time employee, the administrator would just re-assign the user a different role.

Identity Server uses roles to apply access control instructions. When first installed, Identity Server configures access control instructions (ACIs) that define administrator permissions. These ACIs are then designated in roles (such as Organization Admin Role and Organization Help Desk Admin Role) which, when assigned to a user, define the user's access permissions.

Users can view their assigned roles only if the Display User's Roles attribute is enabled in the Administration Service. For more information, see "Display User's Roles," on page 159.

Create a Role

1. Navigate to the navigation pane of the Organization where the role will be created.
2. Choose Roles from the View menu.

A set of default roles are created when an organization is configured, and are displayed in the navigation pane.

For descriptions of these roles, see "Dynamic Admin Roles ACIs," on page 155 of the Attribute Reference section.

3. Click New in the navigation pane.

The New Role template appears in the Data pane.

4. Enter a name for the role.
5. Enter a description of the role.
6. Choose the role type from the Type menu.

The role can be either an Administrative role or a Service role. The role type is used by the console to figure out where to start the user in the DIT. An administrative role notifies the console that the possessor of the role has administrative privileges; the service role notifies the console that the possessor is an end user.

7. Choose a default set of permissions to apply to the role from the Access Permission menu.

The permissions provide access to entries within the organization. They are discussed in the section "Default Role Permissions (ACIs)," on page 152. (The default permissions shown are in no particular order.)

Generally, the No Permissions ACI is assigned to Service roles, while Administrative roles are assigned any of the default ACIs.

8. Click Create.

The created role is displayed in the Navigation pane and status information about the role is displayed in the Data pane. You can add or remove services to the Role by clicking the Edit link in the Services display. For more information, see “Role Properties View,” on page 43.

Delete a Role

1. Navigate to the organization that contains the role for deletion.

Choose Organizations from the View menu in Identity Management and select the organization from the navigation pane. The Location path displays the default top-level organization and chosen organization.

2. Choose Roles from the View menu.

3. Select the checkbox next to the name of the role.

4. Click Delete.

Add Users to a Role

1. Navigate to the Organization that contains the role to modify.

2. Choose Roles from the View menu.

3. Select the role to modify and click on the Properties arrow.

4. Choose Users from the View menu in the Data pane.

5. Click Add.

A search window appears in the data pane.

6. Enter a user ID.

Search criteria can also be entered (including first name, last name or active/inactive) if specific user id information is not available.

7. Choose the users from the names returned by selecting the checkbox next to the user name.

8. Click Save.

The Users are now assigned to the role.

Remove Users from a Role

1. Navigate to the Organization that contains the role to modify.

Choose Organizations from the View menu in the Identity Management module and select the organization from the navigation pane.

2. Choose Roles from the View menu.
3. Select the role to modify.
4. Choose Users from the View menu.
5. Select the checkbox of the users for removal.
6. Click Remove.

The users are now removed from the role.

Add a Role to a Policy

Identity Server objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Modifying Policies," on page 88.

Role Properties View

The Roles Properties view allows for customizing the services available to a role, and the access level for the service attributes, on a per-role basis. Using the Roles Properties view, an administrator can customize the Service and End User pages, and create service administrators who only have access to specific services. For example, an administrator can deny write-access to one or more attributes in the user services for a given role, and a user possessing this role will not be able to modify these attributes. A policy administrator role can be created by granting access to all policy services, but denying access to other services. An administrator possessing the policy administrator role will then be able to create and assign policies, but will be denied from performing user management tasks.

To display the Role Profile page, click on the Properties button associated with a given role in the Roles Properties page, as shown in Figure 2-3.

Figure 2-3 Role Profile View

Logout | Help
Welcome amAdmin

Identity Management Service Management Session Management Federation Management

Search

iplanet ▾

View: Roles ▾

Roles (5 rows)

New... Delete ▾ Show

Name
<input type="checkbox"/> Deny Write Access ▾
<input type="checkbox"/> Top-level Admin Role ▾
<input type="checkbox"/> Top-level Help Desk Admin Role ▾
<input type="checkbox"/> Top-level Policy Admin Role ▾
<input type="checkbox"/> People Admin ▾

Deny Write Access

View: General ▾

Description: Deny Write Access

Permission Description: Read only Access to all entries

Services: LDAP, Policy Configuration, Core, Provider Configuration, Authentication Domain Configuration [\[Edit...\]](#)

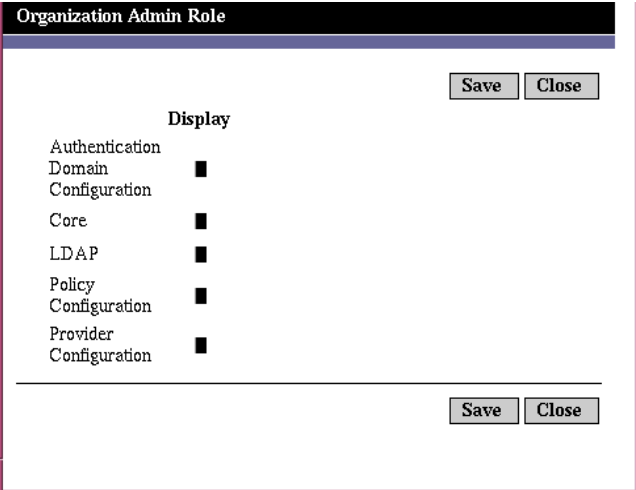
Service Attributes: [\[Edit...\]](#)

Customize Service Access

1. In the Role Properties page, click Edit in the Services listing. The Service Access page is displayed, as shown in Figure 2-4.
2. Choose a service that is to be granted to the role by clicking on the service name in the Display column. By default, a role has access to all services.
3. Click Save.

NOTE When access to a service is denied (not checked), the service will not be displayed in the Identity Server console for the user possessing the role. Additionally, it is not possible to register or unregister a user, assign the service to a user, or create, delete, view or modify the Service template.

Figure 2-4 Service Access Page

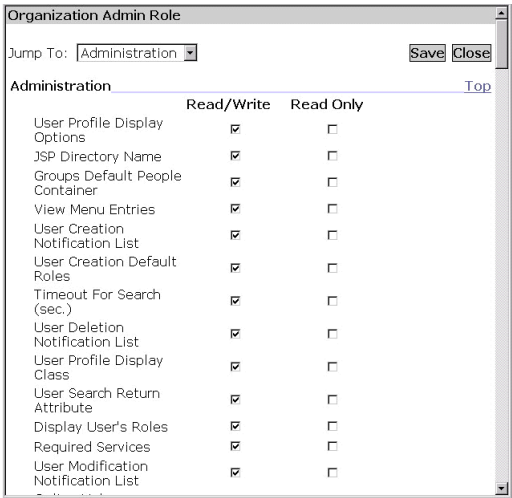


Customize Attribute Access

1. In the Role Properties page, click Edit in the Service Attribute listing. The Attribute Access page is displayed, as shown in Figure 2-5.
2. Use the Jump menu to display the attributes for a particular service.
3. Assign an access level to an attribute by selecting the Read/Write or Read Only check boxes.
4. Click Save.

NOTE If neither the Read/Write or Read Only options are selected for a given attribute, read and write access to that attribute is denied.

Figure 2-5 Attribute Access Page



For more information on specific Service attributes, see Part 3 of this manual, the *Attribute Reference Guide*.

Policies

Policies define rules to help protect an organization’s web resources. Although policy creation, modification and deletion is performed through the Identity Management module, the procedures are described in “Policy Management” on page 83.

Containers

The container entry is used when, due to object class and attribute differences, it is not possible to use an organization entry. It is important to remember that the Identity Server container entry and the Identity Server organization entry are not necessarily equivalent to the LDAP object classes `organizationalUnit` and `organization`. They are abstract Identity entries. Ideally, the organization entry will be used instead of the container entry.

NOTE	The display of containers is optional. To view containers you must select Display Containers in Menu in the Identity Server Administration service. For more information, see “Display Containers In Menu,” on page 78.
-------------	---

Create a Container

1. Navigate to the Organization or Container where the new Container will be created.

Select Containers from the View menu.

2. Click New.

A Container template displays in the Data pane.

3. Enter the name of the Container to be created.

4. Click Create.

Delete a Container

1. Navigate to the organization or container which contains the container to be deleted.

2. Choose Containers from the View menu.

3. Select the checkbox next to the name of the container to be deleted.

4. Click Delete.

NOTE	Deleting a container will delete all objects that exist in that Container. This includes all objects and sub containers.
-------------	--

People Containers

A People Container is the default LDAP organizational unit to which all users are assigned when they are created within an organization. People containers can be found at the organization level and at the people container level as a sub People Container. They can contain only other people containers and users. Additional people containers can be added into the organization, if desired.

NOTE	The display of people containers is optional. To view People Containers you must select Show People Containers in the Identity Server Administration service. For more information, see “Show People Containers,” on page 78.
-------------	---

Create a People Container

1. Navigate to the organization or people container where the new people container will be created.

Select People Containers from the View menu.

2. Click New.

The People Container template displays in the data pane.

3. Enter the name of the people container to be created.

4. Click Create.

Delete a People Container

1. Navigate to the organization or people container which contains the people container to be deleted.

2. Choose People Containers from the View menu.

3. Select the checkbox next to the name of the people container to be deleted.

4. Click Delete.

NOTE	Deleting a people container will delete all objects that exist in that people container. This includes all users and sub people containers.
-------------	---

Group Containers

A Group Container is used to manage groups. It can contain only groups and other group containers. The group container Groups is dynamically assigned as the parent entry for all managed groups. Additional group containers can be added, if desired.

NOTE	The display of group containers is optional. To view group containers you must select Show Group Containers in the Identity Server Administration service. For more information, see “Show Group Containers,” on page 78.
-------------	---

Create a Group Container

1. Navigate to the organization or the group container which contains the group container to be created.
2. Choose group containers from the View menu.
The default Groups was created during the organization’s creation.
3. Click New.
4. Enter a value in the Name field and click Create.
The new group container displays in the navigation pane.

Delete a Group Container

1. Navigate to the organization which contains the group container to be deleted.
2. Choose Group Containers from the View menu.
The default Groups and all created group containers display in the navigation pane.
3. Select the checkbox next to the group container to be deleted.
4. Click Delete Selected.

Chapter 3

Service Configuration

This chapter describes the service management features of Sun ONE Identity Server. The Service Configuration interface provides a way to view, manage and configure all Identity Server services and their values (both default and customized) in addition to configuring Identity Server console display settings. This chapter contains the following sections:

- Definition of a Service
- Identity Server Services Defined
- Attribute Types
- Service Configuration Interface

Definition of a Service

A *service* is a group of attributes defined under a common name. The attributes define the parameters that the service provides to an organization. For instance, in developing a payroll service, a developer might decide to include attributes that define an employee name, an hourly rate and a tax exemption. When the service is registered to an organization, that organization can use these attributes in the configuration of its entries.

Identity Server defines services using Extensible Markup Language (XML). The Service Management Services Document Type Definition (`sms.dtd`) defines the structure of a service XML file. This file can be found in the following directory:

Identity_Server_root/SUNWam/web-apps/services/dtd/

For more information on defining a Identity Server service, see the *Sun One Identity Server Programmer's Guide*.

Identity Server Services Defined

The default services provided with Identity Server are defined by XML files located in the following directory:

Identity_Server_root/SUNWam/web-apps/services/WEB-INF/config/xml

Some of these services, when configured through the Service Configuration interface, define values for the Identity Server application. Others are registered to a specific organization configured within Identity Server and are used to define default values for the organization.

Administration

The Administration service allows for the configuration of the console at both the application level (similar to a *Preferences* or *Options* menu for the Identity Server application) as well as at a configured organization level (*Preferences* or *Options* specific to a configured organization).

Authentication

There are eight authentication services including a base service. This allows the administrator the opportunity to choose the method with which each defined organization would have their users' authorization verified.

Anonymous

This service allows for log in without specifying a user name and password. Anonymous connections have limited access to the server and are customized by the administrator.

Certificate-based

This service allows login through a personal digital certificate (PDC). Sun One Certificate Server can be installed as a Certificate Authority. For more information on Certificate Server, see the documentation set located at http://docs.sun.com/db/coll/S1_slCertificateServer_47.

Core

The Core service is the general configuration base for the Identity Server authentication services. It must be registered and configured to use any of the specific services. It allows the administrator to define default values that will be picked up for those not specifically set in the Anonymous, Certificate-based, LDAP, Membership and RADIUS, SafeWord and Unix services.

LDAP

This service allows for authentication using LDAP bind, an operation which associates a password with a particular LDAP entry.

Membership (Self-Registration)

This service allows a new user to self-register for authentication with a login and password.

NT

This service allows for authenticating users using an Windows NT™ server.

RADIUS

This service allows for authenticating users using an external Remote Authentication Dial-In User Service (RADIUS) server.

SafeWord

This service allows for authenticating users using Secure Computing's SafeWord™ or SafeWord PremierAccess™ authentication servers.

Unix

This service allows for authenticating users using a Unix server.

NOTE	The Unix authentication service is not supported on the Windows 2000 platform.
-------------	--

Authentication Configuration

The Authentication Configuration service allows you to configure authentication on for roles, users and services and organizations to set the rules determining the precedence of the authentication modules.

Client Detection

The Client Detection Service defines attributes to detect the client and perform actions based on client type.

Logging

The Logging service is where the administrator configures values for the Identity Server application logging function. Examples include log file size and log file location.

Naming

The Naming service is used to get and set URLs, plug-ins and configurations as well as request notifications for various other Identity Server services such as session, authentication and logging.

Platform

The Platform service is where additional servers can be added to the Identity Server configuration as well as other options applied at the top level of the Identity Server application.

Policy Configuration

Policy Configuration defines user privileges to web resources, allowing an administrator to allow or deny access to `http` and `https`-based URLs.

SAML

The Security Assertion Markup Language (SAML) service defines a framework for exchanging security assertions among security authorities to achieve interoperability across different platforms, which provide authentication and authorization services.

Session

The Session service defines values for an authenticated user session such as maximum session time and maximum idle time.

User

Default user preferences are defined through the user service. (These include time zone, locale and DN starting view).

Identity Server Security Service

The Identity Server Security Service is automatically loaded after installation. Through the console, administrators and users can use this service to receive security certificates.

In order to enable the Identity Server Security Service, you must:

1. Install Sun ONE Certificate Server 4.7 SP1. For installation instructions, see the Certificate Server 4.7 SPI release notes at <http://docs.sun.com/source/816-6407-10/index.html>.
2. Configure the Certificate Server to enable the Identity Server Security Service. For configuration instructions, see the “Support for Identity Server Single Sign-on (SSO)” section in the Sun One Certificate Server 4.7 SP1 release notes.
3. Define the Identity Server Security Service attributes described in “Identity Server Security Service Attributes,” on page 255.

Identity Server and Certificate Server are integrated using a Single Sign-On (SSO) based token, which is issued by the Identity Server and sent to the Certificate Server. Users that have logged into Identity Server can receive certificates from the Certificate Server by clicking the Get My Certificate button in the User Profile page. The certificate issued to the user allows the user to authenticate with the Certificate-based authentication service.

Attribute Types

The attributes that make up an Identity Server service are classified as one of the following types: *Dynamic*, *Policy*, *User*, *Organization* or *Global*. Using these types to subdivide the attributes in each service allows for a more consistent arrangement of the service schema and easier management of the service parameters.

Dynamic Attributes

A dynamic attribute can be assigned to an 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. For example, a role is created for an organization's employees. This role might contain the organization's address and a fax number, two things that remain static for all employees. When the role is assigned to each employee, these dynamic attributes are inherited by each employee.

User Attributes

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. Examples of user attributes include `userid`, `employee number` and `password`. User attributes can be added or removed from the User service by modifying the `dpUser.xml` file. For more information, see the *Sun One Identity Server Programmer's Guide*.

Organization Attributes

Organization attributes are only assigned to organizations. In that respect, they work as dynamic attributes, yet they differ from dynamic attributes, as they are not inherited by entries in the subtrees. Additionally, no object classes are associated with organization attributes. Attributes listed in the authentication services are defined as organization attributes because authentication is done at the organization level rather than at a subtree or user level.

Global Attributes

Global attributes are applied across the Identity Server configuration. They can not be applied to users, roles or organizations as the goal of global attributes is to customize the Identity Server application. There is only one instance of a global attribute in the Identity Server configuration. There are no object classes associated with global attributes. Examples of global attributes include log file size, log file location, port number or a server URL that Identity Server can use to access data.

Policy Attributes

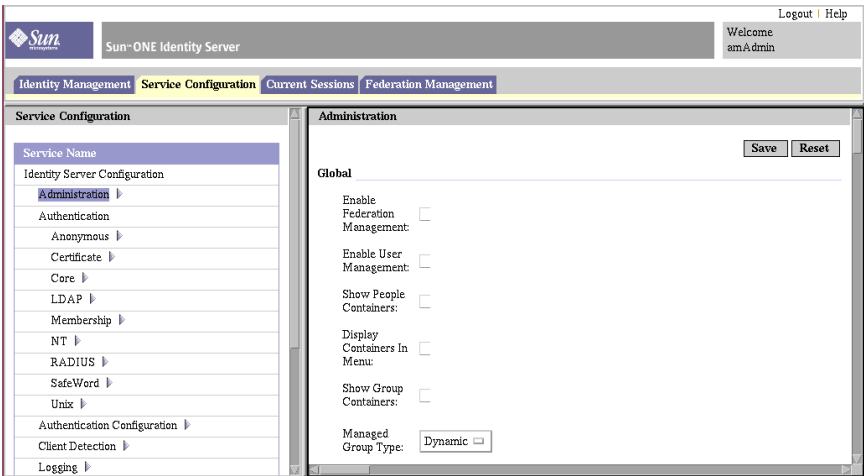
Policy attributes are privilege attributes. Policy attributes are configured through the Identity Management interface as discussed in Chapter 6, “Policy Management. Once a policy is configured, it may be assigned to roles or organizations. That is the only difference between dynamic and policy attributes; 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.

Service Configuration Interface

Services are configured and managed through the Service Configuration module. Organization-specific services which are not covered by the Identity Server default service packages can be written using XML (based on the Identity Server services document type definition or DTD) and added into the interface under the Other Configuration heading. Instructions on how this is done can be found in Part 3, “Attribute Reference Guide” which describes the default services and the definitions of their corresponding attributes.

The Service Configuration module is for displaying service configurations on a global level. In other words, it is a view of the default configurations of all available services in Identity Server, whether registered or not. When a service is registered and activated by an organization, the initial default data assigned to the service is displayed under the service’s Service Configuration page. Figure 3-1 is a screenshot of the graphical user interface.

Figure 3-1 Service Configuration View



Access the Service Configuration view by choosing the Service Configuration module. The navigation pane will display a list of all defined Identity Server services. To set the global default values for a service, select the Properties arrow next to the name of the service. The attributes for the service will be displayed in the Data pane.

Chapter 4

Current Sessions

This chapter describes the session management features of Sun ONE Identity Server. The Session Management module provides a solution for viewing user session information and managing user sessions. It keeps track of various session times as well as allowing the administrator to terminate a session.

The Current Sessions Interface

The Current Sessions module interface allows an administrator, with the appropriate permissions, to view the session information for any user who is currently logged in to Identity Server.

Figure 4-1 Current Sessions Interface

The screenshot displays the Sun ONE Identity Server web interface. At the top, the Sun logo and 'Sun ONE Identity Server' are visible. A navigation bar includes links for 'Identity Management', 'Service Configuration', 'Current Sessions' (which is highlighted), and 'Federation Management'. In the top right corner, there are links for 'Logout' and 'Help', and a 'Welcome amAdmin' message.

The main content area is titled 'Current Sessions'. On the left, there is a 'Server Name' field containing the URL 'http://redline.red.iplanet.com:58080'. On the right, there is a section for 'User Sessions (2 rows)'. This section includes a 'Terminate Session' button, a search input field, and a 'Filter' button. Below these is a table with the following data:

User Id	Time Left	Max Session Time	Idle Time	Max Idle Time
amAdmin	116	120	0	30
user1	119	120	0	30

Session Management Pane

The Session Management pane displays the name of the Identity Server that is currently being administered.

Session Information Window

The Session Information window displays all of the users who are currently logged into Identity Server, and displays the session time for each user. The display fields are:

User ID	Displays the user ID of the user who is currently logged in.
Time Left	Displays the amount of time (in minutes) remaining that the user has for that session before having to reauthenticate.
Max Session Time	Displays the maximum time (in minutes) before the session expires and the user must reauthenticate to regain access.
Idle Time	Displays the time (in minutes) that have expired without the user performing any action.
Max Idle Time	Displays the maximum time (in minutes) that a user can remain idle before having to reauthenticate.

The time limits are defined by the administrator in the Session Management Service. See “Session Attributes” on page 245 for more information.

You can display a specific user session, or a specific range of user sessions, by entering a string in the User ID field and clicking Filter. Wildcards are permitted.

Clicking the Refresh button will update the user session display.

Terminating a Session

Administrators with appropriate permissions can terminate a user session at any time. To do so:

1. Select the user session that you wish to terminate.
2. Click Terminate.

Chapter 5

Federation Management

This chapter describes the Federation Management interface features of the Sun ONE Identity Server. The Federation Management interface provides a way to view, manage and configure the metadata pertaining to the authentication domains and providers. This chapter contains the following sections:

- Liberty Alliance and Federated Identity
- Federation Management Concepts
- Federation Management Process
- Managing Authentication Domains and Providers

NOTE Example data for the attribute fields described in this chapter can be found in the following default location:

Identity_Server_root/opt/SUNWam/samples/liberty

Liberty Alliance and Federated Identity

On the internet, one person might have a multitude of accounts set up to access various business, community and personal service providers. For example, different names, passwords and preferences might be set up for a news portal, a bank account, a retailer, and an email account. A *local identity* refers to the set of attributes that an individual might set up with one service provider. These attributes serve to uniquely identify the individual with that provider and may include a name, phone number, social security number, address, credit records, bank balances or bill payment information.

Because the internet is fast becoming the prime vehicle for business, community and personal interactions, it has become necessary to fashion a system for online users to aggregate their local identities, enabling them to have one *network identity*. This system is *identity federation*. Identity federation allows a user to associate, connect or bind multiple internet service providers' local identities. A network identity allows users to login at one service provider's site and then go to an affiliated site without having to re-authenticate or re-establish their identity. The Liberty Alliance Project was forged to make identity federation a reality.

Federation Management Concepts

To understand the information contained in this chapter, you should familiarize yourself with the following concepts.

Authentication Domain

An authentication domain is a group of affiliated service providers and one or more identity providers. The providers that belong to an authentication domain share a *trusted* relationship.

Service Provider

Service providers are organizations offering web-based services to users. This includes practically any organization on the Internet (for example, internet portals, retailers, transportation providers, financial institutions, entertainment companies, non-profit organizations, and governmental agencies).

Identity Provider

Identity providers are organizations that specialize in providing authentication services. In the Liberty context, authentication completed by an identity provider will be honored by all its affiliates.

Remote Provider

A remote provider is a service provider or identity provider that is not hosted by the current installation of Identity Server, but is Liberty-enabled, either by another (remote) installation of Identity Server, or by another implementation of the Liberty specification.

Hosted Provider

A hosted provider is either a service provider or an identity provider that is Liberty-enabled by the current, or present, installation of Identity Server.

Metadata

Metadata is the set of required data for configuring the policies that govern the behavior of a service provider or identity provider. Liberty specifications define the metadata attributes for service providers and identity providers.

Account Federation

Account Federation enables service providers and identity providers to unite the otherwise distinct accounts of users.

Federated Identity

Federation allows users to retain their individual accounts with services and identity providers, and establishes an affiliation between the accounts, which facilitates the exchange information about the account on the user's behalf.

Single Sign-on

Single sign-on (SSO) is established when a user authenticates with an identity provider and accesses multiple affiliated service providers, without having to reauthenticate.

Federation Termination

Users will have the ability to terminate federations. Federation termination results in breaking affiliation established between the user's service provider account and the identity provider account at the time of federation.

Single Logout

When a user logs out from an identity provider, the user will effectively be logged out from all affiliated service providers within an authentication domain. Logout information is sent to the identity provider when it is initiated from a service provider. Users can initiate logout from either a service provider or the identity provider.

Common Domain

Due to the constraints in using cookies, a web service can only share cookies with other web services that are in the same DNS domain. Therefore, an identity provider in one DNS domain cannot write a cookie that a service provider in another DNS domain can read. To rectify this, the Liberty specification advocates the use of a common domain. This domain is common to all the members of the authentication domain and thus accessible to all parties. All of the cookies in Liberty-based communication will be set by the common domain DNS to make it available to all of the providers within an authentication domain.

Authentication Context

Authentication Context is the information that the service provider may require, in addition to the authentication assertion itself, before it makes an entitlement decision.

If a service provider is to rely on the authentication of a principal by an identity provider, the service provider may require additional information, beyond the authentication assertion itself, before the authentication can be put in a trust context. This information could include:

- Initial user identification mechanisms (Examples: face-to-face, online, shared secret)
- Mechanisms for storing and protecting credentials (Examples: smartcard, password rules)
- Authentication mechanism (Examples: password, certificate-based SSL)

Not all authentication assertions are the same; a particular authentication assertion will be characterized by the values for each of these variables, which is captured in an Authentication Context

Name Identifier

Identity federation maps a user's account information across a number of service and identity provider organizations. The user's identity is exchanged between the identity and service providers as a *name identifier*, and is stored in the Directory Server data store.

The Liberty Alliance Project

The vision of the Liberty Alliance Project is to enable individuals and organizations to more easily conduct transactions while protecting the individual's identity. To accomplish this, the Liberty Alliance has established specifications for identity federation that enables:

- Opt-in account linking where users can choose to federate different internet service provider accounts.
- Simplified single sign-on where a user can log in and authenticate with one provider's federated account and navigate to another account without having to log in again.
- Authentication context where organizations with linked accounts communicate the type and level of authentication that should be used when the user logs in.

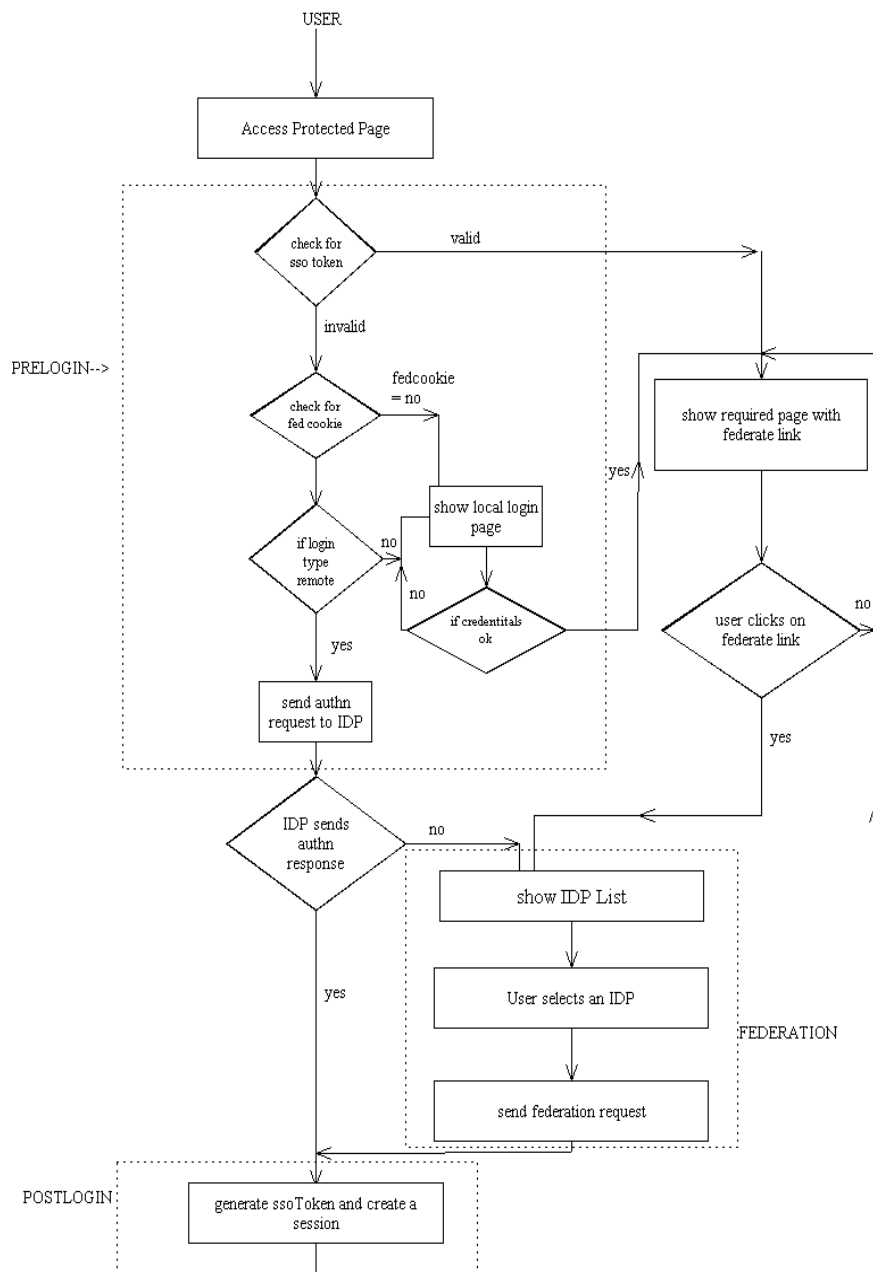
- Global log-out where a user logs out of the site to which they initially logged in and is automatically logged out of all sites that maintain a live session.
- A client feature which can be implemented in fixed and wireless devices to facilitate use of the Liberty specifications.

These capabilities can be achieved when commercial or non-commercial organizations join together into an authentication domain based on Liberty-enabled technology and operational agreements. This is referred to as an *authentication domain*. The authentication domain includes service providers (who offer web-based services to users), identity providers (service providers who also offer federated authentication), and the users themselves. Once an authentication domain is established, users can federate any or all identities they might have with the service providers that have joined this domain, enabling them to make use of the federated authentication capabilities.

Federation Management Process

Out of the box, Identity Server has two options for user or application authentication. The first is the Identity Server Authentication Service and the second is the Liberty-enabled Federation Management Service. In an Identity Server scenario, when a user or application tries to access a resource protected by the Identity Server, the user is redirected to the Authentication Service via a Login page for access authorization. When the user provides credentials, the authentication module verifies them and either allows or denies access.

In a scenario where the Identity Server is Liberty-enabled and a user or application attempts to access a protected resource, the user is redirected to a Pre-Login page which invokes the Federation Management Service's Pre-Login servlet. This servlet searches for either a valid Identity Server single sign-on token or a valid Federation Cookie (which indicates that a user has federated his account using this Identity Server provider). If an SSO token is found, the user's Federation information is retrieved, and the user is authenticated; a Federation Cookie is also set and the user is returned to the target resource. If a Federation Cookie is found, the user is directed to the Federation Single Sign-On Service which provides an Authentication Assertion allowing the user access to the target resource. If neither of these items is found, the user is redirected to the Identity Server Authentication Service where, upon successful authentication, the user is directed to the Post-Login page which invokes the Post-Login servlet. This servlet processes the user's Identity Server authentication and initiates the Federation Management Single Sign-On Service which, once again, provides an Authentication Assertion to allow the user access to the target resource. Figure 5-1 on page 68 illustrates this flow.

Figure 5-1 Liberty-enabled Identity Server Authentication Process Flow

Managing Authentication Domains and Providers

The Federation Management module provides an interface for creating, modifying, and deleting authentication domains, remote providers and hosted providers. To set up a basic Federation Management model, you would do the following:

1. Create an authentication domain.
2. Create one or more hosted providers that belong to the created authentication domain.
3. Create one or more remote providers that belong to the created authentication domain. You must also include the metadata for the remote providers. The metadata could be an XML document that is compliant with the Liberty schema.
4. Establish a trusted relationship between the providers. A hosted provider can choose to trust a subset of providers, either hosted or remote, that belong to the same authentication domain.

The following sections explain how to create and configure authentication domains, remote providers, and hosted providers.

Authentication Domains

This section describes how to create, modify, and delete authentication domains.

Creating An Authentication Domain

1. Choose Authentication Domain from the View menu in the Federation Management module.
2. Click New in the navigation frame.

The Create Authentication Domain is displayed in the Data frame.

3. In the Create Authentication Domain window, enter the name of the Authentication Domain.
4. Enter a value for the description of the Authentication Domain.
5. Enter a value for the Writer Service URL.

Writer Service URL specifies the location of the Writer service that writes the cookie from the Common Domain. For example, if sun.com is the common domain, the URL could be:

`http://sun.com:8080/liberty/WriterServlet`

6. Enter a value for the Reader Service URL.

The Reader Service URL specifies the location of the service that reads the cookie from the Common Domain.

7. Choose a status of active or inactive.

The default is active. This can be changed at any time during the life of the Authentication Domain by selecting the Properties icon. Choosing inactive disables Liberty communication within authentication domain, (including all remote and hosted providers within the authentication domain), with respect to the current installation of Identity Server.

8. Click Create.

The new Authentication Domain displays in the navigation frame.

Modifying An Authentication Domain

1. Click on the properties arrow next to the Authentication Domain you wish to modify.

The properties of the Authentication Domain display in the Data frame.

2. Modify the properties of the Authentication Domain.
3. Click Save.

Deleting An Authentication Domain

Deleting an authentication domain does not delete the providers that belong to it. If providers belong to an authentication domain that has been deleted, they remain part of the authentication domain until they are explicitly removed. Additional providers can not be added to an authentication domain that has been deleted.

1. Choose Authentication Domains from the View menu in the Federation Management module.

All created Authentication Domains display in the navigation frame.

2. Check the box next to the name of the Authentication Domain to be deleted.
3. Click Delete Selected.

NOTE There is no warning message when performing a delete.

Providers

This section describes how to create, modify and delete remote and hosted providers.

Creating Remote Providers

A remote provider is an entity that receives metadata from a principal, which is an organization or an individual who interacts with the system. To create a remote provider:

1. Choose Remote Provider from the View menu in the Federation Management module.

By default, when a Provider is created, it will be a service provider. You can optionally decide to create the remote provider as an identity provider by selecting the option described in Step 14.

2. Click New. The Create Remote Provider window is displayed.
3. Enter a value for the Name of the Provider.
4. Enter a value for the Provider ID.

The Provider ID should specify the URL identifier of the provider. It must be unique across all remote and hosted providers.

5. Enter the Provider Succinct ID.

The Provider Succinct ID uniquely identifies a service provider to an identity provider. It is also used by the identity provider to determine service provider-specific information such as return URLs, public key/certificates, and so on.

This field accepts a 40-character hexadecimal value. The Succinct ID should be an SHA1 encoded string. It is recommended that the provider ID string should be used as the value to encode, as it will ensure that it is unique. To generate the SHA1 encoding, use the OpenSSL command line tool syntax:

```
$ echo <providerID> | openssl sha1
```

6. Specify Active or Inactive status.

Active status enables this remote provider to participate in federation and SSO. Inactive status makes this remote provider unavailable, and will not respond to any requests.

7. Enter the Security Key.

The Security Key defines the Security Certificate alias. The certificates are stored in the JKS keystore against an alias. This alias (the Security Key) is used to fetch the required certificate.

8. Enter the SOAP End Point URL.

This field specifies the location for the receiver of SOAP requests. This is used to communicate on the back-channel (non-browser communication) through SOAP.

9. Enter the Single Logout Service URL.

The Single Logout Service URL is used by a service provider or identity provider to send and receive logout requests.

10. Enter the Single Logout Service Return URL.

This specifies the URL to which logout requests are redirected after processing.

11. Enter the Federation Termination Service URL.

This field specifies the URL to which federation termination requests are sent.

12. Enter a value for the Federation Termination Service Return URL.

This field specifies the URL to which federation termination requests are redirected after processing.

13. Enter the Assertion Consumer URL.

This field defines the service provider end-point to which an identity provider will send SAML assertions.

14. Decide if the remote provider is to be defined as an identity provider. By default, all providers are service providers. If selected, the Is Identity Provider option will additionally define the remote provider as an identity provider.

15. Define the Single Sign-On Service URL.

This field defines the identity provider URL to which the service provider sends requests during federation and SSO. This field only needs to be defined if the Is Identity Provider option is enabled.

16. Click Create.

The new Provider displays in the navigation frame.

Modifying Remote Providers

Once a remote host is created, you can modify it at any time. To do so:

1. Select Remote Providers from the View menu in the Navigation frame.
2. Choose the provider profile you wish to modify, and click on the Edit arrow.

By default the General view is displayed in the Navigation pane. All of the fields displayed in the General view contain the data that was entered during the creation of the remote provider. If you modify any of the fields, click Save to save the changes.

3. To modify the Service Provider fields, choose Service Provider from the View menu.

The Assertion Consumer URL field contains data that was entered during the creation of the remote provider. However, there are three additional fields that you can modify:

Federation Termination Protocol	<p>You can choose SOAP or HTTP/Redirect.</p> <p>This field specifies if the SOAP or HTTP/Redirect profile is to be used to notify of federation termination. This can be changed at any time during the life of the Provider.</p>
Single Logout Protocol	<p>You can choose SOAP, HTTP Get, or HTTP Redirect.</p> <p>This field specifies if SOAP or HTTP Redirect is to be used to notify a logout event. If the remote provider is defined as an identity provider, it can also send the notification using the HTTP Get profile. This can be changed at any time during the life of the provider.</p>
Authentication Request Signed	<p>This option, if enabled, specifies that the remote provider send signed authentication and federation requests. The identity provider will not process unsigned requests originated from the service provider.</p>

4. Click Save to save the changes.

5. If the remote provider was defined as an identity provider during creation, you can modify the fields by selecting Identity Provider in the View menu.

The data contained in these fields were entered at creation. If you modify the fields, click Save to save the changes.

6. Select Authentication Domains in the View menu to edit the authentication domains to which the remote provider will belong.

Use the direction arrows to move a selected authentication domain into the Available list. Click Save. This will assign the provider to the authentication domain. A provider can belong to one or more authentication domains, however a provider without any authentication domains specified can not participate in Liberty communications.

7. Choose Trusted Providers from the View menu.

Choose the trusted providers. The remote provider will only accept requests (such as Federation requests, Federation Termination Requests, SSO requests, and so forth) originating from this set of providers. The requests from other providers will be ignored. Click Save.

Creating Hosted Providers

A hosted provider is an entity that creates, maintains, and manages identity information for principals and provides principal authentication to other service providers within an authentication domain. To create a hosted provider:

1. Choose Hosted Provider from the View menu in the Federation Management module.

By default, when a Provider is created, it will be a service provider. You can optionally decide to create the remote provider as an identity provider by selecting the option described in Step 9.

2. Click New. The Create Remote Provider window is displayed.
3. Enter a value for the Name of the provider.
4. Enter the Alias for the provider.

For each of the hosted providers, the alias provided in this field is added to a string called meta Alias. This string is then added to the automatically populated URLs for the hosted providers. These URLs are called metadata URLs. In the following examples, `sunAlias` is the alias for the provider:

Federation Termination Service URL:

```
http://www.sun.com:58080/amserver/ProcessTermination/metaAlias/sunAlias
```

SOAP Endpoint URL:

```
http://www.sun.com:58080/amserver/SOAPReceiver/metaAlias/sunAlias
```

5. Enter a value for the Provider ID.

The Provider ID specifies the URL identifier of the provider. It must be unique across all remote and hosted providers.

6. Enter the Provider Succinct ID.

The Provider Succinct ID uniquely identifies a service provider to an identity provider. It is also used by the identity provider to determine service provider-specific information, such as return URLs, public key/certificates, and so on.

The Succinct ID should be an SHA1 encoded string. It is recommended that the provider ID string should be used as the value to encode, as it will ensure that it is unique. To generate the SHA1 encoding, use the OpenSSL command line tool syntax:

```
$ echo <providerID> | openssl sha1
```

7. Enter the Cookie Domain.

This field specifies the cookie domain if Identity Server is installed on a multi-hosted domain and each of the providers use different domains for their cookie settings.

8. Specify Active or Inactive status.

Active status enables the remote provider to participate in federation and SSO. Inactive status makes the remote provider unavailable, and will not respond to any requests.

9. Decide if the remote provider is to be defined as an identity provider. By default, all providers are service providers. If selected, the Is Identity Provider option will additionally define the remote provider as an identity provider.

10. Enter the Security Key.

The Security Key defines the Security Certificate alias. The certificates are stored in the JKS keystore against an alias. This alias (the Security Key) is used to fetch the required certificate.

11. Enter the Provider URL.

This field specifies the URL from which the metadata will be sent.

12. Click Create.

The new provider is displayed in the navigation frame.

Modifying Hosted Providers

1. Choose the provider profile you wish to modify, and click on the Edit arrow.

By default the General view is displayed in the Navigation pane. All of the fields displayed in the General view contain the data that was entered during the creation of the hosted provider. If you modify any of the fields, click Save to save the changes.

2. To modify the Service Provider fields, choose Service Provider from the View menu.

The Assertion Consumer URL field contains data that was entered during the creation of the remote provider. However, there are four additional fields that you can modify:

Federation Termination Protocol	<p>You can choose SOAP or HTTP/Redirect.</p> <p>This field specifies if the SOAP or HTTP/Redirect profile is to be used for federation termination notifications. This can be changed at any time during the life of the Provider.</p>
Single Logout Protocol	<p>You can choose SOAP, HTTP Get, or HTTP Redirect.</p> <p>This field specifies if SOAP or HTTP Redirect is to be used to notify a logout event. If the remote provider is defined as an identity provider, it can also send the notification using the HTTP Get profile. This can be changed at any time during the life of the provider.</p>
Authentication Request Signed	<p>This option, if enabled, specifies that the remote provider send signed authentication and federation requests. The identity provider will not process unsigned requests originated from the service provider.</p>

Authentication Context	<p>This provides an interface to set the Liberty authentication context to the Identity Server authentication levels. The authentication contexts are:</p> <ul style="list-style-type: none">Previous-SessionTime-Sync-TokenSmartcardMobileUnregisteredSmartcard-PKI MobileContractPasswordPassword-ProtectedTransportMobileDigitalIDSoftware-PKI
------------------------	---

3. Click Save to save the changes.

4. If the remote provider was defined as an identity provider during creation, you can modify the fields by selecting Identity Provider in the View menu. Most of the data contained in these fields were entered at creation.

In this view, you can additionally define the Identity Provider Authentication Context.

The fields are as follows:

Supported	Specifies if the identity provider supports the authentication context. The identity provider should support at least one authentication context.
Context Reference	Defines the name of the authentication context. There are ten contexts defined in the Liberty protocol.
Key	<p>The query string sent to the /UI/Login (the Identity Server authentication servlet) will contain a key-value pair identifying the authentication mechanisms to be used. The possible key values are:</p> <ul style="list-style-type: none"> • Module • Level • Role • Service • User
Value	Defines the value of the key-value pair for the authentication mechanism.
Priority	Indicates the ordering determined by the identity provider for the Liberty-defined authentication contexts. If the identity provider does not support the authentication context requested by the service provider during the authentication request, it can use any other authentication context which is either at the same or higher priority level.

5. Click Save to save the changes.

6. Select Authentication Domains in the View menu to edit the authentication domains to which the remote provider will belong.

Use the direction arrows to move a selected authentication domain into the Available list. Click Save. This will assign the provider to the authentication domain. A provider can belong to one or more authentication domains, however a provider without any authentication domains specified can not participate in Liberty communications.

7. Choose Trusted Providers from the View menu.

Choose the trusted providers. The remote provider will only accept request originated from this set of providers. The requests from other providers will be ignored. To create the list of trusted providers, select the providers from the Available field and use the Add button to add them to the Selected field. (You can remove providers by using the Remove button.) Click Save.

8. Choose Identity Server Configuration Attributes.

The fields are as follows:

Authentication Type	Remote/Local: Specifies if the hosted provider should contact an identity provider for authentication upon receiving an authentication request (Remote), or if authentication should be done by the hosted provider itself (Local).
Authentication Federate Profile	Specifies the profile used by the hosted provider for sending authentication requests. Identity Server provides the following protocols: <ul style="list-style-type: none"> • Browser Post - specifies a front-channel (http POST-based) protocol • Browser Artifact - Backchannel (non-browser) SOAP-based protocol.
Organization DN	Specifies the storage location of the DN of the organization if each hosted provider chooses to manage users across different organizations leading to a hosted model.
Forced Authentication	Indicates if the identity provider must reauthenticate (even during a live session) when an authorization request is received.

Is Passive	If selected, this specifies that the identity provider must not interact with the principal and must interact with the user.
Responds With	This is required for the service provider. It defines what the service provider should expect to receive from an identity provider in response to an authentication request. It can be any QNAME, but by default, the value is set to <code>lib:AuthenticationStatement</code> .
Liberty Version URI	Specifies the version of the Liberty specification.
Name Registration Indicator	Allows the option for a service provider to participate in name registration. Name registration is a profile by which service providers specify a principal's name identifier that an identity provider will use when communicating to the service provider.
Home Page URL	Specifies the home page of the provider.
Default Authentication Context	Specifies the authentication context to be used if the identity provider does not receive it as part of a service provider request. It also specifies the authentication context used by the service provider when an unknown user tries to access a protected resource. The default values are: <ul style="list-style-type: none"> • Previous-Session • Time-Sync-Token • Smartcard • MobileUnregistered • Smartcard-PKI MobileContract • Password • Password-ProtectedTransport • MobileDigitalID • Software-PKI
Assertion Interval	Specifies the validity interval for the assertion issued by an identity provider. A principal will remain authenticated by the identity provider until the assertion interval expires.
Cleanup Interval	Specifies the interval of time to clear assertions that are stored in the identity provider.

Artifact Timeout	Specifies the timeout of an identity provider for assertion artifacts.
Assertion Limit	Specifies the number of assertions an identity provider can issue, or that can be stored.

9. Click Save to save the changes.

Deleting Providers

1. Choose Provider from the View menu in Federation Management.
All created Providers display in the Navigation frame.
2. Check the boxes of the Providers you want to delete.
3. Click Delete Selected.

NOTE	There is no warning message when performing a delete.
-------------	---

Chapter 6

Policy Management

This chapter describes the policy service management features of Sun ONE Identity Server. Policy management provides a way to view, manage and configure all Identity Server policies.

This chapter contains the following sections:

- What is a Policy?
- Policy Types
- Policy Management

What is a Policy?

Every business has a need to protect its resources. This is done by configuring and managing rules that define who can do what to which resource. The Identity Server Policy Service enables an organization to set up these rules or *policies*.

A *policy* defines permissions that allow an administrator to assign security levels based on an organization's needs and the conditions created within the policy. This policy, when possessed by an object, defines which resources within an organization that the object is able to access. A single policy can define either binary or non-binary decisions. A binary decision is *yes/no*, *true/false* or *allow/deny*; most policies are of this type. A non-binary decision represents the value of an attribute. For example, a mail service might include a `mailboxQuota`

attribute with a maximum storage value set for each user. A policy service administers this restriction ensuring that each user's quota is not exceeded. In general, a policy is configured to define what an object can do to which resource and under what conditions.

Identity Server ships with one policy service, the URL Policy Agent, and one sample mail service. For more information on the sample mail service and writing new policy schema, see the *Sun One Identity Server Programmer's Guide*.

Policy Types

There are two types of policy that can be configured using Identity Server: a *normal* policy or a *referral* policy. A normal policy consists of *rules*, *subjects* and *conditions*. A referral policy consists of *rules* and *referrals* to organizations.

Normal Policy

In Identity Server, a policy that defines access permissions is referred to as a *normal* policy. A normal policy consists of *rules*, *subjects* and *conditions*.

A *rule* consists of a *resource*, and one or more sets of an *action* and a *value*. A resource defines the object that is being protected; an action is the name of an operation that can be performed on the resource and a value defines the permission.

NOTE It is acceptable to define an action without resources.

Policies are not assigned to identities. Instead, *subjects* are assigned to policies. A subject is the identity object to which the policy is assigned and applied.

A *condition* defines the situations in which a policy is applicable. For example, a 7 am to 10 am condition in a policy means that the policy is applicable only from 7 am to 10 am.

NOTE The terms referral, rule, resource, subject, condition, action and value correspond to the elements *Referral*, *Rule*, *ResourceName*, *Subject*, *Condition*, *Attribute* and *Value* in the `policy.dtd`. They are explained further in the *Sun One Identity Server Programmer's Guide*.

Referral Policy

An administrator might typically need to delegate one 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 one or more *rules* and one or more *referrals*. A rule defines the resource whose policy evaluation is being referred. The referral defines the organization to which the policy evaluation is being referred.

NOTE

The referred-to organization can define or evaluate policies only for those resources (or their sub-resources) that have been referred to it. This restriction, however, does not apply to the root organization. Therefore, an administrator must define management policies at the root level organization only.

There are two types of referrals bundled with Identity Server: peer organization and suborganization. They delegate to an organization on the same level and an organization on a sub-level, respectively. See “Creating Policies for Peer and Suborganizations,” on page 95 for more information.

Policy Management

You can create, delete, and modify policies through the Policy API, through the `amadmin` command line tool, and through the Identity Server console.

This chapter focuses on creating policies through the console. For more information on `amadmin`, see “The `amadmin` Command Line Tool” on page 123. For more information on the Policy API, see the “Policy Service” chapter in the *Sun One Identity Server Programmer's Guide*.

Policies are configured using the Identity Management interface. This interface provides a means for:

- The Top-Level Administrator to view, create, delete and modify policies for a specific service that can be used across all organizations.
- An organization or suborganization administrator to view, create, delete and modify policies for specific use by the organization.

In general, policy is created at the organization (or suborganization) level to be used throughout the organization's tree.

Figure 6-1 Policy View

The screenshot displays the Sun ONE Identity Server web interface. At the top, there's a header with the Sun logo, 'Sun ONE Identity Server', and user information 'Welcome amAdmin' with 'Logout' and 'Help' links. Below the header is a navigation bar with tabs: 'Identity Management' (selected), 'Service Configuration', 'Current Sessions', and 'Federation Management'. A 'Search' link is also present. The main content area is divided into two panes. The left pane, titled 'sun', shows a 'View:' dropdown set to 'Policies', a 'Policies (1 row)' section with 'New...' and 'Delete' buttons, and a table with one row containing 'policy1'. The right pane, titled 'New Policy', contains a 'Type of Policy' dropdown with 'Normal' selected, a 'Name' text field with 'New Policy' entered, and 'Create' and 'Cancel' buttons. A note '* Indicates a required field' is visible.

Registering Policy Configuration Services

Registering a policy configuration service is the same as registering any type of service; it is done within the Identity Management interface. By default, the Policy Configuration service is automatically registered to the top-level organization. Any policy service you create must be registered to all organizations. To register a policy configuration service:

1. Navigate to the Identity Management interface.

When the console opens, the default interface is Identity Management.

2. Choose the organization for which you would like to create policy.

If logged in as the Top-Level Administrator, make sure that the location of the Identity Management module is the top-level organization where all configured organizations are visible. The default top-level organization is defined during installation.

3. Choose Services from the View menu.

If the organization already has registered services, they will be displayed in the navigation pane.

4. Click Register in the navigation pane.

A listing of services not yet registered to this organization is displayed in the data pane.

5. From the Register Services window, opened in the Data pane, choose Policy Configuration and click register.

The Policy Configuration Service is added to the list of services in the Navigation pane.

6. Configure the policy service by clicking the Properties arrow. If the policy template has not yet been configured, you will need to create a service template for the newly registered policy service.

To configure the policy service, click Create. Modify the Policy Configuration attributes. See “Policy Configuration Attributes,” on page 229 for a description of these attributes. Click Save.

The policy configuration service is now registered to the chosen organization.

NOTE suborganizations must register their policy services independently of their parent organization. In other words, the suborganization `o=suborg,dc=sun,dc=com` will not inherit the policy configuration service from its parent `dc=sun,dc=com`.

Creating Policies

Policies are created through the Identity Management interface. To create a policy:

1. Navigate to the Identity Management interface.
2. Choose the organization for which you would like to create a policy.

Ensure that the location of the Policy Management window is correct for your organization.

3. Choose Policies from the View menu.

By default, the Organizations view is visible in the View menu. All suborganizations configured, if any, will be visible below it. If creating policies for a suborganization, choose the suborganization and then choose Policies from the View menu.

4. Click New in the navigation pane. The New Policy window opens.

5. Select the type of policy, normal or referral, that you wish to create.

If a referral policy that refers to a suborganization does not exist, you will not be able to create any policies for suborganizations. For more information, see “Creating Policies for Peer and Suborganizations,” on page 95.

It is not necessary to define all of the fields for normal or referral policies at this time. You may create the policy, then add rules, subjects, referrals, and so forth, later. For information on configuring normal and referral policies, see “Modifying Policies,” on page 88.

6. Type a name for the policy and click Next.

The new policy rule window opens under the policy name created.

7. By default, the General view is displayed.

The General view displays the name of the policy and allows you to enter a description of the policy that is to be created.

8. Click Create to complete the policy’s configuration.

Modifying Policies

Once a normal or referral policy is created, you can modify the rules, subjects, conditions and referrals.

1. From the Identity Management interface, select Policies from the View menu.

The policies that were created for that organization are displayed.

2. Choose the policy you wish to modify and click the Properties arrow. The Edit Policy window is opened in the Data pane.

By default, the General view is displayed.

Modify a Normal Policy

Through the Identity Management interface, you can create a policy that defines access permissions. Such a policy is referred to as a *normal* policy. A normal policy can consist of multiple rules, subjects, and conditions. This section lists and defines the default fields that you can specify when creating a normal policy.

Adding Rules

Rules define the resource, actions and action values of the policy. To add rules to a normal policy:

1. From the Identity Management interface, select Policies from the View.
The policies that were created for that organization are displayed.
2. Choose the policy you wish to modify and click the Properties arrow. The Edit Policy window is opened in the Data pane.
By default, the General view is displayed.
3. To define rules for the policy, select Rules from the View menu and click Add.
If more than one policy service exists, they will be listed in the Navigation pane. Choose the policy service for which you wish to create a policy and click Next. The Add Rule window is displayed.
4. Define the resource, actions and action values in the Rules fields.
The fields are:

Service	Displays the policy service for the policy to be created.
Name	This field allows you to enter the name of the rule.
Resource Name	<p>This field allows you to enter the name of a resource. For example:</p> <p><code>http://www.sunone.com</code></p> <p>Currently, the only resources that can be enforced are <code>http://</code> and <code>https://</code> addresses.</p> <p>You must enter a full domain name. Wildcards and IP addresses are not allowed.</p> <p>For the URL Policy Agent service, if a port number is not entered, the default port number is 80 for <code>http://</code>, and 443 for <code>https://</code>.</p>
Select Actions	<p>For the URL Policy Agent Service, you can select either or both of the following default actions:</p> <ul style="list-style-type: none"> • GET • POST

Select Action Values

For the URL Policy Agent Service, you can choose one of the following action values:

- `allow` lets you access the resource matching the resource defined in the rule.
- `deny` denies access to the resource matching the resource defined in the rule.

NOTE

If the policy service is defined so that an action does not need resource definitions, the resource field will not be displayed. If the service contains both types of actions (some requiring resources, some without resources), an option is displayed to select rules with actions requiring no resources, or rules with actions requiring resources.

5. Click Create to save the rule.
6. Repeat steps 1 - 5 to create additional rules.
7. All of the rules created for that policy are displayed in the table in the Rules view. Click Save to add the rules to the policy.

To remove a rule from a policy, select the rule and click Remove.

You can edit any rule definition by clicking on the Edit link next to the rule name.

Adding Subjects

Subjects define the subject to which the policy will apply. To add subjects to a policy:

1. To define the subject for the policy, select Subject from the View menu and click Add.
2. Select one of the default subject identities:
 - Identity Server Roles
 - LDAP Groups
 - LDAP Roles
 - LDAP Users

- Organization

Click Next to continue.

3. Enter a name for the subject. Click Add.
4. Perform a search in order to display the identities to add to the subject.
The default (*) search pattern will display the qualified entries.
5. Select the identities that you wish to add for the subject and click Create.
6. All of the subjects created for that policy are displayed in the table in the Subjects view. Select the subjects that you wish to add to the policy and click Save.

To remove a subject from a policy, select the subject and click Remove.

You can edit any subject definition by clicking on the Edit link next to the rule name.

Adding Conditions

Conditions allows you to define constraints on the policy. For example, if you are defining policy for a paycheck application, you can define a condition on this action limiting access to the application only during specific hours. Additionally, you may wish to define a condition that only grants this action if the request originates from a given set of IP addresses or from a company intranet. To add conditions to a normal policy:

1. To define conditions for the policy, select Conditions from the View menu. Click Add to add a new condition, or click the Edit link to edit an existing condition.
2. Select one of the following default conditions:
 - Authentication Level
 - Authentication Scheme
 - IP Address
 - Time

Click Next.

3. Define the values for a given condition in the Rules fields. The fields are:

Authentication Level

name	This field allows you to enter the name of the condition.
authentication level	The authentication level value indicates how much to trust authentications.

Authentication Scheme

name	This field allows you to enter the name of the condition.
authentication scheme	This field allows you to choose from the pull-down menu the authentication scheme for the condition.

IP Address

name	This field allows you to enter the name of the condition.
IP Address To/From	This field allows you to specify the range of the IP address
DNS Name	This field allows you to specify the DNS name.

Time

name	This field allows you to enter the name of the condition.
Date To/From	This field allows you to specify the range of the date.
Time	This field allows you to specify the range of time within a day.
Day	This field allows you to specify a range of days.
Timezone	This field allows you specify a timezone, either standard or custom.

4. Once you have defined the condition, click Create.
5. All of the conditions created for that policy are displayed in the table in the Conditions view. Select the conditions that you wish to add to the policy and click Save.

To remove a condition from a policy, select the condition and click Remove.

You can edit any condition definition by clicking on the Edit link next to the rule name.

Modify a Referral Policy

Through the Identity Management interface you can delegate an organization's policy definitions and decisions to another organization. (You can also delegate policy decisions for a resource 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.

Adding Rules

Rules define the resource of the policy. To add rules to a referral policy:

1. To define rules for the policy, select Rules from the View menu. Click Add to add a new rule, or click the Edit link to edit an existing rule.
2. Define the resource in the Rules fields. The fields are:

Service	Displays the policy service for the policy to be created.
Name	This field allows you to enter the name of the rule.
Resource Name	<p>This field allows you to enter the name of a resource. For example:</p> <p><code>http://www.sun.com</code></p> <p>Currently, the only resources that can be enforced are <code>http://</code> and <code>https://</code> addresses.</p> <p>You must enter a full domain name. Wildcards and IP addresses are not allowed.</p> <p>For the Policy URL Agent service, if a port number is not entered, the default port number is 80 for <code>http://</code>, and 443 for <code>https://</code>.</p>

3. Click Create to save the rule.
4. Repeat steps 1 - 3 to create additional rules.
5. All of the rules created for that policy are displayed in the table in the Rules view. Select the rules that you wish to add to the policy and click Save.

To remove a rule from a policy, select the rule and click Remove.

You can edit any rule definition by clicking on the Edit link next to the rule name.

Adding Referrals

The referral defines the organization to which the policy evaluation is being referred. By default, there are two types of referrals: peer organization and suborganization. They delegate to an organization on the same level and an organization on a sub-level, respectively.

To add a referral:

1. To define referrals for the policy, select Referrals from the View menu. Click Add to add a new referral, or click the Edit link to edit an existing referral.
2. Define the resource in the Rules fields. The fields are:

referral	Displays the current referral.
Name	This field allows you to enter the name of the referral.

3. Click Create to save the referral.

To remove a referral from a policy, select the referral and click Remove.

You can edit any referral definition by clicking on the Edit link next to the rule name.

Creating Policies for Peer and Suborganizations

In order to create policies for peer or suborganizations, you must first create a referral policy in the parent (or another peer) organization. The referral policy must contain, in its rule definition, the resource prefix that is being managed by the suborganization. Once the referral policy is created in the parent organization (or another peer organization), normal policies can be created at the suborganization (or peer organization).

The Identity Server policy framework does not allow the creation of referral policies if the action name does not contain resource names. In other words, if the action does not include any resource names, policies can only be created under the root organization, not under the suborganization.

In this example, `o=isp` is the parent organization, `o=sun.com` is the suborganization and manages resources and sub-resources of `http://www.sun.com`. To create a policy for this suborganization, follow these steps:

1. Create a referral policy at `o=isp`. For information on referral policies, see the procedure “Modify a Referral Policy,” on page 93.

The referral policy must define `http://www.sun.com` as the resource in the rule, and must contain a `SubOrgReferral` with `sun.com` as the value in the referral.

2. Go to the Organization view and navigate to the suborganization `sun.com`.
3. Ensure that the policy configuration service is registered at the suborganization level, `sun.com`. For information, see “Registering Policy Configuration Services,” on page 86.
4. Now that the resource is referred to `sun.com` by `isp`, normal policies can be created for the resource `http://www.sun.com`, or for any resource starting with `http://www.sun.com`.

See the procedure “Modify a Normal Policy,” on page 88 for information on creating normal policies.

To define policies for other resources managed by `sun.com`, additional referral policies must be created at `isp`.

Authentication Options

Sun ONE Identity Server provides a framework for authentication, a process which verifies the identities of users accessing applications within an enterprise. A user must pass an authentication process before accessing the Identity Server console, or any other Identity Server-protected resource. Authentication is implemented through plug-ins that validate the user's identity. (This plug-in architecture is described more fully in the *Sun One Identity Server Programmer's Guide*.)

The Identity Server console is used to set the default values, to register authentication services, to create an authentication template and to enable the service. This chapter provides an overview of the authentication services and instructions for registering them. It contains the following sections:

- Core Authentication
- Anonymous Authentication
- Certificate-based Authentication
- LDAP Directory Authentication
- Membership Authentication
- NT Authentication
- RADIUS Server Authentication
- SafeWord Authentication
- Unix Authentication
- Authentication Configuration
- Authentication By Authentication Level
- Authentication By Module

Core Authentication

Identity Server provides, by default, nine different authentication services, as well as a Core authentication service. The Core authentication service provides overall configuration for the authentication service. Before registering and enabling Anonymous, Certificate-based, LDAP, Membership, NT, RADIUS, SafeWord, and Unix authentication, the Core authentication must be registered and enabled. Chapter 17, “Core Authentication Attributes” contains a detailed listing of the Core attributes.

To Register and Enable the Core Service

1. Navigate to the navigation pane of the Organization for which the Core service is to be registered.
2. Choose Services from the View menu.
3. Click Register in the navigation pane.

A list of available services displays in the data pane.

4. Select the checkbox for Core Authentication and click Register.

The Core Authentication service will appear in the navigation pane assuring the administrator that it has been registered.

5. Click the Core Authentication Properties arrow.

The message *No template available for this service* appears in the Data pane.

6. Click Create.

The Core attributes appear in the data pane. Modify the attributes as necessary. An explanation of the Core attributes can be found in Chapter 17, “Core Authentication Attributes” or by clicking the Help link in the upper right hand corner of the console.

Anonymous Authentication

By default, when this module is enabled, a user can log in to Identity Server as an *anonymous* user. A list of anonymous users can also be defined for this module by configuring the Valid Anonymous User List attribute (see page 166). Granting anonymous access means that it can be accessed without providing a password. Anonymous access can be limited to specific types of access (for example, access for read or access for search) or to specific subtrees or individual entries within the directory.

To Register and Enable Anonymous Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator.

1. Navigate to the navigation pane of the Organization for which Anonymous Authentication is to be registered.
2. Choose Services from the View menu.

The Core service, if already registered, displays in the navigation pane. If it is not already registered, it can be done concurrently with the Anonymous Authentication service.

3. Click Register in the navigation pane.

A list of available services displays in the data pane.

4. Select the checkbox for Anonymous Authentication and click Register.

The Anonymous Authentication service will appear in the navigation pane assuring the administrator that it has been registered.

5. Click the Anonymous Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.

6. Click Create.

The Anonymous Authentication attributes appear in the data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 15, “Anonymous Authentication Attributes” or by clicking the Help link in the upper right hand corner of the console.

7. Click Save.

The Anonymous Authentication service has been enabled.

Logging In Using Anonymous Authentication

In order to log in using Anonymous Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define Anonymous Authentication. This ensures that when the user logs in using

`http://<hostname>:<port>/<deploy_URI>/UI/Login?module=Anonymous` the user will see the Anonymous Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

NOTE	The Default Anonymous User Name attribute value in the Anonymous Authentication service is <code>anonymous</code> . This is the name users use to log in. A default Anonymous User must be created within the organization. The user id should be identical to the user name specified in the Anonymous Authentication attributes.
-------------	--

Certificate-based Authentication

Certificate-based Authentication involves using a personal digital certificate (PDC) to identify and authenticate a user. A PDC can be configured to require a match against a PDC stored in Directory Server, and verification against a Certificate Revocation List.

There are a number of things that need to be accomplished before registering the Certificate-based Authentication service to an organization. First, the Sun ONE Web Server that is installed with the Identity Server needs to be secured and configured for Certificate-based Authentication. Before enabling the Certificate-based service, see Chapter 5, “Securing Your Web Server” in the *iPlanet Web Server 6.0 Administrator's Guide* for these initial Web Server configuration steps. This document can be found at the following location:

`http://docs.sun.com/db/doc/816-5691-10`

NOTE Each user that will authenticate using the certificate-based service must request a PDC for their browser. Instructions are different depending upon the browser used. See your browser's documentation for more information.

To Register and Enable Certificate-based Authentication

You must log in to Identity Server as the Organization Administrator.

1. Navigate to the navigation pane of the Organization for which Certificate-based Authentication is to be registered.
2. Choose Services from the View menu.

The Core service, if already registered, displays in the navigation pane. If it is not already registered, it can be done concurrently with the Certificate-based Authentication service.
3. Click Register in the navigation pane.

A list of available services displays in the data pane.
4. Select the checkbox for Certificate-based Authentication and click Register.

The Certificate-based Authentication service will appear in the navigation pane assuring the administrator that it has been registered.
5. Click the Certificate-based Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.
6. Click Create.

The Certificate-based Authentication attributes appear in the data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 16, "Certificate Authentication Attributes" or by clicking the Help link in the upper right hand corner of the console.
7. Click Save.

Logging In Using Certificate-based Authentication

In order to log in using Certificate-based Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define Certificate-based Authentication. This ensures that when the user logs in using

`http://<hostname>:<port>/<deploy_URI>/UI/Login?module=Cert`, they will see the Certificate-based Authentication login window. Based on the authentication type that is being used (such as role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

LDAP Directory Authentication

With the LDAP Authentication service, when a user logs in, he or she is required to bind to the LDAP Directory Server with a specific user DN and password. This is the default authenticating module for all organization-based authentication. If the user provides a user id and password that are in the Directory Server, the user is allowed access to, and is set up with, a valid Identity Server session. LDAP Authentication is enabled by default when Identity Server is installed. The following instructions are provided in the event that the service is disabled.

To Register and Enable LDAP Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator.

1. Navigate to the navigation pane of the Organization for which LDAP Authentication is to be registered.
2. Choose Services from the View menu.

The Core service, if already registered, displays in the navigation pane. If it is not already registered, it can be done concurrently with the LDAP Authentication service.

3. Click Register in the navigation pane.

A list of available services displays in the data pane.

4. Select the checkbox for LDAP Authentication and click Register.

The LDAP Authentication service will appear in the navigation pane assuring the administrator that it has been registered.

5. Click the LDAP Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.

6. Click Create.

The LDAP Authentication attributes appear in the data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 18, “LDAP Authentication Attributes” or by clicking the Help link in the upper right hand corner of the console.

7. Enter the password in the Password for Root User Bind attribute. By default, the `amldapuser` password that was entered during installation is used as the bind user.

To use a different bind user, change the DN of the user in the DN For Root User Bind attribute, and enter the password for that user in the Password for Root User Bind attribute.

8. Click Save.

The LDAP Authentication service has been enabled.

Logging In Using LDAP Authentication

In order to log in using LDAP Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define LDAP Authentication. This ensures that when the user logs in using `http://<hostname>:<port>/<deploy_URI>/UI/Login?module=LDAP`, they will see the LDAP Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Enabling LDAP Authentication Failover

The LDAP authentication attributes include a value field for both a primary and a secondary Directory Server. Identity Server will look to the second server for authentication if the primary server becomes unavailable. For more information, see the LDAP attributes “Primary LDAP Server and Port,” on page 186 and “Secondary LDAP Server and Port,” on page 186.

Membership Authentication

Membership authentication is implemented similarly to personalized sites such as `my.site.com`, or `mysun.sun.com`. When this service is enabled, a user creates an account and personalizes it without the aid of an administrator. With this new account, the user can access it as a registered user. The user can also access the viewer interface, saved on the user profile database as authorization data and user preferences.

To Register and Enable Membership Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator.

1. Navigate to the navigation pane of the Organization for which Membership Authentication is to be registered.
2. Choose Services from the View menu.

The Core service, if already registered, displays in the navigation pane. If it is not already registered, it can be done concurrently with the Membership Authentication service.

3. Click Register in the navigation pane.

A list of available services displays in the data pane.

4. Select the checkbox for Membership Authentication and click Register.

The Membership Authentication service will appear in the navigation pane assuring the administrator that it has been registered.

5. Click the Membership Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.

6. Click Create.

The Membership Authentication attributes appear in the data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 19, “Membership Authentication Attributes” or by selecting the Help link in the upper right hand corner of the console.

7. Enter the password in the Password for Root User Bind attribute. By default, the `amldapuser` password that was entered during installation is used as the bind user.

To use a different bind user, change the DN of the user in the DN For Root User Bind attribute, and enter the password for that user in the Password for Root User Bind attribute.

8. Click Save.

The Membership Authentication service has been enabled.

Logging In Using Membership Authentication

In order to log in using Membership Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define Membership Authentication. This ensures that when the user logs in using

`http://<hostname>:<port>/<deploy_URI>/UI/Login?module=Membership`, they will see the Membership Authentication login (Self Registration) window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

NT Authentication

Identity Server can be configured to work with an NT /Windows 2000 server that is already installed. Identity Server provides the client portion of NT authentication. The NT server may exist on the system on which Identity Server is installed, or on a separate system.

1. Configure the NT server.

For detailed instructions, see the NT server documentation.

2. Before you can register and enable the NT authentication service, you must obtain and install a Samba client to communicate with Identity Server on your Solaris system. For more information, see “NT Authentication Attributes,” on page 197.
3. Add the module class to the Pluggable Auth Module Classes attribute in the Core Authentication Service. To do so:
 - a. Select Service Configuration in the Identity Server Console.
 - b. Click on the Properties arrow for the Core Authentication Service.
 - c. Add the module class for the NT authentication service in the Pluggable Auth Module attribute. For example:

```
com.sun.identity.authentication.modules.nt.NT
```

For more information on this attribute, see “Pluggable Auth Module Classes,” on page 174.

4. Register and enable the NT authentication service.

To Register and Enable NT Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator.

1. Navigate to the navigation pane of the Organization for which NT Authentication is to be registered.
2. Choose Services from the View menu.

The Core service, if already registered, displays in the navigation pane. If it is not already registered, it can be done concurrently with the NT Authentication service.

3. Click Register in the navigation pane.

A list of available services displays in the data pane.

4. Select the checkbox for NT Authentication and click Register.

The NT Authentication service will appear in the navigation pane assuring the administrator that it has been registered.

5. Click the NT Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.

6. Click Create.

The NT Authentication attributes appear in the data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 20, “NT Authentication Attributes” or by selecting the Help link in the upper right hand corner of the console.

7. Click Save.

The NT Authentication service has been enabled.

Logging In Using NT Authentication

In order to log in using NT Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define NT Authentication. This ensures that when the user logs in using `http://<hostname>:<port>/<deploy_URI>/UI/Login?module=NT`, they will see the NT Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

RADIUS Server Authentication

Identity Server can be configured to work with a RADIUS server that is already installed. This is useful if there is a legacy RADIUS server being used for authentication in your enterprise. Enabling the RADIUS authentication service is a two-step process.

1. Configure the RADIUS server.

For detailed instructions, see the RADIUS server documentation.

2. Register and enable the RADIUS authentication service.

To Register and Enable RADIUS Authentication

You must log in to Identity Server as the Organization Administrator.

1. Navigate to the navigation pane of the Organization for which RADIUS Authentication is to be registered.

2. Choose Services from the View menu.

The Core service, if already registered, displays in the navigation pane. If it is not already registered, it can be done concurrently with the RADIUS Authentication service.

3. Click Register in the navigation pane.

A list of available services displays in the data pane.

4. Select the checkbox for RADIUS Authentication and click Register.

The RADIUS Authentication service will appear in the navigation pane assuring the administrator that it has been registered.

5. Click the RADIUS Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.

6. Click Create.

The RADIUS Authentication attributes appear in the data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 21, “RADIUS Authentication Attributes” or by selecting the Help link in the upper right hand corner of the console.

7. Click Save.

The RADIUS Authentication service has been enabled.

Logging In Using RADIUS Authentication

In order to log in using RADIUS Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define RADIUS Authentication. This ensures that when the user logs in using `http://<hostname>:<port>/<deploy_URI>/UI/Login?module=RADIUS`, they will see the RADIUS Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

SafeWord Authentication

Identity Server can be configured to handle SafeWord Authentication requests to Secure Computing's SafeWord™ or SafeWord PremierAccess™ authentication servers. Identity Server provides the client portion of SafeWord authentication. The SafeWord server may exist on the system on which Identity Server is installed, or on a separate system.

To Register and Enable SafeWord Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator.

1. Navigate to the navigation pane of the Organization for which SafeWord Authentication is to be registered.
2. Choose Services from the View menu.

The Core service, if already registered, displays in the navigation pane. If it is not already registered, it can be done concurrently with the SafeWord Authentication service.

3. Click Register in the navigation pane.

A list of available services displays in the data pane.

4. Select the checkbox for SafeWord Authentication and click Register.

The SafeWord Authentication service will appear in the navigation pane, assuring the administrator that it has been registered.

5. Click the SafeWord Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.

6. Click Create.

The SafeWord Authentication attributes appear in the data pane. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 22, "SafeWord Authentication Attributes," or by clicking the Help link on the upper right corner of the console.

7. Click Save.

The SafeWord Authentication service has been enabled.

Logging In Using SafeWord Authentication

In order to log in using SafeWord Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define SafeWord Authentication. This ensures that when the user logs in using `http://<hostname>:<port>/<deploy_URI>/UI/Login?module=SAFEWORD`, they will see the SafeWord Authentication login window. Based on the authentication type that is being used (such as role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Unix Authentication

Identity Server can be configured to process authentication requests against Unix userids and passwords known to the Solaris system on which Identity Server is installed. While there is only one organizational attribute, and a few global attributes for Unix authentication, there are some system-oriented considerations.

In order to authenticate locally-administered userids (see `admintool (1M)`), root access is required. If Identity Server is installed to run as `nobody`, or a userid other than root, then the `<install_dir>/SUNWam/bin/doUnix` process must still execute as root. The `passwd` entry in the `/etc/nsswitch.conf` file determines whether the `/etc/passwd` and `/etc/shadow` files, or NIS are consulted for authentication.

Unix Authentication makes use of an authentication *helper*, which is a separate process from the main Identity Server process. Upon startup, this helper listens on a port for configuration information. There is only one Unix helper per Identity Server to serve all of its organizations. The Unix authentication service is not available on the Windows platform.

To Register and Enable Unix Authentication

You must log in to the Identity Server as Top-Level Administrator for the following steps.

1. Select the Service Configuration module.
2. Click on the Unix Authentication Properties arrow in the Service Name list.

Several Global and one Organization attributes are displayed. Because one Unix helper serves all of the Identity Server server's organizations, most of the Unix attributes are global. An explanation of these attributes can be found in Chapter 23, "Unix Authentication Attributes," or by clicking the Help link in the upper right corner of the console.

3. Click Save to save the new values for the attributes.

You may log in to Identity Server as the Organization Administrator to enable Unix Authentication for an organization.

4. Navigate to the navigation pane of the Organization for which Unix Authentication is to be registered.

5. Choose Services from the View menu.

The Core service, if already registered, displays in the Navigation pane. If it is not already registered, it can be done concurrently with the Unix Authentication service.

6. Click Register in the navigation pane.

A list of available services displays in the data pane.

7. Select the checkbox for Unix Authentication and click Register.

The Unix Authentication service will appear in the Navigation pane, assuring the administrator that it has been registered.

8. Click the Unix Authentication Properties arrow.

The message *No template available for this service* appears in the data pane.

9. Click Create.

The Unix Authentication organization attribute appears in the data pane. Modify the Authentication Level attribute as necessary. An explanation of this attribute can be found in Chapter 23, "Unix Authentication Attributes," or by clicking the Help link in the upper right corner of the console.

10. Click Save.

The Unix Authentication service has been enabled.

Logging In Using Unix Authentication

In order to log in using Unix Authentication, the Core Authentication service attribute “Organization Authentication Modules,” on page 176 must be modified to define Unix Authentication. This ensures that when the user logs in using `http://<hostname>:<port>/<deploy_URI>/UI/Login?module=Unix`, the user will see the Unix Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Authentication Configuration

The Authentication Configuration service is used to define authentication modules for any of the following authentication types:

- organization
- role
- service
- user

Once an authentication module is defined for one of these authentication types, the module can be configured to supply redirect URLs, as well as a post-processing Java class specification, based on a successful or failed authentication process.

Before an authentication module can be configured, the Core authentication service attribute Organization Authentication Modules must be modified to include the specific authentication module name.

Authentication Configuration User Interface

The Authentication Configuration services allows you to define one or more authentication services (or *modules*) that a user must pass before being allowed access to the console or any secured resource within Identity Server. Organization, role, service, and user-based authentication use a common user interface to define the authentication modules. (Instructions for access the Authentication Configuration interface for specific object types are described in subsequent sections).

1. Click on the Edit link next to the object's Authentication Configuration attribute to display the Module List window.
2. This window lists the authentication modules that have been assigned to the object. If no modules exist, click Add to display the Add Module window.

The Add Module Window contains three files to define:

Module Name	<p>This pull-down list allows you to select the authentication modules (including custom modules that may be added) available to Identity Server. By default, the modules are:</p> <ul style="list-style-type: none">• LDAP• Cert• Anonymous• SafeWord• Membership• NT• RADIUS• Unix
-------------	---

Flag

This pull-down menu allows you specify the authentication module requirements. It can be one of:

- **REQUIRED** - The authentication module is required to succeed. If it succeeds or fails, authentication continues to proceed down the authentication module list.
- **REQUISITE** - The authentication module is required to succeed. If it succeeds, authentication continues down the authentication module list. If it fails, control returns to the application (authentication does not proceed down the authentication module list.)
- **SUFFICIENT** - The authentication module is not required to succeed. If it does succeed, control immediately returns to the application (authentication does not proceed down the authentication module list.). If it fails, authentication continues down the list.
- **OPTIONAL** - The authentication module is not required to succeed. If it succeeds or fails, authentication still continues to proceed down the list.

These flags establish an enforcement criteria for the authentication module for which they are defined. There is hierarchy for enforcement, with **REQUIRED** being the highest, and **OPTION** being the lowest.

For example, if an administrator defines an LDAP module with the **REQUIRED** flag, then the user's credential must pass the LDAP authentication requirements to access a given resource.

If you add multiple authentication modules and for each module the Flag is set to **REQUIRED**, the user must pass all authentication requirements before being granted access.

For more information on the flag definitions, refer to the JAAS (Java Authentication and Authorization Service) located at:

<http://java.sun.com/security/jaas/doc/module.html>

Option Allows for additional options for the for the module as a key=value pair. Multiple options are separated by a space.

Figure 7-1 Add Module List Window For A User

The screenshot shows a dialog box titled "Add Module". It contains the following fields and controls:

- Module Name: ***: A dropdown menu with "LDAP" selected.
- Flag: ***: A dropdown menu with "REQUIRED" selected.
- Option:**: A text input field with a cursor inside.
- Buttons**: "OK" and "Cancel" buttons at the bottom right.

3. Once the fields are selected, click OK to return to the Module List window. The authentication modules you have defined are listed in this window. Click Save.

You can add as many authentication modules to this list as you wish. Adding multiple authentication modules is called *chaining*. If you are chaining authentication modules, note that the order in which they are listed defines the order of hierarchy of enforcement.

To change the order of the authentication modules:

- a. Click the Reorder button.
- b. Select the module you wish to reorder.
- c. Use the Up and Down buttons to place it in the desired position.

Figure 7-2 Module List Window For A User

The screenshot shows a window titled "Module List - user1". Inside the window, there are three buttons at the top: "Add...", "Delete", and "Reorder". Below these buttons is a table with three columns: "Module", "Flag", and "Option". The table contains three rows of data, each with a checkbox in the "Module" column.

Module	Flag	Option
<input type="checkbox"/> LDAP	REQUIRED	
<input type="checkbox"/> Cert	REQUIRED	
<input type="checkbox"/> Anonymous	OPTIONAL	

At the bottom right of the window, there are two buttons: "Save" and "Close".

4. To remove any authentication module from the list, select the checkbox next to the authentication module and click Delete.

Authentication Configuration for Organizations

Authentication modules are set for an organization by first registering the Core Authentication service to the organization.

To configure the organization's authentication attributes:

1. Navigate to the organization for which you will configure the authentication attributes.
2. Select Services from the View menu.
3. Click the Core Properties arrow in the service listing.
The Core authentication attributes are displayed in the data frame.
4. Click the edit link next to the Admin Authenticator attribute. This allows you to define the authentication services for administrators only. An administrator is a user who needs access to the Identity Server console. This attribute can be used if the authentication module for administrators needs to be different from the module for end users. The default authentication module is LDAP.

Once you have defined the authentication services, click Save to save the changes, and click Close to return to the Core Authentication attributes for organizations.

5. Click the Edit link next to the Organization Authentication Configuration attribute. This allows you to define authentication modules for all users within the organization. The default authentication module is LDAP.
6. Once you have defined the authentication services, click Save to save the changes, and click Close to return to the Core Authentication attributes for organizations.

Authentication Configuration for Roles

Authentication modules are set for roles after registering the Authentication Configuration service at the role level.

1. Navigate to the organization for which you will configure the authentication attributes.
2. Choose Roles from the View menu.
3. Select the role for which to set the authentication configuration and click on the Properties arrow.

The role's properties are displayed in the data frame.

4. Select Services from the View menu in the data frame.
5. Modify the Authentication Configuration attributes as necessary. An explanation of these attributes can be found in Chapter 24, "Authentication Configuration Attributes," or by clicking the Help link in the upper right corner of the console.
6. Click Save.

NOTE	If you are creating a new role, the Authentication Configuration service is not automatically assigned to it. Make sure that you select the Authentication Configuration service option at the top of the role profile page before you create it.
-------------	---

Authentication Configuration for Services

Authentication modules are set for services after registering the Authentication Configuration service. To do so:

1. Choose Services from the View menu in the Identity Management module.

The list of registered services are displayed. If the Authentication Configuration service is not registered, continue with the steps below. If the service is registered, skip to step Step 4.

2. Click Register in the Navigation Pane.

A list of available services is displayed in the data pane.

3. Select the checkbox for Authentication Configuration and click Register.

The Authentication Configuration service will appear in the navigation pane assuring the administrator that it has been registered.

4. Click the Authentication Configuration Properties arrow.

The Service Instance List is displayed in the in the data pane.

5. Click on the service instance for which to configure the authentication modules.

6. Modify the authentication configuration attributes and click Save. An explanation of these attributes can be found in Chapter 24, "Authentication Configuration Attributes," or by clicking the Help link in the upper right corner of the console.

Authentication Configuration for Users

1. Choose Users from the View menu in the Identity Management module.

The list of users is displayed in the navigation pane.

2. Select the user you wish to modify and click the Properties arrow.

The user profile is displayed in the data pane.

NOTE If you are creating a new user, the Authentication Configuration service is not automatically assigned to the user. Make sure that you select the Authentication Configuration service option at the top of the user profile page before you create the user. If this option is not selected, the user will not inherit the authentication configuration defined at for the role.

3. To ensure that the Authentication Configuration service is assigned to the user, Select Services from the View menu. If assigned, the Authentication Configuration service will be listed as an assigned service.
4. Select User from the View menu in the data pane.
5. Click on the Edit link next to the User Authentication Configuration attribute to define the authentication modules for the user.
6. Click Save.

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.

The authentication level will be set on a user's SSO token after the user has successfully authenticated to the module. If the user is required to authenticate to multiple authentication modules, and does so successfully, the highest authentication level value will be set in user's SSO token.

If a user attempts to access a service, the service can determine if the user is allowed access by checking the authentication level in user's SSO token. It then redirects the user to the go through the authentication modules with a set authentication level.

Users can also access authentication modules with specific authentication level. For example, a user performs a login 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 be 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.

Authentication By Module

Users can access a specific authentication module using the following syntax:

```
http://<hostname>:<port>/<deploy_uri>/UI/Login?module=<module_name>
```

Before the authentication module can be accessed, the Core authentication service attribute `Organization Authentication Modules` must be modified to include the authentication module name. If the authentication module name is not included in this attribute, the “authentication module denied” page will be displayed when the user attempts to authenticate. For more information, see “Organization Authentication Modules,” on page 176.

Command Line Reference Guide

This is the Command Line Reference Guide, part two of the Sun ONE Identity Server Administration Guide. This section contains the following chapters:

- The amadmin Command Line Tool
- The amserver Command Line Tool
- The ampassword Command Line Tool
- The am2bak Command Line Tool
- The bak2am Command Line Tool
- The VerifyArchive Command Line Tool

All of the command line tools described in this section can be found in the following default location:

```
Identity_Server_root/opt/SUNWam/bin
```


The amadmin Command Line Tool

This chapter provides information on the `amadmin` command line tool and contains the following sections:

- The amadmin Command Line Tool
- Creating Policies with amadmin

The amadmin Command Line Executable

The primary purposes of the command line executable `amadmin` is to load XML service files into the Directory Server and to perform batch administrative tasks on the DIT. `amadmin` can be found in `Identity_Server_root/SUNWam/bin` and is used to:

- Load XML service files - Administrators load services into Sun One™ Identity Server that use the XML service file format defined in the `sms.dtd`. All services must be loaded using `amadmin`; they cannot be imported through the Identity Server console.

NOTE	XML service files are stored in the Directory Server as static <i>blobs</i> of XML data that is referenced by Identity Server. This information is not used by Directory Server which only understands LDAP.
-------------	--

- Perform batch updates to the DIT - Administrators can perform batch updates to the Directory Server DIT using the batch processing XML file format defined in the `amadmin.dtd`. For example, if an administrator wants to create 10 organizations, 1000 users, and 100 groups, it can be done in one attempt by putting the requests in one or more batch processing XML files and loading them using `amadmin`. More information on this can be found in the “Service Management” chapter in the *Sun One Identity Server Programmer’s Guide*.

NOTE `amadmin` only supports a subset of features that the Sun One™ Identity Server console supports and is not intended as a replacement. It is recommended that the console be used for small administrative tasks while `amadmin` is used for larger administrative tasks.

`amadmin` will not enforce the account expiry date set for a user when it tries to bind to Directory Server to authenticate the user.

The amadmin Syntax

There are a number of structural rules that must be followed in order to use `amadmin`. The generic syntaxes for using the tool are:

- `amadmin -u | --runasdn <dnname> -w | --password <password> [-l | --locale <localename>] [[-v | --verbose] | [-d | --debug]] -t | --data <xmlfile1> [<xmlfile2> ...]`
- `amadmin -u | --runasdn <dnname> -w | --password <password> [-l | --locale <localename>] [[-v | --verbose] | [-d | --debug]] -s | --schema <xmlfile1> [<xmlfile2> ...]`
- `amadmin -u | --runasdn <dnname> -w | --password <password> [-l | --locale <localename>] [[-v | --verbose] | [-d | --debug]] -r | --deleteService <serviceName1> [<serviceName2> ...]`
- `amadmin -u | --runasdn <dnname> -w | --password <password> or -f | --password file <passwordfile> [-l | --locale <localename>] [[-v | --verbose] | [-d | --debug]] -m | --session <servername> <pattern>`
- `amadmin -h | --help`
- `amadmin -n | --version`

NOTE Two hyphens must be entered exactly as shown in the syntax.

amadmin Options

Following are definitions of the `amadmin` command line parameter options:

--runasdn

`--runasdn` is used to authenticate the user to the LDAP server. The argument is a value equal to that of the Distinguished Name (DN) of the user authorized to run `amadmin`; for example

```
--runasdn uid=amAdmin,ou=People,o=iplanet.com,o=isp.
```

The DN can also be formatted by inserting spaces between the domain components and double quoting the entire DN such as: `--runasdn "uid=amAdmin, ou=People, o=iplanet.com, o=isp"`.

--password

`--password` is a mandatory option and takes a value equal to that of the password of the DN specified with the `--runasdn` option.

--locale

`--locale` is an option that takes a value equal to that of the name of the locale. This option can be used for the customization of the message language. If not provided, the default locale, `en_US`, is used.

--continue

`--continue` is an option that will continue to process the XML files even if there are errors. For example, if there are three XML files to be loaded at the same time, and the first XML file fails, `amadmin` will continue to load the remaining files.

--session

`--session (-m)` is an option to manage the sessions, or to display the current sessions. When specifying `--runasdn`, it must be the same as the DN for the super user in `AMConfig.properties`, or just ID for the top-level admin user.

The following example will display all sessions for a particular service host name,:

```
amadmin -u uid=amadmin,ou=people,dc=iplanet,dc=com -v -w
12345678 -m http://sun.com:58080
```

The following example will display a particular user's session:

```
amadmin -u uid=amadmin,ou=people,dc=iplanet,dc=com -v -w
12345678 -m http://sun.com:58080 <username>
```

The `username` attribute is case sensitive. So, for example, if you specify `amadmin` instead of `amAdmin` as the `username`, nothing will be returned.

You can terminate a session by entering the corresponding index number, or enter multiple index numbers (with spaces) to terminate multiple sessions.

--debug

`--debug` is an option that will write messages to the `amAdmin` file created under the `identity_server_root/SUNWam/web-apps/services/debug` directory. These messages are technically-detailed but not i18n-compliant.

--verbose

`--verbose` is an option that prints to the screen the overall progress of the `amadmin` command. It does not print to a file the detailed information. Messages output to the command line are i18n-compliant.

--data

`--data` is an option that takes as its value the name of the batch processing XML file being imported. One or more XML files can be specified. This XML file can create, delete and read various directory objects as well as register and unregister services. For more information on what types of XML files can be passed to this option, see the “Servic Management” chapter in the *Sun ONE Identity Server Programmer’s Guide*.

--schema

`--schema` is an option that loads the attributes of an Sun One™ Identity Server service into the Directory Server. It takes as an argument an XML service file in which the service attributes are defined. This XML service file is based on the `sms.dtd`. One or more XML files can be specified.

NOTE	Either the <code>--data</code> or <code>--schema</code> option must be specified, depending on whether configuring batch updates to the DIT, or loading service schema and configuration data.
-------------	--

--deleteservice

`--deleteservice` is an option for deleting a service and its schema only.

--serviceName

`--serviceName` is an option that takes a value equal to the service name which is defined under the `Service name=...` tag of an XML service file. This portion is displayed in Code Example 8-1 on page 127.

Code Example 8-1 Portion of sampleMailService.xml

```

...
<ServicesConfiguration>
  <Service name="sampleMailService" version="1.0">
    <Schema
      serviceHierarchy="/other.configuration/sampleMailService"
      i18nFileName="sampleMailService"
      i18nKey="iplanet-am-sample-mail-service-description">
    ...

```

--help

--help is an argument that displays the syntax for the amadmin command.

--version

--version is an argument that displays the utility name, product name, product version and legal notice.

Creating Policies with amadmin

Policies can be administered through amadmin, however they cannot be modified using amadmin directly. To modify the policy, you must first delete the policy and then add the modified policy using amadmin.

To add policies using amadmin, the policy XML file must be developed following the policy.dtd. (policy.dtd is described in the *Sun One Identity Server Programmer's Guide*) Once the policy's XML file is developed, you can use the following command to load it:

```

<install-dir>/SUNWam/bin/amadmin

--runasdn "uid=amAdmin,ou=People,<default_org>,<root_suffix>"
--password <password>
--data <policy.xml>

```

When creating policies through amadmin, ensure that the authentication module is registered with the organization while creating authentication scheme condition; that the corresponding LDAP objects (organizations, groups, roles and users) exist while creating Organization, LDAP groups', LDAP roles' and LDAP users' subjects; that Identity Server roles exist while creating IdentityServerRoles subjects; and that the relevant organizations exist while creating sub organization or peer organization referrals.

The amserver Command Line Tool

This chapter provides information on the `amserver` command line tool. This chapter contains the following sections:

- The `amserver` Command Line Executable
- Using `amserver` for Multi-Server Installer Administration

The amserver Command Line Executable

The `amserver` command line executable is to create, start, stop, and delete additional Identity Server instances on the Solaris platform. `amserver` on the Windows 2000 platform only allows for starting and stopping Identity Server.

amserver Syntax

The generic syntax for the tools is:

```
./amserver { create | delete [<instance_name>] | startall | start |  
stop | stopall | version }
```

amserver Commands for Solaris

create

`create` is a command that is used to create a new instance of Identity Server. The `amserver` script should be run as root. To create instances run `amserver` script `./amserver create`. Detailed steps for creating multiple server instances are described in “Using `amserver` for Multi-Server Installer Administration,” on page 130.

startall

`startall` is a command that is used to start all the Identity server instances. To start individual instance run:

```
Identity_Server_root/SUNWam/bin/amserver.<instance_name> start
```

stopall

`stopall` is a command that is used to stop all the Identity Server instances. To stop individual identity server instance run:

```
/opt/SUNWam/bin/amserver.<instance_name> stop
```

delete

`delete` is a command that will delete all the files created by the `create` option.

amserver Commands for Windows 2000

amserver on the Windows 2000 platform only supports the following commands:

start

`start` is a command that starts the Identity Server.

stop

`stop` is a command that stops the Identity Server.

restart

`restart` is a command that restarts Identity Server

`amserver` cannot stop or start Directory Server if its not installed with Identity Server packages. You may need to start it manually

Using amserver for Multi-Server Installer Administration

You can use the `amserver` command line utility to install and administer multiple instances of Identity Server. Before installing multiple instances of Identity Server, you must log in as root. The scripts described in the steps below can be found in `<Identity_Server_root>/SUNWam/bin`.

To install multiple instances:

1. Create a new server instance through amServer by entering `./amserver create`.

For example, if you were to create instances named `instance1` which will listen to port `81`., the output of the script output may look like the following:

```
#####
#####

Please enter the name of the server instance: instance1
Please enter the port number: 81
Do you want to create more server instances? y/[n]
Installing... please wait...

#####
##
```

- a. A directory is then created for each web server instance. Example:

```
<Identity_Server_root>/SUNWam/servers/https-<instance_name>
```

- b. The Identity Server applications are deployed to the following location:

```
<Identity_Server_root>/SUNWam/servers/web-apps-<instance_name>
```

- c. The `<Identity_Server_root>/SUNWam/bin` directory holds the instance specific version of amServer. For example:

```
amserver.<instance_name>
```

- d. A copy of the Identity Server configuration file is created in

```
<Identity_Server_root>/SUNWam/lib/AMConfig-<instance_name>.properties.
```

- e. The file `/etc/rc3.d`, holds the instance specific version of the initialization files:

```
S55amserver.<instance_name>
```

```
K55amserver.<instance_name>
```

NOTE Do not use “_” (under score_) or “.” (period) in the creation of the instance name

2. Start all Identity Server instances, including the original server instance, by entering:

```
./amserver startall
```

You can alternatively use the following command to start individual servers:

```
<Identity_Server_root>/SUNWam/bin/amserver.<instance_name> start
```

You should now be able to invoke the Identity Server login screens for all instances through your browser.

3. Stop all server instances, including the original, by entering:

```
./amserver stopall
```

Alternatively, you can use the following command to stop individual servers:

```
<Identity_Server_root>/SUNWam/bin/amserver.<instance_name> stop
```

4. Invoke the Delete Command option by entering:

```
./amserver delete
```

All of the files created by the Create command should be removed. If you use the Identity Server Uninstall utility, the files generate by the scripts are not removed.

5. Specify your directories for the debug files by entering:

```
Edit
<Identity_Server_root>/SUNWam/lib/AMConfig-<instance_name>.properties
```

Make sure that you change the `com.ipplanet.services.debug.directory` property to your designated directory.

6. Invoke the `ammultiserverinstall` utility by using the following syntax:

```
ammultiserverinstall [ server-instance-name ] [ port-number ]
```

For applications that require the installation of multiple instances of Identity Server, but prefer a non-interactive interface, use the `ammultiserverinstall` utility. If the `ammultiserverinstall` fails, it will exit with a value of 1.

7. `amserver` will automatically add server instances to the Platform Server List.

8. Configure Identity Server to run in SSL mode. Instructions for this found in Appendix A, “Configuring Identity Server in SSL Mode” of this manual.
9. Enter the following command to start all of the Identity Server instances:

```
./amserver startall
```

Alternatively, you can use the following command to start individual Identity Server instances:

```
./amserver-<instance> start
```


The ampassword Command Line Tool

This chapter provides information on the `amPassword` command line tool and contains the following sections:

- The ampassword Command Line Executable
- Running ampassword on SSL

The ampassword Command Line Executable

Identity Server contains an `ampassword` utility under `$installroot/bin`. This utility allows you change the Identity Server password for the administrator or user.

The ampassword Syntax

The generic syntax for using the `ampassword` tool is:

```
ampassword -a | --admin [ -o | --old <oldPassword> -n | --new  
<newPassword> ]
```

```
ampassword -p | --proxy [ -o | --old <oldPassword> -n | --new  
<newPassword> ]
```

```
ampassword -e | --encrypt [ password ]
```

NOTE Two hyphens must be entered exactly as shown in the syntax.

ampassword Options

--admin

--admin is used to change the admin password.

--proxy

--proxy is used to change the proxy password. It corresponds to the proxy user (user type proxy in serverconfig.xml.)

--encrypt

--encrypt is used to encrypt the password. It is printed to the command line.

Running ampassword on SSL

To run ampassword with Identity Server running in Secure-Socket Layer (SSL) mode:

1. Modify the serverconfig.xml file, located in the following directory:

Identity_Server_root/SUNWam/config/ums

2. Change port the server attribute to the SSL port which Identity Server is running.
3. Change the type attribute to SSL.

For example:

```
<iPlanetDataAccessLayer>

<ServerGroup name="default" minConnPool="1" maxConnPool="10">

  <Server name="Server1" host="sun.com" port="636" type="SSL"
/>

  <User name="User1" type="proxy">

    <DirDN>

      cn=puser,ou=DSAME Users,dc=iplanet,dc=com
```



```
</DirDN>

<DirPassword>

    AQIC5wM2LY4Sfcy+AQBQxghVwhBE92i78cqf

</DirPassword>

</User> ...
```

Running ampassword on SSL

The am2bak Command Line Tool

This chapter provides information on the `am2bak` command line tool and contains the following section:

- The `am2bak` Command Line Executable

The am2bak Command Line Executable

Identity Server contains an `am2bak` utility under `<Identity_Server_root>/SUNWam/bin`. This utility performs a backup of either all or optional components of Identity Server. Directory Server must be running while taking the log backup.

The am2bak Syntax

The generic syntax for using the `am2bak` tool for the Solaris operating system is:

```
./am2bak [ -v | --verbose ] [ -k | --backup <backup-name> ] [ -l |
--location <location> ] [[-c | --config] | [-b | --debug] | [-g |
--log] | [-t | --cert] | [-d | --ds] | [-a | --all]]*
./am2bak -h | --help
./am2bak -n | --version
```

The generic syntax for using the `am2bak` tool for the Windows 2000 operating system is:

```
am2bak [ -v | --verbose ] [ -k | --backup <backup-name> ] [ -l |
--location <location> ] [[-c | --config] | [-b | --debug] | [-g |
--log] | [-t | --cert] | [-d | --ds] | [-a | --all]]*
am2bak -h | --help
```

```
am2bak -n | --version
```

NOTE Two hyphens must be entered exactly as shown in the syntax.

am2bak Options

--verbose

--verbose is used to run the backup utility in verbose mode.

--backup <backup-name>

--backup <backup-name> defines the name of the backup file. The default is *ambak*.

--location

--location specifies the directory location of the backup. The default location is *<Identity_Server_root>/backup*.

--config

--config specifies backup only for configuration files.

--debug

--debug specifies backup only for debug files.

--log

--log specifies backup only for log files.

--cert

--cert specifies backup only for certificate database files.

--ds

--ds specifies backup only for the Directory Server.

--all

--all specifies a complete backup of the entire Identity Server.

--help

--help is an argument that displays the syntax for the *am2bak* command.

--version

--version is an argument that displays the utility name, product name, product version and legal notice.

Backup Procedure

1. Login as root.

The user running this script must have root access.

2. Run the script ensuring that the correct path is used, if necessary.

The script will backup the following Solaris™ Operating Environment files:

o Configuration and Customization Files:

- identity_server_root/SUNWam/config/
- identity_server_root/SUNWam/locale/
- identity_server_root/SUNWam/servers/httpacl
- identity_server_root/SUNWam/lib/*.properties (Java property files)
- identity_server_root/SUNWam/bin/amserver.<instance-name>
- identity_server_root/SUNWam/servers/https-<all_instances>
- identity_server_root/SUNWam/servers/web-apps-<all_instances>
- identity_server_root/SUNWam/web-apps/services/WEB-INF/config
- identity_server_root/SUNWam/web-apps/services/config
- identity_server_root/SUNWam/web-apps/applications/WEB-INF/classes
- identity_server_root/SUNWam/web-apps/applications/console
- /etc/rc3.d/K55amserver.<all_instances>
- /etc/rc3.d/S55amserver.<all_instances>
- directory_server_root/slapd-<host>/config/schema/
- directory_server_root/slapd-<host>/config/slapd-collations.conf
- directory_server_root/slapd-<host>/config/dse.ldif

o Log And Debug Files:

- var/opt/SUNWam/logs (Sun One™ Identity Server log files)

- `var/opt/SUNWam/install` (Identity Server installation log files)
- `var/opt/SUNWam/debug` (Identity Server debug files)
- **Certificates:**
 - `identity_server_root/SUNWam/servers/alias`
 - `directory_server_root/alias`

The script will also backup the following Microsoft® Windows 2000 operating system files:

- **Configuration and Customization Files:**
 - `identity_server_root/web-apps/services/WEB-INF/config/*`
 - `identity_server_root/locale/*`
 - `identity_server_root/web-apps/applications/WEB-INF/classes/*.properties` (java property files)
 - `identity_server_root/servers/https-<host>/config/jvm12.conf`
 - `identity_server_root/servers/https-<host>/config/magnus.conf`
 - `identity_server_root/servers/https-<host>/config/obj.conf`
 - `directory_server_root/slapd-<host>/config/schema/*.ldif`
 - `directory_server_root/slapd-<host>/config/slapd-collations.conf`
 - `directory_server_root/slapd-<host>/config/dse.ldif`
- **Log And Debug Files:**
 - `var/opt/logs` (Sun One™ Identity Server log files)
 - `var/opt/debug` (Sun One™ Identity Server debug files)
- **Certificates:**
 - `identity_server_root/servers/alias`
 - `identity_server_root/alias`

The bak2am Command Line Tool

This chapter provides information on the `bak2am` command line tool and contains the following section:

- The `bak2am` Command Line Executable

The bak2am Command Line Executable

Identity Server contains an `bak2am` utility under `<Identity_Server_root>/SUNWam/bin`. This utility performs a restore of the Identity Server components that were backed-up by the `am2back` utility.

The bak2am Syntax

The generic syntax for using the `bak2am` tool for the Solaris operating system is:

```
./bak2am [ -v | --verbose ] -z | --gzip <tar.gz-file>
./bak2am [ -v | --verbose ] -t | --tar <tar-file>
./bak2am -h | --help
./bak2am -n | --version
```

The generic syntax for using the `bak2am` tool for the Windows 2000 operating system is:

```
bak2am [ -v | --verbose ] -d | --directory <directory-name>
bak2am -h | --help
bak2am -n | --version
```

NOTE Two hyphens must be entered exactly as shown in the syntax.

bak2am Options

--gzip <backup-name>

`--gzip` specifies the full path and filename of the backup file in `tar.gz` format. By default, the path is `<Identity_Server_root>/backup`. This option is for Solaris only.

--tar <backup-name>

`--tar` specifies the full path and filename of the backup file in `tar` format. By default, the path is `<Identity_Server_root>/backup`. This option is for Solaris only.

--verbose

`--verbose` is used to run the backup utility in verbose mode.

--directory

`--directory` specifies the backup directory. By default, the path is `<Identity_Server_root>/backup`. This option is for Windows 2000 only.

--help

`--help` is an argument that displays the syntax for the `bak2am` command.

--version

`--version` is an argument that displays the utility name, product name, product version and legal notice.

1. Login as root.

The user running this script must have root access.

2. Untar the input tar file.

This was generated when the backup script was run.

The VerifyArchive Command Line Tool

This chapter provides information on the `VerifyArchive` command line tool and contains the following section:

- The `VerifyArchive` Command Line Executable

The VerifyArchive Command Line Executable

The purpose of `VerifyArchive` is to verify the log archives. A log archive is a set of timestamped logs and their corresponding key stores (keystores contain the keys used to generate the MACs and the Digital Signatures which are used to detect tampering of the log files). Verification of an archive detects possible tampering and/or deletion of any file in the archive.

`VerifyArchive` extracts all of the archive sets, and all files belonging to each archive set, for a given `logName`. When executed, `VerifyArchive` searches each log record to for tampering. If tampering is detected, it prints a message specifying which file and the number of the record that has been tampered with..

`VerifyArchive` also checks for any files that have been deleted from the archive set. If a deleted file is detected, it prints a message explaining that verification has failed. If no tampering or deleted files are detected, it returns a message explaining that the archive verification has been successfully completed.

VerifyArchive Syntax

All of the parameters options are required. The syntax is as follows:

```
VerifyArchive -l <logName> -p <path> -u <uname> -w <password>
```

VerifyArchive Options

logName

`logName` refers to the name of the log which is to be verified (such as, `amConsole`, `amAuthentication` and so forth..). `VerifyArchive` verifies the both the access and error logs for the given `logName`. For example, if `amConsole` is specified, the verifier verifies the `amConsole.access` and `amConsole.error` files. Alternatively, the `logName` can be specified as `amConsole.access` or `amConsole.error` to restrict the verification of those logs only.

path

`path` is the full directory path where the log files are stored.

uname

`uname` is the user id of the Identity Server administrator.

password

`password` is the password of the Identity Server administrator.

Attribute Reference Guide

This is the Attribute Reference Guide, part three of the Sun ONE Identity Server Administration Guide. It discusses the configured attributes within Identity Server's default services. This part contains the following chapters:

- Administration Attributes
- Anonymous Authentication Attributes
- Certificate Authentication Attributes
- Core Authentication Attributes
- LDAP Authentication Attributes
- Membership Authentication Attributes
- NT Authentication Attributes
- RADIUS Authentication Attributes
- SafeWord Authentication Attributes
- Unix Authentication Attributes
- Authentication Configuration Attributes
- Client Detection Attributes
- Logging Attributes
- Naming Attributes
- Platform Attributes
- Policy Configuration Attributes
- SAML Attributes
- Session Attributes

- User Attributes
- Identity Server Security Service Attributes

Administration Attributes

The Administration Service consists of global and organization 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 can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application. Values applied to the organization attributes are default values for each organization configured and can be changed when the service is registered to the organization. The organization attributes are not inherited by entries of the organization. The Administration Attributes are divided into:

- Global Attributes
- Organization Attributes

Global Attributes

The global attributes in the Administration Service are:

- Enable Federation Management
- Enable User Management
- Show People Containers
- Display Containers In Menu
- Show Group Containers
- Managed Group Type
- Default Role Permissions (ACIs)
- Domain Component Tree Enabled

- Admin Groups Enabled
- Compliance User Deletion Enabled
- Dynamic Admin Roles ACIs
- User Profile Service Classes

Enable Federation Management

When selected, this field enables Federation Management. It is selected by default. To disable this feature, deselect the field, log out of the Identity Server console, and then log back in. The Federation Management Service tab will not appear in the console.

Enable User Management

When selected as True, this field enables User Management. This is selected by default.

Show People Containers

This attribute specifies whether to display People Containers in the Identity Server console. If this option is selected, the menu choice People Containers displays in the View menu for Organizations, Containers and Group Containers. People Containers will be seen at the top-level only for a flat DIT.

People containers are organizational units containing user profiles. It is recommended that you use a single people container in your DIT and leverage the flexibility of roles to manage accounts and services. The default behavior of the Identity Server console is to hide the People Container. However, if you have multiple people containers in your DIT, select Show People Containers to display People Containers as managed objects in the Identity Server console.

Display Containers In Menu

This attribute specifies whether to display any containers in the View menu of the Identity Server console. The default value is `false`. An administrator can optionally chose either:

- `false` (checkbox not selected) — Containers are not listed among the choices on the View menu at the top-level for organizations and other containers.
- `true` (checkbox selected) — Containers are listed among the choices on the View menu at the top-level and for organizations and other containers.

Show Group Containers

This attribute specifies whether to show Group Containers in the Identity Server console. If this option is selected, the menu choice Group Containers displays in the View menu for organizations, containers, and group containers. Group containers are organizational units for groups.

Managed Group Type

This option specifies whether subscription groups created through the console are static or dynamic. The console will either create and display subscription groups that are static or dynamic, not both. (Filtered groups are always supported regardless of the value given to this attribute.) The default value is dynamic.

- A static group explicitly lists each group member using the `groupOfNames` or `groupOfUniqueNames` object class. The group entry contains the `uniqueMember` attribute for each member of the group. Members of static groups are manually added; the user entry itself remains unchanged. Static groups are suitable for groups with few members.
- A dynamic group uses a `memberOf` attribute in the entry of each group member. Members of dynamic groups are generated through the use of an LDAP filter which searches and returns all entries which contain the `memberOf` attribute. Dynamic groups are suitable for groups that have a very large membership.
- A filtered group uses an LDAP filter to search and return members that meet the requirement of the filter. For instance, the filter can generate members with a specific uid (`uid=g*`) or email address (`email=*@sun.com`). In these examples, the LDAP filter would return all users whose uid begins with `g` or whose email address ends with `sun.com`, respectively. Filtered groups can only be created within the User Management view by choosing Membership by Filter.

An administrator can select one of the following:

- **Dynamic** — Groups created through the Membership By Subscription option will be dynamic.
- **Static** — Groups created through the Membership By Subscription option will be static.

NOTE The Managed Group Type option is only available when Identity Server is installed using the default mode.

Default Role Permissions (ACIs)

This attribute defines a list of default access control instructions (ACIs) or *permissions* that are used to grant administrator privileges when creating new roles. One of these ACIs is selected depending on the level of privilege desired. Identity Server ships with four default role permissions:

No Permissions

No permissions are to be set on the role.

Organization Admin

The Organization Administrator has read and write access to all entries in the configured organization.

Organization Help Desk Admin

The Organization Help Desk Administrator has read access to all entries in the configured organization and write access to the `userPassword` attribute.

Organization Policy Admin

The Organization Policy Administrator has read and write access to all policies in the organization. The Organization Policy Administrator can not create a referral policy to a peer organization.

NOTE

Roles are defined using the format `aci_name | aci_desc | dn:aci ## dn:aci ## dn:aci where:`

- *aci_name* is the name of the ACI.
- *aci_desc* is a description of the access these ACIs allow. For maximum usability, assume the reader of this description does not understand ACIs or other directory concepts.

aci_name and *aci_desc* are i18n keys contained in the `amAdminUserMsgs.properties` file. The values displayed in the console come from the `.properties` file, and the keys are used to retrieve those values.

- *dn:aci* represents pairs of DNs and ACIs separated by `##`. 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.

Domain Component Tree Enabled

The Domain Component tree (DC tree) is a specific DIT structure used by many Sun ONE components to map between DNS names and organizations' entries.

When this option is enabled, the DC tree entry for an organization is created, provided that the DNS name of the organization is entered at the time the organization is created. The DNS name field will appear in the Organization Create page. This option is only applicable to top-level organizations, and will not be displayed for suborganizations.

Any status change made to the `inetdomainstatus` attribute through the Identity Server SDK in the organization tree will update the corresponding DC tree entry status. (Updates to status that are not made through the Identity Server SDK will not be synchronized.) For example, if a new organization, `sun`, is created with the DNS name attribute `sun.com`, the following entry will be created in the DC tree:

```
dc=sun,dc=com,o=internet,<root suffix>
```

The DC tree may optionally have its own root suffix configured by setting `com.ipplanet.am.domaincomponent` in `AMCONFIG.properties`. By default, this is set to the Identity Server root. If a different suffix is desired, this suffix must be created using LDAP commands. The ACIs for administrators that create organizations required modification so that they have unrestricted access to the new DC tree root.

Admin Groups Enabled

This option specifies whether to create the `DomainAdministrators` and `DomainHelpDeskAdministrators` groups. If selected (`true`), these groups are created and associated with the Organization Admin Role and Organization Help Desk Admin Role, respectively. Once created, adding or removing a user to one of these associated roles automatically adds or removes the user from the corresponding group. This behavior, however, does not work in reverse. Adding or removing a user to one of these groups will not add or remove the user in their associated roles.

The `DomainAdministrators` and `DomainHelpDeskAdministrators` groups are only created in organizations that are created after this option is enabled.

NOTE	This option does not apply to suborganizations, with the exception of the <code>root org</code> . At the <code>root org</code> , the <code>ServiceAdministrators</code> and <code>ServiceHelpDesk Administrators</code> groups are created and associated with the Top-level Admin and Top-level Help Desk Admin roles, respectively. The same behavior applies.
-------------	--

Compliance User Deletion Enabled

This option specifies whether a user's entry will be deleted, or just marked as deleted, from the directory. When a user's entry is deleted and this option is selected (`true`), the user's entry will still exist in the directory, but will be marked as deleted. User entries that are marked for deletion are not returned during Directory Server searches. If this option is not selected, the user's entry will be deleted from the directory.

Dynamic Admin Roles ACIs

This attribute defines the access control instructions for the administrator roles that are created dynamically when a group or organization is configured using Identity Server. These roles are used for granting administrative privileges for the specific grouping of entries created. The default ACIs can be modified only under this attribute listing.

CAUTION Administrators at the Organization level have a wider scope of access than do group administrators. But, by default, when a user is added to a group administrator role, that user can change the password of anyone in the group. This would include any organization administrator who is a member of that group.

Container Help Desk Admin

The Container Help Desk Admin role has read access to all entries in an organizational unit and write access to the `userPassword` attribute in user entries only in this container unit.

Organization Help Desk Admin

The Organization Help Desk Administrator has read access to all entries in an organization and write access to the `userPassword` attribute.

NOTE When a suborganization is created, remember that the administration roles are created in the suborganization, not in the parent organization.

Container Admin

The Container Admin role has read and write access to all entries in an LDAP organizational unit. In Identity Server, the LDAP organizational unit is often referred to as a container.

Organization Policy Admin

The Organization Policy Administrator has read and write access to all policies, and can create, assign, modify, and delete all policies within that organization.

People Container Admin

By default, any user entry in an newly created organization is a member of that organization's People Container. The People Container Administrator has read and write access to all user entries in the organization's People Container. Keep in mind that this role DOES NOT have read and write access to the attributes that contain role and group DNs therefore, they cannot modify the attributes of, or remove a user from, a role or a group.

NOTE	Other containers can be configured with Identity Server to hold user entries, group entries or even other containers. To apply an Administrator role to a container created after the organization has already been configured, the Container Admin Role or Container Help Desk Admin defaults would be used.
-------------	---

Group Admin

The Group Administrator has read and write access to all members of a specific group, and can create new users, assign users to the groups they manage, and delete the users the that they have created.

When a group is created, the Group Administrator role is automatically generated with the necessary privileges to manage the group. The role is not automatically assigned to a group member. It must be assigned by the group's creator, or anyone that has access to the Group Administrator Role.

NOTE	Group Admins do not have access to add or remove groups. Regular users can subscribe to groups because of a privileged proxy account (<code>guser</code>) to search for subscribable groups.
-------------	--

Top-level Admin

The Top-level Administrator has read and write access to all entries in the top-level organization. In other words, this Top-level Admin role has privileges for every configuration principal within the Identity Server application.

Organization Admin

The Organization Administrator has read and write access to all entries in an organization. When an organization is created, the Organization Admin role is automatically generated with the necessary privileges to manage the organization.

User Profile Service Classes

This attribute lists the services that will have a custom display in the User Profile page. The default display generated by the console may not be sufficient for some services. This attribute creates a custom display for any service, giving full control over what and how the service information is displayed. The syntax is as follows:

`<service name> | <relative url>`

NOTE Services that are listed in this attribute will not display in the User Create pages. Any data configuration for a custom service display must be performed the User Profile pages.

Organization Attributes

The organization attributes in the administration service are:

- Groups Default People Container
- Groups People Container List
- Display User's Roles
- User Profile Display Class
- Display User's Groups
- User Group Self Subscription
- User Profile Display Options
- User Creation Default Roles
- View Menu Entries
- Maximum Results Returned From Search
- Timeout For Search (sec.)
- JSP Directory Name
- Online Help Documents
- Required Services
- User Search Key
- User Search Return Attribute

- User Creation Notification List
- User Deletion Notification List
- User Modification Notification List
- Maximum Entries Per Page

Groups Default People Container

This field specifies the default People Container where users will be placed when they are created. There is no default value. A valid value is the DN of a people container. See the note under Groups People Container List attribute for the People Container fallback order.

Groups People Container List

This field specifies a list of People Containers from which a Group Administrator can choose when creating a new user. This list can be used if there are multiple People Containers in the directory tree. (If no People Containers are specified in this list or in the Groups Default People Container field, users are created in the default Identity Server people container, `ou=people`.) There is no default value for this field. The syntax for this attribute is as follows:

<group name> | <dn of people container>

For a Group Administrator to have access to the relevant People Container, this attribute must be set before creating the group.

NOTE When a user is created, this attribute is checked for a container in which to place the entry. If the attribute is empty, the Groups Default People Container attribute is checked for a container. If the latter attribute is empty, the entry is created under `ou=people`.

User Profile Display Class

This attribute specifies the Java class used by the Identity Server console when it displays the User Profile pages.

Display User's Roles

This option specifies whether to display a list of roles assigned to a user as part of their user profile page. If the value is `false` (not selected), the user profile page shows the user's roles only for administrators. The default value is `false`.

Display User's Groups

This option specifies whether to display a list of groups assigned to a user as part of their user profile page. If the value is `false` (not selected), the user profile page shows the user's groups only for administrators. The default value is `false`.

User Group Self Subscription

This option specifies whether users can add themselves to groups that are open to subscription. If the value is `false`, the user profile page allows the user's group membership to be modified only by an administrator. The default value is `false`.

NOTE This option applies only when the Display User's Groups option is selected.

User Profile Display Options

This menu specifies which service attributes will be displayed in the user profile page. An administrator can select from the following:

- `UserOnly` — Display viewable User schema attributes for services assigned to the user.

User service attribute values are viewable by the user when the attribute contains the keyword `Display`. See the *Sun One Identity Server Programmer's Guide* for details.

- `Combined` — Display viewable User and Dynamic schema attributes for services assigned to the user.

User Creation Default Roles

This listing defines roles that will be assigned to newly created users automatically. There is no default value. An administrator can input the DN of one or more roles.

NOTE	This field only takes a full Distinguished Name address, not a role name.
-------------	---

View Menu Entries

This field lists the Java classes of services that will be displayed in the View menu at the top of the console. The syntax is `i18N key | java class name`. (The `i18N` key is used for the localized name of the entry in the View menu.)

Maximum Results Returned From Search

This field defines the maximum number of results returned from a search. The default value is 100.

CAUTION	Do not set this value above 500. The search will be refused.
----------------	--

Timeout For Search (sec.)

This field defines the amount of time (in number of seconds) that a search will continue before timing out. It is used to stop potentially long searches. After the maximum search time is reached, an error is returned. The default is 5 seconds.

JSP Directory Name

This field specifies the name of the directory that contains the `.jsp` files used to construct the console, to give an organization a different appearance (customization). The `.jsp` files need to be copied into the directory that is specified in this field.

Online Help Documents

This field lists the online help links that will be created on the main Identity Server help page. This allows other applications to add their online help links in the Identity Server page. The format for this attribute is as follows:

linki18nkey | html page to load when clicked | i18n properties file

For example:

`IdentityServer Help | /dpAdminHelp.html | amAdminModuleMsgs`

Required Services

This field lists the services that are dynamically added to the users' entries when they are created. Administrators can choose which services are added at the time of creation.

This attribute is not used by the console, but by the Identity Server SDK. Users that are dynamically created will be assigned the services listed in this attribute.

User Search Key

This attribute defines the attribute name that is to be searched upon when performing a simple search in the Navigation page. The default value for this attribute is `cn`. For example, if this attribute uses the default:

If you enter `j*` in the Name field in the Navigation frame, users whose names begins with "j" or "J" will be displayed.

User Search Return Attribute

This attribute defines the attribute name used when displaying the users returned from a simple search. The default of this attribute is `cn`, and the full name of the user will be displayed.

User Creation Notification List

This field defines a list of email addresses that will be sent notification when a new user is created.

The notification list also accepts different locales by using the `:locale` option. For example, to send the notification to an administrator in France:

```
someuser@sun.com|self|admin@sun.com:fr
```

See Table 17-1 on page 180 for a list of locales.

NOTE The sender email ID can be changed by modifying property 497 in `amProfile.properties`, which is located, by default, at `<Identity_Server_Root/opt/SUNWam/locale`. The default sender ID is `DSAME`.

User Deletion Notification List

This field defines a list of email addresses that will be sent notification when a user is deleted. The notification list also accepts different locales by using the `:locale` option. For example, to send the notification to an administrator in France:

```
someuser@sun.com|self|admin@sun.com:fr
```

See Table 17-1 on page 180 for a list of locales.

NOTE The sender email ID can be changed by modifying property 497 in `amProfile.properties`, which is located, by default, at `<Identity_Server_Root/opt/SUNWam/locale`. The default sender ID is `DSAME`.

User Modification Notification List

This field defines a list of attributes and email addresses associated with the attribute. When a user modification occurs on an attribute defined in the list, the email address associated with the attribute will be sent notification. Each attribute can have a different set of addresses associated to it. If multiple email addresses are specified, they are separated by the “|” character.

The `self` keyword may be used in place of one of the addresses. This sends mail to the user whose profile was modified.

For example:

```
manager someuser@sun.com|self|admin@sun.com
```

Mail will be sent to the address specified in the `manager` attribute, `someuser@sun.com`, `admin@sun`, the person who modified the user (`self`).

The notification list also accepts different locales by using the `:locale` option. For example, to send the notification to an administrator in France:

```
manager someuser@sun.com|self|admin@sun.com:fr
```

See Table 17-1 on page 180 for a list of locales.

NOTE	The attribute name is the same as it appears in the Directory Server schema, and not as the display name in the console.
-------------	--

Maximum Entries Per Page

This attribute allows you to define the maximum rows that can be displayed per page. The default is 25. For example, if a user search returns 100 rows, there will be 4 pages with 25 rows displayed in each page.

Anonymous Authentication Attributes

The Anonymous Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Anonymous Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Anonymous Authentication attributes are:

- Authentication Level
- Valid Anonymous User List
- Default Anonymous User Name

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may chose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Auth Level," on page 179 for details.
-------------	---

Valid Anonymous User List

This field contains a list of user IDs that have permission to login without providing credentials. If a user's login name matches a user ID in this list, access is granted and the session is assigned to the specified user ID. If the user's login name does not match a user ID in this list, anonymous access is still granted, but the session is assigned to the user ID specified in the Default Anonymous User Name field.

NOTE In order to login with a user ID defined in Valid Anonymous User List, the user must use the following URL:

```
http://<hostname>:<port>/<DEPLOY_URI>/Login?module=
Anonymous&org=<org_name>&username=<user_id>
```

Default Anonymous User Name

This field defines the user ID that a session is assigned to if the login name does not match a user ID in the Valid Anonymous User List field. The default value is `anonymous`. An Anonymous user must also be created in the organization.

NOTE In order to login using the anonymous authentication service, the user defined in Default Anonymous User Name must use the following URL:

```
http://<hostname>:<port>/<DEPLOY_URI>/Login?module=
Anonymous&org=<org_name>
```

Certificate Authentication Attributes

The Certificate Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Certificate Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Certificate Authentication attributes are:

- Match Certificate in LDAP
- Attribute In Subject DN To Use To Search LDAP
- Match Certificate to CRL
- Attribute In Issuer DN To Use To Search CRL
- Enable OCSP Validation
- LDAP Server and Port
- LDAP Start Search DN
- LDAP Server Principal User
- LDAP Server Principal Password
- LDAP Attribute for Profile ID
- SSL On For LDAP Access
- Field in Cert to Use to Access User Profile
- Other Field In Cert To Use To Access User Profile
- Authentication Level

Match Certificate in LDAP

This option specifies whether to check if the user certificate presented at login is stored in the LDAP Server. If no match is found, the user is denied access. If a match is found and no other validation is required, the user is granted access. The default is that the Certificate Authentication service does not check for the user certificate.

NOTE	A certificate stored in the Directory Server is not necessarily valid; it may be on the certificate revocation list. See “Match Certificate to CRL,” on page 168.
-------------	---

Attribute In Subject DN To Use To Search LDAP

This field specifies the attribute of the certificate's `SubjectDN` value that will be used to search LDAP for certificates. The actual value will be used for the search. The default is `CN`.

Match Certificate to CRL

This option specifies whether to compare the user certificate against the Certificate Revocation List (CRL) in the LDAP Server. This check is performed against a user certificate after a matching user profile is found (see “Match Certificate in LDAP,” on page 168). If the certificate is on the CRL, the user is denied access; if not, the user is allowed to proceed. This attribute is, by default, not enabled.

NOTE	Certificates should be revoked when the owner of the certificate has changed status and no longer has the right to use the certificate or when the private key of a certificate owner has been compromised.
-------------	---

Attribute In Issuer DN To Use To Search CRL

This field specifies the attribute of the received certificate's `subjectDN` value that will be used to search LDAP for revoked certificates. This field is used only when the Match Certificate to CRL attribute is enabled. The actual value will be used for the search. The default is `CN`.

Enable OCSP Validation

This parameter enables OCSP validation to be performed by contacting the corresponding OCSP responder. The OCSP responder is decided as follows during runtime:

- If the OCSP responder is set in the `com.sun.identity.authentication.ocsp.repsonder.url` attribute, the value of the attribute will be used as the OCSP responder.
- If the value of the attribute is not set in the `AMConfig.properties` file, the OCSP responder presented in your client certificate is used as the OCSP responder.
- If an OCSP responder can not be found, no OCSP validation will be performed.

NOTE	Before enabling OCSP Validation, make sure that the time of the Identity Server machine and the OCSP responder machine are in sync as close as possible. Also, the time on the Identity Server machine must not be behind the time on the OCSP responder. For example:
-------------	--

OCSP responder machine - 12:00:00 pm

Identity Server machine - 12:00:30 pm

LDAP Server and Port

This field specifies the name and port number of the LDAP server where the certificates are stored. The default value is the host name and port specified when Identity Server was installed. The host name and port of any LDAP Server where the certificates are stored can be used. The format is *hostname:port*.

LDAP Start Search DN

This field specifies the DN of the node where the search for the user's certificate should start. There is no default value. The field will recognize any valid DN. Multiple entries must be prefixed by the local server name.

LDAP Server Principal User

This field accepts the DN of the principal user (usually Directory Manager) for the LDAP server where the certificates are stored. There is no default value for this field which will recognize any valid DN. The principal user must be authorized to read, and search certificate information stored in the Directory Server.

LDAP Server Principal Password

This field carries the LDAP password associated with the user specified in the LDAP Server Principal User field. There is no default value for this field which will recognize the valid LDAP password for the specified principal user.

NOTE	This value is stored as readable text in the directory.
-------------	---

LDAP Attribute for Profile ID

This field specifies the attribute in the Directory Server entry that matches the certificate whose value should be used to identify the correct user profile. There is no default value for this field which will recognize any valid attribute in a user entry (cn, sn, and so on) that can be used as the user ID.

SSL On For LDAP Access

This option specifies whether to use SSL to access the LDAP server. The default is that the Certificate Authentication service does not use SSL for LDAP access.

Field in Cert to Use to Access User Profile

This menu specifies which field in the certificate's Subject DN should be used to search for a matching user profile. For example, if you choose `email address`, the certificate authentication service will search for the user profile that matches the attribute `emailAddr` in the user certificate. The user logging in then uses the matched profile. The default field is `subject CN`. The list contains:

- email address
- issuer DN

- issuer CN
- issuer O
- serial number
- subject CN
- subject DN
- subject O
- subject UID
- other

Other Field In Cert To Use To Access User Profile

If the value of the Field in Cert to Use to Access User Profile attribute is set to `other`, then this field specifies the attribute that will be selected from the received certificate's `subjectDN` value. The authentication service will then search the user profile that matches the value of that attribute.

Authentication Level

The authentication level is set separately for each authentication module. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may choose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Auth Level,” on page 179 for details.
-------------	---

Core Authentication Attributes

The Core Authentication service is the basic service for the Anonymous, Certificate, LDAP, Membership, Safeword, Unix and RADIUS authentication services as well as any custom authentication service created with the Authentication SPI. Core authentication must be configured as a service for each organization that wishes to use any form of authentication. The Core Authentication attributes consist of global and organization 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 can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The values applied to the organization attributes under Service Configuration become the default values for the Core Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Core Authentication attributes are separated into:

- Global Attributes
- Organization Attributes

Global Attributes

The organization attributes in the Core Authentication service are:

- Pluggable Auth Module Classes
- Supported Auth Modules for Clients
- LDAP Connection Pool Size
- LDAP Connection Default Pool Size

Pluggable Auth Module Classes

This field specifies the Java classes of the authentication services available to any organization configured within the Identity Server platform. By default, this includes LDAP, SafeWord, Anonymous, Application, Membership, Unix, Certification, and RADIUS. Identity Server also includes a public SPI that can be used to add other authentication services. To define new services, this field must take a text string specifying the full class name (including package name) of each new authentication service.

Supported Auth Modules for Clients

This attribute specifies a list of supported authentication modules for a specific client. The format is as follows:

```
clientType | module1,module2,module3
```

This attribute is in effect when Client Detection is enabled.

LDAP Connection Pool Size

This attribute specifies the minimum and maximum connection pool to be used on a specific server and port. This attribute is for LDAP and Membership authentication services only. The format is as follows:

```
host:port:min:max
```

NOTE This connection pool is different than the SDK connection pool configured in `serverconfig.xml`.

LDAP Connection Default Pool Size

This attribute sets the default minimum and maximum connection pool to be used with all LDAP authentication module configurations. If an entry for the host and port exists in the LDAP Connection Pool Size attribute, the minimum and maximum settings will be used from LDAP Connection Default Pool Size.

Organization Attributes

The organization attributes in the Core Authentication service are:

- Organization Authentication Modules
- User Profile
- Admin Authenticator
- User Profile Dynamic Creation Default Roles
- Persistent Cookie Mode
- Persistent Cookie Max Time (seconds)
- People Container For All Users
- Alias Search Attribute Name
- Default Auth Level
- User Naming Attribute
- Default Auth Locale
- Organization Authentication Configuration
- Login Failure Lockout Mode
- Login Failure Lockout Count
- Login Failure Lockout Interval (minutes)
- Email Address to Send Lockout Notification
- Warn User After N Failure
- Login Failure Lockout Duration (minutes)
- Lockout Attribute Name
- Lockout Attribute Value
- Default Success Login URL
- Default Failure Login URL
- Authentication PostProcessing Class
- User Name Generator Mode
- Pluggable User Name Generator Class

Organization Authentication Modules

This list specifies the authentication modules available to the organization. Each administrator can choose the type of authentication for their specific organization. Multiple authentication modules provide flexibility, but users must be sure that their login setting is appropriate for the selected authentication module. The default authentication is LDAP. The authentication services included with Identity Server are:

- LDAP
- Cert
- Anonymous
- Membership
- NT
- SafeWord
- RADIUS
- Unix

NOTE	The Administrator must create and notify the core and authentication module templates in a created organization for that organization to function properly.
-------------	---

User Profile

This option allows you to specify options for a user profile.

- **Required** - This specifies that on successful authentication, the user needs to have a profile in Directory Server for the authentication service to issue an SSOToken.
- **Dynamically Created** - This specifies that on successful authentication, the authentication service will create the user profile if one does not already exist. The SSOToken will then be issued.
- **Ignore** - This specifies that the user profile is not required by the authentication service to issue the SSOToken for a successful authentication.

Admin Authenticator

Clicking the edit link will allow you to define the authentication service for administrators only. An administrator is a user who needs access to the Identity Server console. This attribute can be used if the authentication module for administrators needs to be different from the module for end users.

User Profile Dynamic Creation Default Roles

This field specifies the roles assigned to a new user whose profiles are created if Dynamic Creation is selected through the feature “User Profile,” on page 176. There is no default value. The administrator must specify the DNs of the roles that will be assigned to the new user.

NOTE	The role specified must be under the organization for which authentication is being configured.
-------------	---

Persistent Cookie Mode

This option determines whether users can restart the browser and still return to their authenticated session. User sessions can be retained by enabling Persistent Cookie Mode. When Persistent Cookie Mode is enabled, a user session does not expire until its persistent cookie expires, or the user explicitly logs out. The expiration time is specified in Persistent Cookie Max Time (seconds). The default value is that Persistent Cookie Mode is not enabled and the authentication service uses only memory cookies.

NOTE	A persistent cookie must be explicitly requested by the client using the <code>idSPCookie=yes</code> parameter in the login URL. Once the persistent cookie has been set, the <code>idSPCookie</code> parameter expires.
-------------	--

Persistent Cookie Max Time (seconds)

This field specifies the interval after which a persistent cookie expires. (Persistent Cookie Mode must be enabled by selecting its checkbox.) The interval begins when the user’s session has been successfully authenticated. The default value is 2147483 (time in seconds). The field will take any integer value between 0 and 2147483.

People Container For All Users

After successful authentication by a user, their profile is retrieved. The value in this field specifies where to search for the profile. Generally, this value will be the DN of the default People Container. All user entries added to an organization are automatically added to the organization's default People Container. The default value is `ou=People`, and generally, this is completed with the organization name(s) and root suffix. The field will take a valid DN for any organizational unit.

NOTE

Authentication searches for a user profile by:

- Searching under the default People Container, then
- Searching under the default organization, then
- Searching for the user in the default organization using the Alias Search Attribute Name attribute.

The final search is for SSO cases where the user name used to authenticate may not be the naming attribute in the profile. For example, user may authenticate using Safeword ID of `jn10191`, but their profile is `uid=jamie`.

Alias Search Attribute Name

After successful authentication by a user, their profile is retrieved. This field specifies a second LDAP attribute to search from if a search on the first LDAP attribute, specified in "User Naming Attribute," on page 179, fails to locate a matching user profile. Primarily, this attribute will be used when the user identification returned from an authentication module is not the same as that specified in User Naming Attribute. For example, a RADIUS server might return `abc1234` but the user name is `abc`. There is no default value for this attribute. The field will take any valid LDAP attribute (for example, `cn`).

Default Auth Level

The authentication level value indicates how much to trust authentications. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application can use the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level.

The authentication level should be set within the organization's specific authentication template. The Default Auth Level value described here will apply only when no authentication level has been specified in the Authentication Level field for a specific organization's authentication template. The Default Auth Level default value is 0, the lowest authentication level. (The value in this attribute is not used by Identity Server but by any external application that may chose to use it.)

User Naming Attribute

After successful authentication by a user, their profile is retrieved. The value of this attribute specifies the LDAP attribute to use for the search. By default, Identity Server assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`) specify the attribute name in this field.

Default Auth Locale

This field specifies the default language subtype to be used by the authentication service. The default value is `en_US`. A listing of valid language subtypes can be found in Table 17-1.

In order to use a different locale, all authentication templates for that locale must first be created. A new directory must then be created for these templates. See the *Sun One Identity Server Programmer's Guide* for more information.

Table 17-1 Supported Language Locales

Language Tag	Language
af	Afrikaans
be	Byelorussian
bg	Bulgarian
ca	Catalan
cs	Czechoslovakian
da	Danish
de	German
el	Greek
en	English
es	Spanish
eu	Basque
fi	Finnish
fo	Faroese
fr	French
ga	Irish
gl	Galician
hr	Croatian
hu	Hungarian
id	Indonesian
is	Icelandic
it	Italian
ja	Japanese
ko	Korean
nl	Dutch
no	Norwegian
pl	Polish
pt	Portuguese
ro	Romanian

Table 17-1 Supported Language Locales (*Continued*)

Language Tag	Language
ru	Russian
sk	Slovakian
sl	Slovenian
sq	Albanian
sr	Serbian
sv	Swedish
tr	Turkish
uk	Ukrainian
zh	Chinese

Organization Authentication Configuration

This attribute sets the authentication module for the organization. The default authentication module is LDAP. One or more authentication modules can be selected by clicking the Edit link. If more than one module is selected, then the user will have to successfully authenticate to all of selected modules.

Login Failure Lockout Mode

This feature specifies whether to disallow a user to re-authenticate (lockout) if that user has initially failed to authenticate. Selecting this attribute will enable the lockout. By default, the lockout feature is not enabled.

Login Failure Lockout Count

This attribute defines the number of attempts that a user may try to authenticate, within the time interval defined in Login Failure Lockout Interval (minutes), before being locked out.

For example, if Login Failure Lockout Count is set to 5, and Login Failure Lockout Interval (minutes) is set to 5, then a user has five chances within five minutes to authenticate before being locked out.

Login Failure Lockout Interval (minutes)

This attribute defines (in minutes) the amount of time in which the number of authentication attempts (as defined in Login Failure Lockout Count) can be completed, before a user is locked out.

For example, if Login Failure Lockout Count is set to 5, and Login Failure Lockout Interval (minutes) is set to 5, then a user has five chances within five minutes to authenticate before being locked out.

Email Address to Send Lockout Notification

This attribute specifies an email address that will receive notification if a user lockout occurs.

Warn User After N Failure

This attribute specifies the number of authentication failures that can occur before Identity Server sends a warning message that the user will be locked out.

Login Failure Lockout Duration (minutes)

This attribute defines (in minutes) the duration that a user will not be allowed to attempt to re-authenticate, if a lockout has occurred.

If this attribute value is set to 0, and Login Failure Lockout Mode is enabled, the user will be locked out by setting the Lockout Attribute Name in their entry to Lockout Attribute Value.

Lockout Attribute Name

This attribute contains the `inetuserstaus` value that is set in the Lockout Attribute Value attribute. If a user is locked out, and the Login Failure Lockout Duration (minutes) variable is set to 0, `inetuserstatus` will be set to inactive, prohibiting the user from attempting to authenticate.

Lockout Attribute Value

This attribute specifies the `inetuserstatus` value (contained in Lockout Attribute Name) of the user status as either active or inactive. If a user is locked out, and the Login Failure Lockout Duration (minutes) variable is set to 0, `inetuserstatus` will be set to inactive, prohibiting the user from attempting to authenticate.

Default Success Login URL

This field specifies the URL to which users are redirected after successful authentication. The field will take any valid URL.

Default Failure Login URL

This field specifies the URL to which users are redirected if authentication is unsuccessful. The field will take any valid URL.

Authentication PostProcessing Class

This field specifies the name of the Java class used to customize post authentication processes for successful or unsuccessful logins. Example:

```
com.abc.authentication.PostProcessClass
```

User Name Generator Mode

This attribute is used by the Membership authentication module. If this attribute field is enabled, the Membership module is able to generate user IDs for a specific user if the user ID already exists. The user IDs are generated from the Java class specified in Pluggable User Name Generator Class.

Pluggable User Name Generator Class

The field specifies the name of the Java class that will be used to generate user IDs when User Name Generator Mode is enabled.

LDAP Authentication Attributes

The LDAP Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the LDAP Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The LDAP Authentication attributes are:

- Primary LDAP Server and Port
- Secondary LDAP Server and Port
- DN to Start User Search
- DN for Root User bind
- Password for Root User Bind
- Password For Root User Bind (Confirm)
- User Naming Attribute
- User Entry Search Attributes
- User Search Filter
- Search Scope
- Enable SSL to LDAP Server
- Return User DN To Auth
- Authentication Level

Primary LDAP Server and Port

This field specifies the host name and port number of the primary LDAP server specified during Identity Server installation. This is the first server contacted for LDAP authentication. The format is `hostname:port`. (If there is no port number, assume 389.) Multiple entries must be prefixed by the local server name.

Secondary LDAP Server and Port

This field specifies the host name and port number of a secondary LDAP server available to the Identity Server platform. If the primary LDAP server does not respond to a request for authentication, this server would then be contacted. If this server goes down, Identity Server will switch back to the primary server. The format is also `hostname:port`. Multiple entries must be prefixed by the local server name.

CAUTION When authenticating users from a Directory Server that is remote from the Identity Server enterprise, it is important that both the Primary and Secondary LDAP Server Ports have values. The value for one Directory Server location can be used for both fields.

DN to Start User Search

This field specifies the DN of the node where the search for a user would start. (For performance reasons, this DN should be as specific as possible.) The default value is the root of the directory tree. Any valid DN will be recognized. Multiple entries must be prefixed by the local server name.

NOTE If multiple users match the same search, authentication will fail.

DN for Root User bind

This field specifies the DN of the user that will be used to bind to the Directory Server specified in the Primary LDAP Server and Port field as administrator. The authentication service needs to bind as this DN in order to search for a matching user DN based on the user login ID. There is no default value. Any valid DN will be recognized.

Password for Root User Bind

This field carries the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid LDAP password will be recognized.

Password For Root User Bind (Confirm)

Confirmation of the password.

User Naming Attribute

After successful authentication by a user, the user's profile is retrieved. The value of this attribute is used to perform the search. The field specifies the LDAP attribute to use. By default, Identity Server assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`) specify the attribute name in this field.

NOTE	The user search filter will be a combination of the Search Filter attribute and the User Entry Naming Attribute.
-------------	--

User Entry Search Attributes

This field lists the attributes to be used to form the search filter for a user that is to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber` and `mail`, the user could authenticate with any of these names.

User Search Filter

This field specifies an attribute to be used to find the user under the DN to Start User Search field. It works with the User Entry Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.

Search Scope

This menu indicates the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in the attribute “DN to Start User Search,” on page 186. The default value is `SUBTREE`. One of the following choices can be selected from the list:

- `OBJECT` - Searches only the specified node
- `ONELEVEL` - Searches at the level of the specified node and one level down
- `SUBTREE` - Search all entries at and below the specified node

Enable SSL to LDAP Server

This option enables SSL access to the Directory Server specified in the Primary and Secondary LDAP Server and Port field. By default, the box is not checked and the SSL protocol will not be used to access the Directory Server.

Return User DN To Auth

When the Identity Server directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the `userId`, and no search is necessary. Normally, an authentication module returns only the `userId`, and the authentication service searches for the user in the local Identity Server LDAP. If an external LDAP directory is used, this option is typically not enabled.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may chose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Auth Level,” on page 179 for details.
-------------	---

Membership Authentication Attributes

The Membership Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Membership Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Membership Authentication attributes are:

- Minimum Password Length
- Default User Roles
- User Status After Registration
- Primary LDAP Server and Port
- Secondary LDAP Server and Port
- DN to Start User Search
- DN for Root User bind
- Password for Root User Bind
- Password for Root User Bind (Confirm)
- User Naming Attribute
- User Entry Search Attributes
- User Search Filter
- Search Scope
- Enable SSL to LDAP Server
- Return User DN To Auth

- Authentication Level

Minimum Password Length

This field specifies the minimum number of characters required for a password set during self-registration. The default value is 8.

If this value is changed, it should also be changed in the registration and error text in the following file:

```
<IdentityServer_root>/locale/amAuthMembership.properties  
(PasswdMinChars entry)
```

Default User Roles

This field specifies the roles assigned to new users whose profiles are created through self-registration. There is no default value. The administrator must specify the DNs of the roles that will be assigned to the new user.

NOTE	The role specified must be under the organization for which authentication is being configured.
-------------	---

User Status After Registration

This menu specifies whether services are immediately made available to a user who has self-registered. The default value is *Active* and services are available to the new user. By selecting *Inactive*, the administrator chooses to make no services available to a new user.

Primary LDAP Server and Port

This field specifies the host name and port number of the primary Directory Server. This is the first server searched for membership authentication. The default value is the Directory Server URL specified during Identity Server installation. The format is *hostname:port*. If you use multiple entries, the entries must be prefixed by the local server name.

Secondary LDAP Server and Port

This field specifies the host name and port number of the secondary Directory Server. If the primary server does not respond to a request for authentication, this server would then be contacted. There is no default value for this field. The format is `hostname:port`. If you use multiple entries, the entries must be prefixed by the local server name.

DN to Start User Search

This field specifies the DN of the node where the search for a user would start. (For performance reasons, this DN should be as specific as possible.) The default value is the root of the directory tree. Any valid DN will be recognized. If you use multiple entries, the entries must be prefixed by the local server name.

NOTE If multiple users match the same search, authentication will fail.

DN for Root User bind

This field specifies the DN of the user that will be used to bind to the Directory Server specified in the Primary LDAP Server and Port field as administrator. The authentication service needs to bind as this DN in order to search for a matching user DN based on the user login ID. There is no default value. Any valid DN will be recognized.

Password for Root User Bind

This field carries the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid LDAP password will be recognized.

Password for Root User Bind (Confirm)

Confirmation of the password.

User Naming Attribute

This field specifies the attribute used for the naming convention of user entries. By default, Identity Server assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`) specify the attribute name in this field.

User Entry Search Attributes

This field lists the attributes to be used to form the search filter for a user that is to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber` and `mail`, the user could authenticate with any of these names.

User Search Filter

This field specifies an attribute to be used to find the user under the DN to Start User Search field. It works with the User Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.

Search Scope

This menu indicates the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in the attribute "DN to Start User Search," on page 193. The default value is `SUBTREE`. One of the following choices can be selected from the list:

- `OBJECT` — Searches only the specified node
- `ONELEVEL` — Searches at the level of the specified node and one level down
- `SUBTREE` — Search all entries at and below the specified node

Enable SSL to LDAP Server

This option enables SSL access to the Directory Server specified in the Primary and Secondary LDAP Server and Port field. By default, the box is not checked and the SSL protocol will not be used to access the Directory Server.

Return User DN To Auth

When the Identity Server directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the `userId`, and no search is necessary. Normally, an authentication module returns only the `userId`, and the authentication service searches for the user in the local Identity Server LDAP. If an external LDAP directory is used, this option is typically not enabled.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may chose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Auth Level,” on page 179 for details.
-------------	---

NT Authentication Attributes

The NT Authentication Attributes are organization attributes. The values applied to them under Service Configuration become the default values for the NT Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The NT Authentication attributes are:

- NT Authentication Domain
- NT Authentication Host
- NT Module Authentication Level

NT Authentication Domain

This attribute defines the Domain name to which the user belongs.

NT Authentication Host

This attribute defines the NT authentication hostname. The hostname should be the netBIOS name, as opposed to the fully qualified domain name (FQDN). By default, the first part of the FQDN is the netBIOS name.

If the DHCP (Dynamic Host Configuration Protocol) is used, you would put a suitable entry in the HOSTS file on the Windows 2000 machine.

Name resolution will be performed based on the netBIOS name. If you do not have any server on your subnet supplying netBIOS name resolution, the mappings should be hardcoded.

For example, the hostname should be `example1` not `example1.company1.com`.

NT Module Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may chose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Auth Level,” on page 179 for details.
-------------	---

RADIUS Authentication Attributes

The RADIUS Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the RADIUS Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The RADIUS Authentication attributes are:

- RADIUS Server 1
- RADIUS Server 2
- RADIUS Shared Secret
- RADIUS Shared Secret (Confirm)
- RADIUS Server's Port
- Authentication Level
- Timeout (Seconds)

RADIUS Server 1

This field displays the IP address or host name of the primary RADIUS server. The default IP address is 127.0.0.1. The field will recognize any valid IP address or host name. Multiple entries must be prefixed by the local server name.

RADIUS Server 2

This field displays the IP address or host name of the secondary RADIUS server. It is a failover server which will be contacted if the primary server could not be contacted. The default IP address is 127.0.0.1. Multiple entries must be prefixed by the local server name.

RADIUS Shared Secret

This field carries the shared secret for RADIUS authentication. The shared secret should have the same qualifications as a well-chosen password. There is no default value for this field.

RADIUS Shared Secret (Confirm)

Confirmation of the shared secret for RADIUS authentication.

RADIUS Server's Port

This field specifies the port on which the RADIUS server is listening. The default value is 1645.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may chose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Auth Level,” on page 179 for details.
-------------	---

Timeout (Seconds)

This field specifies the time interval in seconds to wait for the RADIUS server to respond before a timeout. The default value is 3 seconds. It will recognize any number specifying the timeout in seconds.

SafeWord Authentication Attributes

The SafeWord Authentication Attributes are organization attributes. The values applied to them under Service Configuration become the default values for the SafeWord Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization.

This service allows for authenticating users using Secure Computing's SafeWord or SafeWord PremierAccess authentication servers. The SafeWord Authentication attributes are:

- SafeWord Server Specification
- SafeWord System Name
- SafeWord Server Verification Files Path
- SafeWord Logging Level
- SafeWord Log Path
- SafeWord Module Authentication Level

SafeWord Server Specification

This field specifies the SafeWord or SafeWord PremiereAccess server name and port. Port 7482 is set as the default for a SafeWord server. The default port number for a SafeWord PremierAccess server is 5030.

SafeWord System Name

This field specifies the system name configured in the SafeWord server. The default system name is `STANDARD`.

SafeWord Server Verification Files Path

This field specifies the directory into which the SafeWord client library places its verification files. The default is as follows:

```
/var/opt/SUNWam/auth/safeword/serverVerification
```

If a different directory is specified in this field, the directory must exist before attempting SafeWord authentication.

SafeWord Logging Level

This attribute is not used.

SafeWord Log Path

This attribute specifies the directory path and log file name for SafeWord client logging. The default path is as follows:

```
/var/opt/SUNWam/auth/safeword/safe.log
```

If a different path or filename is specified, they must exist before attempting SafeWord authentication.

If more than one organization is configured for SafeWord authentication, and different SafeWord servers are used, then different paths must be specified, or only the first organization where SafeWord authentication occurs will work. Likewise, if an organization changes SafeWord servers, the `swec.dat` file in the specified directory must be deleted before authentications to the newly configured SafeWord server will work.

SafeWord Module Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may chose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Auth Level,” on page 179 for details.
-------------	---

Unix Authentication Attributes

The Unix Authentication Service consists of global and organization 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 can not be applied directly to roles or organizations, as the goal of global attributes is to customize the Identity Server application. Values applied to the organization attributes are default values for each organization configured and can be changed when the service is registered to the organization. The organization attributes are not inherited by entries of the organization. The Unix Authentication Attributes are divided into:

- Global Attributes
- Organization Attribute

NOTE The Unix authentication service is not supported on the Windows 2000 platform.

Global Attributes

The global attributes in the Unix Authentication service are:

- Unix Helper Configuration Port
- Unix Helper Authentication Port
- Unix Helper Timeout (Minutes)
- Unix Helper Threads

Unix Helper Configuration Port

This attribute specifies the port to which the Unix Helper ‘listens’ upon startup for the configuration information contained in the Unix Helper Authentication Port, Unix Helper Timeout (Minutes), and Unix Helper Threads attributes. The default is 58946.

If this attribute is changed, you must also change the `unixHelper.port` entry in the `AMConfig.properties` file, and restart Identity Server.

Unix Helper Authentication Port

This attribute specifies the port to which the Unix Helper ‘listens’ for authentication requests after configuration. The default port is 57946.

Unix Helper Timeout (Minutes)

This attribute specifies the number of minutes that users have to complete authentication. If users surpass the allotted time, authentication automatically fails. The default time is set to 3 minutes.

Unix Helper Threads

This attribute specifies the maximum number of permitted simultaneous Unix authentication sessions. If the maximum is reached at a given moment, subsequent authentication attempts are not allowed until a session is freed up. The default is set to 5.

Organization Attribute

The organization attribute for the Unix Authentication service is:

Unix Module Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. (The value in this attribute is not specifically used by Identity Server but by any external application that may chose to use it.) If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0, the lowest authentication level.

NOTE	If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See “Default Auth Level,” on page 179 for details.
-------------	---

Authentication Configuration Attributes

The Authentication Configuration Attributes are dynamic and organization attributes. These attributes can be defined for an organization, service, or role.

If the role is assigned to a user or a user is assigned to the organization, these attributes, by default, are inherited by the user. The Authentication Configuration Attributes are:

- Authentication Configuration
- Login Success URL
- Login Failure URL
- Authentication Post Processing Class

Authentication Configuration

Clicking on the Edit link will display the Authentication Configuration interface. It allows you to configure the authentication modules for role-based or organization-based authentication.

The following table lists the authentication module configuration options:

Module Name	Allows you to select from the list of default authentication modules available to Identity Server.
-------------	--

Flag

This pull-down menu allows you specify the authentication module requirements. It can be one of:

- **REQUIRED** - The authentication module is required to succeed. If it succeeds or fails, authentication continues to proceed down the authentication module list.
- **REQUISITE** - The authentication module is required to succeed. If it succeeds, authentication continues down the authentication module list. If it fails, control returns to the application (authentication does not proceed down the authentication module list.)
- **SUFFICIENT** - The authentication module is not required to succeed. If it does succeed, control immediately returns to the application (authentication does not proceed down the authentication module list.). If it fails, authentication continues down the list.
- **OPTIONAL** - The authentication module is not required to succeed. If it succeeds or fails, authentication still continues to proceed down the list.

These flags establish an enforcement criteria for the authentication module for which they are defined. There hierarchy for enforcement, with **REQUIRED** being the highest, and **OPTION** being the lowest.

For example, if an administrator defines an LDAP module with the **REQUIRED** flag, then the user's credential must pass the LDAP authentication requirements to access a given resource.

If you add multiple authentication modules and for each module the Flag is set to **REQUIRED**, the user must pass all authentication requirements before being granted access.

For more information on the flag definitions, refer to the JAAS (Java Authentication and Authorization Service) located at:

<http://java.sun.com/security/jaas/doc/module.html>

Option

Allows for additional options for the for the module as a key=value pair. Multiple options are separated by a space.

Login Success URL

This attribute specifies the URL that the user will be redirected to upon successful authentication.

Login Failure URL

This attribute specifies the URL that the user will be redirected to upon unsuccessful authentication.

Authentication Post Processing Class

This attribute defines the name of the Java class used to customize the post authentication process after a login success or failure.

Conflict Resolution Level

This attribute applies to roles only. Conflict Resolution level sets a priority level for the Authentication Configuration attributes for roles that may contain the same user. For example, if User1 is assigned to both Role1 and Role2, you can define a higher priority level for Role1 so when the user attempts authentication Role1 will have the highest priority for success or failure redirects and for post authentication processes.

Client Detection Attributes

The Client Detection Attributes are global attributes. The values applied to them are applied across the 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 Identity Server application.) The Client Detection Attributes are:

- Client Types
- Default Client Type
- Client Detection Class
- Client Detection Enabled

Client Types

This attribute specifies a key to access client-specific properties.

Default Client Type

This attribute defines the default client type derived from the list of client types in the Client Types attribute. The default is `genericHTML`.

Client Detection Class

This attribute defines the client detection class for which all client detection requests are routed. The string returned by this attribute should match one of the client types listed in the Client Types attribute. The default client detection class is `com.ipланet.services.cdm.ClientDetectionDefaultImpl`.

Client Detection Enabled

This attribute allows you to enable client detection. If client detection is enabled (selected), every request is routed through the class specified in the Client Detection Class attribute.

Logging Attributes

The Logging Attributes are global attributes. The values applied to them are applied across the Sun ONE Identity Server configuration and are inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The Logging Attributes are:

- Max Log Size
- Number of History Files
- Log Location
- Logging Type
- Database User Name
- Database User Password
- Database User Password (Confirm)
- Database Driver Name
- Configurable Log Fields
- Log Verification Time
- Log Signature Time
- Secure Logging
- Maximum Number of Records
- Remote Buffer Size
- Number Of Files Per Archive

Max Log Size

This attribute accepts a value for the maximum size (in bytes) of a Identity Server log file. A number up to one million can be input in the value field. The default value is 1000000.

Number of History Files

This attribute has a value equal to the number of backup log files that will be retained for historical analysis. Any integer can be input depending on the partition size and available disk space of the local system. The default value is 3.

Log Location

The file-based logging function needs a location where log files can be stored. This field accepts a full directory path to that location. The default location is:

```
<IdentityServer_root>/opt/SUNWam/logs/
```

If a non-default directory is being used, this directory must have write permission to the user under which Identity Server is running.

NOTE	Any changes in logging attribute values require a restart of the Identity Server before the changes are activated.
-------------	--

Logging Type

This attribute allows you to specify either File, for flat file logging, or DB for database logging.

Database User Name

This attribute accepts the name of the user that will connect to the database when the Logging Type attribute is set to DB.

Database User Password

This attribute accepts the database user password when the Logging Type attribute is set to DB.

Database User Password (Confirm)

Confirmation of the database password.

Database Driver Name

This attribute allows the user to specify the driver that is to be used for the logging implementation class.

Configurable Log Fields

This parameter represents the list of fields that are to be logged. By default, the following fields are logged:

- Hostname
- Loglevel
- LoginID
- Domain
- IPAddress

Log Verification Time

This attribute sets the frequency (in seconds) that the server should verify the logs to detect tampering. The default time is 3600 seconds. This parameter applies to secure logging only.

Log Signature Time

This parameter sets the frequency (in seconds) that the log will be signed. The default time is 900 seconds. This parameter applies to secure logging only.

Secure Logging

This attribute specifies whether or not to enable secure logging. By default, secure logging is off. Secure Logging enables detection of unauthorized changes or tampering of security logs.

Maximum Number of Records

This attribute sets the maximum number of records that the Java LogReader interfaces return, regardless of how many records match the read query. By default, it is set to 500. This attribute can be overridden by the caller of the Logging API through the LogQuery parameter.

Remote Buffer Size

This attribute specifies the maximum amount of log records to be buffered in memory before they are sent to the logging service to be logged. The default is one record.

Number Of Files Per Archive

This attribute is only applicable to secure logging. It specifies when the log files and keystore need to be archived, and the secure keystore regenerated, for subsequent secure logging. The default is five files per logger.

Naming Attributes

The Naming Attributes are global attributes. The values applied to them are carried across the Sun ONE Identity Server configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.)

The Naming Service allows clients to find the correct service URL if the platform is running more than one Identity Server. When a naming URL is found, the naming service will decode the session of the user and dynamically replace the protocol, host, and port with the parameters from the session. This ensures that the URL returned for the service is for the host that the user session was created on. The Naming Attributes are:

- Profile Service URL
- Session Service URL
- Logging Service URL
- Policy Service URL
- Auth Service URL
- SAML Web Profile/Artifact Service URL
- SAML SOAP Service URL
- SAML Web Profile/POST Service URL
- SAML Assertion Manager Service URL

Profile Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/profiles-service
```

This syntax allows for dynamic substitution of the profile URL based on the specific session parameters.

Session Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/sessionservice
```

This syntax allows for dynamic substitution of the session URL based on the specific session parameters.

Logging Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/loggingservice
```

This syntax allows for dynamic substitution of the logging URL based on the specific session parameters.

Policy Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/policyservice
```

This syntax allows for dynamic substitution of the policy URL based on the specific session parameters.

Auth Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/authservice
```

This syntax allows for dynamic substitution of the authentication URL based on the specific session parameters.

SAML Web Profile/Artifact Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/SAMLAwareServlet
```

This syntax allows for dynamic substitution of the SAML web profile/artifact URL based on the specific session parameters.

SAML SOAP Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/SAMLSOAPReceiver
```

This syntax allows for dynamic substitution of the SAML SOAP URL based on the specific session parameters.

SAML Web Profile/POST Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/SAMLPOSTProfileServlet
```

This syntax allows for dynamic substitution of the SAML web profile/POST URL based on the specific session parameters.

SAML Assertion Manager Service URL

This field takes a value equal to

```
%protocol://%host:%port/<DEPLOY_URI>/AssertionManagerServlet/AssertionManagerIF
```

This syntax allows for dynamic substitution of the SAML Assertion Manager Service URL based on the specific session parameters.

Platform Attributes

The Platform Attributes are global attributes. The values applied to them are carried across the Sun ONE Identity Server configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The Platform Attributes are:

- Server List
- Platform Locale
- Cookie Domains
- Login Service URL
- Logout Service URL
- Available Locales
- Client Char Sets

This attribute stores all available locales configured for the platform. Consider an application that lets the user choose their locale. This application would get this attribute from the platform profile and present the list of locales to the user. The user would choose a locale and the application would set this in the user entry `preferredLocale`.

Server List

The naming service reads this attribute at initialization time. This list contains the Identity Server session servers in a single Identity Server configuration. For example, if two Identity 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. At the end of the list, there is a two-byte value that uniquely identifies the server. Each server that is participating in load balancing needs to have a unique identifier. This is also used to shorten the cookie length by mapping the server URL to the server ID. For example:

```
protocol://<server_domain>:<port>|01
```

Additional servers can be added using the format

```
protocol://<server_domain>:<port>|01|<instance something>
```

Platform Locale

The platform locale value is the default language subtype that Identity Server was installed with. The authentication, logging and administration services are administered in the language of this value. The default is `en_US`. See Table 17-1 on page 180 for a listing of all supported language subtypes.

Cookie Domains

This is the list of domains that will be returned in the cookie header when setting a cookie to the user's browser during authentication. If empty, no cookie domain will be set. In other words, the Identity Server session cookie will only be forwarded to the Identity Server itself and no other servers in the domain. If SSO is required with other servers in the domain, this attribute must be set with the cookie domain. If you had two interfaces in different domains on one Identity Server then you would need to set both cookie domains in this attribute. If a load balancer is used, the cookie domain must be that of the load balancer's domain, not the servers behind the load balancer. The default value for this field is the domain of the installed Identity Server.

Login Service URL

This field specifies the URL of the login page. The default value for this attribute is `/<DEPLOY_URI>/UI/Login`.

Logout Service URL

This field specifies the URL of the logout page. The default value for this attribute is `/DEPLY_URI/UI/Logout`.

Available Locales

This attribute stores all available locales configured for the platform. Consider an application that lets the user choose their locale. This application would get this attribute from the platform profile and present the list of locales to the user. The user would choose a locale and the application would set this in the user entry `preferredLocale`.

Client Char Sets

This attribute specifies the character set for different clients at the platform level. It contains a list of client types and the corresponding character sets. The format is as follows:

```
clientType|charset
```

```
clientType2|charset
```

For example:

```
genericHTML|UTF-8
```


Policy Configuration Attributes

The Policy Configuration attributes consist of global and organization 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 can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The values applied to the organization attributes under Service Management become the default values for Policy configuration. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Policy Configuration attributes are separated into:

- Global Attribute
- Organization Attributes

Global Attribute

The global attribute in the Policy Configurative service is:

- Resource Comparator

Resource Comparator

This attribute specifies the resource comparator information, which is used to compare resources specified in a Policy rule definition. Resource comparison is used for both policy creation and evaluation. This attribute contains the following values:

<code>serviceType</code>	Specifies the service to which the comparator should be used.
<code>class</code>	Defines the java class that implements the resource comparison algorithm.
<code>wildcard</code>	Specifies the wildcard that can be defined in resource names
<code>delimiter</code>	Specifies the delimiter to be used in the resource name.
<code>caseSensitivity</code>	Specifies if the comparison of the two resources should consider or ignore case. <code>False</code> ignores case, <code>True</code> considers case.

Organization Attributes

The organization attributes in the Policy Configuration service are:

- LDAP Server and Port
- LDAP Base DN
- LDAP Bind DN
- LDAP Bind Password
- LDAP Bind Password (Confirm)
- LDAP Org Search Filter
- LDAP Org Search Scope
- LDAP Groups Search Filter
- LDAP Groups Search Scope
- LDAP Users Search Filter
- LDAP Users Search Scope
- LDAP Roles Search Filter

- LDAP Roles Search Scope
- LDAP Organization Search Attribute
- LDAP Groups Search Attribute
- LDAP Users Search Attribute
- LDAP Roles Search Attribute
- Maximum Results Returned From Search
- Timeout For Search (seconds)
- LDAP SSL Enabled
- LDAP Connection Pool Minimal Size
- LDAP Connection Pool Maximum Size
- Selected Policy Subjects
- Selected Policy Conditions
- Selected Policy Referrals
- Subjects Result Time To Live

LDAP Server and Port

This field specifies the host name and port number of the primary LDAP server specified during Identity Server installation that will be used to search for Policy subjects, such as LDAP users, LDAP roles, LDAP groups, etc.

The format is `hostname:port`. (If there is no port number, assume 389.) Multiple entries must be prefixed by the local server name.

LDAP Base DN

This field specifies the base DN in the LDAP server from which to begin the search. By default, it is the top-level organization of the Identity Server installation.

LDAP Bind DN

This field specifies the bind DN in the LDAP server.

LDAP Bind Password

This attribute defines the password to be used for binding to the LDAP server. By default, the `amldapuser` password that was entered during installation is used as the bind user.

LDAP Bind Password (Confirm)

Confirmation of the LDAP Bind password.

LDAP Org Search Filter

Specifies the search filter to be used to find organization entries. The default is `(objectclass=sunMangagedOrganization)`.

LDAP Org Search Scope

This attribute defines the scope to be used to find organization entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Groups Search Filter

Specifies the search filter to be used to find group entries. The default is `(objectclass=groupOfUniqueNames)`.

LDAP Groups Search Scope

This attribute defines the scope to be used to find group entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`

- `SCOPE_SUB` (default)

LDAP Users Search Filter

Specifies the search filter to be used to find user entries. The default is `(objectclass=inetorgperson)`.

LDAP Users Search Scope

This attribute defines the scope to be used to find user entries. The scope must be one of the following:

- `SCOPE-BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Roles Search Filter

Specifies the search filter to be used to find entries for roles. The default is `(&(objectclass=ldapsubentry)(objectclass=nsroleddefinitions)`

LDAP Roles Search Scope

This attribute defines the scope to be used to find entries for roles. The scope must be one of the following:

- `SCOPE-BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Organization Search Attribute

This field defines the attribute type for which to conduct a search on an organization. The default is `o`.

LDAP Groups Search Attribute

This field defines the attribute type for which to conduct a search on a group. The default is `cn`.

LDAP Users Search Attribute

This field defines the attribute type for which to conduct a search on a user. The default is `uid`.

LDAP Roles Search Attribute

This field defines the attribute type for which to conduct a search on a role. The default is `cn`.

Maximum Results Returned From Search

This field defines the maximum number of results returned from a search. The default value is 100. If the search limit exceeds the amount specified, the entries that have been found to that point will be returned.

Timeout For Search (seconds)

This attribute specifies the amount of time before a timeout on a search occurs. If the search exceeds the specified time, the entries that have been found to that point will be returned

LDAP SSL Enabled

This attribute specifies whether or not the LDAP server is running SSL. Selected enables SSL, unselected (default) disables SSL.

LDAP Connection Pool Minimal Size

This attribute specifies the minimal size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default is 1.

LDAP Connection Pool Maximum Size

This attribute specifies the maximum size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default is 10.

Selected Policy Subjects

This attribute allows you to select a set of subject types available to be used for policy definition in the organization.

Selected Policy Conditions

This attribute allows you to select a set of conditions types available to be used for policy definition in the organization.

Selected Policy Referrals

This attribute allows you to select a set of referral types available to be used for policy definition in the organization.

Subjects Result Time To Live

This attribute specifies the amount of time (in minutes) that a cached subject result can be used to evaluate the same policy request based on an single sign-on token.

When a policy is initially evaluated for an SSO token, the subject instances in the policy are evaluated to determine whether the policy is applicable to a given user. The subject result, which is keyed by the SSO token ID, is cached in the policy. If another evaluation occurs for the same policy for the same SSO token ID within the time specified in the Subject Result Time To Live attribute, the policy framework retrieves the cached subjects result, instead of evaluating the subject instances. This significantly reduces the time for policy evaluation.

SAML Attributes

The Security Assertion Markup Language (SAML) Attributes are global attributes. The values applied to them are carried across the Sun ONE Identity Server configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.)

For more information about the SAML Service architecture, see the *Sun One Identity Server Programmer's Guide*.

The SAML attributes are as follows:

- Site ID And Site Issuer Name
- Sign Request
- Sign Response
- Sign Assertion
- Artifact Name
- Target Specifier
- Artifact Timeout (seconds)
- Assertion Skew Factor For notBefore Time
- Assertion Timeout (seconds)
- Trusted Partner Sites
- POST To Target URLs

Site ID And Site Issuer Name

This attribute contains a list of entries, with each entry containing an instance ID, site ID, and site issuer name. The default value will be assigned during installation. The format is as follows:

```
instanceid=serverprotocol://servername:portnumber|siteid=<site_id>|  
issuerName=<site_issuer_name>
```

Sign Request

This attribute specifies whether all SAML requests will be digitally signed (XML DSIG) before being delivered. Clicking on this option will enable this feature.

Sign Response

This attribute specifies whether all SAML responses will be digitally signed (XML DSIG) before being delivered. Clicking on this option will enable this feature.

All SAML responses used by the SAML Web Post profile will be digitally signed whether this option is enabled or not enabled.

Sign Assertion

This attribute specifies whether all SAML assertions will be digitally signed (XML DSIG) before being delivered. Clicking on this option will enable this feature.

Artifact Name

This attribute assigns a variable name to a SAML artifact defined in the SAML Service configuration. A SAML artifact is bounded-size data, which identifies an assertion and a source site. It is carried as part of a URL query string and conveyed by a re-direction to the destination site. The default is `SAMLart`.

Target Specifier

This attribute assigns a variable name to the destination site URL used in the re-direct. The default is `Target`.

Artifact Timeout (seconds)

This attribute specifies the timeout for an assertion created for an artifact. The default is 120.

Assertion Skew Factor For notBefore Time

This attribute is used to calculate the notBefore time of an assertion. For example, if the IssueInstant is 2002-09024T21:39:49Z, and the Assertion Skew Factor notBefore Time value is set to 300 seconds (which is the default value), the notBefore attribute of the conditions element for the assertion would be 2002-09-24T21:34:49Z.

Assertion Timeout (seconds)

This attribute specifies the number of seconds before a timeout occurs on an assertion. The default is 60.

Trusted Partner Sites

This attribute stores a partner's information so that one site can establish a trusted relationship to communicate with another partner site.

This attribute contains a list of entries, with each entry containing key/value pairs (separated by “|”). The source ID is required for each entry. For example:

```
SourceID=siteid|SOAPURL=https://servername:portnumber/amserver/SAML  
SOAPReceiver|AuthType=SSL|hostlist=ipaddress
```

The parameters are:

SourceID

The 20-byte sequence defined as part of the SiteId.

target	<p>This parameter is defined in a specific domain, with or without a port number. If you wish to contact a web page hosted in that specific domain, <code>target</code> specifies the redirect to a URL defined by the <code>SAMLUrl</code> or <code>POSTUrl</code> parameters for further processing.</p> <p>If there are two entries (one containing a port number and one not containing a port number) that have the same domain specified in the Trusted Partner Sites attribute, the entry with the port number has a higher priority.</p> <p>For example, if you have the following two trusted partner sites definitions:</p> <pre>target=sun.com SAMLUrl=http://machine1.sun.com:8080/amserver/SAMLAwareServlet</pre> <p>and</p> <pre>target=sun.com:8080 SAMLUrl=http://machine2.sun.com:80/amserver/SAMLAwareServlet</pre> <p>and are seeking a the following page:</p> <pre>http://somemachine.sun.com:8080/index.html</pre> <p>the second definition will be chosen as the SAML service provider because the matching domain and port coexist in the <code>target</code> parameter.</p>
SAMLUrl	Defines the URL that provides the SAML service. The servlet specified in the URL implements the Web-browser SSO with Artifact profile defined in the OASIS-SAML Bindings and Profiles specification.
POSTUrl	Defines the URL that provides the SAML service. The servlet specified in this URL implements the Web-browser SSO with POST profile defined in the OASIS-SAML Binding and Profiles specification.
issuer	Defines the creator of an assertion generated within Identity Server. The syntax is <code>hostname:port</code> .
SOAPUrl	Specifies the SOAP Receiver service.

AuthType	<p>Defines the authentication type used in SAML. It should be one of the following:</p> <ul style="list-style-type: none"> • NOAUTH • BASICAUTH • SSL • SSLWITHBASICAUTH <p>This parameter is optional, and if not specified, the default is NOAUTH.</p> <p>If BASICAUTH or SSLWITHBASICAUTH is specified, the User parameter is required.</p>
User	<p>Defines the uid of the partner which is used to protect the partner's SOAP Receiver.</p>
hostlist	<p>This attribute lists the IP addresses and/or the certAlias for all of the hosts, within the specified partner site, that can send requests to this site. This ensures that the requester is indeed the intended receiver for the SAML artifact.</p>
AccountMapper	<p>Specifies a pluggable class which defines how the subject of an Assertion is related to an identity at the destination site. By default, it is:</p> <pre>com.sun.identity.saml.plugins.DefaultAccountMapper</pre>
attributeMapper	<p>Specifies the class with the path to where the attributeMapper is located. Applications can develop an attributeMapper to obtain either an SSOToken ID or an assertion containing AuthenticationStatement from the query. The mapper is then used to retrieve the attributes for the subject. If no attributeMapper is specified, DefaultAttributeMapper will be used.</p>
actionMapper	<p>Specifies the class with the path to where the actionMapper is located. Applications can develop an actionMapper to obtain either an SSOToken ID or an assertion containing AuthenticationStatement from the query. The mapper is then used to retrieve the authorization decisions for the actions defined in the query. If no actionMapper is specified, DefaultActionMapper will be used.</p>

<code>siteAttributeMapper</code>	Specifies the class with the path where the <code>siteAttributeMapper</code> is located. Applications can develop a <code>siteAttributeMapper</code> to obtain attributes to be included in the assertion during SSO. If no <code>siteAttributeMapper</code> is found, then no attributes will be included in the assertion during SSO.
<code>certAlias=<aliasName></code>	Specifies a <code>certAlias</code> name used for verifying the signature in an assertion, when the assertion is signed by a partner and the certificate of the partner can not be found in the <code>KeyInfo</code> portion of the signed assertion.

The following table lists an example configuration for trusted partner sites. Not all of the parameters are necessary for all use cases, so the optional parameters are contained in brackets.

	Sender	Receiver
artifact	<code>sourceid</code>	<code>sourceid</code>
	<code>target</code>	<code>SOAPUrl</code>
	<code>SAMLUrl</code>	<code>[accountMapper]</code>
	<code>hostlist</code>	<code>[AuthType]</code>
	<code>[siteAttributeMapper]</code>	<code>[User]</code> <code>[certAlias]</code>
POST profile	<code>sourceid</code>	<code>sourceid</code>
	<code>target</code>	<code>issuer</code>
	<code>POSTUrl</code>	<code>[accountMapper]</code>
	<code>[siteAttributeMapper]</code>	<code>[certAlias]</code>
SOAP Request		<code>sourceid</code>
		<code>hostlist</code>
		<code>[attributeMapper]</code>
		<code>[actionMapper]</code>

Sender

Receiver

[certAlias]

[issuer]

POST To Target URLs

If the target URL received through SSO (either artifact profile or POST profile) by the site is listed in this attribute, the assertion or assertions that are received from SSO will be sent to the target URL by an http: FORM POST.

Session Attributes

The Session Attributes are dynamic attributes. The values applied to these attributes are applied to either a role or an organization. If the role is assigned to a user or a user is assigned to the organization, these attributes, by default, are inherited by the user. The Session Attributes are:

- Max Session Time (Minutes)
- Max Idle Time (Minutes)
- Max Caching Time (Minutes)

Default session values are set in Service Configuration for all Identity Server registered organizations. These values can be set differently for separate organizations by registering the session service to the specific organization, creating a template and inputting a value other than the default value.

Max Session Time (Minutes)

This attribute accepts a value in minutes to express the maximum time before the session expires and the user must reauthenticate to regain access. A value of 1 or higher will be accepted. The default value is 120. (To balance the requirements of security and convenience, consider setting the Max Session Time interval to a higher value and setting the Max Idle Time interval to a relatively low value.)

Max Idle Time (Minutes)

This attribute accepts a value (in minutes) equal to the maximum amount of time without activity before a session expires and the user must reauthenticate to regain access. A value of 1 or higher will be accepted. The default value is 30. (To balance the requirements of security and convenience, consider setting the Max Session Time interval to a higher value and setting the Max Idle Time interval to a relatively low value.)

Max Caching Time (Minutes)

This attribute accepts a value (in minutes) equal to the maximum interval before the client contacts Identity Server to refresh cached session information. A value of 0 or higher will be accepted. The default value is 3.

User Attributes

There are two places which house user attributes: the Service Configuration and User Management windows. The Service Configuration window contains default attributes for registered organizations. The User Management window contains user entry attributes.

- User Attributes
- User Profile Attributes
- Unique User IDs

User Attributes

The User Attributes are dynamic attributes. The values applied to dynamic attributes are assigned to a role or an organization that is configured in Identity Server. When the role is assigned to a user or a user is assigned to the organization, the dynamic attributes become a characteristic of the user. The User Attributes are divided into:

- User Preferred Language
- User Preferred Timezone
- Inherited Locale
- Admin DN Starting View
- Default User Status

Default user values are set for all Identity Server registered organizations. These values can be set differently for separate organizations by registering the user service to the specific organization, creating a template and inputting a value other than the default value.

User Preferred Language

This field specifies the user's choice for the text language displayed in the Identity Server console. The default value is `en`. This value maps a set of localization keys to the user session so that onscreen text appears in a language appropriate for the user.

User Preferred Timezone

This field specifies the time zone in which the user accesses the Identity Server console. There is no default value.

Inherited Locale

This field specifies the locale for the user. The default value is `en_US`. Any value from Table 17-1 on page 180 can be used.

Admin DN Starting View

If this user is a Identity Server administrator, this field specifies the node that would be the starting point displayed in the Identity Server console when this user logs in. There is no default value. A valid DN for which the user has, at the least, read access can be used.

CAUTION If the Top-Level Administrator wishes to assign a user the administration privileges to two different groups, the Admin DN Starting View should be specified as the DN of the level above BOTH groups. This holds true for any entries at the same level such as organizations, or groups. This action could result in the user being able to manage an organization, or group that is not specifically assigned to them. It is up to the Top-Level Administrator to decide on the ACI model and where to define the DN Starting View.

Default User Status

This option indicates the default status for any newly created user. This status is superseded by the User Entry status. Only active users can authenticate through Identity Server. The default value is `Active`. Either of the following can be selected from the pull-down menu:

- `Active` – The user can authenticate through Identity Server.
- `Inactive` – The user cannot authenticate through Identity Server, but the user profile remains stored in the directory.

The individual user status is set by registering the User service, choosing the value, applying it to a role and adding the role to the user's profile.

User Profile Attributes

The User Profile Attributes are default attributes for user profiles. These values are set in the User Profile view by an administrator or by the user when they log on. Administrators can add their own user attributes to the user profile or create a new service. For more information see *Sun One Identity Server Programmer's Guide*.

NOTE Identity Server does not enforce uniqueness for attributes within user entries. For example, `userA` and `userB` are both created in the same organization. For both, the email address attribute can be set `jimb@madisonparc.com`. The administrator can configure Sun ONE Directory Server's attribute uniqueness plug-in to help enforce unique attribute values. For more information, see Unique User IDs at the end of this chapter or the *Sun One Directory Server Administrator's Guide*.

First Name

This field takes the first name of the user. (The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Identity Server console.)

Last Name

This field takes the last name of the user. (The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Identity Server console.)

Full Name

This field takes the full name of the user.

Password

This field takes the password for the name specified in the UserId field.

Password (Confirm)

Confirmation of the password.

Email Address

This field takes the email address of the user.

Employee Number

This field takes the employee number of the user.

Telephone Number

This field takes the telephone number of the user.

Home Address

This field can take the home address of the user.

User Status

This option indicates whether the user is allowed to authenticate through Identity Server. Only active users can authenticate through Identity Server. The default value is `Active`. Either of the following can be selected from the pull-down menu:

- `Active` – The user can authenticate through Identity Server.
- `Inactive` – The user cannot authenticate through Identity Server, but the user profile remains stored in the directory.

NOTE

Changing the user status to `Inactive` only affects authentication through Identity Server. The Directory Server uses the `nsAccountLock` attribute to determine user account status. User accounts inactivated for Identity Server authentication can still perform tasks that do not require Identity Server. To inactivate a user account in the directory, and not just for Identity Server authentication, set the value of `nsAccountLock` to `false`. If delegated administrators at your site will be inactivating users on a regular basis, consider adding the `nsAccountLock` attribute to the Identity Server User Profile page. See the *Sun One Identity Server Programmer's Guide* for details.

Account Expiration Date

If this attribute is present, the authentication service will disallow login if the current date and time has passed the specified Account Expiration Date. The format for this attribute is as follows:

(mm/dd/yyyy hh:mm)

User Authentication Configuration

This attribute sets the authentication method for the user. The default authentication method is LDAP. One or more authentication methods can be selected by clicking the Edit link. If more than one method is selected, then the user may have to successfully authenticate to all of selected methods.

User Alias List

The field defines a list of aliases that may be applied to the user.

Preferred Locale

This field specifies the locale for the user. The default value is `en_US`. Any value from Table 17-1 on page 180 can be used.

You can use one of the following attributes in the pull-down menu:

- Ignore
- Customize
- Inherit

Success URL

This attribute specifies the URL that the user will be redirected to upon successful authentication.

Failure URL

This attribute specifies the URL that the user will be redirected to upon unsuccessful authentication.

Unique User IDs

In order to enforce uid uniqueness within the Identity Server application, the plug-in, available in Directory Server, must be configured as follows:

```
dn: cn=uid uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: uid uniqueness
nsslapd-pluginPath: /ids908/lib/uid-plugin.so
```

```

nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=uid
nsslapd-pluginarg1: markerObjectClass=nsManagedDomain
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
nsslapd-pluginVersion: 5.1
nsslapd-pluginVendor: Sun | Netscape Alliance
nsslapd-pluginDescription: Enforce unique attribute values

```

It is recommended that the `nsManagedDomain` object class is used to mark the organization in which uid uniqueness is desired. The plug-in is not enabled by default.

To configure the uniqueness of uids per organization, either add the DN for each organization in the plug-in entry or use the marker object class option and add `nsManagedDomain` to each top-level organization entry.

```

nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=uid
nsslapd-pluginarg1: markerObjectClass=nsManagedDomain

```


Identity Server Security Service Attributes

The Identity Server Security Service Attributes are dynamic attributes. The values applied to these attributes are applied to either a role or an organization. If the role is assigned to a user or a user is assigned to the organization, these attributes, by default, are inherited by the user. The Identity Server Security Service Attributes are:

- Enrollment URL
- Inherited Country

NOTE In order to enable the Identity Server Security Service, you must:

1. Install Sun ONE Certificate Server 4.7 SP1. For installation instructions, see the Certificate Server 4.7 release notes at <http://docs.sun.com/source/816-6407-10/index.html>.
 2. Configure the Certificate Server to enable the Identity Server Security Service. For configuration instructions, see the *Sun ONE Certificate Server Installation and Setup Guide* at http://docs.sun.com/db/coll/S1_s1CertificateServer_47.
 3. Define the Identity Server Security Service attributes described in this chapter.
-

Enrollment URL

This attribute specifies the place holder for the value of the Sun One Certificate Server's enrollment service. The syntax is one of the following:

`http://<hostname>:<non ssl end entity port number>/enrollment`

`https://<hostname>:<ssl end entity port number>/enrollment`

The parameters are as follows:

<i>hostname</i>	The hostname of the server where Certificate Server is installed.
<i>non ssl end entity port number</i>	The non-SSL end entity port number, by default, is 80. It is configured after running <code>Setup</code> for Certificate Server.
<i>ssl end entity port number</i>	The SSL end entity port number, by default, is 443. It is configured after running <code>Setup</code> for Certificate Server.
<code>enrollment</code>	The enrollment servlet in Certificate Server.

Inherited Country

This attribute is a dynamic attribute in the schema that is used by an administrator to assign a country that can be used by all users.

The `country` value is used in generating the CSR (Certificate Signing Request) to be sent to the Certificate Server servlet in order to generate a certificate for the user.

Configuring Identity Server in SSL Mode

Using Secure Socket Layer (SSL) with simple authentication guarantees confidentiality and data integrity.

Identity Server is capable of simultaneous SSL and non-SSL communications. This means that you do not have to choose between SSL or non-SSL communications; you can use both at the same time.

To configure Identity Server in SSL mode, see the following steps:

1. In the Identity Server console, click on the Properties arrow for the top-level organization (created during installation).

The Organization Properties window will display in the Data pane.

2. In the Full DNS Name attribute, change the protocol from `http://` to `https://`.
3. Click Save to save the changes.
4. In the Identity Server console, go to the Service Configuration module and select the Platform service. In the Server List attribute, remove the `http://` protocol, and add the `https://` protocol. Click Save.
5. Log on to the Sun ONE Web Server console. The default port is 58888.
6. Select the Web Server instance on which Identity Server is running, and click Manage.

This displays a pop-up window explaining that the configuration has changed. Click OK.

7. Click on the Apply button located top right corner of the screen.

8. Click Apply Settings.

The Web Server should automatically restart. Click OK to continue.

9. Stop the select Web Server instance.

10. Click the Security Tab.

11. Click on Create Database.

12. Enter the new database password and click OK.

Ensure that you write down the database password for later use.

13. Once the Certificate Database has been created, click on Request a Certificate.

14. Enter the data in the fields provided in the screen.

The Key Pair Field Password field is the same as you entered in Step 12. In the location field, you will need to spell out the location completely. Abbreviations, such as CA, will not work.

15. Once the form is submitted, you will see a message such as:

```
--BEGIN CERTIFICATE REQUEST---  
  
afajsdllwgeroisdao1234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdf  
alsfjawoeirjoi2ejowdnlswnvnwofijwoeijfwiepwerfoigeroijepwpfrwl  
  
--END CERTIFICATE REQUEST--
```

16. Copy this text and submit it for the certificate request.

Ensure that you get the Root CA certificate.

17. You will receive a certificate response containing the certificate, such as:

```
--BEGIN CERTIFICATE--  
  
afajsdllwgeroisdao234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdfasdf  
  
alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoigeroijeprwprwl  
  
--END CERTIFICATE--
```

18. Copy this text into your clipboard, or save the text into a file.
19. Go the Web Server console and click on Install Certificate.
20. Click on Certificate for this Server.
21. Enter the Certificate Database password in the Key Pair File Password field.
22. Paste the certificate into the provided text field, or check the radio button and enter the filename in the text box. Click Submit.

The browser will display the certificate, and provide a button to add the certificate.
23. Click Install Certificate.
24. Click Certificate for Trusted Certificate Authority.
25. Install the Root CA Certificate in the same manner described in Step 19 through Step 24.
26. Once you have completed installing both certificates, click on the Preferences tab in the Web Server console.
27. Select Add Listen Socket and then Edit Listen Socket.
28. Change the security status from Off to On, and click OK to submit the changes.
29. Open the `AMConfig.properties` file. By default, the location of this file is `/opt/SUNWam/lib`.
30. Replace all of the protocol occurrences of `http://` to `https://`, except for the Web Server Instance Directory. This is also specified in `AMConfig.properties`, but must remain the same.
31. Save the `AMConfig.properties` file.

- 32.** In the Web Server console, click the ON/OFF button for the Identity Server hosting web server instance.

The Web Server displays a text box in the Start/Stop page.

- 33.** Enter the Certificate Database password in the text field and select Start.

Index

A

- Account Federation 65
- Adding Conditions 91
- Adding Rules 88
- Admin Authenticator 177
- Admin DN Starting View 248
- Administration Attributes 149
 - Global Attributes 149
 - Default Admin Groups Enabled 154
 - Default Compliance User Deletion Enabled 154
 - Default Domain Component Tree Enabled 153
 - Default Role Permissions (ACIs) 152
 - Display Containers In Menu 150
 - Dynamic Admin Roles ACIs 155
 - Managed Group Type 151
 - Show Group Containers 151
 - Show People Containers 150
 - User Profile Service Class 157
 - Organization Attributes 157
 - Display User's Groups 159
 - Display User's Roles 159
 - Groups Default People Container 158
 - Groups People Container List 158
 - JSP Directory Name 160
 - Maximum Entries Per Page 163
 - Maximum Results Returned From Search 160
 - Online Help Documents 161
 - Required Services 161
 - Timeout For Search (sec.) 160
 - User Creation Default Roles 160
 - User Creation Notification List 161
 - User Deletion Notification List 162
 - User Group Self Subscription 159
 - User Modification Notification List 162
 - User Profile Display Class 158
 - User Profile Display Options 159
 - User Search Key 161
 - User Search Return Attribute 161
 - View Menu Entries 160
- Alias Search Attribute Name 178
- am2bak command line tool 139
 - Backup Procedure 141
 - Syntax 139
- amadmin command line tool 123
 - Creating policies with 127
 - Syntax 124
- ampassword command line tool 135
 - Running on SSL 136
 - Syntax 135
- amserver command line tool 129
 - Multi-server installation 130
 - Syntax 129
- Anonymous Authentication 99
 - Logging In With 100
 - Register and Enable 99
- Anonymous Authentication Attributes 165
 - Organization Attributes
 - Authentication Level 165
 - Default Anonymous User Name 166
 - Valid Anonymous User List 166
- Artifact Name 238
- Artifact Timeout 239
- Assertion Skew Factor For notBefore Time 239

- Assertion Timeout 239
- Attribute In Issuer DN To Use To Search CRL 168
- Attribute in Subject DN To Use To Search LDAP 168
- Attributes
 - Attribute Types 56
 - Dynamic Attributes 56
 - Global Attributes 57
 - Organization Attributes 57
 - Policy Attributes 57
 - User Attributes 56
- Auth Service URL 222
- Authentication
 - By Authentication Level 119
 - By Module 120
- Authentication Configuration 112, 211
 - For Organizations 116
 - For Roles 117
 - For Services 118
 - For Users 118
 - User Interface 112
- Authentication Configuration Attributes 211
 - Organization Attributes
 - Authentication Configuration 211
 - Authentication Post Processing Class 213
 - Conflict Resolution Level 213
 - Login Failure URL 213
 - Login Success URL 213
- Authentication Context 66
- Authentication Domain 64
- Authentication Domains
 - Creating 69
 - Deleting 70
 - Modifying 70
- Authentication Level
 - Anonymous Authentication 165
 - Certificate Authentication 171
 - LDAP Authentication 188
 - Membership Authentication 195
 - RADIUS Authentication 200
 - SafeWord Module Authentication Level 205
 - Unix Module Authentication Level 209
- Authentication Post Processing Class 183, 213
- Available Locales 227

B

- bak2am command line tool 143
 - Syntax 143

C

- Certificate Authentication Attributes 167
 - Organization Attributes
 - Attribute In Issuer DN To Use To Search CRL 168
 - Attribute in Subject DN To Use To Search LDAP 168
 - Authentication Level 171
 - Enable OCSP Validation 169
 - Field in Cert to Use to Access User Profile 170
 - LDAP Attribute for Profile ID 170
 - LDAP Server and Port 169
 - LDAP Server Principal Password 170
 - LDAP Server Principal User 170
 - LDAP Start Search DN 169
 - Match Certificate in LDAP 168
 - Match Certificate to CRL 168
 - Other Field in Cert to Use to Access User Profile 171
 - SSL On For LDAP Access 170
- Certificate Server
 - Documentation 20
- Certificate-based Authentication 100
 - Logging In With 102
 - Register and Enable 101
- Client Char Sets 227
- Client Detection Attributes 215
 - Global Attributes
 - Client Detection Class 215
 - Client Detection Enabled 216
 - Client Types 215
 - Default Client Type 215
- Client Detection Class 215
- Client Detection Enabled 216
- Client Types 215
- Command line tools
 - am2bak 139
 - Backup procedure 141

- Syntax 139
- amadmin 123
 - Creating policies with 127
 - Syntax 124
- ampassword 135
 - Running on SSL 136
 - Syntax 135
- amsrver 129
 - Multi-server installation 130
 - Syntax 129
- bak2am 143
 - Syntax 143
- VerifyArchive 145
 - Syntax 145
- Common Domain 65
- Configurable Log Fields 219
- Confirm Password 250
- Conflict Resolution Level 213
- Console See Identity Server Console
- Containers 46
 - Creating 47
 - Deleting 47
- Cookie Domains 226
- Core Authentication
 - Global Attributes 173
 - LDAP Connection Default Pool Size 174
 - LDAP Connection Pool Size 174
 - Pluggable Auth Module Classes 174
 - Supported Auth Modules for Clients 174
- Organization Attributes 175
 - Admin Authenticator 177
 - Alias Search Attribute Name 178
 - Authentication Post Processing Class 183
 - Default Auth Level 179
 - Default Auth Locale 179
 - Default Failure Login URL 183
 - Default Success Login URL 183
 - Email Address to Send Lockout Notification 182
 - Lockout Attribute Name 182
 - Lockout Attribute Value 183
 - Login Failure Lockout Count 181
 - Login Failure Lockout Duration 182
 - Login Failure Lockout Interval 182
 - Login Failure Lockout Mode 181

- Organization Authentication
 - Configuration 181
 - Organization Authentication Menu 176
 - People Container For All Users 178
 - Persistent Cookie Max Time (seconds) 177
 - Persistent Cookie Mode 177
 - User Name Generator Mode 183
 - User Naming Attribute 179
 - User Profile 176
 - User Profile Dynamic Creation Default Roles 177
 - Warn User After N Failure 182
- Core Authentication Attributes 173
- Core Authentication Service 98
 - Register and Enable 98
- Current Sessions
 - Interface 59
 - Session Management
 - Terminating a Session 61
 - Session Management Window 60

D

- Database Driver Name 219
- Database User Name 218
- Database User Password 219
- Default Anonymous User Name 166
- Default Auth Level 179
- Default Auth Locale 179
- Default Client Type 215
- Default Failure Login URL 183
- Default Role Permissions (ACIs) 152
- Default Success Login URL 183
- Default User Roles 192
- Default User Status 248, 249
- Developer Information 21
- Display Containers In Menu 150
- Display User's Groups 159
- Display User's Roles 159
- DN for Root User Bind
 - LDAP Authentication 186
 - Membership Authentication 193

- DN to Start User Search
 - LDAP Authentication 186
 - Membership Authentication 193
- Documentation
 - Certificate Server 20
 - Overview 18
 - Proxy Server 20
 - Terminology 19
 - Web Server 20
- Downloads
 - Sun ONE Software 20
- DSAME Console
 - Data Pane 30
- Dynamic Admin Roles ACIs 155
- Dynamic Attributes
 - Admin DN Starting View 248
 - Default User Status 248, 249
 - Enrollment URL 255
 - Inherited Country 256
 - Max Caching Time (Minutes) 246
 - Max Idle Time (Minutes) 246
 - Max Session Time (Minutes) 245
 - User Preferred Language 248
 - User Preferred Locale 248
 - User Preferred Timezone 248
- Dynamic Groups 151

E

- Email Address 250
- Email Address to Send Lockout Notification 182
- Employee Number 250
- Enable OCSP Validation 169
- Enable SSL to LDAP Server
 - LDAP Authentication 188, 194
- Enrollment URL 255

F

- Federated Identity 65
- Federation Management 63

- Authentication Domains
 - Creating 69
 - Deleting 70
 - Modifying 70
- Concepts
 - Account Federation 65
 - Authentication Context 66
 - Authentication Domain 64
 - Common Domain 65
 - Federated Identity 65
 - Federation Termination 65
 - Hosted Provider 64
 - Identity Provider 64
 - Metadata 65
 - Name Identifier 66
 - Remote Provider 64
 - Service Provider 64
 - Single Logout 65
 - Single Sign-on 65
- Federated Identity 63
- Hosted Providers
 - Creating 74
 - Deleting 81
 - Modifying 76
- Liberty Alliance Project 66
- Remote Providers
 - Creating 71
 - Deleting 81
 - Modifying 73
- Federation Termination 65
- Field in Cert to Use to Access User Profile 170
- Filtered Groups 151
- First Name 249
- Full Name 250

G

- Global Attributes 173
 - Admin Groups Enabled 154
 - Artifact Name 238
 - Artifact Timeout 239
 - Assertion Skew Factor For notBefore Time 239
 - Assertion Timeout 239
 - Auth Service URL 222

- Available Locales 227
- Client Char Sets 227
- Client Detection Class 215
- Client Detection Enabled 216
- Client Types 215
- Compliance User Deletion Enabled 154
- Configurable Log Fields 219
- Cookie Domains 226
- Database Driver Name 219
- Database User Name 218
- Database User Password 219
- Default Client Type 215
- Default Role Permissions (ACIs) 152
- Display Containers In Menu 150
- Domain Component Tree Enabled 153
- Dynamic Admin Roles ACIs 155
- LDAP Connection Default Pool Size 174
- LDAP Connection Pool Size 174
- Log Location 218
- Log Signature Time 219
- Log Verification Time 219
- Logging Service URL 222
- Logging Type 218
- Login Service URL 226
- Logout Service URL 227
- Managed Group Type 151
- Max Log Size 218
- Maximum Number of Records 220
- Number of Files Per Archive 220
- Number of History Files 218
- Platform Locale 226
- Pluggable Auth Module Classes 174
- Policy Service URL 222
- POST To Target URLs 243
- Profile Service URL 221
- Remote Buffer Size 220
- Resource Comparator 230
- SAML Assertion Manager Service URL 223
- SAML SOAP Service URL 223
- SAML Web Profile/Artifact Service URL 223
- SAML Web Profile/POST Service URL 223
- Secure Logging 220
- Server List 225
- Session Service URL 222
- Show Group Containers 151
- Show People Containers 150
- Sign Assertion 238

- Sign Request 238
- Sign Response 238
- Site ID And Site Issuer Name 238
- Supported Auth Modules for Clients 174
- Target Specifier 238
- Trusted Partner Sites 239
- Unix Helper Authentication Port 208
- Unix Helper Configuration Port 208
- Unix Helper Threads 208
- Unix Helper Timeout 208
- User Profile Service Class 157
- Group Containers 48
 - Creating 49
 - Deleting 49
- Groups 36
 - Adding to a Policy 37
 - Create a Managed Group 36
 - Deleting 37
 - Dynamic Groups 151
 - Filtered Groups 151
 - Membership by Filter 36
 - Membership by Subscription 36
 - Static Groups 151
- Groups Default People Container 158
- Groups People Container List 158

H

- Help link 30
- Home Address 250
- Hosted Provider 64
- Hosted Providers
 - Creating 74
 - Deleting 81
 - Modifying 76

I

- Identity Management 31
 - Containers 46
 - Creating 47
 - Deleting 47

- Group Containers 48
 - Creating 49
 - Deleting 49
- Groups 36
 - Adding to a Policy 37
 - Create a Managed Group 36
 - Deleting 37
 - Dynamic Groups 151
 - Filtered Groups 151
 - Membership by Filter 36
 - Membership by Subscription 36
 - Static Groups 151
- Identity Management Interface 31
 - Identity Management View 31
 - User Profile View 32
- Organizations 34
 - Adding to a Policy 35
 - Creating 34
 - Deleting 35
- People Containers 47
 - Creating 48
 - Deleting 48
- Policies 46
- Properties 33
- Roles 40
 - Adding to a Policy 43
 - Adding Users to 42
 - Creating 41
 - Deleting 42
 - Removing Users from 43
 - Role Properties 43
- Services 39
 - Creating a Template 39
 - Registering 39
 - Unregistering 40
- Users 37
 - Adding to a Policy 38
 - Adding to Services, Roles and Groups 38
 - Creating 37
 - Deleting 38
- Identity Provider 64
- Identity Server 25
 - Console 29
 - Features 26
 - Authentication 27
 - Federation Management 26
 - Identity Management 27
 - Identity Server Console 28
 - Policy Management 26
 - SAML 26
 - Service Configuration 26
 - Single Sign-On 27
 - URL Policy Agents 27
 - Installation 28
 - Related Product Information 20
- Identity Server Console
 - Location Pane 29
 - Help link 30
 - Location field 29
 - Logout 30
 - Modules 29
 - Search Link 30
 - Welcome 30
 - Navigation Pane 30
- Identity Server Security Service Attributes 255
 - Dynamic Attributes
 - Enrollment URL 255
 - Inherited Country 256
 - Inherited Country 256

J

- JSP Directory Name 160

L

- Last Name 250
- LDAP Attribute for Profile ID 170
- LDAP Authentication Attributes 185
 - Organization Attributes
 - Authentication Level 188
 - DN for Root User Bind 186
 - DN to Start User Search 186
 - Enable SSL to LDAP Server 188, 194
 - Password for Root User Bind 187, 193
 - Primary LDAP Server and Port 186
 - Return User DN To Auth 188
 - Search Scope 188
 - Secondary LDAP Server and Port 186

- User Entry Naming Attribute 187
- User Entry Search Attributes 187
- User Search Filter 187
- LDAP Base DN 231
- LDAP Bind DN 231
- LDAP Bind Password 232
- LDAP Connection Default Pool Size 174
- LDAP Connection Pool Maximum Size 235
- LDAP Connection Pool Minimal Size 235
- LDAP Connection Pool Size 174
- LDAP Directory Authentication 102
 - Enabling Failover 104
 - Logging In With 103
 - Register and Enable 102
- LDAP Group Search Attribute 234
- LDAP Groups Search Filter 232
- LDAP Groups Search Scope 232
- LDAP Org Search Filter 232
- LDAP Org Search Scope 232
- LDAP Organization Search Attribute 233
- LDAP Roles Search Attribute 234
- LDAP Roles Search Filter 233
- LDAP Roles Search Scope 233
- LDAP Server and Port 169, 231
- LDAP Server Principal Password 170
- LDAP Server Principal User 170
- LDAP SSL Enabled 234
- LDAP Start Search DN 169
- LDAP Users Search Attribute 234
- LDAP Users Search Filter 233
- LDAP Users Search Scope 233
- Liberty See Federation Management
- Lockout Attribute Name 182
- Lockout Attribute Value 183
- Log Location 218
- Log Signature Time 219
- Log Verification Time 219
- Logging Attributes 217
 - Global Attributes
 - Configurable Log Fields 219
 - Database Driver Name 219
 - Database User Name 218
 - Database User Password 219

- Log Location 218
- Log Signature Time 219
- Log Verification Time 219
- Logging Type 218
- Max Log Size 218
- Maximum Number of Records 220
- Number of Files Per Archive 220
- Number of History Files 218
- Remote Buffer Size 220
- Secure Logging 220
- Logging Service URL 222
- Logging Type 218
- Login Failure Lockout Count 181
- Login Failure Lockout Duration 182
- Login Failure Lockout Interval 182
- Login Failure Lockout Mode 181
- Login Failure URL 213
- Login Service URL 226
- Login Success URL 213
- Logout 30
- Logout Service URL 227

M

- Managed Group Type 151
- Managing Identity Server Objects 33
- Match Certificate in LDAP 168
- Match Certificate to CRL 168
- Max Caching Time (Minutes) 246
- Max Idle Time (Minutes) 246
- Max Log Size 218
- Max Session Time (Minutes) 245
- Maximum Entries Per Page 163
- Maximum Number of Records 220
- Maximum Results Returned From Search 160
- Membership Authentication 104
 - Logging In With 105
 - Register and Enable 104
- Membership Authentication Attributes 191
 - Organization Attributes
 - Authentication Level 195
 - Default User Roles 192

- DN for Root User Bind 193
- DN to Start User Search 193
- Minimum Password Length 192
- Primary LDAP Authentication Server 192
- Return User DN to Auth 194
- Search Scope 194
- Secondary LDAP Authentication Server 193
- User Entry Search Attributes 194
- User Naming Attribute 194
- User Search Filter 194
- User Status After Registration 192
- metadata 63
- Minimum Password Length 192

N

- Name Identifier 66
- Naming Attributes 221
 - Global Attributes
 - Auth Service URL 222
 - Logging Service URL 222
 - Policy Service URL 222
 - Profile Service URL 221
 - SAML Assertion Manager Service URL 223
 - SAML SOAP Service URL 223
 - SAML Web Profile/Artifact Service URL 223
 - SAML Web Profile/POST Service URL 223
 - Session Service URL 222
- Normal Policy 84, 88, 91
 - Adding Subjects 90
 - Creating 87
 - Modifying 88
- NT Authentication 105
 - Logging In With 107
 - Organization Attributes
 - NT Authentication Domain 197
 - NT Authentication Host 197
 - NT Module Authentication Level 198
 - Register and Enable 106
- NT Authentication Attributes 197
- NT Authentication Domain 197
- NT Authentication Host 197
- NT Module Authentication Level 198

- Number of Files Per Archive 220
- Number of History Files 218

O

- Online Help Documents 161
- Organization Attributes 157
 - Admin Authenticator 177
 - Alias Search Attribute Name 178
 - Attribute In Issuer DN To Use To Search CRL 168
 - Attribute In Subject DN To Use To Search LDAP 168
- Authentication Configuration 211
- Authentication Level
 - Anonymous Authentication 165
 - Certificate Authentication 171
 - LDAP Authentication 188
 - Membership Authentication 195
 - RADIUS Authentication 200
- Authentication Post Processing Class 183, 213
- Conflict Resolution Level 213
- Default Anonymous User Name 166
- Default Auth Level 179
- Default Auth Locale 179
- Default Failure Login URL 183
- Default Success Login URL 183
- Default User Roles 192
- Display User's Groups 159
- Display User's Roles 159
- DN for Root User Bind
 - LDAP Authentication 186
 - Membership Authentication 193
- DN to Start User Search
 - LDAP Authentication 186
 - Membership Authentication 193
- Email Address to Send Lockout Notification 182
- Enable OCSP Validation 169
- Enable SSL to LDAP Server
 - LDAP Authentication 188, 194
- Field in Cert to Use to Access User Profile 170
- Groups Default People Container 158
- Groups People Container List 158
- JSP Directory Name 160
- LDAP Attribute for Profile ID 170
- LDAP Base DN 231

- LDAP Bind DN 231
- LDAP Bind Password 232
- LDAP Connection Pool Maximum Size 235
- LDAP Connection Pool Minimal Size 235
- LDAP Group Search Attribute 234
- LDAP Groups Search Filter 232
- LDAP Groups Search Scope 232
- LDAP Org Search Filter 232
- LDAP Org Search Scope 232
- LDAP Organization Search Attribute 233
- LDAP Roles Search Attribute 234
- LDAP Roles Search Filter 233
- LDAP Roles Search Scope 233
- LDAP Server and Port 169, 231
- LDAP Server Principal Password 170
- LDAP Server Principal User 170
- LDAP SSL Enabled 234
- LDAP Start Search DN 169
- LDAP Users Search Attribute 234
- LDAP Users Search Filter 233
- LDAP Users Search Scope 233
- Lockout Attribute Name 182
- Lockout Attribute Value 183
- Login Failure Lockout Count 181
- Login Failure Lockout Duration 182
- Login Failure Lockout Interval 182
- Login Failure Lockout Mode 181
- Login Failure URL 213
- Login Success URL 213
- Match Certificate in LDAP 168
- Match Certificate to CRL 168
- Maximum Entries Per Page 163
- Maximum Results Returned From Search 160, 234
- Minimum Password Length 192
- NT Authentication Domain 197
- NT Authentication Host 197
- NT Module Authentication Level 198
- Online Help Documents 161
- Organization Authentication Configuration 181
- Organization Authentication Menu 176
- Other Field in Cert to Use to Access User Profile 171
- Password for Root User Bind
 - LDAP Authentication 187
 - Membership Authentication 193
- People Container For All Users 178
- Persistent Cookie Max Time (seconds) 177
- Persistent Cookie Mode 177
- Primary LDAP Authentication Server 192
- Primary LDAP Server and Port 186
- RADIUS Server 1 199
- RADIUS Server 2 200
- RADIUS Server's Port 200
- RADIUS Shared Secret 200
- Required Services 161
- Return User DN To Auth
 - LDAP Authentication 188
- Return User DN to Auth
 - Membership Authentication 194
- SafeWord Log Path 204
- SafeWord Module Authentication Level 205
- SafeWord Server Specification 203
- SafeWord System Name 204
- Search Scope
 - LDAP Authentication 188
 - Membership Authentication 194
- Secondary LDAP Authentication Server 193
- Secondary LDAP Server and Port 186
- Selected Policy Conditions 235
- Selected Policy Referrals 235
- Selected Policy Subjects 235
- SSL On For LDAP Access 170
- Subjects Result Time To Live 235
- Timeout (Seconds) 201
- Timeout For Search 234
- Timeout For Search (sec.) 160
- Unix Module Authentication Level
 - Unix Module Authentication Level 209
- User Creation Default Roles 160
- User Creation Notification List 161
- User Deletion Notification List 162
- User Entry Naming Attribute 187
- User Entry Search Attributes 187
 - Membership Authentication 194
- User Group Self Subscription 159
- User Modification Notification List 162
- User Name Generator Mode 183
- User Naming Attribute
 - Core Authentication 179
 - Membership Authentication 194
- User Profile 176
- User Profile Display Class 158
- User Profile Display Options 159

- User Profile Dynamic Creation Default Roles 177
- User Search Filter
 - LDAP Authentication 187
 - Membership Authentication 194
- User Search Key 161
- User Search Return Attribute 161
- User Status After Registration 192
- Valid Anonymous User List 166
- View Menu Entries 160
- Warn User After N Failure 182
- Organization Authentication Configuration 181
- Organization Authentication Menu 176
- Organizations 34
 - Adding to a Policy 35
 - Creating 34
 - Deleting 35
- Other Field in Cert to Use to Access User Profile 171

P

- Password 250
- Password for Root User Bind
 - LDAP Authentication 187
 - Membership Authentication 193
- People Container For All Users 178
- People Containers 47
 - Creating 48
 - Deleting 48
- Persistent Cookie Max Time (seconds) 177
- Persistent Cookie Mode 177
- Platform Attributes 225
 - Global Attributes
 - Available Locales 227
 - Client Char Sets 227
 - Cookie Domains 226
 - Login Service URL 226
 - Logout Service URL 227
 - Platform Locale 226
 - Server List 225
- Platform Locale 226
- Pluggable Auth Module Classes 174
- Policy 83
 - Creating 87

- Creating for Peer and Suborganizations 95
- Defined 83
- Normal Policy 84
 - Adding Conditions 91
 - Adding Rules 88
 - Adding Subjects 90
 - Creating 87
 - Modifying 88
- Referral Policy 85
 - Adding Referrals 94
 - Creating 87
 - Modifying 93
- Registering Policy Configuration Service 86
- Policy Configuration Attributes 229
 - Global Attributes
 - Resource Comparator 230
 - Organization Attributes
 - LDAP Base DN 231
 - LDAP Bind DN 231
 - LDAP Bind Password 232
 - LDAP Connection Pool Maximum Size 235
 - LDAP Connection Pool Minimal Size 235
 - LDAP Group Search Attribute 234
 - LDAP Groups Search Filter 232
 - LDAP Groups Search Scope 232
 - LDAP Org Search Filter 232
 - LDAP Org Search Scope 232
 - LDAP Organization Search Attribute 233
 - LDAP Roles Search Attribute 234
 - LDAP Roles Search Filter 233
 - LDAP Roles Search Scope 233
 - LDAP Server and Port 231
 - LDAP SSL Enabled 234
 - LDAP Users Search Attribute 234
 - LDAP Users Search Filter 233
 - LDAP Users Search Scope 233
 - Maximum Results Returned From Search 234
 - Selected Policy Conditions 235
 - Selected Policy Referrals 235
 - Selected Policy Subjects 235
 - Subjects Result Time To Live 235
 - Timeout For Search 234
- Policy Service URL 222
- POST To Target URLs 243
- Primary LDAP Authentication Server 192
- Primary LDAP Server and Port 186

- Professional Services 20
- Profile Service URL 221
- Properties 33
- Proxy Server
 - Documentation 20

R

- RADIUS Authentication Attributes 199
 - Organization Attributes
 - Authentication Level 200
 - RADIUS Server 1 199
 - RADIUS Server 2 200
 - RADIUS Server's Port 200
 - RADIUS Shared Secret 200
 - Timeout (Seconds) 201
- RADIUS Server 1 199
- RADIUS Server 2 200
- RADIUS Server Authentication 107
 - Logging In With 108
 - Register and Enable 107
- RADIUS Server's Port 200
- RADIUS Shared Secret 200
- Referral Policy 85
 - Adding Referrals 94
 - Creating 87
 - Modifying 93
- Registering Policy Configuration Service 86
- Remote Buffer Size 220
- Remote Provider 64
- Remote Providers
 - Creating 71
 - Deleting 81
 - Modifying 73
- Required Services 161
- Resource Comparator 230
- Return User DN To Auth
 - Membership Authentication 194
- Return User DN to Auth Authentication 188
- Roles 40
 - Adding to a Policy 43
 - Adding Users to 42

- Creating 41
- Deleting 42
- Removing Users from 43
- Role Properties 43

S

- SafeWord Authentication 109
 - Logging In With 110
 - Register and Enable 109
- SafeWord Authentication Attributes
 - Organization Attributes
 - SafeWord Log Path 204
 - SafeWord Logging Level 204
 - SafeWord Module Authentication Level 205
 - SafeWord Server Specification 203
 - SafeWord Server Verification Files Path 204
 - SafeWord System Name 204
- SafeWord Log Path 204
- SafeWord Logging Level 204
- SafeWord Module Authentication Level 205
- SafeWord Server Specification 203
- SafeWord Server Verification Files Path 204
- SafeWord System Name 204
- SAML Assertion Manager Service URL 223
- SAML Attributes 237
 - Global Attributes
 - Artifact Name 238
 - Artifact Timeout 239
 - Assertion Skew Factor For notBefore Time 239
 - Assertion Timeout 239
 - POST To Target URLs 243
 - Sign Assertion 238
 - Sign Request 238
 - Sign Response 238
 - Site ID And Site Issuer Name 238
 - Target Specifier 238
 - Trusted Partner Sites 239
- SAML SOAP Service URL 223
- SAML Web Profile/Artifact Service URL 223
- SAML Web Profile/POST Service URL 223
- Search Link 30
- Search Scope

- LDAP Authentication 188
- Membership Authentication 194
- Secondary LDAP Authentication Server 193
- Secondary LDAP Server and Port 186
- Secure Logging 220
- Selected Policy Conditions 235
- Selected Policy Referrals 235
- Selected Policy Subjects 235
- Server List 225
- Service Configuration
 - Service Configuration Module 58
- Service Configuration Interface 57
- Service Provider 64
- Services 39
 - Creating a Template 39
 - Default Services Defined 52
 - Certificate-based Authentication 52
 - Administration 52
 - Anonymous Authentication 52
 - Authentication Configuration 54
 - Client Detection 54
 - Core Authentication 53
 - Identity Server Security Service 55
 - LDAP Authentication 53
 - Logging 54
 - Membership Authentication 53
 - Naming 54
 - NT Authentication 53
 - Platform 54
 - Policy Configuration 54
 - RADIUS Authentication 53
 - SafeWord Authentication 53
 - SAML 55
 - Session 55
 - Unix Authentication 53
 - User 55
 - Defined 51
 - Registering 39
 - Unregistering 40
- Session Attributes 245
 - Dynamic Attributes
 - Max Caching Time (Minutes) 246
 - Max Idle Time (Minutes) 246
 - Max Session Time (Minutes) 245

- Session Service URL 222
- Show Group Containers 151
- Show People Containers 150
- Sign Assertion 238
- Sign Request 238
- Sign Response 238
- Single Logout 65
- Single Sign-on 65
- Site ID And Site Issuer Name 238
- Solaris
 - Patches 20
 - Support 20
- SSL
 - Configuring Identity Server For 257
- SSL On For LDAP Access 170
- Static Groups 151
- Subjects Result Time To Live 235
- Sun ONE
 - Support 20
- Support
 - Professional Services 20
 - Solaris 20
 - Sun ONE 20
- Supported Auth Modules for Clients 174
- Supported Language Locales 180

T

- Target Specifier 238
- Telephone Number 250
- Terminating a Session 61
- Timeout (Seconds) 201
- Timeout For Search 234
- Timeout For Search (sec.) 160
- Trusted Partner Sites 239

U

- Unique User IDs 252

- Unix Authentication 110
 - Logging In With 112
 - Register and Enable 110
- Unix Authentication Attributes 207
 - Global Attributes
 - Unix Helper Authentication Port 208
 - Unix Helper Configuration Port 208
 - Unix Helper Threads 208
 - Unix Helper Timeout 208
 - Organization Attributes
 - Unix Module Authentication Level 209
- Unix Helper Authentication Port 208
- Unix Helper Configuration Port 208
- Unix Helper Threads 208
- Unix Helper Timeout 208
- User Attributes 247
 - Service Management
 - Dynamic Attributes
 - Admin DN Starting View 248
 - Default User Status 248, 249
 - User Preferred Language 248
 - User Preferred Locale 248
 - User Preferred Timezone 248
- User Profile Attributes 249
 - Confirm Password 250
 - Email Address 250
 - Employee Number 250
 - First Name 249
 - Full Name 250
 - Home Address 250
 - Last Name 250
 - Password 250
 - Telephone Number 250
 - Unique User IDs 252
 - User Status 251
- User Creation Default Roles 160
- User Creation Notification List 161
- User Deletion Notification List 162
- User Entry Naming Attribute 187
- User Entry Search Attributes 187
 - Membership Authentication 194
- User Group Self Subscription 159
- User Modification Notification List 162
- User Name Generator Mode 183
- User Naming Attribute
 - Core Authentication 179
 - Membership Authentication 194
- User Preferred Language 248
- User Preferred Locale 248
- User Preferred Timezone 248
- User Profile 176
- User Profile Attributes 249
 - Confirm Password 250
 - Email Address 250
 - Employee Number 250
 - First Name 249
 - Full Name 250
 - Home Address 250
 - Last Name 250
 - Password 250
 - Telephone Number 250
 - Unique User IDs 252
 - User Status 251
- User Profile Display Class 158
- User Profile Display Options 159
- User Profile Dynamic Creation Default Roles 177
- User Search Filter
 - LDAP Authentication 187
 - Membership Authentication 194
- User Search Key 161
- User Search Return Attribute 161
- User Status 251
- User Status After Registration 192
- Users 37
 - Adding to a Policy 38
 - Adding to Services, Roles, and Groups 38
 - Creating 37
 - Deleting 38

V

- Valid Anonymous User List 166
- VerifyArchive command line tool 145
 - Syntax 145
- View Menu Entries 160

W

Warn User After N Failure 182

Web Server

Documentation 20