



Sun Java™ System

Identity Server Administration Guide

2004Q2

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

Part No: 817-5709-10

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

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

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

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

The OPEN LOOK and Sun(TM) Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

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

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

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

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

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

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

Contents

Audience for This Guide	19
Identity Server 2004Q2 Documentation Set	20
Identity Server 2004Q2 Core Documentation	20
Identity Server Policy Agent Documentation	21
Your Feedback on the Documentation	22
Documentation Conventions Used in This Guide	22
Typographic Conventions	22
Terminology	22
Related Information	24
Related Third-Party Web Site References	24
Part I Identity Server Configuration	25
Chapter 1 Identity Server 2004Q2 Configuration Scripts	27
Identity Server 2004Q2 Installation Overview	28
Identity Server amconfig Script Operations	29
Identity Server Sample Silent Mode Input File	30
Deployment Mode Variable	30
Identity Server Configuration Variables	31
Web Container Configuration Variables	34
Sun Java System Web Server 6.1 SP2	34
Sun Java System Application Server 7.0 Update 3	35
BEA WebLogic Server 6.1 SP4 and SP5	37
BEA WebLogic Server 8.1	38
IBM WebSphere 5.1	39

Directory Server Configuration Variables	40
Identity Server amconfig Script	41
Identity Server Deployment Scenarios	42
Deploying Additional Instances of Identity Server	42
To Deploy an Additional Identity Server Instance	42
Reconfiguring an Instance of Identity Server	44
Uninstalling an Identity Server Instance	45
Uninstalling All Identity Server Instances	46
Chapter 2 Identity Server Tuning Scripts	47
The amtune Scripts	47
amtune	48
The amtune-env Configuration File Parameters	49
amtune Parameters	49
AMTUNE_MODE	49
AMTUNE_MODE_OS	50
AMTUNE_MODE_DS	50
AMTUNE_MODE_WEB_CONTAINER	50
AMTUNE_MODE_IDENTITY	50
AMTUNE_DEBUG_FILE_PREFIX	50
AMTUNE_PCT_MEMORY_TO_USE	50
AMTUNE_PER_THREAD_STACK_SIZE	51
AMTUNE_SESSION_MAX_SESSION_TIME_IN_MTS	51
AMTUNE_SESSION_MAX_IDLE_TIME_IN_MTS	52
AMTUNE_SESSION_MAX_CACHING_TIME_IN_MTS	52
Installation Environment Parameters	52
HOSTNAME	52
DOMAINNAME	53
IS_CONFIG_DIR	53
WEB_CONTAINER	53
CONTAINER_BASE_DIR	53
WEB_CONTAINER_INSTANCE_NAME	53
IS_INSTANCE_NAME	54
CONTAINER_INSTANCE_DIR	55
Directory Server Parameters	56
DIRMGR_UID	56
DEFALUT_ORG_PEOPLE_CONTAINER	56
Chapter 3 Configuring Identity Server in SSL Mode	57
Configuring Identity Server With a Secure Sun Java System Web Server	57
Configuring Identity Server with a Secure Sun Java System Application Server	60
Setting Up Application Server With SSL	60

Configuring Identity Server in SSL Mode	64
Configuring Identity Server to Directory Server in SSL Mode	64
Configuring Directory Server in SSL Mode	65
Connecting Identity Server to the SSL-enabled Directory Server	65

Part II Managing Identity Server Through the Console 67

Chapter 4 Identity Management	69
The Identity Server Console	69
Header Pane	70
Navigation Pane	71
Data Pane	71
Identity Management View	72
User Profile View	72
Properties Function	73
The Identity Management Interface	73
Managing Identity Server Objects	74
Organizations	74
To Add an Organization to a Policy	76
Groups	77
To Add or Remove Members to a Static Group	78
To Create a Filtered Group	79
To Add a Group to a Policy	80
Users	81
To Add a User to a Policy	82
Services	83
Roles	84
To Add a Role to a Policy	93
Customizing a Service to a Role	93
To Add a Role to a Policy	94
Policies	95
Agents	95
To Create an Agent	95
Containers	96
People Containers	97
Group Containers	98
Display Options	99
To Change the Display Options	99
Available Actions	100
To Set Available Actions for Users	100

Chapter 5 Service Configuration	101
Definition of a Service	101
Identity Server Services	102
Administration Service	102
Authentication Service	102
Anonymous	102
Certificate-based	103
Core	103
HTTP Basic	103
LDAP	103
Membership (Self-Registration)	103
NT	103
RADIUS	103
SafeWord	104
SecurID	104
Unix	104
Windows Desktop SSO	104
Authentication Configuration Service	104
Client Detection Service	105
Globalization Settings Service	105
Discovery Service	105
Logging Service	105
Naming Service	105
Password Reset Service	106
Platform Service	106
Policy Configuration Service	106
SAML Service	106
Session Service	106
SOAP Binding Service	106
User Service	107
Attribute Types	107
Dynamic Attributes	107
User Attributes	107
Organization Attributes	107
Global Attributes	108
Policy Attributes	108
Service Configuration Interface	108
Chapter 6 Current Sessions	111
The Current Sessions Interface	111
Session Management Frame	112
Session Information Window	112
Terminating a Session	113

Chapter 7 Policy Management	115
Overview	116
Policy Management Feature	116
URL Policy Agent Service	116
Policy Agents	117
The Policy Agent Process	118
Policy Types	119
Normal Policy	119
Rules	119
Subjects	119
Referral Policy	121
Rules	121
Referrals	121
Policy Definition Type Document	121
Policy Element	122
Rule Element	122
ServiceName Element	122
ResourceName Element	123
AttributeValuePair Element	123
Attribute Element	123
Value Element	123
Subjects Element	124
Subject Element	124
Referrals Element	124
Referral Element	125
Conditions Element	125
Condition Element	125
Adding a Policy Service	125
To Add a New Policy Service	126
Creating Policies	126
Creating Policies With amadmin	127
To Create Policies With the Identity Server Console	127
Creating Policies for Peer Organizations and Suborganizations	128
To Create a Policy for a Suborganization	129
Managing Policies	129
Modifying a Normal Policy	129
Modifying a Referral Policy	136
Policy Configuration Service	138
Caching Subject Evaluations	138
amldapuser Definition	138
Adding Policy Configuration Services	139
To Add the Policy Configuration Service	139
Policy-Based Resource Management	140

Limitations	140
Chapter 8 Authentication Options	143
Core Authentication	144
Adding and Enabling the Core Service	144
Anonymous Authentication	145
Adding and Enabling Anonymous Authentication	145
Logging In Using Anonymous Authentication	146
Certificate-based Authentication	146
Adding and Enabling Certificate-based Authentication	147
Adding a Server URL in Platform Server List for Certificate-based Authentication	148
Logging In Using Certificate-based Authentication	148
HTTP Basic Authentication	148
Adding and Enabling HTTP Basic Authentication	149
Logging In Using HTTP Basic Authentication	150
LDAP Directory Authentication	150
Adding and Enabling LDAP Authentication	150
Logging In Using LDAP Authentication	151
Enabling LDAP Authentication Failover	152
Multiple LDAP Configuration	152
Membership Authentication	152
Adding and Enabling Membership Authentication	152
Logging In Using Membership Authentication	153
NT Authentication	154
Installing the Samba Client	154
Adding and Enabling NT Authentication	155
Logging In Using NT Authentication	155
RADIUS Server Authentication	156
Adding and Enabling RADIUS Authentication	156
Logging In Using RADIUS Authentication	157
SafeWord Authentication	158
Adding and Enabling SafeWord Authentication	159
Logging In Using SafeWord Authentication	159
Configuring SafeWord with Sun ONE Application Server	160
SecurID Authentication	161
Adding and Enabling SecurID Authentication	162
Logging In Using SecurID Authentication	162
Unix Authentication	163
Adding and Enabling Unix Authentication	164
Logging In Using Unix Authentication	165
Windows Desktop SSO Authentication	165
Adding and Enabling Windows Desktop SSO Authentication	165
To Create a User in the Windows 2000 Domain Controller	165

To Set Up Internet Explorer	166
To Add and Configure Windows Desktop SSO Authentication	167
Logging In Using Windows Desktop SSO Authentication	168
Authentication Configuration	168
Authentication Configuration User Interface	169
Authentication Configuration for Organizations	171
Authentication Configuration for Roles	172
Authentication Configuration for Services	173
Authentication Configuration for Users	174
Auth Level Authentication	174
Module Based Authentication	175
URL Redirection	175
Authentication Service Failover	176

Chapter 9 Password Reset Service	177
Registering the Password Reset Service	177
To Register Password Reset for Users in a Different Organization	177
Configuring the Password Reset Service	178
To Configure the Service	178
Password Reset Lockout	179
Memory Lockout	179
Physical Lockout	179
Password Reset for End Users	180
Customizing Password Reset	180
Resetting Forgotten Passwords	181
Password Policies	182

Part III Command Line Reference Guide **185**

Chapter 10 The amadmin Command Line Tool	187
The amadmin Command Line Executable	187
The amadmin Syntax	188
amadmin Options	188
Using amadmin for Federation Management	191
Loading the Liberty meta compliance XML into Directory Server	191
Exporting an Entity to an XML File (Without XML Digital Signing)	192
--entityname (--e)	192
--export (-o)	192
Exporting an Entity to an XML File (With XML Digital Signing)	192
--entityname (--e)	193
--exportwithsig (-o)	193

Using amadmin for Resource Bundles	193
Add resource bundle.	193
Get resource strings.	193
Remove resource bundle.	194
Chapter 11 The amserver Command Line Tool	195
The amserver Command Line Executable	195
amserver Syntax	195
Chapter 12 The am2bak Command Line Tool	197
The am2bak Command Line Executable	197
The am2bak Syntax	197
am2bak Options	198
Backup Procedure	199
Chapter 13 The bak2am Command Line Tool	201
The bak2am Command Line Executable	201
The bak2am Syntax	201
bak2am Options	202
Chapter 14 The ampassword Command Line Tool	203
The ampassword Command Line Executable	203
The ampassword Syntax	203
ampassword Options	204
Running ampassword on SSL	204
Chapter 15 The VerifyArchive Command Line Tool	207
The VerifyArchive Command Line Executable	207
VerifyArchive Syntax	208
VerifyArchive Options	208
Chapter 16 The amsecuridd Helper	209
The amsecuridd Helper Command Line Executable	209
amsecuridd Syntax	210
amsecuridd Options	210
Running the amsecuridd helper	210
Required Libraries	211

Part IV Attribute Reference 213

Chapter 17 Administration Service Attributes	215
Global Attributes	215
Enable Federation Management	216
Enable User Management	216
Show People Containers	216
Show Containers In View Menu	217
Show Group Containers	217
Managed Group Type	217
Default Role Permissions (ACIs)	218
No Permissions	218
Organization Admin	218
Organization Help Desk Admin	218
Organization Policy Admin	218
Enable Domain Component Tree	219
Enable Administrative Groups	220
Enable Compliance User Deletion	220
Dynamic Administrative Roles ACIs	220
Container Help Desk Admin	221
Organization Help Desk Admin	221
Container Admin	221
Organization Policy Admin	221
People Container Admin	221
Group Admin	221
Top-level Admin	222
Organization Admin	222
User Profile Service Classes	222
DC Node Attribute List	222
Search Filters for Deleted Objects	223
Default People Container	223
Default Groups Container	223
Default Agents Container	223
Organization Attributes	224
Groups Default People Container	225
Groups People Container List	225
User Profile Display Class	225
End User Profile Display Class	225
Show Roles on User Profile Page	225
Show Groups on User Profile Page	226
Enable User Self Subscription to Group	226
User Profile Display Options	226
User Creation Default Roles	226

Administrative Console Tabs	227
Maximum Results Returned From Search	227
Timeout For Search	227
JSP Directory Name	227
Online Help Documents	227
Required Services	228
User Search Key	228
User Search Return Attribute	228
User Creation Notification List	229
User Deletion Notification List	229
User Modification Notification List	230
Maximum Entries Displayed per Page	230
Event Listener Classes	230
Pre and Post Processing Classes	231
Enable External Attributes Fetch	231
UserID and Password Validation Plugin Class	231
Chapter 18 Anonymous Authentication Attributes	233
Valid Anonymous User List	233
Default Anonymous User Name	234
Enable Case Sensitive User IDs	234
Authentication Level	234
Chapter 19 Certificate Authentication Attributes	237
Match Certificate in LDAP	238
Subject DN Attribute Used to Search LDAP for Certificates	238
Match Certificate to CRL	238
Issuer DN Attribute Used to Search LDAP for CRLs	239
HTTP Parameters for CRL Update	239
Enable OCSP Validation	239
LDAP Server Where Certificates Are Stored	240
LDAP Start Search DN	240
LDAP Server Principal User	240
LDAP Server Principal Password	240
LDAP Attribute for Profile ID	241
Use SSL for LDAP Access	241
Certificate Field Used to Access User Profile	241
Other Certificate Field Used to Access User Profile	241
Trusted Remote Hosts	242
SSL Port Number	242
Authentication Level	242

Chapter 20 Core Authentication Attributes	243
Global Attributes	243
Pluggable Authentication Module Classes	244
Supported Authentication Modules for Clients	244
LDAP Connection Pool Size	244
Default LDAP Connection Pool Size	244
Organization Attributes	245
Organization Authentication Modules	246
User Profile	246
Administrator Authentication Configuration	247
User Profile Dynamic Creation Default Roles	247
Enable Persistent Cookie Mode	247
Persistent Cookie Maximum Time	248
People Container For All Users	248
Alias Search Attribute Name	248
User Naming Attribute	249
Default Authentication Locale	249
Organization Authentication Configuration	250
Enable Login Failure Lockout Mode	251
Login Failure Lockout Count	251
Login Failure Lockout Interval	251
Email Address to Send Lockout Notification	251
Warn User After N Failure	251
Login Failure Lockout Duration	252
Lockout Attribute Name	252
Lockout Attribute Value	252
Default Success Login URL	252
Default Failure Login URL	253
Authentication PostProcessing Class	253
Enable Generate UserID Mode	253
Pluggable User Name Generator Class	253
Default Authentication Level	254
Chapter 21 HTTP Basic Authentication Attributes	255
Authentication Level	255
Chapter 22 LDAP Authentication Attributes	257
Primary LDAP Server	258
Secondary LDAP Server	258
DN to Start User Search	259
DN for Root User Bind	259
Password for Root User Bind	259
Password For Root User Bind (Confirm)	260

LDAP Attribute Used to Retrieve User Profile	260
LDAP Attributes Used to Search for a User to be Authenticated	260
User Search Filter	260
Search Scope	260
Enable SSL Access to LDAP Server	261
Return User DN To Authenticate	261
LDAP Server Check Interval	261
User Creation Attributes List	261
Authentication Level	262
Chapter 23 Membership Authentication Attributes	263
Minimum Password Length	264
Default User Roles	264
User Status After Registration	264
Primary LDAP Server	264
Secondary LDAP Server	265
DN to Start User Search	265
DN for Root User Bind	266
Password for Root User Bind	266
Password for Root User Bind (Confirm)	266
LDAP Attribute Used to Retrieve User Profile	266
LDAP Attributes Used to Search for a User to be Authenticated	266
User Search Filter	267
Search Scope	267
Enable SSL Access to LDAP Server	267
Return User DN To Authenticate	267
Authentication Level	268
Chapter 24 NT Authentication Attributes	269
NT Authentication Domain	270
NT Authentication Host	270
Authentication Level	270
Chapter 25 RADIUS Authentication Attributes	271
RADIUS Server 1	271
RADIUS Server 2	272
RADIUS Shared Secret	272
RADIUS Shared Secret (Confirm)	272
RADIUS Server's Port	272
Timeout	272
Authentication Level	272

Chapter 26 SafeWord Authentication Attributes	275
SafeWord Server	275
SafeWord Server Verification Files Directory	275
SafeWord Logging Level	276
SafeWord Log File	276
Authentication Level	276
Chapter 27 SecurID Authentication Attributes	277
SecurID ACE/Server Configuration Path	277
SecurID Helper Configuration Port	278
SecurID Helper Authentication Port	278
Authentication Level	278
Chapter 28 Unix Authentication Attributes	279
Global Attributes	279
Unix Helper Configuration Port	280
Unix Helper Authentication Port	280
Unix Helper Timeout	280
Unix Helper Threads	280
Organization Attribute	280
Authentication Level	280
Chapter 29 Windows Desktop SSO Authentication Attributes	283
Service Principal	283
Keytab Filename	284
Kerberos Realm	284
Kerberos Server Name	284
Return Principal With Domain Name	284
Authentication Level	284
Chapter 30 Authentication Configuration Service Attributes	287
Authentication Configuration	287
Login Success URL	288
Login Failure URL	289
Authentication Post Processing Class	289
Conflict Resolution Level	289
Chapter 31 Client Detection Service Attributes	291
Client Types	291
Client Manager	292
Default Client Type	294

Client Detection Class	294
Enable Client Detection	294
Chapter 32 Globalization Setting Service Attributes	295
Charsets Supported By Each Locale	295
Charset Aliases	295
Auto Generated Common Name Format	296
Chapter 33 Logging Service Attributes	297
Maximum Log Size	298
Number of History Files	298
Log File Location	298
Logging Type	299
Database User Name	299
Database User Password	299
Database User Password (Confirm)	299
Database Driver Name	299
Configurable Log Fields	299
Log Verification Frequency	300
Log Signature Time	300
Enable Secure Logging	300
Maximum Number of Records	300
Number Of Files Per Archive	300
Buffer Size	301
Buffer Time	301
Enable Time Buffering	301
Chapter 34 Naming Service Attributes	303
Profile Service URL	304
Session Service URL	304
Logging Service URL	304
Policy Service URL	304
Auth Service URL	304
SAML Web Profile/Artifact Service URL	305
SAML SOAP Service URL	305
SAML Web Profile/POST Service URL	305
SAML Assertion Manager Service URL	305
Federation Assertion Manager Service URL	306
Identity SDK Service URL	306
Chapter 35 Password Reset Service Attributes	307
User Validation	308

Secret Question	308
Search Filter	308
Base DN	308
Bind DN	308
Bind Password	309
Password Reset Option	309
Password Change Notification Option	309
Enable Password Reset	309
Enable Personal Question	309
Maximum Number of Questions	309
Force Change Password on Next Login	310
Enable Password Reset Failure Lockout	310
Password Reset Failure Lockout Count	310
Password Reset Failure Lockout Interval	310
Email Address to Send Lockout Notification	310
Warn User After N Failure	311
Password Reset Failure Lockout Duration	311
Password Reset Lockout Attribute Name	311
Password Reset Lockout Attribute Value	311
Chapter 36 Platform Service Attributes	313
Server List	313
Platform Locale	314
Cookie Domains	314
Login Service URL	314
Logout Service URL	315
Available Locales	315
Client Char Sets	315
Chapter 37 Policy Configuration Service Attributes	317
Global Attributes	317
Resource Comparator	318
Continue Evaluation On Deny Decision	318
Organization Attributes	318
LDAP Server and Port	320
LDAP Base DN	321
LDAP Users Base DN	321
Identity Server Roles Base DN	321
LDAP Bind DN	321
LDAP Bind Password	321
LDAP Bind Password (Confirm)	321
LDAP Organization Search Filter	321

LDAP Organization Search Scope	322
LDAP Groups Search Filter	322
LDAP Groups Search Scope	322
LDAP Users Search Filter	322
LDAP Users Search Scope	322
LDAP Roles Search Filter	323
LDAP Roles Search Scope	323
Identity Server Roles Search Scope	323
LDAP Organization Search Attribute	323
LDAP Groups Search Attribute	323
LDAP Users Search Attribute	324
LDAP Roles Search Attribute	324
Maximum Results Returned From Search	324
Timeout For Search	324
Enable LDAP SSL	324
LDAP Connection Pool Minimal Size	324
LDAP Connection Pool Maximum Size	325
Selected Policy Subjects	325
Selected Policy Conditions	325
Selected Policy Referrals	325
Subjects Result Time To Live	325
User Alias Enabled	326
Chapter 38 SAML Service Attributes	327
Site ID And Site Issuer Name	328
Sign SAML Request	328
Sign SAML Response	328
Sign Assertion	328
SAML Artifact Name	328
Target Specifier	329
Artifact Timeout	329
Assertion Skew Factor For notBefore Time	329
Assertion Timeout	329
Trusted Partner Sites	329
POST To Target URLs	333
Chapter 39 Session Service Attributes	335
Global Attributes	335
Maximum Number of Search Results	335
Timeout For Search (Seconds)	336
Dynamic Attributes	336
Max Session Time (Minutes)	336

Max Idle Time (Minutes)	336
Max Caching Time (Minutes)	337
Chapter 40 SOAP Binding Service Attributes	339
Request Handler List	339
Web Service Authenticator	340
Supported Authentication Mechanisms	340
Chapter 41 User Attributes	341
User Service Attributes	341
User Preferred Language	342
User Preferred Timezone	342
Inherited Locale	342
Administrator DN Starting View	342
Default User Status	342
User Profile Attributes	343
First Name	343
Last Name	343
Full Name	343
Password	343
Password (Confirm)	344
Email Address	344
Employee Number	344
Telephone Number	344
Home Address	344
User Status	344
Account Expiration Date	345
User Authentication Configuration	345
User Alias List	345
Preferred Locale	345
Success URL	346
Failure URL	346
Unique User IDs	346
Appendix A Error Codes	349
Identity Server Console Errors	349
Authentication Error Codes	350
Policy Error Codes	354
amadmin Error Codes	355
Glossary	361

About This Guide

The *Sun Java™ System Identity Server 2004Q2 Administration Guide* offers information on how manage Sun Java™ System Identity Server 2004Q2 (formerly Sun™ ONE Identity Server) through the User and Command Line Interface.

This preface contains the following sections:

- [Audience for This Guide](#)
- [Identity Server 2004Q2 Documentation Set](#)
- [Documentation Conventions Used in This Guide](#)
- [Related Information](#)
- [Related Third-Party Web Site References](#)

Audience for This Guide

This *Administration Guide* is intended for use by IT administrators and software developers who implement an integrated identity management and web access platform using Sun Java System servers and software.

Readers of this guide should be familiar with the following concepts and technologies:

- Sun Java System Directory Server
- Lightweight Directory Access Protocol (LDAP) concepts
- Java™ technology
- JavaServer Pages™ (JSP) technology
- HyperText Transfer Protocol (HTTP)

- HyperText Markup Language (HTML)
- eXtensible Markup Language (XML)

Identity Server 2004Q2 Documentation Set

The Identity Server 2004Q2 documentation includes two sets:

- [Identity Server 2004Q2 Core Documentation](#)
- [Identity Server Policy Agent Documentation](#)

Identity Server 2004Q2 Core Documentation

The Identity Server 2004Q2 documentation set contains the following titles:

- *Technical Overview* (<http://docs.sun.com/doc/817-5706>) provides a high-level overview of how Identity Server components work together to consolidate identity management and to protect enterprise assets and web-based applications. It also explains basic Identity Server concepts and terminology.
- *Migration Guide* (<http://docs.sun.com/doc/817-5708>) provides details on how to migrate existing data and Sun Java System product deployments to the latest version of Identity Server. (For instructions about installing Identity Server and other products, see the *Sun Java Enterprise System 2004Q2 Installation Guide* (<http://docs.sun.com/doc/817-5760>).
- *Administration Guide* (<http://docs.sun.com/doc/817-5709>) describes how to use the Identity Server console as well as manage user and service data via the command line.
- *Deployment Planning Guide* (<http://docs.sun.com/doc/817-5707>) provides information on planning an Identity Server deployment within an existing information technology infrastructure.
- *Developer's Guide* (<http://docs.sun.com/doc/817-5710>) offers information on how to customize Identity Server and integrate its functionality into an organization's current technical infrastructure. It also contains details about the programmatic aspects of the product and its API.
- *Developer's Reference* (<http://docs.sun.com/doc/817-5711>) provides summaries of data types, structures, and functions that make up the public Identity Server C APIs.

- *Federation Management Guide* (<http://docs.sun.com/doc/817-6362>) provides information about Federation Management, which is based on the Liberty Alliance Project.
- The *Release Notes* (<http://docs.sun.com/doc/817-5712>) will be available online after the product is released. They gather an assortment of last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Updates to the *Release Notes* and links to modifications of the core documentation can be found on the Identity Server page at the Sun Java System 2004Q2 documentation web site (<http://docs.sun.com/prod/entsys.04q2>). Updated documents will be marked with a revision date.

Identity Server Policy Agent Documentation

Policy agents for Identity Server documents are available on this Web site:

http://docs.sun.com/coll/S1_IdServPolicyAgent_21

Policy agents for Identity Server are available on a different schedule than the server product itself. Therefore, the documentation set for the policy agents is available outside the core set of Identity Server documentation. The following titles are included in the set:

- *Web Policy Agents Guide* documents how to install and configure an Identity Server policy agent on various web and proxy servers. It also includes troubleshooting and information specific to each agent.
- *J2EE Policy Agents Guide* documents how to install and configure an Identity Server policy agent that can protect a variety of hosted J2EE applications. It also includes troubleshooting and information specific to each agent.
- The *Release Notes* will be available online after the set of agents is released. There is generally one *Release Notes* file for each agent type release. The *Release Notes* gather an assortment of last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Updates to the *Release Notes* and modifications to the policy agent documentation can be found on the Policy Agents page at the Sun Java System documentation web site. Updated documents will be marked with a revision date.

Your Feedback on the Documentation

Sun Microsystems and the Identity Server technical writers are interested in improving this documentation and welcomes your comments and suggestions. Use the following web-based form to provide feedback to us:

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

Please provide the full document title and part number in the appropriate fields. The part number can be found on the title page of the book or at the top of the document, and is usually a seven or nine digit number. For example, the part number of the Administration Guide is 817-5709-10.

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 words used in the literal sense.
- Monospace font is used for sample code and code listings, API and language elements (such as function names and class names), filenames, pathnames, directory names, HTML tags, and any text that must be typed on the screen.
- *Italic serif font* is used within code and code fragments to indicate variable placeholders. For example, the following command uses *filename* as a variable placeholder for an argument to the `gunzip` command:

```
gunzip -d filename.tar.gz
```

Terminology

The following terms are used in the Identity Server documentation set:

- *Identity Server* refers to Identity Server and any installed instances of the Identity Server software.

- *Policy and Management services* refers to the collective set of Identity Server components and software that are installed and running on a dedicated deployment container such as a web server.
- *Directory Server* refers to an installed instance of Sun Java System Directory Server.
- *Application Server* refers to an installed instance of Sun Java System Application Server (also known as Sun ONE Application Server.)
- *Web Server* refers to an installed instance of Sun Java System Web Server (also known as Sun ONE Web Server).
- *Web container that runs Identity Server* refers to the dedicated J2EE container (such as Web Server or Application Server) where the Policy and Management Services are installed.
- *IdentityServer_base* represents the base installation directory for Identity Server. The Identity Server 2004Q2 default base installation and product directory depends on your specific platform:
 - Solaris™ systems: `/opt/SUNWam`
 - Linux systems: `/opt/sun/identity`

The product directory is `/SUNWam` for Solaris systems and `/identity` for Linux systems. When you install Identity Server 2004Q2, you can specify a different directory for `/opt` on Solaris systems or `/opt/sun` on Linux systems; however, do not change the `/SUNWam` or `/identity` product directory.

For the base installation directory of the following products, refer to the documentation for the specific product.

- *DirectoryServer_base* represents the base installation directory for Sun Java System Directory Server.
- *ApplicationServer_base* is a variable place holder for the home directory for Sun Java System Application Server.
- *WebServer_base* is a variable place holder for the home directory for Sun Java System Web Server.

Related Information

Useful information can be found at the following locations:

- **Directory Server documentation:**
http://docs.sun.com/coll/DirectoryServer_04q2
- **Web Server documentation:**
http://docs.sun.com/coll/S1_websvr61_en
- **Application Server documentation**
http://docs.sun.com/coll/s1_asseu3_en
- **Web Proxy Server documentation:**
<http://docs.sun.com/prod/s1.webproxys#hic>
- **Download Center:**
<http://www.sun.com/software/download/>
- **Technical Support:**
<http://www.sun.com/service/sunone/software/index.html>
- **Professional Services:**
<http://www.sun.com/service/sunps/sunone/index.html>
- **Sun Enterprise Services, Solaris Patches, and Support:**
<http://sunsolve.sun.com/>
- **Developer Information:**
<http://developers.sun.com/prodtech/index.html>

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Identity Server Configuration

This is part one of the *Sun Java™ System Identity Server 2004Q2 Administration Guide*. It discusses configuration options that you can perform after Identity Server installation. This part contains the following chapters:

- “Identity Server 2004Q2 Configuration Scripts” on page 27
- “Identity Server Tuning Scripts” on page 47
- “Configuring Identity Server in SSL Mode” on page 57

Identity Server 2004Q2 Configuration Scripts

This chapter describes how to configure and deploy Sun Java™ System Identity Server (formerly Sun™ ONE Identity Server) using the `amconfig` script and the sample silent mode input file (`amsamplesilent`). Topics include:

- [“Identity Server 2004Q2 Installation Overview” on page 28](#)
- [“Identity Server Sample Silent Mode Input File” on page 30](#)
 - [Deployment Mode Variable](#)
 - [Identity Server Configuration Variables](#)
 - [Web Container Configuration Variables](#)
 - [Directory Server Configuration Variables](#)
- [“Identity Server amconfig Script” on page 41](#)
- [“Identity Server Deployment Scenarios” on page 42](#)
 - [Deploying Additional Instances of Identity Server](#)
 - [Reconfiguring an Instance of Identity Server](#)
 - [Uninstalling an Identity Server Instance](#)
 - [Uninstalling All Identity Server Instances](#)

Identity Server 2004Q2 Installation Overview

For a new installation, always install the first instance of Identity Server 2004Q2 by running the Sun Java Enterprise System installer. When you run the installer, you can select either of these configuration options for Identity Server:

- The Configure Now option allows you to configure the first instance during the installation by the choices (or default values) that you select on the Identity Server installation panels.
- The Configure Later option installs the Identity Server 2004Q2 components, and then after installation, you must configure them, as described in [Reconfiguring an Instance of Identity Server](#).

For information about the installer, refer to the *Sun Java Enterprise System 2004Q2 Installation Guide* (<http://docs.sun.com/doc/817-5760>).

The Java Enterprise System installer installs the Identity Server 2004Q2 `amconfig` script and sample silent mode input file (`amsamplesilent`) in the *IdentityServer_base/SUNWam/bin* directory on Solaris systems or the *IdentityServer_base/identity/bin* directory on Linux systems.

IdentityServer_base represents the Identity Server base installation directory. On Solaris systems, the default base installation directory is `/opt`, and on Linux systems, it is `/opt/sun`. However, you can specify another directory, if you prefer, when you run the installer.

The `amconfig` script is a top-level script that calls other scripts as needed to perform the requested operation. For more information, see the [Identity Server amconfig Script](#).

The sample silent mode input file (`amsamplesilent`) is an example of an input file that you must specify when you run the `amconfig` script in silent mode.

This sample silent mode input file is an ASCII text file that contains Identity Server configuration variables. Before you run the `amconfig` script, copy (and rename, if you wish) the `amsamplesilent` file, and then edit the variables in the file. The configuration variables are in the following format:

```
variable-name=value
```

For example:

```
DEPLOY_LEVEL=1  
NEW_INSTANCE=true  
SERVER_HOST=ishost.example.com
```

For a list of the variables you can set in a silent mode input file, see the [Identity Server Sample Silent Mode Input File](#).

CAUTION The format of the silent mode input file used when you run the `amconfig` script in silent mode does not follow the same format or necessarily use the same variable names as a Java Enterprise System silent installation state file. This file contains sensitive data, such as the admin password. Make sure to protect or delete this file as appropriate.

Identity Server `amconfig` Script Operations

After you install first instance of Identity Server using the Sun Java Enterprise System installer, you can run the `amconfig` script to perform the following operations, depending on the values of the variables in the silent mode input file:

- Deploy and configure additional instances of Identity Server on the same host system. For example, after you configure an additional instance of a web container, you can then deploy and configure a new Identity Server instance for that web container instance.
- Reconfigure both the first instance and any additional instances of Identity Server.
- Deploy and configure the Identity Server SDK, which enables support for these products:
 - BEA WebLogic Server 6.1 SP4 and SP5
 - BEA WebLogic Server 8.1 SP1
 - IBM WebSphere 5.1
- Deploy and configure specific Identity Server components such as the console or Federation Management module.
- Uninstall instances and components of Identity Server that you deployed using the `amconfig` script.

Identity Server Sample Silent Mode Input File

After you run the Java Enterprise System installer, the Identity Server sample silent mode input file (`amsamplesilent`) is available in the *IdentityServer_base*/SUNWam/bin directory on Solaris systems or the *IdentityServer_base*/identity/bin directory on Linux systems.

To set configuration variables, first copy and rename the `amsamplesilent` file. Then set the variables in the copy for the operation you want to perform.

This sample silent mode input file contains the following configuration variables:

- [Deployment Mode Variable](#)
- [Identity Server Configuration Variables](#)
- [Web Container Configuration Variables](#)
- [Directory Server Configuration Variables](#)

Deployment Mode Variable

[Table 1-1](#) describes the values for the required `DEPLOY_LEVEL` variable. This variable determines the operation you want the `amconfig` script to perform.

Table 1-1 Identity Server `DEPLOY_LEVEL` Variable

Operation	<code>DEPLOY_LEVEL</code> Variable Value and Description
Install	1 = Full Identity Server installation for a new instance (default)
	2 = Install Identity Server console only
	3 = Install Identity Server SDK only
	4 = Install SDK only and configure the container
	5 = Install Federation Management module only
	6 = Install server only
Uninstall (unconfigure)	11 = Full uninstall
	12 = Uninstall console only
	13 = Uninstall SDK only
	14 = Uninstall SDK only and unconfigure the container
	15 = Uninstall Federation Management module
	16 = Uninstall server only

Table 1-1 Identity Server DEPLOY_LEVEL Variable (*Continued*)

Operation	DEPLOY_LEVEL Variable Value and Description
Re-install	21 = Full re-install
(also referred to as re-deploy or re-configure)	32 = Re-install console only
	31 = Re-install SDK only
	33= Re-install SDK console only
	35 = Re-install Federation Management module
	26 = Re-install server only

Identity Server Configuration Variables

[Table 1-2](#) describes the Identity Server configuration variables.

Table 1-2 Identity Server Configuration Variables

Variable	Description
BASEDIR	Base installation directory for Identity Server packages. Default: PLATFORM_DEFAULT For Solaris systems, PLATFORM_DEFAULT is /opt For Linux systems, PLATFORM_DEFAULT is /opt/sun
SERVER_HOST	Fully qualified host name of the system where Identity Server is running (or will be installed). For a remote SDK installation, set this variable to the host where Identity Server is (or will be) installed and not the remote client host.
SERVER_PORT	Identity Server port number. Default: 58080 For a remote SDK installation, set this variable to the port on the host where Identity Server is (or will be) installed and not the remote client host.
SERVER_PROTOCOL	Server protocol: http or https. Default: http For a remote SDK installation, set this variable to the protocol on the host where Identity Server is (or will be) installed and not the remote client host.
CONSOLE_HOST	Fully qualified host name of the server where the console is installed. Default: Value provided for the Identity Server host (SERVER_HOST variable)
CONSOLE_PORT	Port of the web container where the console is installed and listens for connections. Default: Value provided for the Identity Server port (SERVER_PORT variable)
CONSOLE_PROTOCOL	Protocol of the web container where the console is installed. Default: Server protocol (SERVER_PROTOCOL variable)

Table 1-2 Identity Server Configuration Variables (*Continued*)

Variable	Description
CONSOLE_REMOTE	Set to true if the console is remote from the Identity Server services. Otherwise, set to false. Default: false
DS_HOST	Fully qualified host name of Directory Server.
DS_PORT	Directory Server port. Default: 389.
DS_DIRMGRDN	Directory manager DN: the user who has unrestricted access to Directory Server. Default: "cn=Directory Manager"
DS_DIRMGRPASSWD	Password for the directory manager (DS_DIRMGRDN variable). See the note about special characters in the description of ADMINPASSWD .
ROOT_SUFFIX	Initial or root suffix of the directory. You must make sure that this value exists in the Directory Server you are using. See the note about special characters in the description of ADMINPASSWD .
ADMINPASSWD	Password for the administrator (<code>amadmin</code>). Must be different from the password for <code>amldapuser</code> . Note: If the password contains special characters such as a slash (/) or backslash (\), the special character must be enclosed by single quotes ('). For example: <code>ADMINPASSWD='\\\\\\#####/'</code> However, the password cannot have a single quote as one of the actual password characters.
AMLDAPUSERPASSWD	Password for <code>amldapuser</code> . Must be different from the password for <code>amadmin</code> . See the note about special characters in the description of ADMINPASSWD .
CONSOLE_DEPLOY_URI	URI prefix for accessing the HTML pages, classes and JAR files associated with the Identity Server Administration Console subcomponent. Default: <code>/amconsole</code>
SERVER_DEPLOY_URI	URI prefix for accessing the HTML pages, classes, and JAR files associated with the Identity Management and Policy Services Core subcomponent. Default: <code>/amserver</code>
PASSWORD_DEPLOY_URI	URI that determines the mapping that the web container running Identity Server will use between a string you specify and a corresponding deployed application. Default: <code>/ampassword</code>
COMMON_DEPLOY_URI	URI prefix for accessing the common domain services on the web container. Default: <code>/amcommon</code>

Table 1-2 Identity Server Configuration Variables (*Continued*)

Variable	Description
COOKIE_DOMAIN	Names of the trusted DNS domains that Identity Server returns to a browser when it grants a session ID to a user. At least one value should be present. In general, the format is the server's domain name preceded with a period. Example: <code>.example.com</code>
JAVA_HOME	Path to the Java 2 home directory. Default: <code>/usr/jdk/entsys-j2se</code>
AM_ENC_PWD	Password encryption key: String that Identity Server uses to encrypt user passwords. Default: <code>none</code> Important: If you are deploying multiple instances of Identity Server or the remote SDK, all instances must use the same password encryption key. When you deploy an additional instance, copy the value from the <code>am.encrypted.pwd</code> property in the <code>AMConfig.properties</code> file for the first instance.
PLATFORM_LOCALE	Locale of the platform. Default: <code>en_US</code> (US English)
NEW_OWNER	New owner for the Identity Server files after installation. Default: <code>root</code>
NEW_GROUP	New group for the Identity Server files after installation. Default: <code>other</code> For a Linux installation, set <code>NEW_GROUP</code> to <code>root</code> .
XML_ENCODING	XML encoding. Default: <code>ISO-8859-1</code>
NEW_INSTANCE	Specifies whether the configuration script should deploy Identity Server to a new user-created web container instance: <ul style="list-style-type: none"> <code>true</code> = To deploy Identity Server to a new user-created web container instance other than the instance created by the Java Enterprise System installer. <code>false</code> = To re-configure an instance. Default: <code>false</code>

Web Container Configuration Variables

To specify the web container for Identity Server, set the `WEB_CONTAINER` variable in the silent mode input file, as described in [Table 1-3](#).

Table 1-3 Identity Server `WEB_CONTAINER` Variable

Value	Web Container
WS6 (default)	Sun Java System Web Server 6.1 SP2
AS7	Sun Java System Application Server 7.0 Update 3
WL6	BEA WebLogic Server 6.1 SP4 and SP5 (with the Identity Server SDK only)
WL8	BEA WebLogic Server 8.1 (with the Identity Server SDK only)
WAS5	IBM WebSphere 5.1 (with the Identity Server SDK only)

Sun Java System Web Server 6.1 SP2

[Table 1-4](#) describes the configuration variables for Web Server 6.1 SP2 in the silent mode input file.

Table 1-4 Web Server 6.1 SP2 Configuration Variables

Variable	Description
WS61_INSTANCE	Name of the Web Server instance on which Identity Server will be deployed or un-deployed. Default: <code>https-<i>web-server-instance-name</i></code> where <i>web-server-instance-name</i> is the Identity Server host (<code>SERVER_HOST</code> variable)
WS61_HOME	Web Server base installation directory. Default: <code>/opt/SUNWwbsvr</code>
WS61_PROTOCOL	Protocol used by the Web Server instance set by the <code>WS61_INSTANCE</code> variable where Identity Server will be deployed: http or https. Default: Identity Server protocol (<code>SERVER_PROTOCOL</code> variable)
WS61_HOST	Fully qualified host name for the Web Server instance (<code>WS61_INSTANCE</code> variable). Default: Identity Server host instance (<code>SERVER_HOST</code> variable)
WS61_PORT	Port on which Web Server listens for connections. Default: Identity Server port number (<code>SERVER_PORT</code> variable)

Table 1-4 Web Server 6.1 SP2 Configuration Variables (*Continued*)

Variable	Description
WS61_ADMINPORT	Port on which the Web Server Administration Server listens for connections. Default: 58888
WS61_ADMIN	User ID of the Web Server administrator. Default: "admin"
WS61_IS_SECURE	Specifies whether a secure port is enabled: <ul style="list-style-type: none"> true: Secure port is enabled (HTTPS protocol). If the container is SSL enabled, the configuration script will use the SSL_PASSWORD variable to start the server without requiring user intervention. false: Secure port is not enabled (HTTP protocol). Default: false (not enabled)

Sun Java System Application Server 7.0 Update 3

[Table 1-5](#) describes the configuration variables for Application Server 7.0 Update 3 in the silent mode input file.

Table 1-5 Application Server 7.0 Update 3 Configuration Variables

Variable	Description
AS70_HOME	Path to the directory where Application Server 7.0 is installed. Default: /opt/SUNWappserver7
AS70_PROTOCOL	Protocol used by the Application Server instance: http or https. Default: Identity Server protocol (SERVER_PROTOCOL variable)
AS70_HOST	Fully qualified domain name (FQDN) on which the Application Server instance listens for connections. Default: Identity Server host (SERVER_HOST variable)
AS70_PORT	Port on which Application Server instance listens for connections. Default: Identity Server port number (SERVER_PORT variable)
AS70_ADMINPORT	Port on which the Application Server administration server listens for connections. Default: 4848
AS70_ADMIN	Name of the user who administers the Application Server administration server for the domain into which Application Server is being displayed. Default: admin

Table 1-5 Application Server 7.0 Update 3 Configuration Variables (*Continued*)

Variable	Description
AS70_ADMINPASSWD	<p>Password for the Application Server administrator for the domain into which Application Server is being displayed.</p> <p>See the note about special characters in the description of ADMINPASSWD.</p>
AS70_INSTANCE	<p>Name of the Application Server instance that will run Identity Server.</p> <p>Default: <code>server1</code></p>
AS70_DOMAIN	<p>Path to the Application Server directory for the domain to which you want to deploy this Identity Server instance.</p> <p>Default: <code>domain1</code></p>
AS70_INSTANCE_DIR	<p>Path to the directory where Application Server stores files for the instance.</p> <p>Default: <code>/var/opt/SUNWappserver7/domains/domain1/server1</code></p>
AS70_DOCS_DIR	<p>Directory where Application Server stores content documents.</p> <p>Default: <code>/var/opt/SUNWappserver7/domains/domain1/server1/docroot</code></p>
AS70_IS_SECURE	<p>Specifies whether a secure port is enabled:</p> <ul style="list-style-type: none"> • <code>true</code>: Secure port is enabled (HTTPS protocol). If the container is SSL enabled, the configuration script will use the SSL_PASSWORD variable to start the server without requiring user intervention. • <code>false</code>: Secure port is not enabled (HTTP protocol). <p>Default: <code>false</code> (not enabled)</p> <p>During installation, if the Application Server admin port is SSL enabled, configuration will fail. Do not use the admin server in <code>https</code> mode.</p>

BEA WebLogic Server 6.1 SP4 and SP5

Table 1-6 describes the configuration variables for BEA WebLogic Server 6.1 in the silent mode input file.

Table 1-6 BEA WebLogic Server 6.1 SP4 and SP5 Configuration Variables

Variable	Description
WL61_HOME	WebLogic home directory. Default: <code>/export/boa61a</code>
WL61_PROJECT_DIR	WebLogic project directory. Default: <code>user_projects</code>
WL61_DOMAIN	WebLogic domain name. Default: <code>mydomain</code>
WL61_SERVER	WebLogic server name. Default: <code>myserver</code>
WL61_INSTANCE	WebLogic instance name. Default: <code>WS61_HOME/wlserver6.1</code>
WL61_PROTOCOL	WebLogic protocol. Default: <code>http</code>
WL61_HOST	WebLogic host name.
WL61_PORT	WebLogic port. Default: <code>7001</code>
WL61_SSLPORT	WebLogic SSL port. Default: <code>7002</code>
WL61_ADMIN	WebLogic administrator. Default: <code>"system"</code>
WL61_PASSWORD	WebLogic administrator password. See the note about special characters in the description of ADMINPASSWD .
WL61_JDK_HOME	WebLogic JDK home directory. Default: <code>WS61_HOME/jdk131</code>

BEA WebLogic Server 8.1

[Table 1-7](#) describes the configuration variables for BEA WebLogic Server 8.1 in the silent mode input file.

Table 1-7 BEA WebLogic Server 8.1 Configuration Variables

Variable	Description
WL8_HOME	WebLogic home directory. Default: <code>/export/boa8</code>
WL8_PROJECT_DIR	WebLogic project directory. Default: <code>projects</code>
WL8_DOMAIN	WebLogic domain name. Default: <code>mydomain</code>
WL8_SERVER	WebLogic server name. Default: <code>myserver</code>
WL8_INSTANCE	WebLogic instance name. Default: <code>/export/boa8/weblogic81</code>
WL8_PROTOCOL	WebLogic protocol. Default: <code>http</code>
WL8_HOST	WebLogic host name. Default: <code>none</code>
WL8_PORT	WebLogic port. Default: <code>7001</code>
WL8_SSLPORT	WebLogic SSL port. Default: <code>7002</code>
WL8_ADMIN	WebLogic administrator. Default: <code>"system"</code>
WL8_PASSWORD	WebLogic administrator password. See the note about special characters in the description of ADMINPASSWD .
WL8_JDK_HOME	WebLogic JDK home directory. Default: <code>WL8_HOME/jdk141_03</code>
WL8_IS_SECURE	Specifies whether a secure port is enabled: <ul style="list-style-type: none"> • <code>true</code>: Secure port is enabled (HTTPS protocol). • <code>false</code>: Secure port is not enabled (HTTP protocol). Default: <code>false</code> (not enabled)

IBM WebSphere 5.1

Table 1-8 describes the configuration variables for IBM WebSphere Server 5.1 in the silent mode input file.

Table 1-8 IBM WebSphere 5.1 Configuration Variables

Variable	Description
WAS51_HOME	WebSphere home directory. Default: /opt/WebSphere/AppServer
WAS51_JDK_HOME	WebSphere JDK home directory. Default: /opt/WebSphere/AppServer/java
WAS51_CELL	WebSphere cell. Default: sample
WAS51_DOMAIN	WebSphere domain name. Default: mydomain
WAS51_NODE	WebSphere node name. Default: host name of the server where WebSphere is installed. Default: sample
WAS51_INSTANCE	WebSphere instance name. Default: server1
WAS51_PROTOCOL	WebSphere protocol. Default: http
WAS51_HOST	WebSphere host name. Default: sample
WAS51_PORT	WebSphere port. Default: 9080
WAS51_SSLPORT	WebSphere SSL port. Default: 9081
WAS51_ADMIN	WebSphere administrator. Default: "admin"
WAS51_ADMINPORT	WebSphere administrator port. Default: 9090
WAS51_IS_SECURE	Specifies whether a secure port is enabled: <ul style="list-style-type: none"> • true: Secure port is enabled (HTTPS protocol). • false: Secure port is not enabled (HTTP protocol). Default: false (not enabled)

Directory Server Configuration Variables

Identity Server 2004Q2 supports Sun ONE Directory Server 5.1 and Sun Java System Directory Server 5 2004Q2. [Table 1-9](#) describes the Directory Server configuration variables in the silent mode input file.

Table 1-9 Directory Server Configuration Variables

Variable	Description
DIRECTORY_MODE	<p>Directory Server modes:</p> <p>1 = Use for a new installation of a Directory Information Tree (DIT).</p> <p>2 = Use for an existing DIT. The naming attributes and object classes are the same, so the configuration scripts load the <code>installExisting.ldif</code> and <code>umsExisting.ldif</code> files.</p> <p>The configuration scripts also update the LDIF and properties files with the actual values entered during configuration (for example, <code>BASE_DIR</code>, <code>SERVER_HOST</code>, and <code>ROOT_SUFFIX</code>).</p> <p>This update is also referred to as “tag swapping,” because the configuration scripts replace the placeholder tags in the files with the actual configuration values.</p> <p>3 = Use for an existing DIT when you want to do a manual load. The naming attributes and object classes are different, so the configuration scripts do not load the <code>installExisting.ldif</code> and <code>umsExisting.ldif</code> files. The scripts perform tag swapping (described for mode 2).</p> <p>You should inspect and modify (if needed) the LDIF files and then manually load the LDIF files and services.</p> <p>4 = Use for an existing multi-server installation. The configuration scripts do not load the LDIF files and services, because the operation is against an existing Identity Server installation. The scripts perform tag swapping only (described for mode 2) and adds a server entry in the platform list.</p> <p>5 = Use for an existing upgrade. The scripts perform tag swapping only (described for mode 2).</p> <p>Default: 1</p>
USER_NAMING_ATTR	User naming attribute: Unique identifier for the user or resource within its relative name space. Default: <code>uid</code>
ORG_NAMING_ATTR	Naming attribute of the user's company or organization. Default: <code>o</code>
ORG_OBJECT_CLASS	Organization object class. Default: <code>sunManagedOrganization</code>
USER_OBJECT_CLASS	User object class. Default: <code>inetOrgPerson</code>
DEFAULT_ORGANIZATION	Default organization name. Default: <code>none</code>

Identity Server amconfig Script

After you run the Java Enterprise System installer, the `amconfig` script is available in the `IdentityServer_base/SUNWam/bin` directory on Solaris systems or the `IdentityServer_base/identity/bin` directory on Linux systems.

The `amconfig` script reads a silent install input file and then calls other scripts in silent mode, as needed to perform the requested operation.

To run the `amconfig` script, use this syntax:

```
amconfig [ -s input-file ]
```

where:

`-s` runs `amconfig` in silent mode.

input-file is a silent install input file that contains the configuration variables for the operation you want to perform. For more information, see [Identity Server Sample Silent Mode Input File](#).

If you are deploying a web container for WebLogic Server or WebSphere for use with the Identity Server SDK, the `amconfig` script calls other scripts to perform the configuration, but these scripts do not start (or stop) the respective web container. To start a web container instance, use the WebLogic Server or WebSphere command or procedure that applies to your specific deployment.

NOTE In the Identity Server 2004Q2 release, the following scripts are not supported:

- `amserver` with the `create` argument
- `amserver .instance`

Also, by default `amserver start` starts only the authentication `amsecuridd` and `amunixd` helpers. The `amsecuridd` helper is available only on the Solaris OS SPARC platform.

Identity Server Deployment Scenarios

After you have installed the first instance of Identity Server using the Java Enterprise System installer, you can deploy and configure additional Identity Server instances by editing the configuration variables in the silent mode input file and then running the `amconfig` script.

This section describes the following scenarios:

- [Deploying Additional Instances of Identity Server](#)
- [Reconfiguring an Instance of Identity Server](#)
- [Uninstalling an Identity Server Instance](#)
- [Uninstalling All Identity Server Instances](#)

Deploying Additional Instances of Identity Server

Before you can deploy a new instance of Identity Server, you must create and start the new web container instance using the administration tools for the web container. For information, refer to the specific web container documentation:

- For Web Server 6.1 SP2, see:
http://docs.sun.com/coll/S1_websvr61_en
- For Application Server 7.0 Update 3, see:
http://docs.sun.com/coll/s1_asseu3_en

To Deploy an Additional Identity Server Instance

1. Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 will be the web container for the new instance, log in either as superuser (root) or as the user account for the Web Server Administration Server.
2. Copy the `amsamplesilent` file to a writable directory and make that directory your current directory. For example, you might create a directory named `/newinstances`.

Tip Rename the copy of the `amsamplesilent` file to describe the new instance you want to deploy. For example, the following steps use an input file named `amnews6instance` to install a new instance for Web Server 6.1.

3. Set the following variables in the new `amnews6instance` file:

```
DEPLOY_LEVEL=1  
NEW_INSTANCE=true
```

Set other variables in the `amnews6instance` file as required for the new instance you want to create. For a description of these variables, refer to the tables in the following sections:

- o [Identity Server Configuration Variables](#)
- o [Web Container Configuration Variables](#)
- o [Directory Server Configuration Variables](#)

Important All Identity Server instances must use the same value for the password encryption key. To set the `AM_ENC_PWD` variable for this instance, copy the value from the `am.encrypted.pwd` property in the `AMConfig.properties` file for the first instance.

In case you might need to uninstall this instance later, save the `amnews6instance` file.

4. Run the `amconfig` script, specifying the new `amnews6instance` file. For example, on Solaris systems:

```
cd IdentityServer_base/SUNWam/bin/  
./amconfig -s /newinstances/amnews6instance
```

The `-s` option runs the `amconfig` script in silent mode.

The `amconfig` script calls other configuration scripts as needed, using variables in the `amnews6instance` file to deploy the new instance.

Reconfiguring an Instance of Identity Server

You can reconfigure the first instance of Identity Server that was installed using the Java Enterprise System installer and any additional Identity Server instances that you deployed by running the `amconfig` script.

For example, you might want to reconfigure an instance to change the Identity Server owner and group.

To Reconfigure an Instance of Identity Server

1. Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 is the web container, log in either as `superuser` (`root`) or as the user account for Web Server Administration Server.
2. Copy the silent install input file you used to deploy the instance to a writable directory and make that directory your current directory. For example, to reconfigure an instance for Web Server 6.1, the following steps use an input file named `amnewinstanceforWS61` in the `/reconfig` directory.
3. In the `amnewinstanceforWS61` file, set the `DEPLOY_LEVEL` variable to one of the values described for a [Re-install](#) operation. For example, set `DEPLOY_LEVEL=21` to reconfigure a full installation.
4. In the `amnewinstanceforWS61` file, set the `NEW_INSTANCE` variable to `false`:

```
NEW_INSTANCE=false
```
5. Set other variables in the `amnewinstanceforWS61` file to reconfigure the instance. For example, to change the owner and group for the instance, set the `NEW_OWNER` and `NEW_GROUP` variables to their new values.

For a description of other variables, refer to the tables in the following sections:

- [Identity Server Configuration Variables](#)
 - [Web Container Configuration Variables](#)
 - [Directory Server Configuration Variables](#)
6. Run the `amconfig` script, specifying your edited input file. For example, on Solaris systems:

```
cd IdentityServer_base/SUNWam/bin/  
./amconfig -s /reconfig/amnewinstanceforWS61
```

The `-s` option runs the script in silent mode. The `amconfig` script calls other configuration scripts as needed, using variables in the `amnewinstanceforWS61` file to reconfigure the instance.

Uninstalling an Identity Server Instance

You can uninstall an instance of Identity Server that was installed by running the `amconfig` script. You can also temporarily unconfigure an instance of Identity Server, and unless you remove the web container instance, it is still available for you to re-deploy another Identity Server instance later.

To Uninstall an Instance of Identity Server

1. Log in as an administrator, depending on the web container for the instance. For example, if Web Server 6.1 is the web container, log in either as `superuser` (`root`) or as the user account for Web Server Administration Server.
2. Copy the silent install input file you used to deploy the instance to a writable directory and make that directory your current directory. For example, to unconfigure an instance for Web Server 6.1, the following steps use an input file named `amnewinstanceforWS61` in the `/unconfigure` directory.
3. In the `amnewinstanceforWS61` file, set the `DEPLOY_LEVEL` variable to one of the values described for an **Uninstall (unconfigure)** operation. For example, set `DEPLOY_LEVEL=11` to uninstall (or unconfigure) a full installation.
4. Run the `amconfig` script, specifying your edited input file. For example, on Solaris systems:

```
cd IdentityServer_base/SUNWam/bin/  
./amconfig -s /unconfigure/aminstanceforWS61
```

The `-s` option runs the script in silent mode. The `amconfig` script reads the `amnewinstanceforWS61` file and then uninstalls the instance.

The web container instance is still available if you want to use it to re-deploy another Identity Server instance later.

Uninstalling All Identity Server Instances

This scenario completely removes all Identity Server 2004Q2 instances and packages from a system.

To Completely Remove Identity Server 2004Q2 From a System

1. Log in as or become superuser (root).
2. In the input file you used to deploy the instance, set the `DEPLOY_LEVEL` variable to one of the values described for an **Uninstall (unconfigure)** operation. For example, set `DEPLOY_LEVEL=11` to uninstall (or unconfigure) a full installation.
3. Run the `amconfig` script using the file you edited in [Step 2](#). For example on Solaris systems:

```
cd IdentityServer_base/SUNWam/bin/  
./amconfig -s /newinstances/amnews6instance
```

The `amconfig` script runs in silent mode to uninstall the instance.

Repeat these steps for any other Identity Server instances you want to uninstall, except for the first instance, which is the instance you installed using the Java Enterprise System installer.

4. To uninstall the first instance and remove all Identity Server packages from the system, run the Java Enterprise System uninstaller. For information about the uninstaller, refer to the *Sun Java Enterprise System Installation Guide*.

Identity Server Tuning Scripts

This chapter describes the `amtune` tuning scripts for Sun Java™ System Identity Server 2004Q2 and contains the following sections:

- “The `amtune` Scripts” on page 47
- “The `amtune-env` Configuration File Parameters” on page 49

NOTE For the 2004Q2 release, the `amtune` scripts are only fully functional on Solaris. The Linux and x86 versions of these scripts will function, but they are not as mature on these platforms.

The `amtune` Scripts

The `amtune` scripts allow you to tune the performance of Identity Server, as well as optimize the performance settings for various components of your Identity Server deployment.

The `amtune` scripts are non-interactive, meaning that before you run a script, you must edit the parameters in the `amtune-env` configuration file to specify the tuning you want to perform for your specific environment.

To edit tuning enhancements, modify the parameters in the `amtune-env` file and run the `amtune` script in the following format, where `admin_password` is the Identity Server Admin Client Utility password, and `dirmanager_password` is the Directory Manager (cn=Directory Manager) password:

```
amtune admin_password dirmanager_password
```

If you wish to tune specific components, you can use the component scripts provided in the `/amtune` directory. The component scripts will use the relevant parameters in the `amtune-env` file. The available component scripts are:

- `amtune-as7` - This script tunes the Sun Java System Application Server 7 web container.
- `amtune-identity` - This script tunes the installed instance of Identity Server.
- `amtune-os` - This script tunes the Solaris operating system kernel and TCP/IP parameters.
- `amtune-prepareDSTuner` - This script tunes the Directory Server instance that supports Identity Server. Tuning Directory Server requires extra levels of confirmation. Identity Server should use an existing Directory Server in non-exclusive mode. Regardless of where the Directory Server is installed (locally or remotely), Directory Server is not tuned when you run `amtune`. When you run the script, it creates a tar file named `/tmp/amtune-directory.tar`. By default, the extracted files will be placed in `/tmp` directory. You need to extract this file in the machine on which Directory Server is running on your system and then run the `amtune-directory` script.
- `amtune-ws61` - This script tunes the Sun Java System Web Server 6.1 web container.

For example, if you wish to tune the operating system, use the following format:

```
amtune-os admin_password dirmanager_password
```

The `amtune` scripts and the associated `amtune-env` file can be found in the following directories:

```
IdentityServer_base/SUNWam/bin/amtune (Solaris)
```

```
IdentityServer_base/identity/bin/amtune (Linux)
```

NOTE Throughout the rest of this chapter, only the Solaris directory information will be given. Please note that the directory structure for Linux is different. For more information, please see [“About This Guide” on page 19](#).

amtune

The `amtune` script has two generation modes; one to generate a set of tuning suggestions for an Identity Server deployment and one to implement your tuning specifications. The following modes that can be specified are defined in the `AMTUNE_MODE` parameter in the `amtune-env` file:

- **Review Mode** - When the Review mode is specified, the `amtune` script returns tuning suggestions, but does not make any changes to the deployment environment.

- **Change Mode** - When the Change mode is specified `amtune` will make all of the tuning modifications that you have defined in `amtune-env`, with the exception of Directory Server tuning.

NOTE

Use caution when using the Change mode. After running the script, the web container will need to be restarted, and the `amtune` may recommend a system restart.

Tuning is a very iterative process and can vary between different deployments. While the `amtune` utility tries to apply the best possible tuning parameters, every deployment is unique and might require further customization to suit the requirements for the deployment. It is important for the Identity Server administrator to understand, and review each of the tuning applied to the deployment.

In either mode, a list of tuning recommendations and current values are written to the `amtune` output file and displayed in the terminal window. The location of this file is based on the `AMTUNE_DEBUG_FILE_PREFIX` parameter in `amtune-env`.

The amtune-env Configuration File Parameters

The `amtune-env` configuration file contains parameters to define the tuning options for your Identity Server deployment. This section describes the `amtune-env` parameters.

amtune Parameters

The following parameters are used for component-specific tuning:

AMTUNE_MODE

This parameter defines the following modes:

- `review` - When the Review mode is specified, the `amtune` script returns tuning suggestions, but does not make any changes to the deployment environment.
- `change` - When the Change mode is specified, `amtune` will make all of the tuning modifications that you have defined in `amtune-env`, with the exception of Directory Server tuning.

AMTUNE_MODE_OS

This parameter tunes the Solaris operating system kernel and TCP/IP settings.

AMTUNE_MODE_DS

This parameter tunes the Directory Server instance that supports Identity Server. Tuning Directory Server requires extra levels of confirmation. Identity Server should use an existing Directory Server in non-exclusive mode. Regardless of where the Directory Server is installed (locally or remotely), Directory Server is not tuned when you run `amtune`. When you run the script, it creates a tar file named `/tmp/amtune-directory.tar`. By default, the extracted files will be placed in `/tmp` directory. You need to extract this file in the machine on which Directory Server is running on your system and then run the `amtune-directory` script.

AMTUNE_MODE_WEB_CONTAINER

This parameter tunes the web container into which Identity Server is installed.

AMTUNE_MODE_IDENTITY

This parameter tunes the installed instance of Identity Server.

The following parameters are used for all amtune operations:

AMTUNE_DEBUG_FILE_PREFIX

This parameter defines the debug filename prefix. If this is set to a non-empty value, then all of the operations performed by the `amtune` scripts are logged. The location of the log file is set in the `com.ipplanet.services.debug.directory` parameter in `AMConfig.properties`.

If no value is specified, debugging information is not recorded and all output is sent to the `/dev/null` directory.

AMTUNE_PCT_MEMORY_TO_USE

This parameter defines the amount of available memory used by Identity Server. Currently, Identity Server requires a minimum of 512MB of RAM and can use a maximum of 4 GB, which is the per-process address space limit for 32-bit applications. If you set this parameter to 0 (the lowest value), Identity Server is configured to use 512MB. Conversely, if you set this parameter to 100, the maximum space allowed for Identity Server would be the minimum amount between 4GB and 100% of the system's available RAM. The following values are some of the files tuned based on this setting (for a complete list, see the debug file):

Web container values

`server.xml` file:

- **heap size settings**
- `<JVMOPTIONS>-XX:PermSize` **and** `<JVMOPTIONS>-XX:MaxNewSize`
- `<JVMPTIONS>-XX:Permsize` **and** `<JVMOPTIONS>-XX:MaxPermSize`

`magnus.conf` file:

- `RqThrottle` **setting**

Identity Server AMConfig.properties values

Notification thread pool settings:

- `com.ipplanet.am.notification.threadpool.size`
- `com.ipplanet.am.notification.threadpool.threshold`

SDK cache maximum size setting:

- `com.ipplanet.am.sdk.cache.maxsize`

Session settings:

- `com.ipplanet.am.session.httpSession.enabled`
- `com.ipplanet.am.session.maxSessions`
- `com.ipplanet.am.session.invalidsessionmaxtime`
- `com.ipplanet.am.session.purgedelay`

AMTUNE_PER_THREAD_STACK_SIZE

This parameter sets the available stack spaces per thread. The per thread stack size is used to tune various thread-related parameters in Identity Server and the web container. The default value is 128KB. This value should not be changed.

AMTUNE_SESSION_MAX_SESSION_TIME_IN_MTS

This parameter sets the maximum session time in minutes. The default is 60, however this value may be different for your installation. If the Session service is registered and customized at the any other level, the tuning will not apply.

Setting this parameter to very high or very low values affects the number of active user sessions an Identity Server deployment can support, so this parameter is optional for tuning purposes.

In order to use this parameter, you must ensure that `AM_TUNE_DONT_TOUCH_SESSION_PARAMETERS` is set to `false`.

AMTUNE_SESSION_MAX_IDLE_TIME_IN_MTS

This parameter sets the maximum idle time for a session in minutes. The default is 10, however this value may be different for your installation. If the Session service is registered and customized at the any other level, the tuning will not apply.

Setting this parameter to very high or very low values affects the number of active user sessions an Identity Server deployment can support, so this parameter is optional for tuning purposes.

In order to use this parameter, you must ensure that `AM_TUNE_DONT_TOUCH_SESSION_PARAMETERS` is set to `false`.

AMTUNE_SESSION_MAX_CACHING_TIME_IN_MTS

This parameter sets the maximum session cache time in minutes. The default is 2, however this value may be different for your installation. If the Session service is registered and customized at the any other level, the tuning will not apply.

Setting this parameter to very high or very low values affects the number of active use sessions an Identity Server deployment can support, so this parameter is optional for tuning purposes.

In order to use this parameter, you must ensure that `AM_TUNE_DONT_TOUCH_SESSION_PARAMETERS` is set to `false`.

Installation Environment Parameters

HOSTNAME

This parameter defines the host name of the system on which Identity Server is deployed. If the host name for your environment cannot be obtained using the `hostname` command, comment the following line:

```
HOSTNAME='/bin/hostname'
```

Then, add a line setting the correct hostname. For example:

```
HOSTNAME=machine_name
```

DOMAINNAME

This parameter defines the domain name of the system on which Identity Server is deployed. If the domain name for your environment cannot be obtained using the `domainname` command, comment the following line:

```
DOMAINNAME='/bin/domainname'
```

Then, add a line setting the correct domainname. For example:

```
DOMAINNAME=example.com
```

IS_CONFIG_DIR

This parameter defines the configuration directory for Identity Server. The default location is `IdentityServer_base/SUNWam/config`. Do not change this parameter.

WEB_CONTAINER

This parameter defines the name of the web container on which Identity Server is deployed. It accepts the following values:

- `WS61` - specifies Web Server 6.1 as the web container.
- `AS7` - specifies Application Server 7 as the web container.

Any other value will produce a validation error.

CONTAINER_BASE_DIR

This parameter defines the base directory for the web container on which Identity Server is deployed. If you installed the web container in a non-default location, change this value before running `amtune`.

WEB_CONTAINER_INSTANCE_NAME

This parameter defines the instance of the name of the web container where Identity Server is deployed.

For Java System Web Server web container, the instance name is normally the host name of the Identity Server. If the instance name is different from the hostname, you will need to specify the correct instance name here. For example:

```
/opt/SUNWwbsrvr/https-fully_qualified_hostname
```

In this case, `WEB_CONTAINER_INSTANCE_NAME` can be left as is:

```
WEB_CONTAINER_INSTANCE_NAME=$HOSTNAME
```

If the Web Server installation location is other than the typical value, for example, `/opt/SUNWwbsrvr/https-instance1`, the instance name would be `instance1`.

```
WEB_CONTAINER_INSTANCE_NAME=instance1
```

NOTE You will need to drop the "https-" from the directory name of the install location of JSWS.

For the Application Server web container, the instance name is normally `server1`. For example:

```
/var/opt/SUNWappserver7/domains/domain1/server1/
```

In this case, the instance name is the last part of the install location and is `server1`.

If the Application Server install location is other than the typical value, lets say, if the install location is

```
/var/opt/SUNWappserver7/domains/domain1/server-identity-ssl
```

, the instance name would be `server-identity-ssl`:

```
WEB_CONTAINER_INSTANCE_NAME=server-identity-ssl
```

NOTE You will need to specify the complete instance name for Application Server, typically, the leaf directory in the install path.

IS_INSTANCE_NAME

This parameter is used in determining the property filenames for the Identity Server install. Multiple instances of Identity Server could be deployed in the same machine, but generally, there will be one set of property files per Identity Server instance and the instance name will be appended to the file names.

If there is only one instance of Identity Server on a machine, then the instance name will not be appended to the file names.

For example, there may be a single instance of Identity Server running under the default instance of Web Server:

If your Identity Server is installed on a machine named `server.example.com`, typically your first instance of Web Server will be `https-server.example.com`. The property files for the first Identity Server instance will not have the instance name appended (for example, `AMConfig.properties`).

In the case of multiple instances, there will be different names. For example, there may be three instances of Web Server. The Web Server instances could be `server.example.com-instance1`, `server.example.com-instance2`, `server.example.com-instance3`. If three instances of Identity Server are deployed (one per container instance), then the primary property file names for Identity Server (typically, `AMConfig.properties`) may look like the following:

- `AMConfig-instance1.properties`
- `AMConfig-instance2.properties`
- `AMConfig-instance3.properties`.

You can specify `IS_INSTANCE_NAME=instance1`. `amtune` will resolve the property file names in the following order:

1. `AMConfig-IS_INSTANCE_NAME`
2. `AMConfig-WEB_CONTAINER_INSTANCE_NAME`
3. `AMConfig.properties`

The tool will use the first available property file in the list and use it.

NOTE The web container and the `amadmin` tool should point to the correct instance of Identity Server as well.

For web containers, you will have to explicitly specify the instance name in the `server.xml` configuration file of the web container instance configuration as well. For example:

```
<JVMOPTIONS>-Dserver.name=instance1</JVMOPTIONS>
```

NOTE `amadmin` tool should also point to the correct server name (`java option -Dserver.name=instance1`).

CONTAINER_INSTANCE_DIR

This parameter defines the base directory for the container instance to which Identity Server is deployed. If you have installed the web container in a non-default location, change this value before running `amtune`.

Directory Server Parameters

DIRMGR_UID

This parameter defines the user ID of the Directory Manager. If you change the user ID from the default value (`cn=Directory Manager`), then you must change the value of this parameter.

DEFALUT_ORG_PEOPLE_CONTAINER

This parameter defines the Identity Server instance's default people container location below the top-level organization. This value is used to tune the search base for the LDAP authentication service. The search scope is also modified to the object level and the default search scope is in the subtree level. This parameter is useful when there are no suborganizations in the default organization. If no values are specified, the tuning is skipped.

Configuring Identity Server in SSL Mode

Using Secure Socket Layer (SSL) with simple authentication guarantees confidentiality and data integrity. To enable Identity Server in SSL, mode you would typically:

1. Configure Identity Server with a secure web container
2. Configure Identity Server to a secure Directory Server

The following sections describe these steps:

- [“Configuring Identity Server With a Secure Sun Java System Web Server” on page 57](#)
- [“Configuring Identity Server with a Secure Sun Java System Application Server” on page 60](#)
- [“Configuring Identity Server to Directory Server in SSL Mode” on page 64](#)

Configuring Identity Server With a Secure Sun Java System Web Server

To configure Identity Server in SSL mode with Sun Java System Web Server, see the following steps:

1. 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.

NOTE Be sure to click Save. If you don't, you will still be able to proceed with the following steps, but all configuration changes you have made will be lost and you will not be able to log in as administrator to fix it.

[Step 2](#) through [Step 25](#) describe the Sun Java System Web Server.

2. Log on to the Web Server console. The default port is 58888.
3. 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.
4. Click on the Apply button located top right corner of the screen.
5. Click Apply Settings.
The Web Server should automatically restart. Click OK to continue.
6. Stop the select Web Server instance.
7. Click the Security Tab.
8. Click on Create Database.
9. Enter the new database password and click OK.
Ensure that you write down the database password for later use.
10. Once the Certificate Database has been created, click on Request a Certificate.
11. Enter the data in the fields provided in the screen.
The Key Pair Field Password field is the same as you entered in [Step 9](#). In the location field, you will need to spell out the location completely. Abbreviations, such as CA, will not work. All of the fields must be defined. In the Common Name field, provide the hostname of your Web Server.
12. Once the form is submitted, you will see a message such as:

```
--BEGIN CERTIFICATE REQUEST--
afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdf
alsfjawoeirjoi2ejowdnlkswvnwofijwoeijfwiepweroiwejprwfrwl
--END CERTIFICATE REQUEST--
```

13. Copy this text and submit it for the certificate request.

Ensure that you get the Root CA certificate.

14. You will receive a certificate response containing the certificate, such as:

```
--BEGIN CERTIFICATE--
afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdf
alsfjawoeirjoi2ejowdnlkswvnwofijwoeijfwiepweroiwejprwfrwl
--END CERTIFICATE--
```

15. Copy this text into your clipboard, or save the text into a file.

16. Go to the Web Server console and click on Install Certificate.

17. Click on Certificate for this Server.

18. Enter the Certificate Database password in the Key Pair File Password field.

19. 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.

20. Click Install Certificate.

21. Click Certificate for Trusted Certificate Authority.

22. Install the Root CA Certificate in the same manner described in [Step 16](#) through [Step 21](#).
23. Once you have completed installing both certificates, click on the Preferences tab in the Web Server console.
24. Select Add Listen Socket if you wish to have SSL enabled on a different port. Then, select Edit Listen Socket.
25. Change the security status from Disabled to Enabled, and click OK to submit the changes.

[Step 26](#) through [Step 28](#) describe Identity Server.

26. Open the `AMConfig.properties` file. By default, the location of this file is `etc/opt/SUNWam/config`.
27. 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.
28. Save the `AMConfig.properties` file.
29. 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.
30. Enter the Certificate Database password in the text field and select Start.

Configuring Identity Server with a Secure Sun Java System Application Server

Setting up Identity Server to run on an SSL-enabled Sun Java System Application server is a two-step process. First, secure the Application Server instance to the installed Identity Server, then configure Identity Server itself.

Setting Up Application Server With SSL

To Secure the Application Server Instance:

1. Log into the Sun Java System Application Server console as an administrator by entering the following address in your browser:

```
http://fullservername:port
```

The default port is 4848.
2. Enter the username and password you entered during installation.
3. Select the Application Server instance on which you installed (or will install) Identity Server. The right frame displays that the configuration has changed.
4. Click Apply Changes.
5. Click Restart. The Application Server should automatically restart.
6. In the left frame, click Security.
7. Click the Manage Database tab.
8. Click Create Database, if it is not selected.
9. Enter the new database password and confirm, then click the OK button. Make sure that you write down the database password for later use.
10. Once the Certificate Database has been created, click the Certificate Management tab.
11. Click the Request link, if it is not selected.
12. Enter the following Request data for the certificate
 - a. Select it if this is a new certificate or a certificate renewal. Many certificates expire after a specific period of time and some certificate authorities (CA) will automatically send you renewal notification.
 - b. Specify the way in which you want to submit the request for the certificate.

If the CA expects to receive the request in an E-mail message, check CA E-mail and enter the E-mail address of the CA. For a list of CAs, click List of Available Certificate Authorities.

If you are requesting the certificate from an internal CA that is using the Sun Java System Certificate Server, click CA URL and enter the URL for the Certificate Server. This URL should point to the certificate server's program that handles certificate requests.
 - c. Enter the password for your key-pair file (this is the password you specified in [Step 9](#)).

- d. Enter the following identification information:

Common Name. The full name of the server including the port number.

Requestor Name. The name of the requestor.

Telephone Number. The telephone number of the requestor

Common Name. The fully qualified name of the Sun Java System Application Server on which the digital certificate will be installed.

E-mail Address. The E-mail address of the administrator.

Organization Name. The name of your organization. The certificate authority may require any host names entered in this attribute belong to a domain registered to this organization.

Organizational Unit Name. The name of your division, department, or other operational unit of your organization.

Locality Name (city). The name of your city or town.

State Name. The name of the state or province in which your organization operates if your organization is in the United States or Canada, respectively. Do not abbreviate.

Country Code. The two-letter ISO code for your country. For example, the code for the United States is US.

13. Click the OK button. A message will be displayed, for example:

```
--BEGIN NEW CERTIFICATE REQUEST---  
  
afajsdllwqeroisdaoi234r1kqwelkasjlasnvdknbslajowijalsdkjfalsdf1a  
  
alsfjawoeirjoi2ejowdn1kswnvwnwofijwoeijfwiepwferfoiqeroijepwprfwl  
  
--END NEW CERTIFICATE REQUEST--
```

14. Copy all of this text to a file and click OK. Make sure that you get the Root CA certificate.
15. Select a CA and follow the instructions on that authority's web site to get a digital certificate. You can get the certificate from CMS, Verisign or Entrust.net

16. After you receive your digital certificate from the certificate authority, you can copy the text into your clipboard, or save the text into a file.
17. Go to the Sun Java System Application Server console and click on the Install link.
18. Select Certificate For This Server.
19. Enter the Certificate Database password in the Key Pair File Password field. (It is the same password you entered in [Step 9](#)).
20. Paste the certificate into the provided text field, Message text (with headers), or enter the filename in the Message that is in this file text box. Select the appropriate radio button.
21. Click OK button. The browser displays the certificate, and provides a button to add the certificate.
22. Click Add Server Certificate.
23. Install the Root CA Certificate in the same manner described in [Step 10](#) through [Step 22](#). However, in [Step 18](#), select Certificate for Trusted Certificate Authority.
24. Once you have completed installing both certificates, expand the HTTP Server node in the left frame
25. Select HTTP Listeners under HTTP Server.
26. Select `http-listener-1`. The browser displays the socket information.
27. Change the value of the port used by `http-listener-1` from the value entered while installing application server, to a more appropriate value such as 443.
28. Select SSL/TLS Enabled.
29. Select Certificate Nickname.
30. Specify the Return server. This should match the common name specified in [Step 12](#).
31. Click Save.
32. Select the Application Server instance on which you will install the Sun Java System Identity Server software. The right frame shows that the configuration has changed.
33. Click Apply Changes.
34. Click Restart. The application server should automatically restart.

Configuring Identity Server in SSL Mode

To configure Identity Server in SSL mode:

1. In the Identity Server console, go to the Service Configuration module and select the Platform service. In the Server List attribute, add the same URL with the HTTPS protocol and an SSL-enabled port number. Click Save.

NOTE If a single instance of Identity Server is listening on two ports (one in Http and one in Https) and you try to access Identity Server with a stalled cookie, Identity Server will become unresponsive. This is not a supported configuration.

2. Open the `AMConfig.properties` file from the following default location:
`/etc/opt/SUNWam/config.`
3. Replace all of the protocol occurrences of `http://` to `https://` and change the port number to an SSL-enabled port number.
4. Save the `AMConfig.properties` file.
5. Restart the Application Server.

Configuring Identity Server to Directory Server in SSL Mode

To provide secure communications over the network, Identity Server includes the LDAPS communications protocol. LDAPS is the standard LDAP protocol, but it runs on top of the Secure Sockets Layer (SSL). In order to enable SSL communication, you must first configure the Directory Server in SSL mode and then connect Identity Server to Directory Server. The basic steps are as follows:

1. Obtain and install a certificate for your Directory Server, and configure the Directory Server to trust the certification authority's (CA) certificate.
2. Turn on SSL in your directory.
3. Configure the authentication, policy and platform services to connect to an SSL-enabled Directory Server.
4. Configure Identity Server to securely connect to the Directory Server backend.

Configuring Directory Server in SSL Mode

In order to configure the Directory Server in SSL mode, you must obtain and install a server certificate, configure the Directory Server to trust the CA's certificate and enable SSL. Detailed instructions on how to complete these tasks are included in Chapter 11, "Managing Authentication and Encryption" in the *Directory Server Administration Guide*. This document can be found in the following location:

<http://docs.sun.com/doc/817-5221>

You can also download a PDF of the manual from the following location:

http://docs.sun.com/coll/DirectoryServer_04q2

If your Directory Server is already SSL-enabled, go to the next section for details on connecting Identity Server to Directory Server.

Connecting Identity Server to the SSL-enabled Directory Server

Once the Directory Server has been configured for SSL mode, you need to securely connect Identity Server to the Directory Server backend. To do so:

1. In the Identity Server Console, go to the LDAP Authentication service in the Service Configuration module.
 - a. Change the Directory Server port to the SSL port.
 - b. Select the Enable SSL Access to LDAP Server attribute.
2. Go to the Membership Authentication service in the Service Configuration module.
 - a. Change the Directory Server port to the SSL port.
 - b. Select the Enable SSL Access to LDAP Server attribute.
3. Go to the Policy Configuration Authentication service in the Service Configuration module.
 - a. Change the Directory Server port to the SSL port.
 - b. Select the Enable LDAP SSL attribute.

4. Open the `serverconfig.xml` in a text editor. The file is in the following location:

`etc/opt/SUNWam/config`

- a. In the `<Server>` element, change the following values:

`port` - enter the port number of the secure port to which Identity Server listens (636 is the default).

`type`- change SIMPLE to SSL.

- b. Save and close `serverconfig.xml`.

5. Open the `AMConfig.properties` file from the following default location:

`IdentityServer_base/SUNWam/config`.

Change the following properties:

- a. `Directory Port = 636 (if using the default)`

- b. `ssl.enabled = true`

- c. Save `AMConfig.properties`.

6. Restart the server

Managing Identity Server Through the Console

This is part two of the *Sun Java™ System Identity Server 2004Q2 Administration Guide*. It discusses the Identity Server graphical user interface and how to navigate through it. This section contains the following chapters:

- “Identity Management” on page 69
- “Service Configuration” on page 101
- “Current Sessions” on page 111
- “Policy Management” on page 115
- “Authentication Options” on page 143
- “Password Reset Service” on page 177

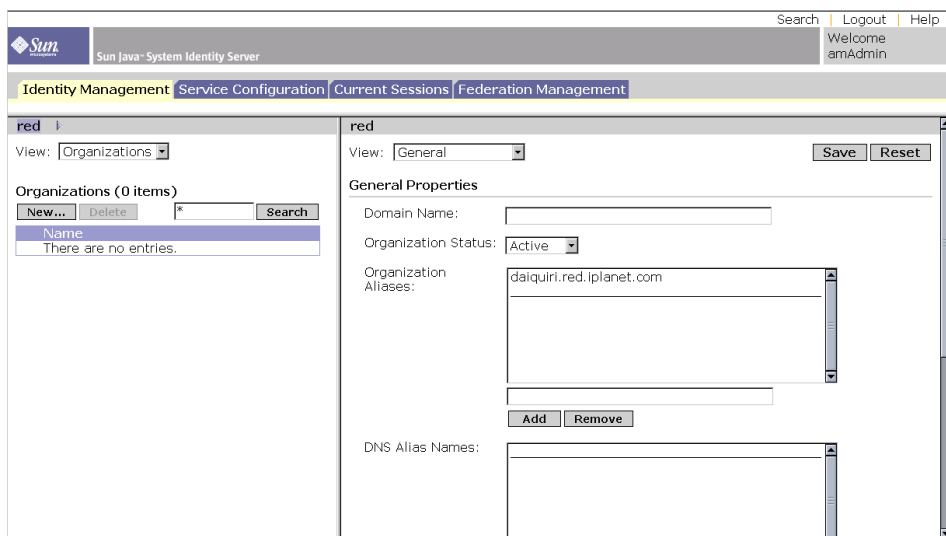
Identity Management

This chapter describes the identity management features of Sun Java™ System Identity Server 2004Q2. 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 Server Console” on page 69](#)
- [“The Identity Management Interface” on page 73](#)
- [“Managing Identity Server Objects” on page 74](#)

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 4-1 The Identity Server Console

Header Pane

The Header pane runs along the top of the console. The tabs in the Header pane allow 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* field displays the name of the user that is currently running the console with a link to the 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 IV](#) of this documentation, the [Attribute Reference](#).

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

Navigation Pane

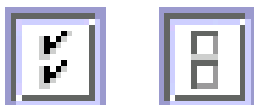
The Navigation pane is the left portion of the Identity Server console. The *Directory Object* portion (within the grey box) displays the name of the directory object that is currently open and its *Properties* link. (Most objects displayed in the Navigation pane will have a corresponding *Properties* link. Selecting this link will render the entry's attributes in the Data pane 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.

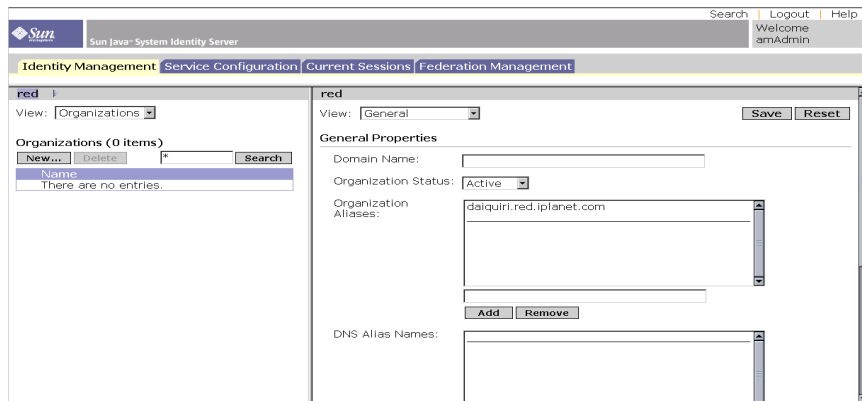


There are two basic views of the Identity Server graphical user interface. Depending on the roles of the user logging in, they might gain access to the Identity Management view or the User Profile view.

Identity Management View

When a user with an administrative role authenticates to the Identity Server, the 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 creating, deleting and managing objects (users, organizations, policies, and so forth), and configuring services.

Figure 4-2 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 the user's own User Profile. In this view the user can modify the values of the attributes particular to the user's 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 *Identity Server Developer's Guide*.

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.

See the *Identity Server Developer's Guide* for information on how to extend an entry's properties.

Figure 4-3 User Profile View

The screenshot displays the 'User Profile View' in the Sun Java System Identity Server. The interface is divided into a header and a main content area. The header contains the Sun logo, the text 'Sun Java System Identity Server', and a user greeting 'Welcome scott carver' with 'Logout' and 'Help' links. The main content area features a form for editing user details. The form includes the following fields and values: 'First Name' (Joey), 'Last Name' (Ramone), 'Full Name' (Joey Ramone), 'Password' (masked with asterisks), 'Password (confirm)' (masked with asterisks), 'Email Address' (joeyramone@example.com), 'Employee Number' (05191951), 'Telephone Number' (1-234-555-5303), 'Home Address' (empty), and 'User Alias List' (empty table with 'Add' and 'Remove' buttons). 'Save' and 'Reset' buttons are positioned at the top right of the form area.

The Identity Management Interface

The Identity Management interface allows for the creation and management of identity-related objects. User, role, group, policies, organization, suborganization and container objects and more 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.

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 policies, container objects, and agents). This section explains the object types and details how to configure them.

For most Identity Server object types, you can optionally configure Display Options and Available Actions to show or hide the way in which the web interfaces are displayed in the Identity Server console. Configuration is done at the organization and role levels and users inherit the configuration from the organization in which they reside and the are roles that are assigned to them. These settings are described at the end of this chapter.

Organizations

An Organization 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.

To Create an Organization

1. Choose Organizations from the View menu in the Identity Management module.
2. Click New in the Navigation pane.
3. Enter the values for the fields. Only Name is required. The fields are:

Name. Enter a value for the name of the Organization.

Domain Name. Enter the full Domain Name System (DNS) name for the organization, if it has one.

Organization Status. 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 user access when logging in to the organization.

Organization Aliases. This field defines alias names for the organization, allowing you to use the aliases for authentication with a URL login. For example, if you have an organization named `exampleorg`, and define `123` and `abc` as aliases, you can log into the organization using any of the following URLs:

```
http://machine.example.com/UI/Login?org=exampleorg
```

```
http://machine.example.com/UI/Login?org=abc
```

```
http://machine.example.com/UI/Login?org=123
```

Organization alias names must be unique throughout the organization. You can use the Unique Attribute List to enforce uniqueness.

DNS Alias Names. Allows you to add alias names for the DNS name for the organization. This attribute only accepts “real” domain aliases (random strings are not allowed). For example, if you have a DNS named `example.com`, and define `example1.com` and `example2.com` as aliases for an organization named `exampleorg`, you can log into the organization using any of the following URLs:

```
http://machine.example.com/UI/Login?org=exampleorg
```

```
http://machine.example1.com/UI/Login?org=exampleorg
```

```
http://machine.example2.com/UI/Login?org=exampleorg
```

Unique Attribute List. Allows you to add a list of unique attribute names for users in the organization. For example, if you add a unique attribute name specifying an email address, you would not be able to create two users with the

same email address. This field also accepts a comma-separated list. Any one of the attribute names in the list defines uniqueness. For example, if the field contains the following list of attribute names:

```
PreferredDomain, AssociatedDomain
```

and PreferredDomain is defined as `http://www.example.com` for a particular user, then the entire comma-separated list is defined as unique for that URL.

Uniqueness is enforced for all suborganizations.

4. Click OK.

The new organization displays in the Navigation pane. To edit any of the properties that you defined during creation of the organization, click the Properties arrow of the organization you wish to edit, select General from the View menu in the Data pane, edit the properties and click OK. You can use the [Display Options](#) and [Available Actions](#) views to customize the appearance of the Identity Server console and to specify the behavior for any users that authenticate to this organization.

To 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 Search.

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 and you can not perform an undo.

To 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 "[Managing Policies](#)" on page 129.

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. Groups that exist within other groups are called *sub-groups*. Sub-groups are child nodes that “physically” exist within a parent group.

Identity Server also supports *nested groups*, which are “representations” of existing groups contained in a single group. As opposed to sub-groups, nested groups can exist anywhere in the DIT. They allow you to quickly set up access permissions for a large number of users.

When you create a group, you can create groups that use Membership By Subscription (*static group*) or Membership By Filter (*filtered groups*). This controls the way in which users are added to the group. Users can only be added to static groups. Dynamic groups control the addition of users through a filter. Nested or sub-groups, however, can be added to both.

Static Group (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 217.

NOTE By default, the managed group type is dynamic. You can change this default in the Administration service configuration.

Filtered Group (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.

NOTE Identity Server should be configured with Directory Server to use the referential integrity plug-in. When the referential integrity plug-in is enabled, it performs integrity updates on specified attributes immediately after a delete or rename operation. This ensures that relationships between related entries are maintained throughout the database. Database indexes enhance the search performance in Directory Server. For more information on enabling the plug-in, see the *Sun Java System Identity Server Migration Guide*.

To Create a Static Group

1. Navigate to the organization, group or group container where the group will be created.
2. Choose Groups from the View menu.
3. Click New.
4. Select Membership By Subscription for the group type from within the Data pane.
5. Enter a name for the group in the Name field. Click Next.
6. Select the Users Can Subscribe to this Group attribute to allow users to subscribe to the group themselves.
7. If you have defined multiple group containers in your DIT and the Show Group Containers attribute (from the Administration Service) is not enabled, you can select the Parent Group Container to which the static group will belong. Otherwise, this field is not displayed.
8. Click Finish.

Once the group is created, you can edit the Users Can Subscribe to this Group attribute by selecting General from the View menu in the Data pane.

To Add or Remove Members to a Static Group

1. Click the Properties arrow next to the group to which you will add members.

2. In the Data pane, select Members from the View menu.

Choose an action to perform in the Select Action menu. The actions you can perform are as follows:

New User. This action creates a new user and automatically adds the user to the group when the user information is saved.

Add User. This action adds an existing user to the group. When you select this action, you create a search criteria which will specify users you wish to add. The fields used to construct the criteria use either an *ANY* or *ALL* operator. *ALL* returns users for all specified fields. *ANY* returns users for any one of the specified fields. If a field is left blank, it will match all possible entries for that particular attribute. From the returned list of users, select the users you wish to add and click OK.

Add Group. This action adds a nested group to the current group. When you select this action, you create a search criteria, including search scope, the name of the group (the "*" wildcard is accepted), and you can specify whether users can subscribe to the group themselves. From the returned list of groups, select the group you wish to add and click OK.

Remove Members. This action will remove members from the group, but will not delete them. Select the member you wish to remove and select Remove Member from the action menu.

Delete Members. This action will permanently delete the member you select.

To Create a Filtered 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 Membership By Filter for the group type from within the Data pane.
5. Enter a name for the group in the Name field. Click Next.

6. Construct the LDAP search filter.

By default, Identity Server displays the Basic search filter interface. The Basic fields used to construct the filter use either an ANY or ALL operator. ALL returns users for all specified fields. ANY returns users for any one of the specified fields. If a field is left blank it will match all possible entries for that particular attribute.

Alternatively, you can select the Advanced button to define the filter attributes yourself. For example,

```
(&(uid=user1)(|(inetuserstatus=active)!(inetuserstatus=*)))))
```

When you click Finish, all users matching the search criteria are automatically added to the group.

To Add or Remove Members to a Filtered Group

1. Click the Properties arrow next to the group to which you will add members.
2. In the Data pane, select Members from the View menu.

Choose an action to perform in the Select Action menu. The actions you can perform are as follows:

Add Group. This action adds a nested group to the current group. When you select this action, you create a search criteria, including search scope, the name of the group (the "*" wildcard is accepted), and you can specify whether the group allows users to subscribe to them. From the returned list of groups, select the group you wish to add and click OK.

Remove Members. This action will remove members from the group, but will not delete them. Select the member you wish to remove and click OK.

Delete Members. This action will permanently delete the member you select.

To 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 ["Managing Policies" on page 129](#).

Users

A *user* represents an individual's identity. Through the Identity Server Identity Management module, users can be created and deleted in organizations, containers and groups and can be added or removed from roles and/or groups. You can also assign services to the user.

NOTE If a user in a suborganization is created with the same `userid` as `amadmin`, the login will fail for `amadmin`. If such a problem arises, the administrator should change the user's `userid` through the Directory Server console. This enables the administrator to login to the default organization. Additionally, the DN to Start User Search in the authentication service can be set to the people container DN to ensure that a unique match is returned during the login process.

To Create a User

1. Navigate to the organization, container or people container where the user is to be created.
2. Choose Users from the View menu.
3. Click New.

This displays the New User page in the Data pane.

4. Select the services to be assigned to the user.

The only services that are displayed are those that contain user attributes and have been added to the organization to which the user will belong. Click Next.

5. If you have defined multiple (more than two) group containers in your DIT and the Show Group Containers attribute (from the Administration Service) is not enabled, you can select the people container to which the static group will belong from the User Creation page. Otherwise, this field is not displayed.
6. Enter values for the required attributes.

Information on the user profile attributes can be found in [“User Attributes” on page 341](#).

7. Click OK.

To Add a User to Roles and Groups

1. Navigate to the Organization for the user that is to 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 Roles or Groups. Only the roles and groups that have already been assigned to the user are displayed. Click Add to see the list of available roles and groups from which to choose.
5. Select the role or group that to which you wish to add the user, and click Save.

To Add a Service to a User

1. Navigate to the Organization for the user that is to be modified.
2. Choose Users from the View menu in the Navigation pane.
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.
5. Click Add to select the services you wish to assign to the user.
6. Click Save.

To Delete a User

1. Navigate to the location 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.

NOTE There is no warning message before the delete operation, and it can not be undone.

To 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 [“Managing Policies” on page 129](#).

Services

Activating a *service* for an organization or container (containers behave the same as organizations) is a two step process. In the first step you need to add the service to the organization. After you add the service, you must configure a template configured specifically for that organization. For additional information, see [Chapter 5, “Service Configuration”](#)

NOTE New services 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 *Identity Server Developer's Guide*.

To Add 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 Add.

The Data pane will display a list of services available to add to this organization.

4. Select the checkbox next to each service to be added.
5. Click OK. The services that have been added are displayed in the Navigation pane.

NOTE Only the services that are added to the parent organization are displayed at the suborganization level.

To Create a Template for a Service

1. Navigate to the organization or role where the added service exists.

Choose Organizations from the View menu in the Identity 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 *A template does not currently exist for this service. Do you want to create one now?*

4. Click Yes.

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” on page 213](#).

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

To Remove 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 Remove.

NOTE Services can not be removed from the parent organization level if they are registered at the sub organization level.

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 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 the Identity Server console. Other Directory Server types can be used in a policy’s subject definition. For more information on policy subjects, see [“Creating Policies” on page 126](#).

Users can possess one or more roles. For example, a contractor role which has attributes from the Session Service and the Password Reset Service might be created. When new contractors start, the administrator can assign them this role rather than setting separate attributes in the contractor entry. If the contractor is working in the Engineering department and requires services and access rights applicable to an engineering employee, the administrator could assign the contractor to the engineering role as well as the contractor 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 [“Show Roles on User Profile Page” on page 225](#).

NOTE

Identity Server should be configured with Directory Server to use the referential integrity plug-in. When the referential integrity plug-in is enabled, it performs integrity updates on specified attributes immediately after a delete or rename operation. This ensures that relationships between related entries are maintained throughout the database. Database indexes enhance the search performance in Directory Server. For more information on enabling the plug-in, see the *Sun Java System Identity Server Migration Guide*.

Similar to groups, roles can be created by a filter, or be created statically.

Static Role. In contrast to a filtered role, a static role can be created without adding users at the point of the role's creation. This gives you more control when adding specific users to a given role.

Filtered Role. A filtered role is a dynamic role created through the use of an LDAP filter. All users are funneled through the filter and assigned to the role at the time of the role's creation. The filter looks for any attribute value pair (for example, `ca=user*`) in an entry and automatically assign the users that contain the attribute to the role.

To Create a Static Role

1. In the Navigation pane go 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. The default roles are:

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 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.

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.

3. Click New in the Navigation pane. The New Role template appears in the Data pane.
4. Select Static Role and enter a name. Click Next.
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 determine and here to start the user in the Identity Server console. 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. The default permissions shown are in no particular order. The permissions are:

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.

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

8. Click Finish.

The created role is displayed in the Navigation pane and status information about the role is displayed in the Data pane.

You can optionally configure the Display Options and Available Actions by selecting them in the View menu. For more information, see [Display Options](#) and [Available Actions](#) at the end of this chapter.

To Add Users to a Static Role

1. Select the role to modify and click on the Properties arrow.
2. Choose Users from the View menu in the Data pane.
3. Click Add.

4. Enter the information for the search criteria. You can choose to search for users based on one or more the displayed fields The fields are:

Return Users By. Allows you to specify the value returned by the search.

Match. Allows you to include an operator for any the fields you wish to include for the filter. *ALL* returns users for all specified fields. *ANY* returns users for any one of the specified fields.

User ID. Search for a user by User ID.

First Name. Search for users by their first name.

Last Name. Search for users by their last name.

Full Name. Search for users by their full name.

User Status. Search for users by their status (active or inactive).

5. Click Next to begin the search. The results of the search are displayed.
6. Choose the users from the names returned by selecting the checkbox next to the user name.
7. Click Finish.

The Users are now assigned to the role.

To Create a Filtered Role

1. In the Navigation pane, go 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. The default roles are:

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.

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.

3. Click New in the Navigation pane. The New Role template appears in the Data pane.
4. Select Filtered Role and enter the name. Click Next.
5. Enter a description for 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 determine and where to start the user in the Identity Server console. 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.
8. The permissions provide access to entries within the organization. The default permissions shown are in no particular order. The permissions are:

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.

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

9. Enter the information for the search criteria. The fields are:

Match. Allows you to include an operator for any the fields you wish to include for the filter. **ALL** returns users for all specified fields. **ANY** returns users for any one of the specified fields.

User ID. Search for a user by User ID.

First Name. Search for users by their first name.

Last Name. Search for users by their last name.

Full Name. Search for users by their full name.

User Status. Search for users by their status (active or inactive).

Alternatively, you can select the Advanced button to define the filter attributes yourself. For example,

```
(&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*)))))
```

Click Reset to clear the filter properties, or click Cancel to cancel the role creation process.

10. Click Finish to initiate the search based on the filter criteria. The users defined by the filter criteria are automatically assigned to the role.

You can optionally configure the Display Options and Available Actions by selecting them in the View menu. For more information, see [Display Options](#) and [Available Actions](#) at the end of this chapter.

NOTE You can add users to static roles through the Role profile page and/or the User profile page.

To 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 next to each user to be removed.

6. Click Remove.

The users are now removed from the role.

To 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 [“Managing Policies” on page 129](#).

Customizing a Service to a Role

You can customize the services available to a role, and the access level for the service attributes, on a per-role basis. Each of the available services can be customized for a role by setting role-specific values to the attributes. You can also grant access for each of the services and to the services' attributes. There may be services that you wish only to be accessed by a specific type of user (for example, managers). To accomplish this, all users are assigned the service, but only the Manager type belonging to the role is allowed access to the specific service.

The same logic applies to service attributes. A user's account consists of many attributes, some of which the user may not be allowed to access; for example the account expiration date. The administrator of the account can be granted access to this attribute, but the user (the account owner) is not. Customizing the service and attribute access is accomplished through the role's Service view in the Navigation pane.

You must first add the services at the organization level in order to display the services. Users that are added to the role will inherit the role's service attributes.

To Configure Services

1. In the role's Service view, go to the section labeled Service Configuration for this Role.
2. Choose a service that is to be granted to the role by clicking on the Edit link next to the service name.

If you have not created a service template, you will be prompted to do so. Click Yes.

3. Modify the Service attributes. For more information on specific Service attributes, see Part 3 of this manual, the *Attribute Reference Guide*.
4. 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.
-------------	---

To Customize Attribute Access

1. In the role's Services view, go to the section labeled Service Access for this Role.
2. Choose the enable or disable status for the service you wish to modify. Enable allows the access modifications. Disable disallows the access modifications.
3. Click the Modify Access link.
4. Assign an access level to an attribute by selecting the Read/Write or Read Only check boxes.
5. Click Save.

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

6. Click Save.

To 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 [“Managing Policies” on page 129](#).

To Delete a Role

1. Navigate to the organization that contains the role to be deleted.
2. 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.
3. Choose Roles from the View menu.
4. Select the checkbox next to the name of the role.
5. Click Delete.

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 [“Creating Policies” on page 126](#).

Agents

Identity Server Policy Agents protect content on web servers and web proxy servers from unauthorized intrusions. They control access to services and web resources based on the policies configured by an administrator.

The *agent* object defines a Policy Agent profile, and allows Identity Server to store authentication and other profile information about a specific agent that is protecting an Identity Server resource. Through the Identity Server console, administrators can view, create, modify and delete agent profiles.

To Create an Agent

1. Navigate to the organization that contains the agent to be created.
2. Choose Agents from the View menu.
3. Click New.
4. Enter the values for the fields. Only Name is required. The fields are:

Name. Enter the name or identity of the agent. This is the name that the agent will use to log into Identity Server. Multi-byte names are not accepted.

Password. Enter the agent password. This password must match the password used by the agent during LDAP authentication.

Confirm Password. Confirm the password.

Description. Enter a brief description of the agent. For example, you can enter the agent instance name or the name of the application it is protecting.

Agent Key Value. Set the agent properties with a key/value pair. This property is used by Identity Server to receive agent requests for credential assertions about users. Currently, only one property is valid and all other properties will be ignored. Use the following format:

agentRootURL=http://server_name:port/

Device Status. Enter the device status of the agent. If set to Active, the agent will be able to authenticate to and communicate with Identity Server. If set to Inactive, the agent will not be able to authenticate to Identity Server.

5. Click OK.

To Delete an Agent

1. Navigate to the organization that contains the agent to be deleted.
2. Choose Agent from the View menu.
3. Select the checkbox next to the name of the agent.
4. Click Delete.

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 Service Configuration module. For more information, see [“Show Containers In View Menu” on page 217](#).

To 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 OK.

You can optionally configure the Display Options and Available Actions by selecting them in the View menu. For more information, see [Display Options](#) and [Available Actions](#) at the end of this chapter.

To 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 Service Configuration module. For more information, see ["Show People Containers" on page 216](#).

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 OK.

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 Service Configuration module. For more information, see [“Show Group Containers” on page 217](#).

To 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 OK. The new group container displays in the Navigation pane.

To 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.

Display Options

For organizations, roles and containers, you can use Display Options view to customize the way in which identity server objects are displayed in the Identity Server console. Not all display options are available for all object types.

To Change the Display Options

1. Click on the Properties arrow of the organization for which you would like to change the display options.
2. Select Display Options from the View menu in the Data pane.
3. Edit the properties in the General section. The properties are:

Generate Full Name Attribute. Select this attribute to enable Identity Server to always generate the user's full name, which is formed from the first and last name values in the user's profile.

Always Select First Entry. Select this attribute for a search so that it automatically selects the first item of a given identity object type in the Navigation pane and displays it in the Data pane.

User Profile Page Title. Choose an attribute from this pull-down menu to be used for the title in the User Profile Page.

Disable Initial Search. This value disables the initial Identity Server search for one or more identity object types. Disabling the initial Search may enhance performance and reduce the likelihood of a timeout error.

4. Change the display options in the Display Configuration of Identity Server Directory Objects section. This section allows you to customize how Identity Server containers and objects are displayed. The Identity Server Directory Containers option allows you to specify which object views are displayed in the Navigation pane's View menu. The Identity Server Directory Objects field allows you to specify which object views are displayed in the Data pane's View menu.

5. Click Save.

Available Actions

For certain Identity Server object types, you can define user access rights through the Available Actions view.

To Set Available Actions for Users

1. Click on the Properties arrow of the Identity object for which you would like to set available actions.
2. Select Available Actions from the View menu in the Data pane.
3. Choose the action type available for any Identity Server object. The action type defines the user's accessibility for each object. The action types are:

No Access. The user has no access to this object.

View. The user has read-only access to this object.

Modify. The user can modify and view this object.

Delete. The user can modify, view and delete this object.

Full Access. The user can create, modify, view and delete this object.

4. Click Save. To change the values to their previously saved state, click Reset.

Service Configuration

This chapter describes the service management features of Sun Java™ System Identity Server 2004Q2. 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” on page 101](#)
- [“Identity Server Services” on page 102](#)
- [“Attribute Types” on page 107](#)
- [“Service Configuration Interface” on page 108](#)

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 directories:

```
IdentityServer_base/SUNWam/dtd/ (Solaris)
```

```
IdentityServer_base/identity/dtd (Linux)
```

NOTE Throughout the rest of this chapter, only the Solaris directory information will be given. Please note that the directory structure for Linux is different. For more information, please see [“About This Guide” on page 19](#).

For more information on defining a Identity Server service, see the *Identity Server Developer’s Guide*.

Identity Server Services

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

```
etc/opt/SUNWam/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 Service

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 Service

There are multiple authentication modules, including a base module. This allows the administrator the opportunity to choose the method with which each defined organization can verify the user’s authorization.

Anonymous

This authentication 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 authentication service allows login through a personal digital certificate (PDC).

Core

This authentication 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.

HTTP Basic

This authentication service uses basic authentication, which is the HTTP protocol's built-in authentication support. In order to use this service, the LDAP authentication services needs to be registered. This will not work from the C API.

LDAP

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

Membership (Self-Registration)

This authentication service allows a new user to self-register for authentication with a login and password. For self-registration, no authentication is required.

NT

This authentication service allows for authenticating users using an Windows NT™/2000™ server. In order to actualize the NT Authentication module, Samba Client (smbclient) 2.2.2 must be downloaded and installed (for Linux, you can use the Samba Client that ships with the operating system).

RADIUS

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

In order for the RADUIS Authentication service to work correctly with Sun Java System Application Server, you must configure Application Server's `server.policy` file. Instructions for this can be found in [“Authentication Options” on page 143](#).

SafeWord

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

In order for the SafeWord Authentication service to work correctly with Sun Java System Application Server, you must configure Application Server's `server.policy` file. Instructions for this can be found in [“Authentication Options” on page 143](#).

SecurID

This authentication service allows for authenticating users using RSA ACE/Server® authentication software and SecurID® authenticators. This service is not supported on Solaris x86.

NOTE In this version of Identity Server, the SecurID Authentication service is not supported for the Linux operating system.

Unix

This authentication service allows for authenticating users using a Unix® server, using a user's UNIX identification and password.

Windows Desktop SSO

This authentication service allows a user who has already authenticated to a Kerberos Distribution Center (KDC) to authenticate to Identity Server without re-submitting the login criteria (Single Sign-on).

Authentication Configuration Service

The Authentication Configuration service allows you to configure authentication for roles, users and services and organizations to set the rules determining the precedence of the authentication modules. You can also configure service-based authentication through this service.

Client Detection Service

The Client Detection service allows Identity Server to detect the client type of an accessing browser and allows the administrator to add and configure devices based on the client type.

Globalization Settings Service

The Globalization Settings contain properties to configure Identity Server for different character sets.

Discovery Service

This service is used by Identity Server's Federation Management module. For more information on this service, please see the *Identity Server Federation Management Guide*.

Liberty Personal Profile Service

This service is used by Identity Server's Federation Management module. For more information on this service, please see the *Identity Server Federation Management Guide*.

Logging Service

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 Service

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.

Password Reset Service

The Password Reset service allows users to receive a forgotten password or reset their password for access to a given service or application protected by Identity Server. The Password Reset service attributes, defined by the top-level administrator, control user validation credentials (in the form of “secret questions”), control the mechanism for new or existing password notification, and sets possible lockout intervals for incorrect user validation.

Platform Service

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 Service

The Policy Configuration service defines values to be used by Policy framework during policy management and policy evaluation.

SAML Service

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 Service

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

SOAP Binding Service

This service is used by Identity Server’s Federation Management module. For more information on this service, please see the *Identity Server Federation Management Guide*.

User Service

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

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 `amUser.xml` file. For more information, see the *Identity Server Developer'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.

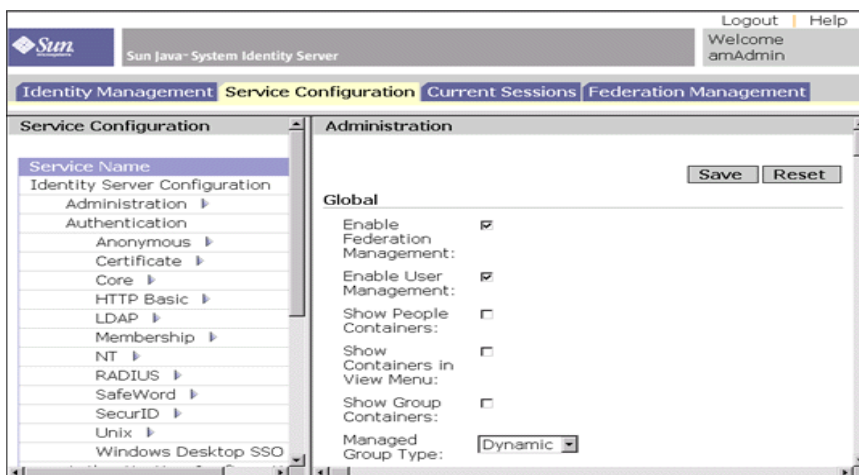
Policy Attributes

Policy attributes specify the access control actions (or privileges) associated with a service. They become a part of the rules when rules are added to a policy. Policy attributes are required in the service schema if you would like to manage access control of the service using Identity Server policies.

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 IV, “Attribute Reference”](#) 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 5-1](#) is a screenshot of the graphical user interface.

Figure 5-1 Service Configuration View

Access the Service Configuration view by choosing the Service Configuration module. The Navigation frame 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 frame.

Current Sessions

This chapter describes the session management features of Sun Java™ System Identity Server 2004Q2. 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. System administrators should ignore the Load Balancer servers listed in the Platform Server list.

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 6-1 Current Sessions Interface

The screenshot shows the Sun Java System Identity Server interface. At the top, there is a navigation bar with the following tabs: Identity Management, Service Configuration, Current Sessions (selected), and Federation Management. The main content area is titled 'Current Sessions' and shows the URL 'http://daiquiri.red.iplanet.com:80'. Below this, there is a 'Server Name' dropdown menu with the same URL selected. The main content area displays 'User Sessions (1 item)' and a table with the following columns: User ID, Time Left (minutes), Max Session Time (minutes), Idle Time (minutes), and Max Idle Time (minutes). The table contains one row for user 'amAdmin' with values: 119, 120, 0, and 30. There is a 'Terminate Session' button and a 'Filter' input field above the table.

User ID	Time Left (minutes)	Max Session Time (minutes)	Idle Time (minutes)	Max Idle Time (minutes)
amAdmin	119	120	0	30

Session Management Frame

The Session Management frame displays the name of the Identity Server that is currently being managed.

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) that the user can be logged in before the session expires and must reauthenticate to regain access.

Idle Time. Displays the time (in minutes) that the user has been idle.

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 Service Attributes” on page 335](#) 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.

The Current Sessions Interface

Policy Management

This chapter describes the Policy Management feature of Sun Java™ System Identity Server 2004Q2. Identity Server's Policy Management feature provides a means for: the Top-level administrator or Top-level policy administrator to view, create, delete and modify policies for a specific service that can be used across all organizations. It also provides a way for an organization or suborganization administrator or policy administrator to view, create, delete and modify policies for specific use by the organization.

This chapter contains the following sections:

- [“Overview” on page 116](#)
- [“Policy Management Feature” on page 116](#)
- [“Policy Types” on page 119](#)
- [“Policy Definition Type Document” on page 121](#)
- [“Creating Policies” on page 126](#)
- [“Managing Policies” on page 129](#)
- [“Policy Configuration Service” on page 138](#)
- [“Policy-Based Resource Management” on page 140](#)

Overview

A *policy* defines rules that specify access privileges to an organization's protected resources. Businesses possess resources, applications and services that they need to protect, manage and monitor. Policies control the access permissions and usage of these resources by defining when and how a user can perform an action on a given resource. A policy, when applied to an object, defines the resources that a particular object can access.

NOTE An object is a principal. A *principal* can be an individual, a corporation, a role, or a group; anything that can have an identity. For more information, see the Java™ 2 Platform Standard Edition Javadocs.

A single policy can define either binary or non-binary decisions. A binary decision is *yes/no*, *true/false* or *allow/deny*. 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. In general, a policy is configured to define what an object can do to which resource and under what conditions.

Policy Management Feature

The Policy Management feature provides a *policy service* for creating and managing policies. The policy service allows administrators to define, modify, grant, revoke and delete permissions to protect resources within the Identity Server deployment. Typically, a policy service includes a data store, a library of interfaces that allows for the creation, administration and evaluation of policies, and a policy enforcer or *policy agent*. Identity Server uses Sun Java System Directory Server for data storage, and provides Java and C APIs for policy evaluation and policy service customization. (see the *Identity Server Developer's Guide* for more information) It also allows administrator to use the Identity Server console for policy management. Identity Server provides one policy service, the URL Policy Agent service, which uses downloadable policy agents to enforce the policies.

URL Policy Agent Service

Out of the box, Identity Server provides the URL Policy Agent service for policy enforcement. This service allows administrators to create and manage policies through a policy enforcer or *policy agent*.

Policy Agents

The Policy Agent is the Policy Enforcement Point (PEP) for a server on which an enterprise's resources are stored. The policy agent is installed separately from Identity Server onto a web server and serves as an additional authorization step when a user sends a request for a web resource that exists on the protected web server. This authorization is in addition to any user authorization request which the resource performs. The agent protects the web server, and in turn, the resource is protected by the authorization plug-in.

For example, a Human Resources web server protected by a remotely-installed Identity Server might have an agent installed on it. This agent would prevent personnel without the proper policy from viewing confidential salary information or other sensitive data. The policies are defined by the Identity Server administrator, stored within the Identity Server deployment and used by the policy agent to allow or deny users access to the remote web server's content.

The most current Sun Java System Identity Server Policy Agents can be downloaded from the Sun Microsystems Download Center.

More information on installing and administrating the policy agents can be found in the *Sun Java System Identity Server J2EE Policy Agents Guide* or *Web Policy Agents Guide*.

NOTE Policy is evaluated in no particular order although as they are evaluated, if one action value evaluates to *deny*, subsequent policies are not evaluated, unless the Continue Evaluation On Deny Decision attribute is enabled in the Policy Configuration service. For more information, see ["Policy Configuration Service Attributes"](#) on page 317.

Policy agents enforce decisions only on web URLs (<http://...>). However, agents can be written using the Java and C Policy Evaluation APIs to enforce policy on other resources.

In addition, the Resource Comparator attribute in the Policy Configuration Service would also need to be changed from its default configuration to:

```
serviceType=Name_of_LDAPService|class=com.sun.identity.policy.plugins.SuffixResourceName|
wildcard=*|delimiter=,|caseSensitive=false
```

Alternately, providing an implementation such as LDAPResourceName to implement com.sun.identity.policy.interfaces.ResourceName and configuring the Resource Comparator appropriately would also work.

NOTE The fields for the Resource Comparator attribute are explained in “Policy Configuration Service Attributes” on page 317.

The Policy Agent Process

The process for protected web resources begins when a web browser requests a URL that resides on a server protected by the policy agent. The server’s installed policy agent intercepts the request and checks for existing authentication credentials (a session token).

If the agent has intercepted a request and validated the existing session token, the following process is followed.

1. If the session token is valid, the user is allowed or denied access. If the token is invalid, the user is redirected to the Authentication Service, as outlined in the following steps.
2. The Authentication Service verifies that the credentials are also valid and issues a token.
3. Once the user’s credentials are properly authenticated, the agent issues a request to the Naming Service which defines the URLs used to access Identity Server’s internal services.
4. The Naming Service returns locators for the policy service, and the agent sends a request to the Policy Service to get policy decisions applicable to the user.
5. Based on the policy decisions for the resource being accessed, the user is either allowed or denied access. If advice on the policy decision indicates a different authentication level or authentication mechanism, the agent redirects the request to the Authentication Service until all criteria is validated.

Assuming the agent has intercepted a request for which there is no existing session token, the agent redirects the user to their default login page even if the resource is protected using a different authentication method.

NOTE Policy-based resource authorization and user authentication are separate types of authentication. More information on this can be found in [“Policy-Based Resource Management”](#) on page 140.

Policy Types

There are two types of policies 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*.

Rules

A *rule* contains a resource, one or more actions, and a value. The rule, basically, defines the policy.

- A *resource* defines the specific object that is being protected; for instance, an HTML page or a user's salary information accessed using a human resources service.
- An *action* is the name of an operation that can be performed on the resource; examples of web server actions are POST or GET. An allowable action for a human resources service might be `canChangeHomeTelephone`.
- A *value* defines the permission for the action, for example, allow or deny.

NOTE It is acceptable to define an action without resources.

Subjects

A *subject* defines the user or collection of users (for instance, a group or those who possess a specific role) that the policy affects. Subjects are assigned to policies. The general rule for subjects is that the policy would apply only if the user is a member of at least one subject in the policy. The default subjects are:

- Authenticated Users
- Identity Server Roles
- LDAP Groups
- LDAP Roles
- LDAP Users

- Organization
- Web Services Client

Identity Server Roles Versus LDAP Roles

An Identity Server role is created using Identity Server. These roles have object classes mandated by Identity Server. An LDAP role is any role definition that uses the Directory Server role capability. These roles have object classes mandated by Directory Server role definition. All Identity Server roles can be used as Directory Server roles. However, all Directory Server roles are not necessarily Identity Server roles. LDAP roles can be leveraged from an existing directory by configuring the [Policy Configuration Service](#). Identity Server roles can only be accessed through the hosting Identity Server Policy Service. Evaluating membership in Identity Server roles will be faster as it accesses the Identity Server SDK and cache. The LDAP Role Search filter can be modified in the Policy Configuration Service to narrow the scope and improve performance.

Nested Roles

Nested roles can be evaluated correctly as LDAP Roles in the subject of a policy definition.

Conditions

A condition 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. Or, 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.

The condition might additionally be used to configure different policies on different URIs on the same domain. For example,

`http://org.example.com/hr/*.jsp` can only be accessed by `org.example.net` from 9am to 5 p.m., yet `http://org.example.com/finance/*.jsp` can be accessed by `org.example2.net` from 5 a.m. to 11 p.m. This can be achieved by using an IP Condition along with a Time Condition. And specifying the rule resource as `http://org.example.com/hr/*.jsp`, the policy would apply to all the JSPs under `http://org.example.com/hr` including those in the sub directories.

NOTE	The terms referral, rule, resource, subject, condition, action and value correspond to the elements <i>Referral</i> , <i>Rule</i> , <i>ResourceName</i> , <i>Subject</i> , <i>Condition</i> , <i>Attribute</i> and <i>Value</i> in the <code>policy.dtd</code> .
-------------	--

Referral Policy

An administrator may need to delegate one organization's policy definitions and decisions to another organization. (Alternatively, 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*.

Rules

A rule defines the resource whose policy definition and evaluation is being referred.

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. See [“Creating Policies for Peer Organizations and Suborganizations” on page 128](#) for more information.

NOTE The referred-to organization can define or evaluate policies only for those resources (or sub-resources) that have been referred to it. This restriction, however, does not apply to the root organization.

Policy Definition Type Document

Once a policy is created and configured, it is stored in Directory Server in XML. In Directory Server, the XML-encoded data is stored in one place. Although policy is defined and configured using the `amadmin.dtd` (or the console), it is actually stored in Directory Server as XML that is based on the `policy.dtd`. The `policy.dtd` contains the policy element tags extracted from the `amadmin.dtd` (without the policy creation tags). So, when the Policy Service loads policies from Directory Server, it parses the XML based on the `policy.dtd`. The `amadmin.dtd` is only used when creating policy with the command line. This section describes the structure of `policy.dtd`. The `policy.dtd` exists in the following location:

`IdentityServer_base/SUNWam/dtd` (Solairs)

`IdentityServer_base/identity,dtd` (Linux)

NOTE Throughout the rest of this chapter, only the Solaris directory information will be given. Please note that the directory structure for Linux is different. For more information, please see ["About This Guide" on page 19](#).

Policy Element

Policy is the root element that defines the permissions or *rules* of a policy and to whom/what the rule applies or the *subject*. It also defines whether or not the policy is a *referral* (delegated) policy and whether there are any restrictions (or *conditions*) to the policy. It may contain one or more of the following sub-elements: *Rule*, *Conditions*, *Subjects*, or *Referrals*. The required XML attribute is *name* which specifies the name of the policy. The *referralPolicy* attribute identifies whether or not the policy is a referral policy; it defaults to a normal policy if not defined. Optional XML attributes include *name* and *description*.

NOTE When tagging a policy as *referral*, subjects and conditions are ignored during policy evaluation. Conversely, when tagging a policy as *normal*, any Referrals are ignored during policy evaluation.

Rule Element

The *Rule* element defines the specifics of the policy and can take three sub-elements: *ServiceName*, *ResourceName*, or *AttributeValuePair*. It defines the type of service or application for which the policy has been created as well as the resource name and the actions which are performed on it. A rule can be defined without any actions; for example, a referral policy rule doesn't have any actions.

NOTE It is acceptable to have a defined policy that does not include a defined *ResourceName* element.

ServiceName Element

The *ServiceName* element defines the name of the service to which the policy applies. This element represents the service type. It contains no other elements. The value is exactly as that defined in the service's XML file (based on the sms.dtd). The XML service attribute for the *ServiceName* element is the name of the service (which takes a string value).

ResourceName Element

The *ResourceName* element defines the object that will be acted upon. The policy has been specifically configured to protect this object. It contains no other elements. The XML service attribute for the *ResourceName* element is the name of the object. Examples of a *ResourceName* might be `http://www.sunone.com:8080/images` on a web server or `ldap://sunone.com:389/dc=example,dc=com` on a directory server. A more specific resource might be `salary://uid=jsmith,ou=people,dc=example,dc=com` where the object being acted upon is the salary information of John Smith.

AttributeValuePair Element

The *AttributeValuePair* element defines an action and its values. It is used as a sub-element to *Subject Element*, *Referral Element* and *Condition Element*. It contains both the *Attribute* and *Value* elements and no XML service attributes.

Attribute Element

The *Attribute* element defines the name of the action. An action is an operation or event that is performed on a resource. POST or GET are actions performed on web server resources, READ or SEARCH are actions performed on directory server resources. The *Attribute* element must be paired with a *Value* element. The *Attribute* element itself contains no other elements. The XML service attribute for the *Attribute* element is the name of the action.

Value Element

The *Value* element defines the action values. Allow/deny or yes/no are examples of action values. Other action values can be either boolean, numeric, or strings. The values are defined in the service's XML file (based on the `sms.dtd`). The *Value* element contains no other elements and it contains no XML service attributes.

CAUTION Deny rules always take precedence over allow rules. For example, if one policy denies access and another allows it, the result is a deny (provided all other conditions for both policies are met). It is recommended that deny policies be used with extreme caution as they can lead to potential conflicts. If explicit deny rules are used, policies assigned to a user through different subjects (such as role and/or group membership) may result in denied access. Typically, the policy definition process should only use allow rules. The default deny may be used when no other policies apply.

Subjects Element

The *Subjects* sub-element identifies a collection of objects to which the policy applies; this overview collection is chosen based on membership in a group, ownership of a role or individual users. It takes the *Subject* sub-element. The XML attributes that can be defined are:

name. This defines a name for the collection.

description. This defines a description of the subject

includeType. This is not currently used.

Subject Element

The *Subject* sub-element identifies a collection of objects to which the policy applies; this collection pinpoints more specific objects from the collection defined by the *Subjects* element. Membership can be based on roles, group membership or simply a listing of individual users. It contains a sub-element, the [AttributeValuePair Element](#). The required XML attribute is `type`, which identifies a generic collection of objects from which the specifically defined subjects are taken. Other XML attributes include `name` which defines a name for the collection and `includeType` which defines whether the collection is as defined, for whether the policy applies to users who are NOT members of the subject.

NOTE When multiple subjects are defined, at least one of the subjects should apply to the user for the policy to apply. When a subject is defined using `includeType` set to `false`, the user should not be a member of that subject.

Referrals Element

The *Referrals* sub-element identifies a collection of policy referrals. It takes the *Referral* sub-element. The XML attributes it can be defined with are `name` which defines a name for the collection and `description` which takes a description.

Referral Element

The *Referral* sub-element identifies a specific policy referral. It takes as a sub-element the [AttributeValuePair Element](#). Its required XML attribute is `type` which identifies a generic collection of assignments from which the specifically defined referrals are taken. It can also include the `name` attribute which defines a name for the collection.

Conditions Element

The *Conditions* sub-element identifies a collection of policy restrictions (time range, authentication level, et.al.). It must contain one or more of the *Condition* sub-element. The XML attributes it can be defined with are `name` which defines a name for the collection and `description` which takes a description.

NOTE The conditions element is an optional element in a policy.

Condition Element

The *Condition* sub-element identifies a specific policy restriction (time range, authentication level, et.al.). It takes as a sub-element the [AttributeValuePair Element](#). Its required XML attribute is `type` which identifies a generic collection of restrictions from which the specifically defined conditions are taken. It can also include the `name` attribute which defines a name for the collection.

Adding a Policy Service

By default, Identity Server provides the URL Policy Agent service (`iPlanetAMWebAgentService`). This service is defined in an XML file located in the following directory:

```
etc/opt/SUNWam/config/xml/
```

You can, however add additional policy services to Identity Server. Once the policy service is created, you add it to Identity Server through the `amadmin` command line utility.

To Add a New Policy Service

1. Develop the new policy service in an XML file based on the `sms.dtd`. Identity Server provides two policy service XML files that you may wish to use as the basis for the new policy service file:

`amWebAgent.xml` - This is the XML file for the default URL Policy Agent service. It is located in `etc/opt/SUNWam/config/xml/`.

`SampleWebService.xml` - This is the sample policy service file located in `etc/opt/SUNWam/samples/policy`.

2. Save the XML file to the directory from which you will load the new policy service. For example:

```
etc/opt/SUNWam/config/xml/newPolicyService.xml
```

3. Load the new policy service with the `amadmin` command line utility. For example:

```
IdentityServer_base/SUNWam/bin/amadmin
--runasdn "uid=amAdmin,ou=People,default_org,root_suffix
--password password
--schema etc/opt/SUNWam/config/xml/newPolicyService.xml
```

4. After you load the new policy service, you can define rules for the policy definitions through the Identity Server console or by loading a new policy through `amadmin`.

Creating Policies

You can create, modify and delete policies through the Policy API and the Identity Server console, and create and delete policies through the `amadmin` command line tool. This section focuses on creating policies through the `amadmin` command line utility and through the Identity Server console. For more information on the Policy APIs, see the *Identity Server Developer's Guide*.

Policies are generally created through an XML file and added to Identity Server through the `amadmin` command line utility and then managed through the Identity Server console (although policies can be created through the console). This is because policies cannot be modified using `amadmin` directly. To modify a policy, you must first delete the policy from Identity Server and then add the modified policy using `amadmin`.

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

Creating Policies With amadmin

1. Create the policy's XML file based on the `policy.dtd`. This file is located in the following directory:

```
IdentityServer_base/SUNWam/dtd
```

2. Once the policy's XML file is developed, you can use the following command to load it:

```
IdentityServer_base/SUNWam/bin/amadmin
```

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

To add multiple policies simultaneously, place the policies in one XML file, as opposed to having one policy in each XML file. If you load policies with multiple XML files in quick succession, the internal policy index may become corrupted, and some policies may not participate in policy evaluation.

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 organizations', 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.

Please note that in the text of Value elements in `SubOrgReferral`, `PeerOrgReferral`, `Organization subject`, `IdentityServerRoles subject`, `LDAPGroups subject`, `LDAPRoles subject` and `LDAPUsers subject` need to be the full DN.

To Create Policies With the Identity Server Console

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 frame. 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 that suborganization.
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.
6. Type a name for the policy and click OK.
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 Save to complete the policy's configuration.

Creating Policies for Peer Organizations 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. Also, the Policy Configuration service should be registered and the template created in the suborganizations. 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).

In this example, `o=isp` is the parent organization, `o=example.com` is the suborganization and manages resources and sub-resources of `http://www.example.com`.

To Create a Policy for a Suborganization

1. Create a referral policy at `o=isp`. For information on referral policies, see the procedure [“Modifying a Referral Policy” on page 136](#).

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

2. Go to the Organization view and navigate to the suborganization `example.com`.
3. Ensure that the policy configuration service is registered at the suborganization level, `example.com`. For information, see [“Adding Policy Configuration Services” on page 139](#).
4. Now that the resource is referred to `sun.com` by `isp`, normal policies can be created for the resource `http://www.example.com`, or for any resource starting with `http://www.example.com`.

See the procedure [“Modifying a Normal Policy” on page 129](#) for information on creating normal policies.

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

Managing Policies

Once a normal or referral policy is created and added to Identity Server, you can manage the policy through the Identity Server console by modifying the rules, subjects, conditions and referrals.

Modifying 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.

To Modify Rules

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 frame.

By default, the General view is displayed. The attributes contained in the General view are described in [“Creating Policies” on page 126](#).

3. Select Rules from the View menu and click New.

If more than one service exists, they will be listed in the Data pane. Choose the service for which you wish to create a policy and click Next. The New Rule window is displayed.

4. Define the resource, actions and action values in the Rules fields. The fields are:

Type. Displays the service for the policy to be created. The default is URL Policy Agent.

Rule Name. Enter the name of the rule.

Resource Name. Enter the name of a resource. For example:

`http://www.example.com`

Currently, Policy Agents only support `http://` and `https://` resources and do not support IP addresses in place of the hostname.

Wildcards are supported for resource names, port number and protocol. For example:

`http*://*:*/*.*.html`

For the URL Policy Agent service, if a port number is not entered, the default port number is 80 for `http://`, and 443 for `https://`.

To allow the management of resource for all servers installed on a specific machine, you can define the resource as `http://host*:*.*`. Additionally, you can define the following resource to grant an administrator to a specific organization authority for all of the services in that organization:

`http://*.subdomain.domain.topleveldomain`

Select Actions. For the URL Policy Agent Service, you can select either or both of the following default actions:

- 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.

Denial rules always take precedence over allow rules in a policy. For example, if you have two policies for a given resource, one denying access and the other allowing access, the result is a deny access (provided that the conditions for both policies are met). It is recommended that deny policies be used with extreme caution as they may lead to potential conflicts between the policies. Typically, the policy definition process should only use allow rules, and use the default deny when no policies apply to accomplish the deny case.

If explicit deny rules are used, policies that are assigned to a given user through different subjects (such as role and/or group membership) may result in denied access to a resource even if one or more of the policies allow access. For example, if there is a deny policy for a resource applicable to an Employee role and there is another allow policy for the same resource applicable to Manager role, policy decisions for users assigned both Employee and Manager roles would be denied.

One way to resolve such problems is to design policies using Condition plug-ins. In the case above, a “role condition” that applies the deny policy to users authenticated to the Employee role and applies the allow policy to users authenticated to the Manager role helps differentiate the two policies. Another way could be to use the `authentication level` condition, where the Manager role authenticates at a higher authentication level. See [“To Add or Modify Conditions” on page 134](#) for more information.

NOTE

If the 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 Finish to save the rule. This only saves the configuration in memory. Follow step 7 to complete the process.
6. Repeat steps 1 through 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.

To Modify Subjects

1. To define the subject for the policy, select Subject from the View menu and click New.
2. Select one of the default subject types:

Authenticated Users. This subject type implies that any user with a valid SSO Token is a member of this subject.

Identity Server Roles. This subject type implies that any member of an Identity Server role is a member of this subject. An Identity Server role is created using Identity Server. These roles have object classes mandated by Identity Server. Identity Server roles can only be accessed through the hosting Identity Server Policy Service.

LDAP Groups. This subject type implies that any member of an LDAP group is member of this subject.

LDAP Roles. This subject type implies that any member of an LDAP role is a member of this subject. An LDAP Role is any role definition that uses the Directory Server role capability. These roles have object classes mandated by Directory Server role definition. The LDAP Role Search filter can be modified in the Policy Configuration Service to narrow the scope and improve performance.

LDAP Users. This subject type implies that any LDAP user is a member of this subject.

Organization. This subject type implies that any member of an organization is a member of this subject.

Web Services Client. This subject type implies that a web service client (WSC) identified by the SSO Token is a member of this subject, if the DN of any principal contained in the SSO Token matches any selected value of this subject. Valid values are the DNs of trusted certificates in the local JKS keystore, which correspond to the certificates of trusted WSCs. This subject has dependency on the Liberty Web Services Framework and should be used only by Liberty Service Providers to authorize WSCs.

Make sure that you have created the keystore before you add this Subject to a policy. Information on setting up the keystore can be found in the following location:

`IdentityServer_base/SUNWam/samples/saml/xmlsig/keytool.html`

Click Next to continue.

3. Enter a name for the subject.

4. Select or deselect the Exclusive field.

If this field is not selected (default), the policy applies to the identity that is a member of the subject. If the field is selected, the policy applies the identity that is not a member of the subject.

If multiple subjects exist in the policy, the policy applies to the identity when at least one of the subjects implies that the policy applies to the given identity.

5. Perform a search in order to display the identities to add to the subject. This step is not applicable for the Authenticated Users subject.

The default (*) search pattern will display all qualified entries.

6. Select the individual identities you wish to add for the subject, or click Add All to add all of the identities at once. Click Add to move the identities to the Select List Box.

7. Click Finish.

8. The subject's names, type and exclusive status are displayed in the table in the Subjects view. Click Save.

To remove a subject from a policy, select the subject and click Delete, then Save.

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

To Add or Modify Conditions

1. Select Conditions from the View menu. Click New 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
 - LE Authentication Level
 - Session

- Time

For Authentication Level, the policy applies if the user's authentication level is greater than or equal to the Authentication level set in the condition. For LE Authentication Level, the policy applies if the user's authentication level is less than or equal to the Authentication level set in the condition

3. Click Next.
4. Define the values for a given condition. The fields are:

Name. Enter the name of the condition.

Authentication Level

Authentication level. Indicate the level of trust for authentication. The available authentication levels are displayed in the authentication level and authentication module table.

Authentication Scheme

Authentication scheme. Choose the authentication scheme for the condition from the pull-down menu. These authentication schemes are taken from the Core service template in the organization authentication modules.

IP Address

IP Address From/To. Specifies the range of the IP address.

DNS Name. Specifies the DNS name. This field can be a fully qualified hostname or a string in one of the following formats:

domainname

**.domainname*

Time

Date From/To. Specifies the range of the date.

Time. Specifies the range of time within a day.

Day. Specifies a range of days.

Timezone. Specifies a timezone, either standard or custom. Custom timezones can only be a timezone ID recognized by Java (for example, PST). If no value is specified, the default value is the Timezone set in the Identity Server JVM.

Session

Max Session Time. Specifies the maximum user session time during which a policy applies.

Terminate Session. If selected, the user session will be terminated if the session time exceeds the maximum allowed as defined in the Max Session Time field.

5. Once you have defined the condition, click Finish.

All of the conditions created for that policy are displayed in the table in the Conditions view.

6. Click Save.

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

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

Modifying 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.

To Modify Rules

1. Select Rules from the View menu. Click New to add a new rule, or click the Edit link to edit an existing rule.
2. Select the Service Type. Click Next if you are creating a new rule.
3. Define the resource in the Rules fields. The fields are:

Type. Displays the policy service for the policy to be created.

Rule Name. Enter the name of the rule.

Resource Name. Enter the name of a resource. For example:

`http://www.sunone.com`

Currently, Policy Agents only support `http://` and `https://` resources and do not support IP addresses in place of the hostname.

Wildcards are supported for resource names, port number and protocol.

For the URL Policy Agent service, if a port number is not entered, the default port number is 80 for `http://`, and 443 for `https://`.

To allow the management of resource for all servers installed on a specific machine, you can define the resource as `http://host*:*`. Additionally, you can define the following resource to grant an administrator to a specific organization authority for all of the services in that organization:

```
http://*.subdomain.domain.topleveldomain
```

4. Click Finish.
5. Repeat steps 1 through 4 to create additional rules.

All of the rules created for that policy are displayed in the table in the Rules view.

6. Click Save.

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

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

To Add Referrals

1. Select Referrals from the View menu. Click New 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 type.

Name. Enter the name of the referral.

Containing. Specifies a filter for the organization names that will be displayed in the Value field. By default, it will display all organization names.

Value. Select the organization name of the referral.

3. Click OK and Save.

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

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

Policy Configuration Service

The Policy Configuration service is used to configure policy-related attributes for each organization through the Identity Server console. You can also define resource name implementations and Directory Server data stores for use with the Identity Server Authentication service.

Caching Subject Evaluations

To improve policy evaluation performance, subject evaluations are cached for a period of minutes as defined by the Subjects Result Time To Live attribute in the Policy Configuration service. These cached policy decisions are referred to until the time defined in the Subjects Result Time To Live attribute has elapsed. Once this time has been reached, the next time the policy is evaluated its decision would reflect the user's changed state, if applicable (for example, if the user has been removed from a group).

amldapuser Definition

amldapuser is a user created during installation that is used to bind and search Directory Server during LDAP and Membership authentication. It is also used in the Policy Configuration service. Once the LDAP, Membership or Policy Configuration services are registered to an organization, the password for this user (configured during installation) must be entered. For more information, see the *Sun Java System Identity Server Migration Guide* .

Adding Policy Configuration Services

Adding a policy configuration service is the same as adding any type of service; it is done within the Identity Management interface. By default, the Policy Configuration service is automatically added to the top-level organization. Any policy service you create must be added to all organizations. Whenever you add the policy configuration service, you must enter the LDAP bind password in the template.

To Add the 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 frame.

4. Click Add in the Navigation frame.

A listing of services not yet registered to this organization is displayed in the Data frame.

5. From the Add Services window, opened in the Data frame, choose Policy Configuration and click OK.

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

6. Click the Properties arrow to configure the policy service.

- a. If the policy template has not yet been configured, you will need to create a service template for the newly registered policy service.
- b. To configure the policy service, click Create.
- c. Modify the Policy Configuration attributes. See [“Policy Configuration Service Attributes”](#) on page 317 for a description of these attributes.

7. Click Save.

The policy configuration service is now added 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`.

Policy-Based Resource Management

Some organizations require an advanced authentication scenario where a user authenticates against a particular module based on the resource that they are attempting to access. Policy-based resource management is a function of Identity Server in which the user does not need to pass their default authentication module in order to access a web resource.

Limitations

Policy-based resource management contains the following limitations:

1. All policies applicable to the resource require the same authentication scheme or level of authentication. For example, if `abc.html` is defined in a policy for the LDAP authentication module, it can not be defined in a policy for the Certificate-based authentication module.
2. Level and scheme are the only conditions that can be defined for this policy.
3. This feature does not work across different DNS domains.

To Configure Policy-based Resource Management

Once both Identity Server and a policy agent have been installed, policy-based resource management can be configured. To do this, it is necessary to point Identity Server to the Gateway servlet.

1. Open `AMAgent.properties`.

`AMAgent.properties` can be found (in a Solaris environment) in `/etc/opt/SUNWam/agents/config/`.

2. Comment out the following line:

```
#com.sun.am.policy.am.loginURL =  
http://identity_server_host.domain_name:port/amserver/UI/Login.
```

3. Add the following line to the file:

```
com.sun.am.policy.am.loginURL =  
http://identity_server_host.domain_name:port/amserver/gateway
```

4. Restart the agent.

Authentication Options

Sun Java™ System Identity Server 2004Q2 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 *Identity Server Developer's Guide*.)

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

- [“Core Authentication” on page 144](#)
- [“Anonymous Authentication” on page 145](#)
- [“Certificate-based Authentication” on page 146](#)
- [“HTTP Basic Authentication” on page 148](#)
- [“LDAP Directory Authentication” on page 150](#)
- [“Membership Authentication” on page 152](#)
- [“NT Authentication” on page 154](#)
- [“RADIUS Server Authentication” on page 156](#)
- [“SafeWord Authentication” on page 158](#)
- [“SecurID Authentication” on page 161](#)
- [“Unix Authentication” on page 163](#)
- [“Windows Desktop SSO Authentication” on page 165](#)

- “Authentication Configuration” on page 168
- “Auth Level Authentication” on page 174
- “Module Based Authentication” on page 175
- “URL Redirection” on page 175

Core Authentication

Identity Server provides, by default, eleven different authentication services, as well as a Core authentication service. The Core authentication service provides overall configuration for the authentication service. Before adding and enabling Anonymous, Certificate-based, HTTP Basic, LDAP, Membership, NT, RADIUS, SafeWord, SecurID, Windows Desktop SSO and Unix authentication, the Core authentication must be added and enabled. Both the Core and LDAP Authentication services are automatically enabled for the default organization. [Chapter 20, “Core Authentication Attributes”](#) contains a detailed listing of the Core attributes.

Adding and Enabling the Core Service

1. Go to the organization for which the Core service is to be added.
2. Choose Services from the View menu.
3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

5. Click the Core Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 20, “Core Authentication Attributes”](#) or by clicking the Help link in the upper right 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 233](#)). 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.

Adding and Enabling Anonymous Authentication

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

1. Go to the organization for which Anonymous Authentication is to be added.
2. Choose Services from the View menu.

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

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

5. Click the Anonymous Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 18, “Anonymous Authentication Attributes”](#) or by clicking the Help link in the upper right 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 246](#) must be modified to enable and select Anonymous Authentication. This ensures that when the user logs in using

`http(s)://hostname:port/SERVER_DEPLOY_URI/UI/Login?module=Anonymous&org=org_name`. To login without the Anonymous Authentication login window, use the following syntax:

```
http(s)://hostname:port/SERVER_DEPLOY_URI/UI/Login?module=Anonymous&org=org_name&Login.Token1=user_id
```

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 `anonymous`. 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. This can optionally be case sensitive.

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 adding the Certificate-based Authentication service to an organization. First, the web container 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 6, “Using Certificates and Keys” in the *Sun ONE Web Server 6.1 Administrator’s Guide* for these initial Web Server configuration steps. This document can be found at the following location:

<http://docs.sun.com/db/prod/slwebsrv#hic>

Or, see the *Sun ONE Application Sever Administrator’s Guide to Security* at the following location:

<http://docs.sun.com/db/prod/slappsrv#hic>

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

Adding and Enabling Certificate-based Authentication

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

1. Go to the organization for which Certificate-based Authentication is to be added.
2. Choose Services from the View menu.

The Core service, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Certificate-based Authentication service.

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

4. Select the checkbox for Certificate-based Authentication and click Add.

The Certificate-based Authentication service will appear in the Navigation pane assuring the administrator that it has been added.

5. Click the Certificate-based Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 19, “Certificate Authentication Attributes”](#) or by clicking the Help link in the upper right corner of the console.

7. Click Save.

Adding a Server URL in Platform Server List for Certificate-based Authentication

In order to add this service, you must log in to Identity Server as the Organization Administrator and have Identity Server and the web container configured for SSL and with client authentication enabled. For more information, see [“Configuring Identity Server in SSL Mode” on page 57](#).

Logging In Using Certificate-based Authentication

In order to make certificate-based authentication the default authentication method, the Core Authentication service attribute [Organization Authentication Modules](#) (see [page 246](#)) must be modified. This ensures that when the user logs in using `https://hostname:port/deploy_URI/UI/Login?module=Cert`, the user 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.

HTTP Basic Authentication

This module uses basic authentication, which is the HTTP protocol’s built-in authentication support. The web server issues a client request for username and password, and sends that information back to the server as part of the authorized request. Identity Server retrieves the username and password and then internally authenticates the user to the LDAP authentication module. In order for HTTP Basic to function correctly, the LDAP authentication module must be added (adding the

HTTP Basic module alone will not work). For more information, see [“Adding and Enabling LDAP Authentication” on page 150](#). Once the user successfully authenticates, he/she will be able to re-authenticate without being prompted for username and password.

Adding and Enabling HTTP Basic Authentication

You must log in to Identity Server as the Organization Administrator or Top-Level Administrator and have the LDAP authentication service already registered.

1. Go to the organization for which HTTP Basic Authentication is to be added.
2. Choose Services from the View menu.

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

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

4. Select the checkbox for HTTP Basic Authentication and click Add.

The HTTP Basic Authentication service will appear in the Navigation pane assuring the administrator that it has been added.

5. Click the HTTP Basic Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.

6. Click Create.

The HTTP Basic Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in [Chapter 21, “HTTP Basic Authentication Attributes”](#) or by clicking the Help link in the upper right corner of the console.

7. Click Save.

The HTTP Basic Authentication service has been enabled.

Logging In Using HTTP Basic Authentication

In order to log in using LDAP Authentication, the Core Authentication service attribute “[Organization Authentication Modules](#)” on [page 246](#) must be modified to enable and select HTTP Basic authentication. This ensures that when the user logs in using

`http://hostname:port/server_deploy_URI/UI/Login?module=HTTPBasic`, the user will see the 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. If authentication fails, a new instance should be opened and the user should login again. To logout completely after using HTTP Basic authentication, all of the existing browser instances must be closed, and a new browser instance must be started.

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. Both the Core and LDAP Authentication services are automatically enabled for the default organization. The following instructions are provided in the event that the service is disabled.

Adding and Enabling LDAP Authentication

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

1. Go to the organization for which LDAP Authentication is to be added.
2. Choose Services from the View menu.

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

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

5. Click the LDAP Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 22, “LDAP Authentication Attributes”](#) or by clicking the Help link in the upper right corner of the console.

7. Enter the password in the Password for User Bind attribute. By default, the `amldapuser` password that was entered during installation is used as the bind user. If your Directory Server allows anonymous access to read user entries, you can skip this step.

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 246 must be modified to enable and select LDAP Authentication. This ensures that when the user logs in using `http://hostname:port/server_deploy_URI/UI/Login?module=LDAP`, the user 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” on page 258](#) and [“Secondary LDAP Server” on page 258](#).

Multiple LDAP Configuration

As a form of failover or to configure multiple values for an attribute when the Identity Server console only provides one value field, an administrator can define multiple LDAP configurations under one organization. Although these additional configurations are not visible from the console, they work in conjunction with the primary configuration if an initial search for the requesting user’s authorization is not found. For information on multiple LDAP configuration, see [“Multi LDAP Configuration” in the *Identity Server Developer’s Guide*](#).

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 added user. The user can also access the viewer interface, saved on the user profile database as authorization data and user preferences.

Adding and Enabling Membership Authentication

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

1. Go to the organization for which Membership Authentication is to be added.
2. Choose Services from the View menu.

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

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

5. Click the Membership Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 23, “Membership Authentication Attributes”](#) or by selecting the Help link in the upper right 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 246](#) must be modified to enable and select Membership Authentication. This ensures that when the user logs in using

`http://hostname:port/deploy_URI/UI/Login?module=Membership`, (note case sensitivity) the user 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 Authentication service is only supported on the Solaris platform.

1. Configure the NT server. For detailed instructions, see the NT server documentation.
2. Before you can add 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 269](#).
3. Add and enable the NT authentication service.

Installing the Samba Client

In order to activate the NT Authentication module, Samba Client 2.2.2 must be downloaded and installed to the following directory:

```
IdentityServer_base/SUNWam/bin
```

Samba Client is a file and print server for blending Windows and UNIX machines together without requiring a separate Windows NT/2000 Server. More information, and the download itself, can be accessed at <http://www.sun.com/software/download/products/3e3af224.html>.

Red Hat Linux ships with a Samba client, located in the following directory:

```
/usr/bin
```

In order to authenticate using the NT Authentication service for Linux, copy the client binary to the following Identity Server directory:

```
IdentityServer_base/sun/identity/bin
```

NOTE If you have multiple interfaces, extra configuration is required. Multiple interfaces can be set by configuration in the `smb.conf` file so it passes to the `mbclient`.

Adding and Enabling NT Authentication

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

1. Go to the organization for which NT Authentication is to be added.

2. Choose Services from the View menu.

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

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

5. Click the NT Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 24, “NT Authentication Attributes”](#) or by selecting the Help link in the upper right 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 246](#) must be modified to enable and select NT Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=NT`, the user 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.

Adding and Enabling RADIUS Authentication

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

1. Go to the organization for which RADIUS Authentication is to be added.
2. Choose Services from the View menu.

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

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

5. Click the RADIUS Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 25, “RADIUS Authentication Attributes”](#) or by selecting the Help link in the upper right 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 246](#) must be modified to enable and select RADIUS Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=RADIUS`, the user 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.

Configuring RADUIS with Sun ONE Application Server

When the RADUIS client forms a socket connection to its server, by default, only the connect permission of the SocketPermissions is allowed in the Application Server’s `server.policy` file. In order for RADUIS authentication to work correctly, permissions need to be granted for the following actions:

- accept
- connect
- listen
- resolve

To grant a permission for a socket connection, you must add an entry into Application Server’s `server.policy` file. A SocketPermission consists of a host specification and a set of actions specifying ways to connect to that host. The host is specified as the following:

```
host = hostname | IPaddress:portrange:portrange = portnumber |
-portnumberportnumber-portnumber
```

The host is expressed as a DNS name, as a numerical IP address, or as localhost (for the local machine). The wildcard “*” may be included once in a DNS name host specification. If it is included, it must be in the left-most position, as in `*.example.com`.

The port (or portrange) is optional. A port specification of the form $N-$, where N is a port number, signifies all ports numbered N and above. A specification of the form $-N$ indicates all ports numbered N and below.

The `listen` action is only meaningful when used with a `localhost`. The `resolve` (resolve host/IP name service lookups) action is implied when any of the other actions are present.

For example, when creating `SocketPermissions`, note that if the following permission is granted to some code, it allows that code to connect to `port 1645` on `machine1.example.com`, and to accept connections on that port:

```
permission java.net.SocketPermission machine1.example.com:1645,
"connect,accept";
```

Similarly, if the following permission is granted to some code, it allows that code to accept connections on, connect to, or listen to any port between 1024 and 65535 on the local host:

```
permission java.net.SocketPermission "machine1.example.com:1645",
"connect,accept";

permission java.net.SocketPermission "localhost:1024-",
"accept,connect,listen";
```

NOTE Granting code permission to accept or make connections to remote hosts may cause problems, because malevolent code can then more easily transfer and share confidential data among parties who may not otherwise have access to the data. Make sure to give only appropriate permissions by specifying exact port number instead of allowing a range of port numbers

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.

Adding and Enabling SafeWord Authentication

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

1. Go to the organization for which SafeWord Authentication is to be added.
2. Choose Services from the View menu.

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

3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

5. Click the SafeWord Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 26, “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 246](#) must be modified to enable and select SafeWord Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=SAFEWORD`, the

user 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.

Configuring SafeWord with Sun ONE Application Server

When the SafeWord client forms a socket connection to its server, by default, only the `connect` permission of the `SocketPermissions` is allowed in the Application Server's `server.policy` file. In order for SafeWord authentication to work correctly, permissions need to be granted for the following actions:

- `accept`
- `connect`
- `listen`
- `resolve`

To grant a permission for a socket connection, you must add an entry into Application Server's `server.policy` file. A `SocketPermission` consists of a host specification and a set of actions specifying ways to connect to that host. The host is specified as the following:

```
host = (hostname | IPaddress)[:portrange] portrange = portnumber |
-portnumberportnumber-[portnumber]
```

The host is expressed as a DNS name, as a numerical IP address, or as `localhost` (for the local machine). The wildcard "*" may be included once in a DNS name host specification. If it is included, it must be in the left-most position, as in `*.example.com`.

The port (or portrange) is optional. A port specification of the form `N-`, where `N` is a port number, signifies all ports numbered `N` and above. A specification of the form `-N` indicates all ports numbered `N` and below.

The `listen` action is only meaningful when used with a `localhost`. The `resolve` (resolve host/IP name service lookups) action is implied when any of the other actions are present.

For example, when creating `SocketPermissions`, note that if the following permission is granted to some code, it allows that code to connect to port 1645 on `machine1.example.com`, and to accept connections on that port:

```
permission java.net.SocketPermission machine1.example.com:5030,
"connect,accept";
```

Similarly, if the following permission is granted to some code, it allows that code to accept connections on, connect to, or listen to any port between 1024 and 65535 on the local host:

```
permission java.net.SocketPermission "machine1.example.com:5030",
"connect,accept";

permission java.net.SocketPermission "localhost:1024-",
"accept,connect,listen";
```

NOTE Granting code permission to accept or make connections to remote hosts may cause problems, because malevolent code can then more easily transfer and share confidential data among parties who may not otherwise have access to the data. Make sure to give only appropriate permissions by specifying exact port number instead of allowing a range of port numbers

SecurID Authentication

Identity Server can be configured to handle SecureID Authentication requests to RSA's ACE/Server authentication servers. Identity Server provides the client portion of SecurID authentication. The ACE/Server may exist on the system on which Identity Server is installed, or on a separate system. In order to authenticate locally-administered userids (see `admintool (1M)`), root access is required.

SecurID Authentication makes use of an authentication *helper*, `amsecuridd`, which is a separate process from the main Identity Server process. Upon startup, this helper listens on a port for configuration information. If Identity Server is installed to run as `nobody`, or a userid other than root, then the `IdentityServer_base/SUNWam/share/bin/amsecuridd` process must still execute as root. For more information on the `amsecuridd` helper, see [“The amsecuridd Helper” on page 209](#).

NOTE For this release of Identity Server, the SecurID Authentication service is not available for the Linux or Solaris x86 platforms.

Adding and Enabling SecurID Authentication

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

1. Go to the organization for which SecurID Authentication is to be added.
2. Choose Services from the View menu.

The Core service, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the SecurID Authentication service.
3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.
4. Select the checkbox for SecurID Authentication and click Add.

The SecurID Authentication service will appear in the Navigation pane, assuring the administrator that it has been added.
5. Click the SecurID Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

The SecurID Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in [Chapter 27, “SecurID Authentication Attributes”](#), or by clicking the Help link on the upper right corner of the console.
7. Click Save.

The SecurID Authentication service has been enabled.

Logging In Using SecurID Authentication

In order to log in using SecurID Authentication, the Core Authentication service attribute [“Organization Authentication Modules” on page 246](#) must be modified to enable and select SecurID Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=SecurID`, the

user will see the SecurID 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 or Linux 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.

Unix Authentication makes use of an authentication *helper*, `amunixd`, 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.

If Identity Server is installed to run as `nobody`, or a userid other than root, then the `IdentityServer_base/SUNWam/share/bin/amunixd` process must still execute as root. The Unix authentication module invokes the `amunixd` daemon by opening a socket to `localhost:58946` to listen for Unix authentication requests. To run the `amunixd` helper process on the default port, enter the following command:

```
./amunixd
```

To run `amunixd` on a non-default port, enter the following command:

```
./amunixd [-c portnm] [ipaddress]
```

The `ipaddress` and `portnumber` is located in the `UnixHelper.ipadr`s (in IPV4 format) and `UnixHelper.port` attributes in `AMConfig.properties`. You can run `amunixd` through the `amserver` command line utility (`amserver` runs the process automatically, retrieving the `portnumber` and `ipaddress` from `AMConfig.properties`).

The `passwd` entry in the `/etc/nsswitch.conf` file determines whether the `/etc/passwd` and `/etc/shadow` files, or NIS are consulted for authentication.

Adding and Enabling 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 28, "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. Go to the organization for which Unix Authentication is to be added.

5. Choose Services from the View menu.

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

6. Click Add in the Navigation pane.

A list of available services displays in the Data pane.

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

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

8. Click the Unix Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* 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 28, "Unix Authentication Attributes"](#), or by clicking the Help link in the upper right corner of the console.

10. Click Save. The Unix Authentication service is enabled.

Logging In Using Unix Authentication

In order to log in using Unix Authentication, the Core Authentication service attribute “[Organization Authentication Modules](#)” on [page 246](#) must be modified to enable and select 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.

Windows Desktop SSO Authentication

The Windows Desktop SSO Authentication service is a Kerberos-based authentication plug-in module used for Windows 2000™. It allows a user who has already authenticated to a Kerberos Distribution Center (KDC) to authenticate to Identity Server without re-submitting the login criteria (Single Sign-on).

Adding and Enabling Windows Desktop SSO Authentication

Enabling Windows Desktop SSO Authentication is a three-step process:

1. Create a User in the Windows 2000 Domain Controller.
2. Setup Internet Explorer.
3. Add and Configure the Windows Desktop SSO Authentication service.

To Create a User in the Windows 2000 Domain Controller

1. In the domain controller, create a user account for the Identity Server authentication service.
 - a. From the Start menu, go to Programs>Administration Tools.
 - b. Select Active Directory and Computers.
 - c. Create a new user with the Identity Server host name as the User ID (login name). The Identity Server host name should not include the domain name.

2. Associate the user account with a service provider name and export the keytab files to the system in which Identity Server is installed. To do so, run the following commands:

```
ktpass -princ host/hostname.domainname@DCDOMAIN -pass password -mapuser  
userName-out hostname.host.keytab
```

```
ktpass -princ HTTP/hostname.domainname@DCDOMAIN -pass password -mapuser  
userName-out hostname.host.keytab
```

The `ktpass` command accepts the following parameters:

hostname. The host name (without the domain name) on which Identity Server runs.

domainname. The Identity Server domain name.

DCDOMAIN. The domain name of the domain controller. This may be different from the Identity Server domain name.

password. The password of the user account. Make sure that password is correct, as `ktpass` does not verify passwords.

userName. The user account ID. This should be the same as `hostname`.

NOTE Make sure that both keytab files are kept secure.

3. Restart the server.

To Set Up Internet Explorer

1. In the Tool menu, go to Internet Options>Advanced/Security>Security.
2. Select the Integrated Windows Authentication option.
3. Go to Security>Local Internet.
 - a. Select Custom Level. In the User Authentication/Logon panel, select the Automatic Logon Only in Intranet Zone option.
 - b. Go to Sites and select all of the options.
 - c. Click Advanced and add the Identity Server to the local zone (if it is not added already).

NOTE These steps apply to Microsoft Internet Explorer™ 6 and later. If you are using an earlier version, make sure that Identity Server is in the browser's internet zone and enable Native Windows Authentication.

To Add and Configure Windows Desktop SSO Authentication

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

1. Go to the organization for which Windows Desktop SSO Authentication is to be added.
2. Choose Services from the View menu.

The Core service, if already added, displays in the Navigation pane. If it is not already added, it can be done concurrently with the Windows Desktop SSO Authentication service.
3. Click Add in the Navigation pane.

A list of available services displays in the Data pane.
4. Select the checkbox for Windows Desktop SSO Authentication and click Add.

The Windows Desktop SSO Authentication service will appear in the Navigation pane assuring the administrator that it has been added.
5. Click the Windows Desktop SSO Authentication Properties arrow.

The message *A template does not currently exist for this service. Do you want to create one now?* appears in the Data pane.
6. Click Create.

The Windows Desktop SSO Authentication attributes appear in the Data pane. Modify the attributes as necessary. An explanation of these attributes can be found in [Chapter 29, “Windows Desktop SSO Authentication Attributes”](#) or by selecting the Help link in the upper right corner of the console.
7. Click Save. The Windows Desktop SSO Authentication service is enabled.

Logging In Using Windows Desktop SSO Authentication

In order to log in using Windows Desktop SSO Authentication, the Core Authentication service attribute “[Organization Authentication Modules](#)” on [page 246](#) must be modified to enable and select Windows Desktop SSO Authentication. This ensures that when the user logs in from a host which is part of the Windows 2000 Domain Controller and has logged in as a domain user using `http://hostname:port/deploy_URI/UI/Login?module=WindowsDesktopSSO`, the user will be authenticated. 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. 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:

- LDAP
- Cert
- Anonymous
- SafeWord
- SecurID
- HTTP Basic
- Membership
- NT
- RADIUS
- Unix
- Windows Desktop SSO

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 8-1 Add Module List Window For A User

Add Authentication Module

Module Name: *

Enforcement Criteria: *

Option:

* Indicates a required field

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

NOTE If you enter `amadmin` credentials in any of the modules in a chain, you will receive the `amadmin` profile. Authentication does not check for alias mapping in this case, nor does it check for modules in the chain.

Authentication Configuration for Organizations

Authentication modules are set for an organization by first adding 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 pane.

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 adding 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 pane.

4. Select Services from the View menu in the Data pane.

5. Modify the Authentication Configuration attributes as necessary. An explanation of these attributes can be found in [Chapter 30, “Authentication Configuration Service 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. When role-based auth is enabled, the LDAP authentication module can be left as the default, as there is no need to configure Membership.

Authentication Configuration for Services

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

1. Choose Services from the View menu in the Identity Management module.
The list of added services are displayed. If the Authentication Configuration service is not added, continue with the steps below. If the service is added, skip to step [Step 4](#).
2. Click Add in the Navigation pane.
A list of available services is displayed in the Data pane.
3. Select the checkbox for Authentication Configuration and click Add.
The Authentication Configuration service will appear in the Navigation pane assuring the administrator that it has been added.
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 30, “Authentication Configuration Service 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.

Auth Level Authentication

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 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.

Module Based Authentication

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 246](#).

URL Redirection

In the Authentication Configuration service, you can assign URL redirection for successful or unsuccessful authentication. The URLs, themselves, are defined in the Login Success URL and Login Failure URL attributes in this service. In order to enable URL redirection, you must add the Authentication Configuration service to your organization to make it available to configure for a role, organization, or user.

Make sure that you add an authentication module, such as LDAP - REQUIRED, when adding the Authentication Configuration service. For information on adding the Authentication Configuration service for identity objects, see [“Authentication Configuration” on page 168](#).

Authentication Service Failover

Authentication service failover automatically redirects an authentication request to a secondary server if the primary server fails because of a hardware or software problem or if the server is temporarily shut down.

An authentication context must first be created on an instance of Identity Server where the authentication service is available. If this instance of Identity Server is not available, an authentication context can then be created on a different instance of Identity Server through the authentication failover mechanism. The authentication context will check for server availability in the following order:

1. The authentication service URL is passed to the AuthContext API. For example:

```
AuthContext(orgName, url)
```

If this API is used, it will only use the server referenced by the URL. No failover will occur even if the authentication service is available on that server.

2. The authentication context will check the server defined in the `com.ipplanet.am.server*` attribute of the `AMConfig.properties` file.
3. If step 2 fails, then the authentication context queries the platform list from a server where the Naming service is available. This platform list is automatically created when multiple instances of Identity Server are installed (generally, for failover purposes) sharing a one instance of Directory Server.

For example, if the platform list contains URLs for `Server1`, `Server2` and `Server3`, then the authentication context will loop through `Server1`, `Server2` and `Server3` until authentication succeeds on one of them.

The platform list may not always be obtained from the same server, as it depends on the availability of the Naming service. Furthermore, Naming service failover may occur first. Multiple Naming service URLs are specified in the `com.ipplanet.am.naming.url` property (in `AMConfig.properties`). The first available Naming service URL will be used to identify the server, which will contain the list of servers (in its platform server list) on which authentication failover will occur.

Password Reset Service

Sun Java™ System Identity Server 2004Q2 provides a Password Reset service to allow users to reset their password for access to a given service or application protected by Identity Server. The Password Reset service attributes, defined by the top-level administrator, control user validation credentials (in the form of *secret questions*), control the mechanism for new or existing password notification, and sets possible lockout intervals for incorrect user validation.

This chapter contains the following sections:

- [“Registering the Password Reset Service” on page 177](#)
- [“Configuring the Password Reset Service” on page 178](#)
- [“Password Reset for End Users” on page 180](#)

Registering the Password Reset Service

The Password Reset service does not need to be registered for the organization in which the user resides. If the Password Reset service does not exist in the organization in which the user resides, it will inherit the values defined for the service in the Service Configuration module.

To Register Password Reset for Users in a Different Organization

1. In the Identity Management module, choose Organizations and select the organization for which you wish to register the service.

2. Click Register in the Navigation frame.

A list of available services displays in the Data frame.

3. Select the checkbox for Password Reset and click Register.

The Password Reset service will appear in the Navigation frame assuring the administrator that it has been registered.

Configuring the Password Reset Service

Once the Password Reset service has been registered, the service must be configured by a user with administrator privileges.

To Configure the Service

1. Select the organization for which the Password Reset service is registered.
2. Click the Password Reset Properties arrow.

The message “No template available for this service” appears in the Data frame. Click Create.

3. The Password Reset attributes appear in the Data frame allowing you to define requirements for the Password Reset service. Make sure that the Password Reset service is enabled (it is by default). At a minimum, the following attributes must be defined:

- User Validation
- Secret Question
- Bind DN
- Bind Password

The Bind DN attribute must contain a user with privileges for resetting the password (for example, Help Desk Administrator). Due a limitation in Directory Server, Password Reset does not work when the bind DN is `cn=directory manager`.

The remaining attributes are optional. Descriptions of the Password Reset attributes can be found in “Password Reset Service Attributes” on page 307 or by clicking the Help link in the upper right corner of the console.

NOTE Identity Server automatically installs the Password Reset web application for random password generation. However, you can write your own plug-in classes for password generation and password notification. See the following Readme.html files in the following locations for samples for these plug-in classes.

PasswordGenerator:

IdentityServer_base/SUNWam/samples/console/PasswordGenerator

NotifyPassword:

IdentityServer_base/SUNWam/samples/console/NotifyPassword

4. Select the Personal Question Enabled attribute if the user is to define his/her unique personal questions. Once the attributes are defined, click Save.

Password Reset Lockout

The Password Reset service contains a lockout feature that will restrict users to a certain number of attempts to correctly answer their secret questions. The lockout feature is configured through the Password Reset service attributes. Descriptions of these attributes can be found in [“Password Reset Service Attributes” on page 307](#). Password Reset supports two types of lockout, memory lockout and physical lockout.

Memory Lockout

This is a temporary lockout and is in effect only when the value in the [Password Reset Failure Lockout Duration](#) attribute is greater than zero and the [Enable Password Reset Failure Lockout](#) attribute is enabled. This lockout will prevent users from resetting their password through the Password Reset web application. The lockout lasts for the duration specified in Password Reset Failure Lockout Duration, or until the server is restarted.

Physical Lockout

This is a more permanent lockout. If the value set in the [Password Reset Failure Lockout Count](#) attribute is set to 0 and the [Enable Password Reset Failure Lockout](#) attribute is enabled, the users' account status is changed to inactive when he or she incorrectly answers the secret questions.

Password Reset for End Users

The following sections describe the user experience for the Password Reset service.

Customizing Password Reset

Once the Password Reset service has been enabled and the attributes defined by the administrator, users are able to log into the Identity Server console in order to customize their secret questions. For example:

1. The user logs into the Identity Server console, providing Username and Password and is successfully authenticated.
2. In the User Profile page, the user selects Password Reset Options. This displays the Available Questions Answer Screen.
3. The user is presented with the available questions that the administrator defined for the service, such as:
 - What is your pet's name?
 - What is your favorite TV show?
 - What is your mother's maiden name?
 - What is your favorite restaurant?
4. The user selects the secret questions, up to the maximum number of questions that the administrator defined for the organization (the maximum amount is defined the Password Reset Service). The user then provides answers to the selected questions. These questions and answers will be the basis for resetting the user's password (see the following section). If the administrator has selected the Personal Question Enabled attribute, text fields are provided, allowing the user to enter a unique secret question and provide an answer.

Figure 9-1 Available Questions Answer Screen with Personal Question Enabled

Password Reset Options for user1

Password Reset Options

This section is used to select the questions used on your forgotten password page. If you forget your password, you will access the forgotten password page, answer the questions that you have selected below, and a new password will be generated for you. You must provide an answer for each question that is selected. You may also provide your own personal question and answer. Up to 3 questions may be selected.

Select	Question	Answer
<input checked="" type="checkbox"/>	what is your pet's name?	raindog
<input type="checkbox"/>	what is your mother's maiden name?	
<input checked="" type="checkbox"/>	what is your favorite baseball team?	giants

OK Cancel

5. The user clicks Save.

Resetting Forgotten Passwords

In the case where users forget their password, Identity Server uses the Password Reset web application to randomly generate new passwords and notify the user of the new password. A typical forgotten password scenario follows:

1. The user logs into the Password Reset web application from a URL given to them by the administrator. For example:

`http://hostname:port/ampassword` (for the default organization)

or

`http://hostname:port/deploy_uri/UI/PWResetUserValidation?org=orgname`,
where *orgname* is the name of the organization.

NOTE If the Password Reset service is not enabled for a parent organization, but is enabled for a sub-organization, users must use the following syntax to access the service:

`http://hostname:
port/deploy_uri/UI/PWResetUserValidation?org=orgname`

2. The user enters the user id.

3. The user is presented with the personal questions that were defined in the Password Reset service and select by the user during customization. If the user has not previously logged into the User Profile page and customized the personal questions, the password will not be generated.

Figure 9-2 Password Questions for User Screen


Sun Java System Identity Server

Password Question for user1

what is your pet's name?

what is your favorite baseball team?

Previous OK

 Copyright © 2004 Sun Microsystems, Inc. All rights reserved. Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries. U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo, Java and Sun[tm] ONE are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Copyright © 2004 Sun Microsystems, Inc. Tous droits réservés. Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les applications de brevet en attente aux États - Unis et dans les autres pays. L'utilisation est soumise aux termes du contrat de licence. Cette distribution peut comprendre des composants développés par des tierces parties. Sun, Sun Microsystems, le logo Sun, Java et Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays.

Once the user answers the questions correctly, the new password is generated and emailed to the user. Attempt notification is sent to the user whether the questions are answered correctly or not. Users must have their email address entered in the User Profile page in order for the new password and attempt notification to be received.

Password Policies

A secure password policy minimizes the risks associated with easily-guessed passwords by enforcing the following:

- Users must change their passwords according to a schedule.
- Users must provide non-trivial passwords.

- Accounts may be locked after a number of binds with the wrong password.

Directory Server provides several ways to set password policy at any node in a tree and there are several ways to set the policy. For details refer following Directory Server documentation:

<http://docs.sun.com/source/816-6700-10/aci.html#14773>

<http://docs.sun.com/source/816-6698-10/useracct.html#14386>

Command Line Reference Guide

This is the Command Line Reference Guide, part three of the Sun Java™ System Identity Server 2004Q2 Administration Guide. This section contains the following chapters:

- “The amadmin Command Line Tool” on page 187
- “The amserver Command Line Tool” on page 195
- “The ampassword Command Line Tool” on page 203
- “The am2bak Command Line Tool” on page 197
- “The bak2am Command Line Tool” on page 201
- “The VerifyArchive Command Line Tool” on page 207
- “The amsecuridd Helper” on page 209

All of the command line tools described in this section can be found in the following default locations”

```
IdentityServer_base/SUNWam/bin (Solairs)
```

```
IdentityServer_base/identity/bin (Linux)
```


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” on page 187](#)

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 `IdentityServer_base/SUNWam/bin` and is used to:

- Load XML service files - Administrators load services into 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 *blobs* of XML data that is referenced by Identity Server. This information is not used by Directory Server which only understands LDAP.

- Perform batch updates of identity objects 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 *Identity Server Developer’s Guide*.

NOTE amadmin only supports a subset of features that the 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.

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 | --passwordfile *passwordfile* [-c | --continue] [-l | --locale *localename*] [[-v | --verbose] | [-d | --debug]] -m | --session *servername* *pattern*
- amadmin -h | --help
- amadmin -n | --version
- amadmin -u | --runasdn *dnname* -w | --password *password* or -f | --passwordfile *passwordfile* [-l | --locale *localename*] [[-v | --verbose] | [-d | --debug]] -a | --addAttributes *serviceName* *schemaType* *xmlfile* [*xmlfile2*] ...

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 (-u)

`--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 (-w)

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

--locale (-l)

`--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 (-c)

`--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 (-m)

`--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
```

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

While using the following option:

```
amadmin -m | --session servername pattern
```

The *pattern* may be a wildcard (*). If this pattern is using a wildcard (*), it has to be escaped with a meta character (\) from the shell.

--debug (-d)

--debug is an option that will write messages to the amadmin file created under the *identity_server_root*/var/opt/SUNWam/debug directory. These messages are technically-detailed but not i18n-compliant. To generate amadmin operation logs, when logging to database, the classpath for the database driver needs to be added manually. For example, add the following lines when logging to mysql in amadmin:

```
CLASSPATH=$CLASSPATH:/opt/IS61/SUNWam/lib/mysql-connector-java-3.0.6-stable-bin.jar
export CLASSPATH
```

--verbose (-v)

--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 (-t)

--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 *Identity Server Developer’s Guide*.

--schema (-s)

--schema is an option that loads the attributes of an 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 --data or --schema option must be specified, depending on whether configuring batch updates to the DIT, or loading service schema and configuration data.

--deleteservice (-r)

--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 10-1 on page 191](#).

Code Example 10-1 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 (-h)

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

--version (-n)

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

Using amadmin for Federation Management

This section lists the parameters of amadmin for use with Federation Management. For more information on Federation Management, see the *Identity Server Federation Management Guide*.

Loading the Liberty meta compliance XML into Directory Server

```
amadmin -u|--runasdn <user's DN>
  -w|--password <password> or -f|--passwordfile <passwordfile>
  -e|--entityname <entity name>
  -g|--import <xmlfile>
```

--runasdn (-u)

The user's DN

--password (-w)

The user's password.

--passwordfile (-f)

The name of file that contains user's password.

--entityname (-e)

The entity name. For example, `http://www.example.com`. An entity should belong to only one organization.

--import (-g)

The name of an XML file that contains the meta information. This file should adhere to Liberty meta specification and XSD.

Exporting an Entity to an XML File (Without XML Digital Signing)

```
amadmin -u | --runasdn <user's DN>
```

```
-w | --password <password> or -f | --passwordfile <passwordfile>
```

```
-e | --entityname <entity name>
```

```
-o | --export <filename>
```

--runasdn (-u)

The user's DN

--password (-w)

The user's password.

--passwordfile (-f)

The name of file that contains user's password.

--entityname (--e)

The name of Entity that resides in the Directory Server

--export (-o)

The name of the file to contain the XML of the entity. XML shall be Liberty meta XSD compliance.

Exporting an Entity to an XML File (With XML Digital Signing)

```
amadmin -u | --runasdn <user's DN>
```

```
-w|--password <password> or -f|--passwordfile <passwordfile>
-e|--entityname <entity name>
-q|--exportwithsig <filename>
```

--runasdn (-u)

The user's DN

--password (-w)

The user's password.

--passwordfile (-f)

The name of file that contains user's password.

--entityname (--e)

The name of Entity that resides in the Directory Server

--exportwithsig (-o)

The name of the file to contain the XML of the entity. This file is digitally signed.
The XML must be Liberty meta XSD compliant.

Using amadmin for Resource Bundles

The following section shows the amadmin syntax for adding, locating and removing resource bundles.

Add resource bundle.

```
amadmin -u|--runasdn <user-dn> -w|--password <user-password>
        -b|--addresourcebundle <name-of-resource-bundle>
        -i|--resourcebundlefilename <resource-bundle-file-name>
        [-R|--resourcelocale] <locale>
```

Get resource strings.

```
amadmin -u|--runasdn <user-dn> -w|--password <user-password>
        -z|--getresourcestrings <name-of-resource-bundle>
```

```
[-R|--resourcelocale] <locale>
```

Remove resource bundle.

```
amadmin -u|--runasdn <user-dn> -w|--password <user-password>  
-j|--deleteresourcebundle <name-of-resource-bundle>  
[-R|--resourcelocale] <locale>
```

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” on page 195](#)
- [“stop is a command that stops the Identity Server.” on page 195](#)

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 { start | stop }
```

start

start is a command that starts the Identity Server.

stop

stop is a command that stops the Identity Server.

The amserver Command Line Executable

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” on page 197](#)

The am2bak Command Line Executable

Identity Server contains an `am2bak` utility under

`IdentityServer_base/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 (-v)

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

--backup *backup-name* (-k)

`--backup backup-name` defines the name of the backup file. The default is `ambak`.

--location (-l)

`--location` specifies the directory location of the backup. The default location is `IdentityServer_base/backup`.

--config (-c)

`--config` specifies backup only for configuration files.

--debug (-b)

`--debug` specifies backup only for debug files.

--log (-g)

`--log` specifies backup only for log files.

--cert (-t)

`--cert` specifies backup only for certificate database files.

--ds (-d)

`--ds` specifies backup only for the Directory Server.

--all (-a)

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

--help (-h)

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

--version (-n)

--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:

- **Configuration and Customization Files:**
 - *IdentityServer_base/SUNWam/config/*
 - *IdentityServer_base/SUNWam/locale/*
 - *IdentityServer_base/SUNWam/servers/httpacl*
 - *IdentityServer_base/SUNWam/lib/*.properties* (Java property files)
 - *IdentityServer_base/SUNWam/bin/amserver.instance-name*
 - *IdentityServer_base/SUNWam/servers/https-all_instances*
 - *IdentityServer_base/SUNWam/servers/web-apps-all_instances*
 - *IdentityServer_base/SUNWam/web-apps/services/WEB-INF/config*
 - *IdentityServer_base/SUNWam/web-apps/services/config*
 - *IdentityServer_base/SUNWam/web-apps/applications/WEB-INF/classes*
 - *IdentityServer_base/SUNWam/web-apps/applications/console*
 - */etc/rc3.d/K55amserver.all_instances*
 - */etc/rc3.d/S55amserver.all_instances*
 - *DirectoryServer_base/slapd-host/config/schema/*
 - *DirectoryServer_base/slapd-host/config/slapd-collations.conf*
 - *DirectoryServer_base/slapd-host/config/dse.ldif*
- **Log And Debug Files:**
 - *var/opt/SUNWam/logs* (Identity Server log files)
 - *var/opt/SUNWam/install* (Identity Server installation log files)

- `var/opt/SUNWam/debug` (Identity Server debug files)
- **Certificates:**
 - `IdentityServer_base/SUNWam/servers/alias`
 - `DirectoryServer_base/alias`

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

- **Configuration and Customization Files:**
 - `IdentityServer_base/web-apps/services/WEB-INF/config/*`
 - `IdentityServer_base/locale/*`
 - `IdentityServer_base/web-apps/applications/WEB-INF/classes/*.properties` (java property files)
 - `IdentityServer_base/servers/https-host/config/jvm12.conf`
 - `IdentityServer_base/servers/https-host/config/magnus.conf`
 - `IdentityServer_base/servers/https-host/config/obj.conf`
 - `DirectoryServer_base/slapd-host/config/schema/*.ldif`
 - `DirectoryServer_base/slapd-host/config/slapd-collations.conf`
 - `DirectoryServer_base/slapd-host/config/dse.ldif`
- **Log And Debug Files:**
 - `var/opt/logs` (Identity Server log files)
 - `var/opt/debug` (Identity Server debug files)
- **Certificates:**
 - `IdentityServer_base/servers/alias`
 - `IdentityServer_base/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” on page 201](#)

The bak2am Command Line Executable

Identity Server contains an `bak2am` utility under `IdentityServer_base/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 `IdentityServer_base/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 `IdentityServer_base/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 `IdentityServer_base/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 ampassword Command Line Tool

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

- [“The ampassword Command Line Executable” on page 203](#)
- [“Running ampassword on SSL” on page 204](#)

The ampassword Command Line Executable

Identity Server contains an `ampassword` utility under `etc/opt/SUNWam/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.

ampasword Options

--admin (-a)

--admin is used to change the admin password.

--proxy (-p)

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

--encrypt (-e)

--encrypt is used to encrypt the password. It is printed to the command line. For example, to encrypt a new dsamuser password, use the following command:

```
ampasword -e newPassword
```

Then, place the new dsamuser password in serverconfig.xml and restart the web container (Web Server or Application Server).

Running ampasword on SSL

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

1. Modify the serverconfig.xml file, located in the following directory:
IdentityServer_base/SUNWam/config/
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> ...
```

ampassword only changes the password in Directory Server. You will have to manually change passwords in the `ServerConfig.xml` and all authentication templates for Identity Server.

Running ampasword on SSL

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” on page 207](#)

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.

NOTE An error may occur if you run `amverifyarchive` as a user without administrator privileges.

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.

The amsecuridd Helper

This chapter provides information on the `amsecuridd` helper and contains the following section:

- [“The amsecuridd Helper Command Line Executable” on page 209](#)
- [“Running the amsecuridd helper” on page 210](#)

The amsecuridd Helper Command Line Executable

The Identity Server SecurID authentication module is implemented using the Security Dynamic ACE/Client C API and the `amsecuridd` helper, which communicates between the Identity Server SecurID authentication module and the SecurID Server. The SecurID authentication module invokes the `amsecuridd` daemon by opening a socket to `localhost:57943` to listen for SecurID authentication requests.

NOTE 57943 is the default port number. If this port number is already used, you can specify a different port number in the [SecurID Helper Authentication Port](#) attribute in the SecurID Authentication module. This port number must be unique across all organizations.

Because the interface to `amsecuridd` is in clear text through `stdin`, only local host connections are permitted. `amsecuridd` uses the SecurID remote API (version 5.x) on the back end for data encryption.

The `amsecuridd` helper listens on port number 58943 (by default) to receive its configuration information. If this port is already used, you can change it in the `securidHelper.ports` attribute in the `AMConfig.properties` file (by default, located in `IdentityServer_base/SUNWam/config/`). The `securidHelp.ports` attribute contains a space-separated list of the ports for each `amsecuridd` helper instance. Restart Identity Sever once the changes to `AMConfig.properties` are saved.

NOTE A separate instance of `amsecuridd` should run for each organization that communicates with a separate ACE/Server (containing different `sdconf.rec` files).

amsecuridd Syntax

The syntax is as follows:

```
amsecuridd [-v] [-c portnum]
```

amsecuridd Options

verbose (-v)

Turns on verbose mode and logs to

`/var/opt/SUNWam/debug/securidd_client.debug`.

configure portnumber (-c portnm)

Configures the listening port number. The default is 58943.

Running the amsecuridd helper

`amsecuridd` is located, by default, in `IdentityServer_base/SUNWam/share/bin`. To run the helper on the default ports, enter the following command (without options):

```
./amsecuridd
```

To run the helper on non-default port, enter the following command:

```
./amsecuridd [-v] [-c portnm]
```

`amsecuridd` can also be run through the `amserver` command line utility, but it will only run on the default ports.

Required Libraries

In order to run the helper, the following libraries are required (most can be found in the operating system in `/usr/lib/`):

- `libnsl.so.1`
- `libthread.so.1`
- `libc.so.1`
- `libdl.so.1`
- `libmp.so.2`
- `librt.so.1`
- `libaio.so.1`
- `libmd5.so.1`

NOTE Set `LD_LIBRARY_PATH` to `IdentityServer_base/Sunwam/lib/` to find `libaceclnt.so`.

The amsecuridd Helper Command Line Executable

Attribute Reference

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

- [“Administration Service Attributes” on page 215](#)
- [“Anonymous Authentication Attributes” on page 233](#)
- [“Certificate Authentication Attributes” on page 237](#)
- [“Core Authentication Attributes” on page 243](#)
- [“HTTP Basic Authentication Attributes” on page 255](#)
- [“LDAP Authentication Attributes” on page 257](#)
- [“Membership Authentication Attributes” on page 263](#)
- [“NT Authentication Attributes” on page 269](#)
- [“RADIUS Authentication Attributes” on page 271](#)
- [“SafeWord Authentication Attributes” on page 275](#)
- [“SecurID Authentication Attributes” on page 277](#)
- [“Unix Authentication Attributes” on page 279](#)
- [“Authentication Configuration Service Attributes” on page 287](#)
- [“Client Detection Service Attributes” on page 291](#)
- [“Globalization Setting Service Attributes” on page 295](#)
- [“Logging Service Attributes” on page 297](#)
- [“Naming Service Attributes” on page 303](#)
- [“Password Reset Service” on page 177](#)

- “Platform Service Attributes” on page 313
- “Policy Configuration Service Attributes” on page 317
- “SAML Service Attributes” on page 327
- “Session Service Attributes” on page 335
- “User Attributes” on page 341

Administration Service Attributes

The Administration Service consists of global and organization attributes. The values applied to the global attributes are applied across the Sun Java System 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” on page 215](#)
- [“Organization Attributes” on page 224](#)

Global Attributes

The global attributes in the Administration Service are:

- [“Enable Federation Management” on page 216](#)
- [“Enable User Management” on page 216](#)
- [“Show People Containers” on page 216](#)
- [“Show Containers In View Menu” on page 217](#)
- [“Show Group Containers” on page 217](#)
- [Managed Group Type](#)
- [Default Role Permissions \(ACIs\)](#)
- [Enable Domain Component Tree](#)

- [“Enable Administrative Groups” on page 220](#)
- [“Enable Compliance User Deletion” on page 220](#)
- [“Dynamic Administrative Roles ACIs” on page 220](#)
- [“User Profile Service Classes” on page 222](#)
- [“DC Node Attribute List” on page 222](#)
- [“Search Filters for Deleted Objects” on page 223](#)
- [“Default People Container” on page 223](#)
- [“Default Groups Container” on page 223](#)
- [“Default Agents Container” on page 223](#)

Enable Federation Management

When selected, this field enables Federation Management. It is selected by default. To disable this feature, deselect the field 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 enabled 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.

Show Containers In View 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 (`mail=*@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.

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.
-

Enable Domain Component Tree

The Domain Component tree (DC tree) is a specific DIT structure used by many Sun Java System 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.iplanet.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.

Enable Administrative Groups

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 the user's 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 `root org`. At the `root org`, the `ServiceAdministrators` and `ServiceHelpDesk Administrators` groups are created and associated with the `Top-level Admin` and `Top-level Help Desk Admin` roles, respectively. The same behavior applies.

Enable Compliance User Deletion

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 Administrative 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 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.

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.

DC Node Attribute List

This field defines the set of attributes that will be set in the DC tree entry when an object is created. The default parameters are:

- maildomainwelcomemessage
- preferredmailhost
- mailclientattachmentquota
- mailroutingsmarthost

- mailroutingsmarthost
- mailroutingsmarthost
- mailaccessproxyreplay
- preferredlanguage
- domainuidseparator
- maildomainmsgquota
- maildomainallowedserviceaccess
- preferredmailmessagestore
- maildomaindiskquota
- maildomaindiskquota
- objectclass=maildomain
- mailroutinghosts

Search Filters for Deleted Objects

This field defines the search filters for objects to be removed when User Compliance Deletion mode is enabled.

Default People Container

This attribute specifies the default people container into which the user is created.

Default Groups Container

This attribute specifies the default groups container into which the group is created.

Default Agents Container

This attribute specifies the default agent container into which the agent is created.

Organization Attributes

The organization attributes in the administration service are:

- “Groups Default People Container” on page 225
- “Groups People Container List” on page 225
- “User Profile Display Class” on page 225
- “Show Roles on User Profile Page” on page 225
- “Show Groups on User Profile Page” on page 226
- “Enable User Self Subscription to Group” on page 226
- “User Profile Display Options” on page 226
- “User Creation Default Roles” on page 226
- “Administrative Console Tabs” on page 227
- “Maximum Results Returned From Search” on page 227
- “Timeout For Search” on page 227
- “JSP Directory Name” on page 227
- “Online Help Documents” on page 227
- “Required Services” on page 228
- “User Search Key” on page 228
- “User Search Return Attribute” on page 228
- “User Creation Notification List” on page 229
- “User Deletion Notification List” on page 229
- “User Modification Notification List” on page 230
- “Maximum Entries Displayed per Page” on page 230
- “Event Listener Classes” on page 230
- “Pre and Post Processing Classes” on page 231
- “Enable External Attributes Fetch” on page 231
- “UserID and Password Validation Plugin Class” on page 231

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:

dn of group | dn of people container

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.

End User Profile Display Class

This attribute specifies the Java class used by the Identity Server console when it displays the End User Profile pages.

Show Roles on User Profile Page

This option specifies whether to display a list of roles assigned to a user as part of the user's 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`.

Show Groups on User Profile Page

This option specifies whether to display a list of groups assigned to a user as part of the user's 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`.

Enable User Self Subscription to Group

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 [Show Groups on User Profile Page](#) 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 *Identity Server Developer'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. The roles can only be Identity Server roles, not LDAP (Directory Server) roles.

Administrative Console Tabs

This field lists the Java classes of modules that will be displayed 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 Use caution when setting this attribute to large value. For sizing limits, see the *Sun Java System Directory Server Installation and Tuning Guide* at the following location:

<http://docs.sun.com/db/doc/816-6697-10>

Timeout For Search

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:

*link*i18n*key | html page to load when clicked | i18n properties file | remote server*

NOTE *remote server* is optional argument that allows you to specify the remote server on which the online help document is located.

For example:

```
IdentityServer Help | /AMAdminHelp.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 and created by the `amadmin` command line utility 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 field defines the attribute name used when displaying the users returned from a simple search. The default of this attribute is `uid cn`. This will display the user ID and the user's full name.

The attribute name that is listed first is also used as the key for sorting the set of users that will be returned. To avoid performance degradation, use an attribute whose value is set in a user's entry.

User Creation Notification List

This field defines a list of email addresses that will be sent notification when a new user is created. Multiple email addresses can be specified, as in the following syntax:

```
e-mail|locale|charset
```

```
e-mail|locale|charset
```

```
e-mail|locale|charset
```

The notification list also accepts different locales by using the `|locale` option. For example, to send the notification to an administrator in France:

```
someuser@example.com|fr|fr
```

See [Table 20-1 on page 249](#) 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 `IdentityServer_base/SUNWam/locale`.

User Deletion Notification List

This field defines a list of email addresses that will be sent notification when a user is deleted. Multiple email addresses can be specified, as in the following syntax:

```
e-mail|locale|charset
```

```
e-mail|locale|charset
```

```
e-mail|locale|charset
```

The notification list also accepts different locales by using the `|locale` option. For example, to send the notification to an administrator in France:

```
someuser@example.com|fr|fr
```

See [Table 20-1 on page 249](#) 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 `IdentityServer_base/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. Multiple email address can be specified, as in the following syntax:

```
attrName e-mail|locale|charset e-mail|locale|charset .....
attrName e-mail|locale|charset e-mail|locale|charset .....
```

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 20-1 on page 249](#) 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 Displayed 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.

Event Listener Classes

This attribute contains a list of listeners that receive creation, modification and deletion events from the Identity Server console.

Pre and Post Processing Classes

This field defines a list of implementation classes through plug-ins that extend the `com.ipplanet.am.sdk.AMCallback` class to receive callbacks during pre and post processing operations for users, organization, roles and groups. The operations are:

- create
- delete
- modify
- add users to roles/groups
- delete users from roles/groups

You must enter the full class name of the plug-in. For example:

```
com.ipplanet.am.sdk.AMCallbacSample
```

You must then change the class path of your web container (from the Identity Server installation base) to include the full path to the location of the plug-in class.

Enable External Attributes Fetch

This option enables callbacks for plug-ins to retrieve external attributes (any external application-specific attribute). External attributes are not cached in the Identity Server SDK, so this attribute allows you enable attribute retrieval per organization level. By default, this option is not enabled.

UserID and Password Validation Plugin Class

This class provides a userID and password validation plugin mechanism.

The methods of this class need to be overridden by the implementation plugin modules that validate the userID and/or password for the user. The implementation plugin modules will be invoked whenever a userID or password value is being added or modified using the Identity Server console, the `amadmin` command line interface, or using the SDK.

The plugins that extend this class can be configured per organization. If a plugin is not configured for an organization, then the plugin configured at the global level will be used.

If the validation of the plugin fails, the plugin module can throw an exception to notify the application to indicate the error in the userID or password supplied by the user.

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:

- [“Valid Anonymous User List” on page 233](#)
- [“Enable Case Sensitive User IDs” on page 234](#)
- [“Default Anonymous User Name” on page 234](#)
- [“Authentication Level” on page 234](#)

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 this list is empty, accessing the following default module login URL will be authenticated as the Default Anonymous User Name:

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name
```

If this list is not empty, accessing Default module login URL (same as above) will prompt the user to enter any valid Anonymous user name

If this list is not empty, the user can log in without seeing the login page by accessing the following URL:

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name&IDToken1=<valid Anonymous username>
```

Default Anonymous User Name

This field defines the user ID that a session is assigned to if Valid Anonymous User List is empty and the following Default module login URL is accessed:

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name
```

The default value is `anonymous`. An Anonymous user must also be created in the organization.

NOTE

If Valid Anonymous User List is not empty, you can login without accessing the login page by using the user defined in Default Anonymous User Name. This can be done by accessing the following URL:

```
protocol://server_host.server_domain:server_port/server_deploy_uri/UI/Login?module=Anonymous&org=org_name&IDToken1=<DefaultAnonymous User Name>
```

Enable Case Sensitive User IDs

If enabled, this option allows for case-sensitivity for user IDs. By default, this attribute is 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. 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.

NOTE

If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

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” on page 238](#)
- [“Subject DN Attribute Used to Search LDAP for Certificates” on page 238](#)
- [“Match Certificate to CRL” on page 238](#)
- [“Issuer DN Attribute Used to Search LDAP for CRLs” on page 239](#)
- [“Enable OCSP Validation” on page 239](#)
- [“LDAP Server Where Certificates Are Stored” on page 240](#)
- [“LDAP Start Search DN” on page 240](#)
- [“LDAP Server Principal User” on page 240](#)
- [“LDAP Server Principal Password” on page 240](#)
- [“LDAP Attribute for Profile ID” on page 241](#)
- [“Use SSL for LDAP Access” on page 241](#)
- [“Certificate Field Used to Access User Profile” on page 241](#)
- [“Other Certificate Field Used to Access User Profile” on page 241](#)
- [“Trusted Remote Hosts” on page 242](#)
- [“SSL Port Number” on page 242](#)

- [“Authentication Level” on page 242](#)

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 238](#). However, the web container may check the validity of the user certificate presented at login.

Subject DN Attribute Used to Search LDAP for Certificates

This field specifies the attribute of the certificate’s `SubjectDN` value that will be used to search LDAP for certificates. This attribute must uniquely identify a user entry. 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. The CRL is located by one of the attribute names in the issuer’s `SubjectDN`. 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.

Issuer DN Attribute Used to Search LDAP for CRLs

This field specifies the attribute of the received certificate's issuer `subjectDN` value that will be used to search LDAP for CRLs. 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`.

HTTP Parameters for CRL Update

This field specifies the HTTP parameters for obtaining a CRL from a servlet for a CRL update. Contact the administrator of your CA for these parameters.

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 `com.sun.identity.authentication.ocspCheck` is true and the OCSP responder is set in the `com.sun.identity.authentication.ocsp.responder.url` attribute, the value of the attribute will be used as the OCSP responder.
- If `com.sun.identity.authentication.ocspCheck` is set to true and 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 `com.sun.identity.authentication.ocspCheck` is set to false or if `com.sun.identity.authentication.ocspCheck` is set to true and 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 Where Certificates Are Stored

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. The format is as follows:

```
servername|search dn
```

For multiple entries

```
servername1|search dn servername2|search dn servername3|search dn...
```

If multiple users are found for the same search, authentication will fail.

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.

Use SSL 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.

Certificate Field Used 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
- subject CN
- subject DN
- subject UID
- other

Other Certificate Field Used to Access User Profile

If the value of the [Certificate Field Used 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.

Trusted Remote Hosts

This attribute defines a list of trusted hosts that can be trusted to send certificates to Identity Server. Identity Server must verify whether the certificate emanated from one of these hosts. This configuration only used with Sun Java System Portal Server.

This attribute accepts the following values:

- **none.** This attribute is disabled. This is set by default.
- **any.** Accepts Portal Server Gateway-style certificate authentication from any client IP address.
- **IP ADDR.** Lists the IP addresses from which to accept Portal Server Gateway-style certificate authentication requests (the IP Address of the Gateway(s)). The attribute is configurable on an organization basis.

SSL Port Number

This attribute specifies the port number for the secure socket layer. Currently, this attribute is only used by the Gateway servlet. Before you add or change an SSL Port Number, see the “Policy-Based Resource Management” section in Chapter 7 of the Identity Server Developer’s Guide.

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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

Core Authentication Attributes

The Core Authentication service is the basic service for all of the default authentication services as well as any custom authentication module attributes. 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 Java System 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 adding the service for the organization. The default values can be changed after adding by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Core Authentication attributes are separated into:

- [“Global Attributes” on page 243](#)
- [“Organization Attributes” on page 245](#)

Global Attributes

The global attributes in the Core Authentication service are:

- [“Pluggable Authentication Module Classes” on page 244](#)
- [“Supported Authentication Modules for Clients” on page 244](#)
- [“LDAP Connection Pool Size” on page 244](#)
- [“Default LDAP Connection Pool Size” on page 244](#)

Pluggable Authentication Module Classes

This field specifies the Java classes of the authentication modules available to any organization configured within the Identity Server platform. By default, this includes LDAP, SafeWord, SecurID, Application, Anonymous, HTTP Basic, Membership, Unix, Certificate, NT, RADIUS and Windows Desktop SSO. You can write custom authentication modules by implementing the AMLoginModule SPI or the JAAS LoginModule SPI. For more information, see the Identity Server Developer's Guide. 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 Authentication 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 LDAP 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`.

Default LDAP Connection 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 not be used from LDAP Connection Default Pool Size.

Organization Attributes

The organization attributes in the Core Authentication service are:

- [“Organization Authentication Modules” on page 246](#)
- [“User Profile” on page 246](#)
- [“Administrator Authentication Configuration” on page 247](#)
- [“User Profile Dynamic Creation Default Roles” on page 247](#)
- [“Enable Persistent Cookie Mode” on page 247](#)
- [“Persistent Cookie Maximum Time” on page 248](#)
- [“People Container For All Users” on page 248](#)
- [“Alias Search Attribute Name” on page 248](#)
- [“Default Authentication Level” on page 254](#)
- [“User Naming Attribute” on page 249](#)
- [“Default Authentication Locale” on page 249](#)
- [“Organization Authentication Configuration” on page 250](#)
- [“Enable Login Failure Lockout Mode” on page 251](#)
- [“Login Failure Lockout Count” on page 251](#)
- [“Login Failure Lockout Interval” on page 251](#)
- [“Email Address to Send Lockout Notification” on page 251](#)
- [“Warn User After N Failure” on page 251](#)
- [“Login Failure Lockout Duration” on page 252](#)
- [“Lockout Attribute Name” on page 252](#)
- [“Lockout Attribute Value” on page 252](#)
- [“Default Success Login URL” on page 252](#)
- [“Default Failure Login URL” on page 253](#)
- [“Authentication PostProcessing Class” on page 253](#)
- [“Enable Generate UserID Mode” on page 253](#)
- [“Pluggable User Name Generator Class” on page 253](#)

Organization Authentication Modules

This list specifies the authentication modules available to the organization. Each administrator can choose the type of authentication for each 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
- HTTP Basic
- Membership
- NT
- SafeWord
- RADIUS
- SecurID
- Unix
- Windows Desktop SSO

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 the local Directory Server installed with Identity 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. The user profile is created in the local Directory Server installed with Identity Server.

- **Ignore** - This specifies that the user profile is not required by the authentication service to issue the SSO Token for a successful authentication.

Administrator Authentication Configuration

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. The modules configured in this attribute are picked up when the Identity Server console is accessed. For example:

```
http://servername.port/console_deploy_uri
```

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 246](#). There is no default value. The administrator must specify the DN 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. This role can be either an Identity Server or LDAP role, but it cannot be a filtered role.

Enable 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 [Enable Persistent Cookie Mode](#). When [Enable 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 Maximum Time](#). 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 `iPSPCookie=yes` parameter in the login URL.

Persistent Cookie Maximum Time

This field specifies the interval after which a persistent cookie expires. ([Enable 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, the user's 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, a user may authenticate using Safeword ID of `jn10191`, but the profile is `uid=jamie`.

Alias Search Attribute Name

After successful authentication by a user, the user's 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 249](#), 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`).

User Naming Attribute

After successful authentication by a user, the user's 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 Authentication 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 20-1](#).

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 "Chapter 3: Authentication Service" in the *Identity Server Developer's Guide* for more information.

Table 20-1 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

Table 20-1 Supported Language Locales (*Continued*)

Language Tag	Language
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
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 pass through the chain of all selected modules.

The modules configured in this attribute are used for authentication when users access the authentication module using the `/server_deploy_uri/UL/Login` format. See the Identity Server Developer's Guide for more information.

Enable Login Failure Lockout Mode

This feature specifies whether a user can attempt a second authentication if the first attempt failed. Selecting this attribute enables a lockout and the user will have only one chance at authentication. By default, the lockout feature is not enabled. This attribute works in conjunction with Lockout-related and notification attributes.

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](#), before being locked out.

Login Failure Lockout Interval

This attribute defines (in minutes) the time between two failed login attempts. If a login fails and is followed by another failed login that occurs within the lockout interval, then the lockout count is incremented. Otherwise, the lockout count is reset.

Email Address to Send Lockout Notification

This attribute specifies an email address that will receive notification if a user lockout occurs. To send email notification to multiple addresses, separate each email address with a space.

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

This attribute enables memory locking. By default, the lockout mechanism will inactivate the User Profile (after a login failure) defined in Lockout Attribute Name. If the value of Login Failure Lockout Duration is greater than 0, then its memory locking and the user account will be locked for the number of minutes specified.

Lockout Attribute Name

This attribute designates any LDAP attribute that is to be set for lockout. The value in Lockout Attribute Value must also be changed to enable lockout for this attribute name. By default, Lockout Attribute Name is empty in the Identity Server Console. The default implementation values are `inetuserstatus` (LDAP attribute) and `inactive` when the user is locked out and Login Failure Lockout Duration is set to 0.

Lockout Attribute Value

This attribute specifies whether lockout is enabled or disabled for the attribute defined in [Lockout Attribute Name](#). By default, the value is set to `inactive` for `inetuserstatus`.

Default Success Login URL

This field accepts a list of multiple values that specify the URL to which users are redirected after successful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML. The Success Login URL is set in the `LoginStatus` element in the `remote-auth.dtd`. See the *Identity Server Developer's Guide* for more information.

Default Failure Login URL

This field accepts a list of multiple values that specify the URL to which users are redirected after an unsuccessful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML. The Failure Login URL is set in the `LoginStatus` element in the `remote-auth.dtd`. See the *Identity Server Developer's Guide* for more information.

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
```

The Java class must implement the following Java interface:

```
com.sun.identity.authentication.spi.AMPostAuthProcessInterface
```

Additionally, you must add the path to where the class is located to the Web Server's Java Classpath attribute.

Enable Generate UserID 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, during the Self Registration process, 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 [Enable Generate UserID Mode](#) is enabled.

Default Authentication 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 value in this attribute is not used by Identity Server but by any external application that may chose to use it.) For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

HTTP Basic Authentication Attributes

The HTTP Basic Authentication attribute is an organization attributes. The values applied to them under Service Configuration become the default values for the HTTP Basic 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 HTTP Basic Authentication attributes is:

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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

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” on page 258](#)
- [“Secondary LDAP Server” on page 258](#)
- [“DN to Start User Search” on page 259](#)
- [“DN for Root User Bind” on page 259](#)
- [“Password for Root User Bind” on page 259](#)
- [“Password For Root User Bind \(Confirm\)” on page 260](#)
- [“LDAP Attribute Used to Retrieve User Profile” on page 260](#)
- [“LDAP Attributes Used to Search for a User to be Authenticated” on page 260](#)
- [“User Search Filter” on page 260](#)
- [“Search Scope” on page 260](#)
- [“Enable SSL Access to LDAP Server” on page 261](#)
- [“Return User DN To Authenticate” on page 261](#)
- [“LDAP Server Check Interval” on page 261](#)
- [“User Creation Attributes List” on page 261](#)
- [“Authentication Level” on page 262](#)

Primary LDAP Server

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.)

If you have Identity Server deployed with multiple domains, you can specify the communication link between specific instances of Identity Server and Directory Server in the following format (multiple entries must be prefixed by the local server name):

```
local_servername|server:port local_servername2|server:port ...
```

For example, if you have two Identity Servers deployed in different locations (L1-machine1-IS and L2-machine2-IS) communicating with different instances of Identity Server (L1-machine1-DS and L2-machine2-DS), it would look the following:

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389  
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

Secondary LDAP Server

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 the primary server is up, 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. If `OBJECT` is selected in the `Search Scope` attribute, the DN should specify one level above the level in which the profile exists.

Multiple entries must be prefixed by the local server name. The format is as follows:

```
servername|search dn
```

For multiple entries

```
servername1|search dn servername2|search dn servername3|search dn...
```

If multiple users are found for 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. The default value is `amldapuser`. Any valid DN will be recognized.

Make sure that password is correct before you logout, because if it is incorrect, you will be locked out. If this should occur, you can login with the super user DN in the `com.iplanet.authentication.super.user` property in the `AMConfig.Properties` file. By default, this is the `amAdmin` account with which you would normally log in, although you will use the full DN. For example:

```
uid_amAdmin,ou=People,IdentityServer_base
```

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.

LDAP Attribute Used to Retrieve User Profile

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 LDAP Attribute Used to Retrieve User Profile.

LDAP Attributes Used to Search for a User to be Authenticated

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 259. 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

CAUTION Users from suborganizations may be able to login even if the sub organization's status is inactive. To avoid this, make sure that the Search Scope and the Base DN are set to the specific organization to which the user belongs.

Enable SSL Access 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, this is not enabled and the SSL protocol will not be used to access the Directory Server. However, if this attribute is enabled, you can bind to a non-SSL server.

Return User DN To Authenticate

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.

LDAP Server Check Interval

This attribute is used for LDAP Server failback. It defines the number of minutes in which a thread will “sleep” before verifying that the LDAP primary server is running.

User Creation Attributes List

This attribute is used by the LDAP authentication module when the LDAP server is configured as an external LDAP server. It contains a mapping of attributes between a local and an external Directory Server. This attribute has the following format:

```
attr1|externalattr1
```

attr2|externalattr2

When this attribute is populated, the values of the external attributes are read from the external Directory Server and are set for the internal Directory Server attributes. The values of the external attributes are set in the internal attributes only when the [User Profile](#) attribute (in the Core Authentication module) is set to “Dynamically Created” and the user does not exist in local Directory Server instance. The newly created user will contain the values for internal attributes, as specified in User Creation Attributes List, with the external attribute values to which they map.

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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

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” on page 264
- “Default User Roles” on page 264
- “User Status After Registration” on page 264
- “Primary LDAP Server” on page 264
- “Secondary LDAP Server” on page 265
- “DN to Start User Search” on page 265
- “DN for Root User Bind” on page 266
- “Password for Root User Bind” on page 266
- “Password for Root User Bind (Confirm)” on page 266
- “LDAP Attribute Used to Retrieve User Profile” on page 266
- “LDAP Attributes Used to Search for a User to be Authenticated” on page 266
- “User Search Filter” on page 267
- “Search Scope” on page 267
- “Enable SSL Access to LDAP Server” on page 267
- “Return User DN To Authenticate” on page 267

- [“Authentication Level” on page 268](#)

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_base/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. Only the roles that can be assigned to the user will be added during self-registration. All other DNs will be ignored. The role can be either an Identity Server role or an LDAP role, but filtered roles are not accepted.

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

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.).

If you have Identity Server deployed with multiple domains, you can specify the communication link between specific instances of Identity Server and Directory Server in the following format (multiple entries must be prefixed by the local server name):

```
local_servername|server:port local_servername2|server:port ...
```

For example, if you have two Identity Servers deployed in different locations (L1-machine1-IS and L2-machine2-IS) communicating with different instances of Identity Server (L1-machine1-DS and L2-machine2-DS), it would look the following:

```
L1-machine1-IS.example.com|L1-machine1-DS.example.com:389  
L2-machine2-IS.example.com|L2-machine2-DS.example.com:389
```

Secondary LDAP Server

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 the primary server is up, 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. If OBJECT is selected in the [Search Scope](#) attribute, the DN should specify one level above the level in which the profile exists.

If you use multiple entries, the entries must be prefixed by the local server name. The format is as follows:

```
servername|search dn
```

For multiple entries

servername1|search dn servername2|search dn servername3|search dn...

If multiple users are found for 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. The default is `amldapuser`. 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.

LDAP Attribute Used to Retrieve User Profile

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.

LDAP Attributes Used to Search for a User to be Authenticated

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 265. 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 Access 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 Authenticate

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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

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.

In order to activate the NT Authentication module, Samba Client 2.2.2 must be downloaded and installed to the following directory:

```
IdentityServer_base/SUNWam/bin
```

Samba Client is a file and print server for blending Windows and UNIX machines together without requiring a separate Windows NT/2000 Server. More information, and the download itself, can be accessed at <http://www.sun.com/software/download/products/3e3af224.html>.

Red Hat Linux ships with a Samba client, located in the following directory:

```
/usr/bin
```

In order to authenticate using the NT Authentication service for Linux, copy the client binary to the following Identity Server directory:

```
IdentityServer_base/identity/bin
```

The NT Authentication attributes are:

- “NT Authentication Domain” on page 270
- “NT Authentication Host” on page 270
- “Authentication Level” on page 270

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`.

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. 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.

NOTE

If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

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” on page 271](#)
- [“RADIUS Server 2” on page 272](#)
- [“RADIUS Shared Secret” on page 272](#)
- [“RADIUS Shared Secret \(Confirm\)” on page 272](#)
- [“RADIUS Server's Port” on page 272](#)
- [“Timeout” on page 272](#)
- [“Authentication Level” on page 272](#)

RADIUS Server 1

This field displays the IP address or fully qualified 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 as in the following syntax:

```
local_servername|ip_address local_servername2|ip_adress ...
```

RADIUS Server 2

This field displays the IP address or fully qualified domain name (FQDN) 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 as in the following syntax:

```
local_servername|ip_address local_servername2|ip_address ...
```

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.

Timeout

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.

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. 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. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

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” on page 275](#)
- [“SafeWord Server Verification Files Directory” on page 275](#)
- [“SafeWord Logging Level” on page 276](#)
- [“SafeWord Log File” on page 276](#)
- [“Authentication Level” on page 276](#)

SafeWord Server

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 Server Verification Files Directory

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 File

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.

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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

SecurID Authentication Attributes

The SecurID Authentication Attributes are organization attributes. The values applied to them under Service Configuration become the default values for the SecurID 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 RSA's ACE/Server authentication server. The SecurID Authentication attributes are:

- [“SecurID ACE/Server Configuration Path” on page 277](#)
- [“SecurID Helper Configuration Port” on page 278](#)
- [“SecurID Helper Authentication Port” on page 278](#)
- [“Authentication Level” on page 278](#)

NOTE In this version of Identity Server, the SecurID Authentication service is not supported for the Linux and x86 operating systems.

SecurID ACE/Server Configuration Path

This field specifies the directory in which the SecurID ACE/Server `sdconf.rec` file is located. The default is as follows:

```
/opt/ace/data
```

If a different directory is specified in this field, the directory must exist before attempting SecurID authentication.

SecurID Helper Configuration Port

This attribute specifies the port on which the SecurID helper 'listens' upon startup for the configuration information contained in the SecurID Helper Authentication Port attribute. The default is 58943.

If this attribute is changed, you must also change the `securidHelper.ports` entry in the `AMConfig.properties` file, and restart Identity Server. The entry in the `AMConfig.properties` file is a space-separated list of the ports for the instances of SecurID helpers. For each organization that communicates with a different ACE/Server (which has a different `sdconf.rec` file), there must be a separate SecurID helper.

SecurID Helper Authentication Port

This attribute specifies the port that the organization's SecurID authentication module will configure its SecurID helper instance to 'listen' for authentication requests. This port number must be unique across all organizations using SecurID or Unix authentication. The default port is 57943.

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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See ["Default Authentication Level" on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

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 Java System 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” on page 279](#)
- [“Organization Attribute” on page 280](#)

NOTE If any of the Unix authentication attributes are modified, both Identity Server and the `amunixd` helper must be restarted.

Global Attributes

The global attributes in the Unix Authentication service are:

- [“Unix Helper Configuration Port” on page 280](#)
- [“Unix Helper Authentication Port” on page 280](#)
- [“Unix Helper Timeout” on page 280](#)
- [“Unix Helper Threads” on page 280](#)

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](#), 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

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:

Authentication Level

The authentication level is set separately for each method of authentication. The value 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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

Windows Desktop SSO Authentication Attributes

The Windows Desktop SSO Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Windows Desktop SSO 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 authentication module requires the Kerberos authentication service provided by a Windows 2000 server running as a domain controller.

The Windows Desktop SSO Authentication attributes are:

- [“Service Principal” on page 283](#)
- [“Keytab Filename” on page 284](#)
- [“Kerberos Realm” on page 284](#)
- [“Kerberos Server Name” on page 284](#)
- [“Return Principal With Domain Name” on page 284](#)
- [“Authentication Level” on page 284](#)

Service Principal

This attribute specifies the Kerberos principal that is used for authentication. Use the following format:

`HTTP/hostname.domainname@dc_domain_name`

hostname and *domainname* represent the hostname and domain name of the Identity Server instance. *dc_domain_name* is the Kerberos domain in which the Windows 2000 Kerberos server (domain controller) resides. It is possibly different from the domain name of the Identity Server.

Keytab Filename

This attribute specifies the Kerberos keytab file that is used for authentication. Use the following format, although the format is not required:

hostname.HTTP.keytab

hostname is the hostname of the Identity Server instance.

Kerberos Realm

This attribute specifies the Kerberos Distribution Center (domain controller) domain name. Depending up on your configuration, the domain name of the domain controller may be different than the Identity Server domain name.

Kerberos Server Name

This attribute specifies the Kerberos Distribution Center (the domain controller) hostname. You must enter the fully qualified domain name (FQDN) of the domain controller.

Return Principal With Domain Name

If enabled, this attributes allows Identity Server to automatically return the Kerberos principal with the domain controller's domain name during authentication.

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. 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.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See [“Default Authentication Level” on page 254](#) for details. For the 2004Q2 release, this feature does not function properly. In previous releases, however, it does.

Authentication Configuration Service Attributes

The Authentication Configuration Service attributes are dynamic and organization attributes. These attributes can be defined for an organization, service, or role. The organization attributes are defined in the Core Authentication module.

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” on page 287](#)
- [“Login Success URL” on page 288](#)
- [“Login Failure URL” on page 289](#)
- [“Authentication Post Processing Class” on page 289](#)

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 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 Service Attributes

The Client Detection Service 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” on page 291](#)
- [“Default Client Type” on page 294](#)
- [“Client Detection Class” on page 294](#)
- [“Enable Client Detection” on page 294](#)

Client Types

In order to detect client types, Identity Server needs to recognize their identifying characteristics. These characteristics identify the properties of all supported types in the form of client data. This attribute allows you to modify the client data through the Client Manager interface. To access the Client Manager, click the Edit link.

Out of the box, Identity Server contains the following client types:

- HDML
- HTML
- JHTML
- VoiceX
- WML

- XHTML
- cHTML
- iHTML
- For descriptions of these client types, see the Sun Java System Portal Server, Mobile Access 2004Q2 Administration Guide at the following location:

<http://docs.sun.com/prod/entsys#hic>

Client Manager

The Client Manager is the interface that lists the base clients, styles and associated properties, and allows you to add and configure devices.

Base Client Types

The Base client types are listed at the top of Client Manager. These client types contain the default properties that can be inherited by all devices that belong to the client type.

Style Profile

The Client Manager groups all available clients, including the Base client type itself, in the Styles pulldown menu. The selected Style (or, parent profile) defines properties that are common to its configured child devices. The devices dynamically inherit the properties of the parent profile

The Current Style Properties link launches a read-only Client Editor window for viewing the style properties.

Device Profile

When a style is selected, the Client Manager displays the device profiles configured for that style. Devices are sorted by user agent (device name) and can be filtered by entering the user agent string in the Filter field (wildcards are accepted).

For each device, you can modify the client properties by clicking on the Edit link located next to each device name. The properties are then displayed in the Client Editor window. To edit the properties, select the following classifications from the pull-down list:

Hardware Platform. Contains properties of the device's hardware, such as display size, supported character sets, and so forth.

Software Platform. Contains properties of the device's application environment, operating system, and installed software.

Network Characteristics. Contains properties describing the network environment, including the supported bearers.

BrowserUA. Contains attributes related to the browser user agent running on the device.

WapCharacteristics. Contains properties of the Wireless Application Protocol (WAP) environment supported by the device.

PushCharacteristicsNames. Contains properties of the WAP environment supported by the device.

Additional Properties. Allows you to add additional properties for the device.

For specific property definitions, see the Open Mobile Alliance Ltd. (OMA) *Wireless Application Protocol, Version 20-Oct-2001* at the following location:

<http://www1.wapforum.org/tech/terms.asp?doc=WAP-248-UAPProf-20011020-a.pdf>

Once the properties have been modified, click Save. The device will display “**” characters to denote that the device has been customized. Use the Default link to remove the customized properties and reset the device back to the default settings.

To add a new device for a style, click the New Device button. The Create New Device window is displayed with the following fields:

Style. Displays the base style for the device, for example HTML.

Device User Agent. Accepts a name for the device.

Click Next to display the following fields:

Client Type Name. Displays the client type, for example HTML. The client type name must be unique across all devices.

The Immediate Parent For This Device. Accepts the parent (base) client type for the device. For example, HTML.

The HTTP User Agent String. Defines the User-Agent in the HTTP request header. For example, Mozilla/4.0.

Click OK and customize the device properties. For specific property definitions, see the Open Mobile Alliance Ltd. (OMA) *Wireless Application Protocol, Version 20-Oct-2001* at the following location:

<http://www1.wapforum.org/tech/>

To duplicate a device and its properties, click the Duplicate link. Device names must be unique. By default, Identity Server will rename the device to `copy_of_devicename`.

To delete any device, click the Delete link listed with the device.

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.sun.mobile.cdm.FEDIClientDetector`. Identity Server also contains `com.iplanet.services.cdm.ClientDetectionDefaultImpl`.

Enable Client Detection

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.

By default, the client detection capability is enabled. If this attribute is not selected, Identity Server assumes that the client is `genericHTML` and will be accessed from a HTML browser.

Globalization Setting Service Attributes

The Globalization Setting Service 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 Globalization Setting Attributes are:

- [“Charsets Supported By Each Locale” on page 295](#)
- [“Charset Aliases” on page 295](#)
- [“Auto Generated Common Name Format” on page 296](#)

Charsets Supported By Each Locale

This attribute lists the charset support for each locale, which indicates the mapping between locale and charset. The format is as follows:

```
locale=localename|charset=charset1;charset2;charset3;...;charsetn
```

You can add, edit, duplicate and remove charsets with the buttons located at the bottom of the attribute.

Charset Aliases

This attribute lists the codeset names (which map to IANA names) that will be used to send the response. These codeset names do not need to match java codeset names. Currently, there is a hash table to map java character sets into IANA charsets and vice versa. The alias format is as follows:

mimeName=*charset* | javaName=*charset*

For example:

```
mimeName=Shift_JIS | javaName=SJIS
```

This implies that both denote same character set.

You can add, edit, duplicate and remove character set aliases with the buttons located at the bottom of the attribute.

Auto Generated Common Name Format

This display option allows you to define the way in which a name is automatically generated, to accommodate name formats for different locales and character sets. The default syntax is as follows (please note that including commas and/or spaces in the definition will display in the name format):

```
en_us = {givenname} {initials} {sn}
```

For example, if you wanted to display a new name format for a user (User One) with a uid (11111) for the Chinese character set, use the following stands:

```
zh = {sn}{givenname}({uid})
```

This would display as:

```
OneUser 11111
```

Logging Service Attributes

The Logging Service attributes are global attributes. The values applied to them are applied across the Sun Java System 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:

- “Maximum Log Size” on page 298
- “Number of History Files” on page 298
- “Log File Location” on page 298
- “Logging Type” on page 299
- “Database User Name” on page 299
- “Database User Password” on page 299
- “Database User Password (Confirm)” on page 299
- “Database Driver Name” on page 299
- “Configurable Log Fields” on page 299
- “Log Verification Frequency” on page 300
- “Log Signature Time” on page 300
- “Enable Secure Logging” on page 300
- “Maximum Number of Records” on page 300
- “Number Of Files Per Archive” on page 300
- “Buffer Size” on page 301
- “Buffer Time” on page 301

- [“Enable Time Buffering” on page 301](#)

Maximum Log Size

This attribute accepts a value for the maximum size (in bytes) of a Identity Server log file. 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 entered depending on the partition size and available disk space of the local system. The default value is 3.

NOTE Entering a value of 0 is interpreted to be the same as a value of 1, meaning that if you specify 0, a backup log file will be created.

Log File 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:

```
/var/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.

When configuring the log location for DB (database) logging (such as, Oracle or MySQL), part of the log location is case sensitive.

For example, if you are logging to an Oracle database, the log location should be:

```
jdbc:oracle:thin:@machine.domain:port:DBName
```

```
jdbc:oracle:thin must be lower case.
```

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:

- Domain
- Hostname
- IPAddress
- LoggedBy
- Loglevel

- LoginID
- ModuleName

Log Verification Frequency

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.

Enable 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.

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.

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.

Buffer Time

This attribute defines the amount of time that the log records will be buffered in memory before they are sent to the logging service to be logged. The default is 3600 seconds.

Enable Time Buffering

When selected as ON, Identity Server will set a time limit for log records to be buffered in memory. The amount of time is set in the [Buffer Time](#) attribute.

Naming Service Attributes

The Naming Service attributes are global attributes. The values applied to them are carried across the Sun Java System 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” on page 304](#)
- [“Session Service URL” on page 304](#)
- [“Logging Service URL” on page 304](#)
- [“Policy Service URL” on page 304](#)
- [“Auth Service URL” on page 304](#)
- [“SAML Web Profile/Artifact Service URL” on page 305](#)
- [“SAML SOAP Service URL” on page 305](#)
- [“SAML Web Profile/POST Service URL” on page 305](#)
- [“SAML Assertion Manager Service URL” on page 305](#)
- [“Federation Assertion Manager Service URL” on page 306](#)
- [“Identity SDK Service URL” on page 306](#)

Profile Service URL

This field takes a value equal to

```
%protocol://%host:%port/Server_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/Server_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/Server_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/Server_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/Server_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/Server_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/Server_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/Server_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/Server_DEPLOY_URI/AssertionManagerServlet/AssertionManagerIF
```

This syntax allows for dynamic substitution of the SAML Assertion Manager Service URL based on the specific session parameters.

Federation Assertion Manager Service URL

This field takes a value equal to

```
%protocol://%host:%port/amserver/FSAssertionManagerServlet/FSAssertionManagerIF
```

This syntax allows for dynamic substitution of the Federation Assertion Manager Service URL based on the specific session parameters.

Identity SDK Service URL

This field takes a value equal to

```
%protocol://%host:%port/amserver/UserManagementServlet/
```

This syntax allows for dynamic substitution of the Identity SDK Service URL based on the specific session parameters.

Password Reset Service Attributes

The Password Reset Service attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Password Reset Service in a given organization. Organization attributes are not inherited by entries in the subtrees of the organization.

The Password Reset attributes are:

- “User Validation” on page 308
- “Secret Question” on page 308
- “Search Filter” on page 308
- “Base DN” on page 308
- “Bind DN” on page 308
- “Bind Password” on page 309
- “Password Reset Option” on page 309
- “Password Change Notification Option” on page 309
- “Enable Password Reset” on page 309
- “Enable Personal Question” on page 309
- “Maximum Number of Questions” on page 309
- “Force Change Password on Next Login” on page 310
- “Enable Password Reset Failure Lockout” on page 310
- “Password Reset Failure Lockout Count” on page 310
- “Password Reset Failure Lockout Interval” on page 310
- “Email Address to Send Lockout Notification” on page 310

- “Warn User After N Failure” on page 311
- “Password Reset Failure Lockout Duration” on page 311
- “Password Reset Lockout Attribute Name” on page 311
- “Password Reset Lockout Attribute Value” on page 311

User Validation

This attribute specifies the value that is used to search for the user whose password is to be reset.

Secret Question

This field allows you to add a list of questions that the user can use to reset his/her password. To add a question, type it in the Secret Question field and click Add. The selected questions will appear in the user's User Profile page. The user can then select a question for resetting the password.

Users may create their own question if the Personal Question Enabled attribute is selected.

Search Filter

This attribute specifies the search filter to be used to find user entries.

Base DN

This attribute specifies the DN from which the user search will start. If no DN is specified, the search will start from the organization DN. You should not use `cn=directorymanager` as the base DN, due to proxy authentication conflicts.

Bind DN

This attribute value is used with Bind Password to reset the user password.

Bind Password

This attribute value is used with Bind DN to reset the user password.

Password Reset Option

This attribute determines the classname for resetting the password. The default classname is:

```
com.sun.identity.password.RandomPasswordGenerator
```

The password reset class can be customized through a plug-in. This class needs to be implemented by the `PasswordGenerator` interface. See the *Identity Server Developer's Guide* for more information.

Password Change Notification Option

This attribute determines the method for user notification of password resetting. The default classname is:

```
com.sun.identity.password.EmailPassword
```

The password notification class can be customized through a plugin. This class needs to be implemented by the `NotifyPassword` interface. See the *Identity Server Developer's Guide* for more information.

Enable Password Reset

Selecting this attribute will enable the password reset feature.

Enable Personal Question

Selecting this attribute will allow a user to create a unique question for password resetting.

Maximum Number of Questions

This value specifies the maximum number of questions to be asked in the password reset page.

Force Change Password on Next Login

When enabled, this option forces the user to change his or her password on the next login. If you want an administrator, other than the top-level administrator, to set the force password reset option, you must modify the Default Permissions ACIs to allow access to that attribute.

Enable Password Reset Failure Lockout

This attribute specifies whether to disallow users to reset their password if that user initially fails to reset the password using the Password Reset application. By default, this feature is not enabled.

Password Reset Failure Lockout Count

This attribute defines the number of attempts that a user may try to reset password, within the time interval defined in Password Reset Failure Lockout Interval, before being locked out.

For example, if Password Reset Failure Lockout Count is set to 5 and Login Failure Lockout Interval is set to 5 minutes, the user has five chances within five minutes to reset the password before being locked out.

Password Reset Failure Lockout Interval

This attribute defines (in minutes) the amount of time in which the number of password reset attempts (as defined in Password Reset Failure Lockout Count) can be completed, before being locked out.

Email Address to Send Lockout Notification

This attribute specifies an email address that will receive notification if a user is locked out from the Password Reset service. Specify multiple email address in a space-separated list.

Warn User After N Failure

This attribute specifies the number of password reset failures that can occur before Identity Server sends a warning message that user will be locked out.

Password Reset Failure Lockout Duration

This attribute defines (in minutes) the duration that user will not be able to attempt a password reset if a lockout has occurred.

Password Reset Lockout Attribute Name

This attribute contains the `inetuserstatus` value that is set in Password Reset Lockout Attribute Value. If a user is locked out from Password Reset, and the Password Reset Failure Lockout Duration (minutes) variable is set to 0, `inetuserstatus` will be set to `inactive`, prohibiting the user from attempting to reset his or her password.

Password Reset Lockout Attribute Value

This attribute specifies the `inetuserstatus` value (contained in Password Reset Lockout Attribute Name) of the user status, as either `active` or `inactive`. If a user is locked out from Password Reset, and the Password Reset Failure Lockout Duration (minutes) variable is set to 0, `inetuserstatus` will be set to `inactive`, prohibiting the user from attempting to reset his or her password.

Platform Service Attributes

The Platform Service attributes are global attributes. The values applied to them are carried across the Sun Java System 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” on page 313
- “Platform Locale” on page 314
- “Cookie Domains” on page 314
- “Login Service URL” on page 314
- “Logout Service URL” on page 315
- “Available Locales” on page 315
- “Client Char Sets” on page 315

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 or failover 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_name`

Only the naming service protocol should be used in this attribute.

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 20-1 on page 249](#) 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 to 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.

NOTE Make sure that the correct cookie domain is entered. If the cookie domain is incorrect, you will not be able to login to Identity Server.

Login Service URL

This field specifies the URL of the login page. The default value for this attribute is `/Service_DEPLOY_URI/UI/Login`.

Logout Service URL

This field specifies the URL of the logout page. The default value for this attribute is `/Service_DEPLOY_URI/UI/Logout`.

Available Locales

This attribute stores all available locales configured for the platform. Consider an application that lets the user choose the user's 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 Service Attributes

The Policy Configuration Service attributes consist of global and organization attributes. The values applied to the global attributes are applied across the Sun Java System 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 Attributes” on page 317](#)
- [“Organization Attributes” on page 318](#)

Global Attributes

The global attributes in the Policy Configurative service are:

- [“Resource Comparator” on page 318](#)
- [“Continue Evaluation On Deny Decision” on page 318](#)

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.

Continue Evaluation On Deny Decision

This attribute specifies whether or not the policy framework should continue evaluating subsequent policies, even if a DENY policy decision exists. If it is unselected (default), policy evaluation would skip subsequent policies once the DENY decision is recognized.

Organization Attributes

The organization attributes in the Policy Configuration service are:

- [“LDAP Server and Port” on page 320](#)
- [“LDAP Base DN” on page 321](#)
- [“LDAP Users Base DN” on page 321](#)
- [“Identity Server Roles Base DN” on page 321](#)
- [“LDAP Bind DN” on page 321](#)
- [“LDAP Bind Password” on page 321](#)
- [“LDAP Bind Password \(Confirm\)” on page 321](#)

- “LDAP Organization Search Filter” on page 321
- “LDAP Organization Search Scope” on page 322
- “LDAP Groups Search Filter” on page 322
- “LDAP Groups Search Scope” on page 322
- “LDAP Users Search Filter” on page 322
- “LDAP Users Search Scope” on page 322
- “LDAP Roles Search Filter” on page 323
- “LDAP Roles Search Scope” on page 323
- “Identity Server Roles Search Scope” on page 323
- “LDAP Organization Search Attribute” on page 323
- “LDAP Groups Search Attribute” on page 323
- “LDAP Users Search Attribute” on page 324
- “LDAP Roles Search Attribute” on page 324
- “Maximum Results Returned From Search” on page 324
- “Timeout For Search” on page 324
- “Enable LDAP SSL” on page 324
- “LDAP Connection Pool Minimal Size” on page 324
- “LDAP Connection Pool Maximum Size” on page 325
- “Selected Policy Subjects” on page 325
- “Selected Policy Conditions” on page 325
- “Selected Policy Referrals” on page 325
- “Subjects Result Time To Live” on page 325
- “User Alias Enabled” on page 326

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* For example:

```
machine1.example.com:389
```

For failover configuration to multiple LDAP server hosts, this value can be a space-delimited list of hosts. The format is *hostname1:port1 hostname2:port2...*

For example:

```
machine1.example1.com:389 machine2.example1.com:389
```

Multiple entries must be prefixed by the local server name. This is to allow specific Identity Servers to be configured to talk to specific Directory Servers.

The format is `servername|hostname:port`

For example:

```
machine1.example1.com|machine1.example1.com:389
```

```
machine1.example2.com|machine1.example2.com:389
```

For failover configuration:

```
IS_Server1.example1.com|machine1.example1.com:389 machine2.example.com1:389
```

```
IS_Server2.example2.com|machine1.example2.com:389 machine2.example2.com:389
```

NOTE This attribute has changed to accept a list of values to support multiple servers. In the 6.0 SP1 release, this attribute only accepted a single value.

This may cause a problem if you attempt to make 6.0SP1 and 6.1 to co-exist in a single deployment environment, specifically for the scenario in which an Identity Server 6.0 SP1 instance points to a 6.1 DIT.

For successful co-existence, ensure that there is only a single LDAP server for this attribute.

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 Users Base DN

This attribute specifies the base DN used by the LDAP Users subject in the LDAP server from which to begin the search. By default, it is the top-level organization of the Identity Server installation base.

Identity Server Roles Base DN

This attribute specifies the base DN used by the Identity Server Roles subject in the LDAP server from which to begin the search. By default, it is the top-level organization of the Identity Server installation base.

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 Organization Search Filter

Specifies the search filter to be used to find organization entries. The default is `(objectclass=sunMangagedOrganization)`.

LDAP Organization 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)

Identity Server Roles Search Scope

This attribute defines the scope to be used to find entries for Identity Server Roles subject. 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

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.

Enable LDAP SSL

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 the 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.

User Alias Enabled

This attribute must be enabled if you create a policy to protect a resource whose subject's member in a remote Directory Server aliases a local user.

This attribute must be enabled, for example, if you create `uid=rmuser` in the remote Directory Server and then add `rmuser` as an alias to a local user (such as `uid=luser`) in Identity Server. When you login as `rmuser`, a session is created with the local user (`luser`) and policy enforcement is successful.

SAML Service Attributes

The Security Assertion Markup Language (SAML) Service attributes are global attributes. The values applied to them are carried across the Sun Java System 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 *Identity Server Developer's Guide*.

The SAML attributes are as follows:

- [“Site ID And Site Issuer Name” on page 328](#)
- [“Sign SAML Request” on page 328](#)
- [“Sign SAML Response” on page 328](#)
- [“Sign Assertion” on page 328](#)
- [“SAML Artifact Name” on page 328](#)
- [“Target Specifier” on page 329](#)
- [“Artifact Timeout” on page 329](#)
- [“Assertion Skew Factor For notBefore Time” on page 329](#)
- [“Assertion Timeout” on page 329](#)
- [“Trusted Partner Sites” on page 329](#)
- [“POST To Target URLs” on page 333](#)

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. A default value will be assigned during installation. The format is as follows:

```
instanceid=serverprotocol://servername:portnumber|siteid=site_id|issuerName=site_issuer_name
```

After configuring for this attribute for SSL (in the both source and destination site), make sure that the `instanceid` protocol is `HTTPS//`.

Sign SAML 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 SAML 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.

SAML 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`. For example using the default `SAMLart` service configuration, the redirect query string could be:

```
http://host:port/deploy_URI/SamlAwareServlet?TARGET=http://URL/&SAMLart=artifact123
```

Target Specifier

This attribute assigns a variable name to the destination site URL used in the re-direct. The default is `Target`.

Artifact Timeout

This attribute specifies the timeout for an assertion created for an artifact. The default is 400.

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 (180 is the default value), the notBefore attribute of the conditions element for the assertion would be `2002-09-24T21:34:49Z`.

Assertion Timeout

This attribute specifies the number of seconds before a timeout occurs on an assertion. The default is 420.

NOTE The total valid duration of an assertion is defined by the values set in both the Assertion Skew Factor For notBefore Time and Assertion Timeout attributes.

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/SAMLSOAPReceiver|AuthType=SSL|hostlist=ipaddress (or, server DNS name, or cert alias)
```

The parameters are:

Table 38-1 Trusted Partner Sites Parameters

SourceID	The 20-byte sequence defined as in the SiteID and Issuer name.
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 <code>Web-browser SSO with Artifact</code> 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 <code>Web-browser SSO with POST</code> 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 URL.

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 require and the SOAPUrl should be HTTPS.</p>
User	<p>Defines the uid of the partner which is used to protect the partner's SOAP Receiver.</p>
version	<p>Defines the SAML version used to send SAML request. Specify either 1.0 or 1.1 for the SAML version. If this parameter is not defined, the following default values are used from AMConfig.properties:</p> <pre>com.example.identity.saml.assertion.version=1.1 com.example.identity.saml.protocol.version=1.1</pre>
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> <p>If the requester's host or client certificate is in this list in the receiver's site, the service will continue. If the host or client certificate does not match any of those hosts or certificates in the hostlist, the SAML service will reject the request.</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.DefaultAccountMa pper</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>

<code>actionMapper</code>	Specifies the class with the path to where the <code>actionMapper</code> is located. Applications can develop an <code>actionMapper</code> to obtain either an <code>SSOToken</code> ID or an assertion containing <code>AuthenticationStatement</code> from the query. The mapper is then used to retrieve the authorization decisions for the actions defined in the query. If no <code>actionMapper</code> is specified, <code>DefaultActionMapper</code> will be used.
<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=<i>aliasName</i></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>

	Sender	Receiver
SOAP Request		sourceid hostlist [attributeMapper] [actionMapper] [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. Avoid using test URLs or any other additional URLs in a POST.

Session Service Attributes

The Session Service attributes are global and dynamic attributes. The values applied to the global attributes 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 values applied to the dynamic 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. 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.

Global Attributes

The global attributes are:

- [“Maximum Number of Search Results” on page 335](#)
- [“Timeout For Search \(Seconds\)” on page 336](#)

Maximum Number of Search Results

This attribute specifies the maximum number of results returned by a session search. The default value is 120.

Timeout For Search (Seconds)

This attribute defines the maximum amount of time before a session search terminates. The default value is 5 seconds.

Dynamic Attributes

The dynamic attributes are:

- [“Max Session Time \(Minutes\)” on page 336](#)
- [“Max Idle Time \(Minutes\)” on page 336](#)
- [“Max Caching Time \(Minutes\)” on page 337](#)

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 Session Time limits the validity of the session. It does not get extended beyond the configured 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. It is recommended that the maximum caching time should always be less than the maximum idle time.

SOAP Binding Service Attributes

The SOAP Binding Service attributes are global attributes. The values applied to them are carried across the Sun Java System 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 SOAP Binding Service attributes are as follows:

- [“Request Handler List” on page 339](#)
- [“Web Service Authenticator” on page 340](#)
- [“Supported Authentication Mechanisms” on page 340](#)

Request Handler List

This attribute stores information about a Web Service Provider (WSP) deployed in Identity Server. It lists entries that contain a key/value pair (separated by “|”). For example:

```
key=disco|class=com.example.identity.liberty.ws.disco.DiscoveryService|soa
pActions=sal sa2 sa2
```

To add a new request handler, click the add button. The key and class parameters are required. The parameters are:

key. This defines the second part of the URI path for the SOAP endpoint of the WSP. The first part is defined as Liberty by the SOAP services. For example, if you define `disco` as the key, the SOAP endpoint for the Discovery service is:

```
protocol://hostname:port/deploy_uri/Liberty/disco
```

class. This parameter specifies the name of the implementation class for the WSP. The Liberty SOAP layer provides a handler interface to be implemented by each WSP to process the requested message and then return a response.

soapActions. This is an optional parameter that specifies supported SOAPActions. If this parameter is not specified, all SOAPActions are supported. If a Web Service Consumer (WSC) sends a request with an unsupported SOAPAction, the request will be rejected by the SOAP layer without passing it one to the corresponding WSP.

Web Service Authenticator

This attribute defines the implementation class for the `WebServiceAuthenticator` interface, which authenticates and generates a credential for a Web Service Consumer (WSC), based on the request.

Supported Authentication Mechanisms

This attribute specifies the authentication mechanisms supported by the SOAP endpoint. By default, all of the mechanisms are selected. If an authentication mechanism is not selected, and a WSC sends a request using that authentication mechanism, the request will be rejected by the SOAP layer without passing it to the corresponding WSP.

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 Service Attributes” on page 341](#)
- [“User Profile Attributes” on page 343](#)
- [“Unique User IDs” on page 346](#)

User Service Attributes

The User Service 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](#)
- [Administrator 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 the on-screen 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 20-1 on page 249](#) can be used.

Administrator 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.

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 *Identity Server Developer'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 Java System 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 Java System 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 `true`. 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 *Identity Server Developer'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. In order to use any aliases configured in this attribute, the LDAP service has to be modified by adding the `iplanet-am-user-alias-list` attribute to the User Entry Search Attributes field in the LDAP service.

Preferred Locale

This field specifies the locale for the user. The default value is `en_US`. Any value from [Table 20-1 on page 249](#) can be used.

You can use one of the following attributes in the pull-down menu:

- Ignore
- Customize
- Inherit

Success URL

This field accepts a list of multiple values that specify the URL to which users are redirected after successful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML

Failure URL

This field accepts a list of multiple values that specify the URL to which users are redirected after an unsuccessful authentication. The format of this attribute is `clientType|URL`, although you can specify only the value of the URL which assumes a default type of HTML

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: 6.1
nsslapd-pluginVendor: Sun | SunONE
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
```

Unique User IDs

Error Codes

This appendix provides a list of the error messages generated by Sun Java System Identity Server. While this list is not exhaustive, the information presented in this chapter will serve as a good starting point for common problems. The tables listed in this appendix provide the error code itself, a description and/or probable cause of the error, and describes the actions that can be taken to fix the encountered problem.

This appendix lists error codes for the following functional areas:

- [Identity Server Console Errors](#)
- [Authentication Error Codes](#)
- [Policy Error Codes](#)
- [amadmin Error Codes](#)

If you require further assistance in diagnosing errors, please contact Sun Technical Support:

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

Identity Server Console Errors

The following table describes the error codes generated and displayed by the Identity Server Console.

Table A-1 Identity Server Console Errors

Error Message	Description/Probable Cause	Action
An error has occurred while deleting the following:	The object may have been removed by another user prior to being removed by the current user.	Redisplay the objects that you are trying to delete and try the operation again.

Table A-1 Identity Server Console Errors

Error Message	Description/Probable Cause	Action
You have entered an invalid URL	This occurs if the URL for an Identity Server console window is entered incorrectly.	
There are no entries matching the search criteria.	The parameters entered in the search window, or in the Filter fields, did not match any objects in the directory.	Run the search again with a different set of parameters
There are no attributes to display.	The selected object does not contain any editable attributes defined in its schema.	
There is no information to display for this service.	The services viewed from the Service Configuration module do not have global or organization based attributes	
Search size limit exceeded. Please refine your search.	The parameters specified in the search have returned more entries than are allowed to be returned	Modify the Maximum Results Returned from a Search attribute in the Administration service to a larger value. You can also modify the search parameters to be more restrictive.
Search time limit exceeded. Please refine your search.	The search for the specified parameters has taken longer than the allowed search time.	Modify the Timeout for Search attribute in the Administration service to a larger value. You can also modify the search parameters, so they are less restrictive, to return more values.
Invalid user's start location. Please contact your administrator.	The start location DN in the users entry is no longer valid	In the User Profile page, change the value of the start DN to a valid DN.
Could not create <i>identity object</i> . User does not have sufficient access.	An operation was executed by a user with insufficient permissions. The permissions a user has defined determines what operations they can perform.	

Authentication Error Codes

The following table describes the error codes generated by the Authentication service. These errors are displayed to the user/administrator in the Authentication module.

Table A-2 Authentication Error Codes

Error Message	Description/Probable Cause	Action
authentication.already.login.	The user has already logged in and has a valid session, but there is no Success URL redirect defined.	Either logout, or set up some login success redirect URL(s) through the Identity Server Console. Use the 'goto' query parameter with the value as Admin Console URL.
logout.failure.	A user is unable to logout of Identity Server.	Restart the server.
uncaught_exception	An authentication Exception is thrown due to an incorrect handler	Check the Login URL for any invalid or special characters.
redirect.error	Identity Server cannot redirect to Success or Failure redirect URL.	Check the web container's error log to see if there are any errors.
gotoLoginAfterFail	This link is generated when most errors occur. The link will send the user to the original Login URL page.	
invalid.password	The password entered is invalid.	Passwords must contain at least 8 characters. Check that the password contains the appropriate amount of characters and ensure that it has not expired.
auth.failed	Authentication failed. This is the generic error message displayed in the default login failed template. The most common cause is invalid/incorrect credentials.	Enter valid and correct user name/password (the credentials required by the invoked authentication module.)
nouser.profile	No user profile was found matching the the entered user name in the given organization. This error is displayed while logging in to the Membership/Self-registration authentication module.	Enter your login information again. If this is your first login attempt, select New User in the login screen.
notenough.characters	The password entered does not contain enough characters. This error is displayed while logging in to the Membership/Self-registration authentication module.	The login password must contain at least 8 characters by default (this number is configurable through the Membership Authentication module).

Table A-2 Authentication Error Codes

Error Message	Description/Probable Cause	Action
useralready.exists	A user already exists with this name in the given organization. This error is displayed while logging in to the Membership/Self-registration authentication module.	User IDs must be unique within the organization.
uidpasswd.same	The User Name and Password fields cannot have the same value. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure that the username and password are different.
nouser.name	No user name was entered. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure to enter the user name.
no.password	No password was entered. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure to enter the password.
missing.confirm.passwd	Missing the confirmation password field. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure to enter the password in the Confirm Password field.
password.mismatch	The password and the confirm password do not match. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure that the password and confirmation password match.
An error occurred while storing the user profile.	An error occurred while storing the user profile. This error is displayed while logging in to the Membership/Self-registration authentication module.	Make sure that the attributes and elements are valid and correct for Self Registration in the Membership.xml file.
orginactive	This organization is not active.	Activate the organization through the Identity Server console by changing the organization status from inactive to active.
internal.auth.error	Internal Authentication Error. This is a generic Authentication error which may be caused by different and multiple environmental and/or configuration issues.	

Table A-2 Authentication Error Codes

Error Message	Description/Probable Cause	Action
usernot.active	The user no longer has an active status.	Activate the user through the Admin Console by changing the user status from <code>inactive</code> to <code>active</code> . if the user is locked out by Memory Locking, restart the server.
user.not.inrole	User does not belong to the specified role. This error is displayed during role-based authentication.	Make sure that the login user belongs to the role specified for the role-based authentication.
session.timeout	The user session has timed out.	Login in again.
authmodule.denied	The specified authentication module is denied.	Make sure that the required authentication module is registered under the required organization, that the template is created and saved for the module, and that the module is selected in the Organization Authentication Modules list in the Core Authentication module.
noconfig.found	No configuration found.	Check the Authentication Configuration service for the required authentication method.
cookie.notpersistent	Persistent Cookie Username does not exist in the Persistent Cookie Domain.	
nosuch.domain	The organization found.	Make sure that the requested organization is valid and correct.
userhasnoprofile.org	User has no profile in the specified organization.	Make sure that the user exists and is valid in the specified organization in the local Directory Server.
reqfield.missing	One of the required fields was not completed. Please make sure all required fields are entered.	Make sure that all required fields are entered.
session.max.limit	Maximum Sessions Limit Reached.	Logout and login again.

Policy Error Codes

The following table describes the error codes generated by the Policy framework and displayed in the Identity Server Console.

Table A-3 Policy Error Codes

Error Message	Description/Probable Cause	Action
illegal_character_/_in_name	Illegal character "/" in the policy name.	Make sure that the policy name does not contain the '/' character.
policy_already_exists_in_org	A rule with the same name already exists.	Use a different name for policy creation.
rule_name_already_present	Another rule with the given name already exists	Use a different rule name for policy creation.
rule_already_present	A rule with the same rule value already exists.	Use a different rule value.
no_referral_can_not_create_policy	No referral exists to the organization.	In order to create policies under a sub organization, you must create a referral policy at its parent organization to indicate what resources can be referred to this sub organization.
ldap_search_exceed_size_limit	LDAP search size limit exceeded. An error occurred because the search found more than the maximum number of results.	Change the search pattern or policy configuration of the organization for the search control parameters. The Search Size Limit is located in the Policy Configuration service.
ldap_search_exceed_time_limit	LDAP search time limit exceeded. An error occurred because the search found more than the maximum number of results.	Change the search pattern or policy configuration of the organization for the search control parameters. The Search Time Limit is located in the Policy Configuration service.
ldap_invalid_password	Invalid LDAP Bind password.	The password for LDAP Bind user defined in Policy Configuration is incorrect. This leads to the inability to get an authenticated LDAP connection to perform policy operations.
app_sso_token_invalid	Application SSO token is invalid.	The server could not validate the Application SSO token. Most likely the SSO token is expired.

Table A-3 Policy Error Codes

Error Message	Description/Probable Cause	Action
user_sso_token_invalid	User SSO token is invalid.	The server could not validate the User SSO token. Most likely the SSO token is expired.
property_is_not_an_Integer	Property value not an integer.	The value for this plugin's property should be an integer.
property_value_not_defined	Property value should be defined.	Provide a value for the given property.
start_ip_can_not_be_greater_than_end_ip	Start IP is larger than End IP	An attempt was made to set end IP Address to be larger than start IP Address in IP Address condition. The Start IP cannot be larger than the End IP.
start_date_can_not_be_larger_than_end_date	Start Date is larger than End Date	An attempt was made to set end Date to be larger than start Date in the policy's Time Condition. The Start Date cannot be larger than the End Date.
policy_not_found_in_organization	Policy not found in organization. An error occurred trying to locate a non-existing policy in an organization.	Make sure that the policy exists under the specified organization.
insufficient_access_rights	User does not have sufficient access. The user does not have sufficient right to perform policy operations.	Perform policy operations with the user who has appropriate access rights.
invalid_ldap_server_host	Invalid LDAP Server host.	Change the invalid LDAP Server host that was entered in the Policy Configuration service.

amadmin Error Codes

The following table describes the error codes generated by the `amadmin` command line tool to Identity Server's debug file.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
nocomptype	1	Too few arguments.	Make sure that the mandatory arguments (<code>--runasdn</code> , <code>--password</code> , <code>--passwordfile</code> , <code>--schema</code> , <code>--data</code> , and <code>--addAttributes</code>) and their values are supplied in the command line.
file	2	The input XML file was not found.	Check the syntax and make sure that the input XML is valid.
nodnforadmin	3	The user DN for the <code>--runasdn</code> value is missing.	Provide the user DN as the value for <code>--runasdn</code> .
noservicename	4	The service name for the <code>--deleteservice</code> value is missing.	Provide the service name as the value for <code>--deleteservice</code> .
nopwdforadmin	5	The password for the <code>--password</code> value is missing.	Provide the password as the value for <code>--password</code> .
nolocalename	6	The locale name was not provided. The locale will default to <code>en_US</code> .	See Default Authentication Locale for a list of locales.
nofile	7	Missing XML input file.	Provide at least one input XML filename to process.
invopt	8	One or more arguments are incorrect.	Check that all arguments are valid. For a set of valid arguments, type <code>amadmin --help</code> .
oprfailed	9	Operation failed.	When <code>amadmin</code> fails, it produces more precise error codes to indicate the specific error. Refer to those error codes to evaluate the problem.
execfailed	10	Cannot process requests.	When <code>amadmin</code> fails, it produces more precise error codes to indicate the specific error. Refer to those error codes to evaluate the problem.
policycreatexception	12	Policy cannot be created.	<code>amadmin</code> produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
policydelexception	13	Policy cannot be deleted.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
smsdelexception	14	Service cannot be deleted.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
ldapauthfail	15	Cannot authenticate user.	Make sure the user DN and password are correct.
parseerror	16	Cannot parse the input XML file.	Make sure that the XML is formatted correctly and adheres to the amAdmin.dtd.
parseiniterror	17	Cannot parse due to an application error or a parser initialization error.	Make sure that the XML is formatted correctly and adheres to the amAdmin.dtd.
parsebuildererror	18	Cannot parse because a parser with specified options cannot be built.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
ioexception	19	Cannot read the input XML file.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
fatalvalidationerror	20	Cannot parse because the XML file is not a valid file.	Check the syntax and make sure that the input XML is valid.
nonfatalvalidationerror	21	Cannot parse because the XML file is not a valid file.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
validwarn	22	XML file validation warnings for the file.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
failedToProcessXML	23	Cannot process the XML file.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
nodataschemawarning	24	Neither --data or --schema options are in the command.	Check that all arguments are valid. For a set of valid arguments, type amadmin --help.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
doctypeerror	25	The XML file does not follow the correct DTD.	Check the XML file for the DOCTYPE element.
statusmsg9	26	LDAP Authentication failed due to invalid DN, password, hostname, or portnumber.	Make sure the user DN and password are correct.
statusmsg13	28	Service Manager exception (SSO exception).	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg14	29	Service Manager exception.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg15	30	Schema file inputstream exception.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg30	31	Policy Manager exception (SSO exception).	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
statusmsg31	32	Policy Manager exception.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
debugerror	33	More than one debug option is specified.	Only one debug option should be specified.
loginFailed	34	Login failed.	amadmin produces exception messages to indicate the specific error. Refer to those messages to evaluate the problem.
levelerr	36	Invalid attribute value.	Check the level set for the LDAP search. It should be either SCOPE_SUB or SCOPE_ONE.
failToGetObjType	37	Error in getting object type.	Make sure that the DN in the XML file is value and contains the correct object type.
invalidOrgDN	38	Invalid organization DN.	Make sure that the DN in the XML file is valid and is an organization object.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
invalidRoleDN	39	Invalid role DN.	Make sure that the DN in the XML file is valid and is a role object.
invalidStaticGroupDN	40	Invalid static group DN.	Make sure that the DN in the XML file is valid and is a static group object.
invalidPeopleContainerDN	41	Invalid people container DN.	Make sure the DN in the XML file is valid and is a people container object.
invalidOrgUnitDN	42	Invalid organizational unit DN.	Make sure that the DN in the XML file is valid and is a container object.
invalidServiceHostName	43	Invalid service host name.	Make sure that the hostname for retrieving valid sessions is correct.
subschemaexception	44	Subschema error.	Subcschema is only supported for global and organization attributes.
serviceschemaexception	45	Cannot locate service schema for service.	Make sure that the sub schema in the XML file is valid.
roletemplateexception	46	The role template can be true only if the schema type is dynamic.	Make sure that the role template in the XML file is valid.
cannotAddusersToFilteredRole	47	Cannot add users to a filtered role.	Made sure that the role DN in the XML file is not a filtered role.
templateDoesNotExist	48	Template does not exist.	Make sure that the service template in the XML file is valid.
cannotAddUsersToDynamicGroup	49	Cannot add users to a dynamic group.	Made sure that the group DN in the XML file is not a dynamic group.
cannotCreatePolicyUnderContainer	50	Policies can not be created in an organization that is a child organization of a container.	Make sure that the organization in which the policy is to be created is not a child of a container.
defaultGroupContainerNotFound	51	The group container was not found.	Create a group container for the parent organization or container.
cannotRemoveUserFromFilteredRole	52	Cannot remove a user from a filtered role.	Make sure that the role DN in the XML file is not filtered role.
cannotRemoveUsersFromDynamicGroup	53	Cannot remove users from a dynamic group.	Make sure that the group DN in the XML file is not a dynamic group.
subSchemStringDoesNotExist	54	The subschema string does not exist.	Make sure that the subschema string exists in the XML file.

Table A-4 amadmin error codes

Error Message	Code	Description/Probable Cause	Action
defaultPeopleContainerNot Found	59	You are trying to add user to an organization or container. And default people container does not exists in an organization or container.	Make sure the default people container exists.
nodefaulturlprefix	60	Default URL prefix is not found following --defaultURLPrefix argument	provide the default URL prefix accordingly.
nometaalias	61	Meta Alias is not found following --metaalias argument	provide the Meta Alias accordingly.
missingEntityName	62	Entity Name is not specified.	provide the entity name.
missingLibertyMetaInputFile	63	File name for importing meta data is missing.	provide the file name that contains meta data.
missingLibertyMetaOutput File	64	File name for storing exported meta data is missing.	provide the file name for storing meta data.
cannotObtainMetaHandler	65	Unable to get a handler to Meta attribute. Specified user name and password may be incorrect.	ensure that user name and password are correct.
missingResourceBundleName	66	Missing resource bundle name when adding, viewing or deleting resource bundle that is store in directory server.	provide the resource bundle name
missingResourceFileName	67	Missing file name of file that contains the resource strings when adding resource bundle to directory server.	Please provide a valid file name.
failLoadLibertyMeta	68	Failed to load liberty meta to Directory Server.	Please check the meta data again before loading it again

Glossary

For a list of terms used in this documentation set, refer to the latest *Sun Java™ Enterprise System Glossary*.

<http://docs.sun.com/doc/816-6873>

SYMBOLS

276

A

Adding Conditions 134

Adding Rules 129

Administration Attributes 215

Global Attributes 215

DC Node Attribute List 222

Default Agents Container 223

Default Groups Container 223

Default People Container 223

Default Role Permissions (ACIs) 218

Dynamic Administrative Roles ACIs 220

Enable Administrative Groups 220

Enable Compliance User Deletion 220

Enable Domain Component Tree 219

Managed Group Type 217

Search Filters for Deleted Objects 223

Show Containers In View Menu 217

Show Group Containers 217

Show People Containers 216

User Profile Service Class 222

Organization Attributes 224

Enable External Attributes Fetch 231

End User Profile Display Class 225

Event Listener Classes 230

Groups Default People Container 225

Groups People Container List 225

JSP Directory Name 227

Maximum Entries Displayed per Page 230

Maximum Results Returned From Search 227

Online Help Documents 227

Pre and Post Processing Classes 231

Required Services 228

Show Groups on User Profile Page 226

Show Roles on User Profile Page 225

Timeout For Search (sec.) 227

User Creation Default Roles 226

User Creation Notification List 229

User Deletion Notification List 229

User Group Self Subscription 226

User Modification Notification List 230

User Profile Display Class 225

User Profile Display Options 226

User Search Key 228

User Search Return Attribute 228

UserID and Password Validation Plugin
Class 231

View Menu Entries 227

Administrator Authentication Configuration 247

Administrator DN Starting View 342

Agents

Deleting 96

Alias Search Attribute Name 248

am.encrypted.pwd property 43

AM_ENC_PWD variable 43

am2bak command line tool 197

Backup Procedure 199

Syntax 197

amadmin command line tool 187

Syntax 188

amconfig script

deployment scenarios 42

operations for 29

syntax for 41

AMConfig.properties file 43

ampassword command line tool 203

Running on SSL 204

Syntax 203

amsamplesilent file 28

amsecuridd Helper

Syntax 210

amsecuridd helper 41

amserver command line tool 195

Syntax 195

amserver script 41

amserver.instance script 41

amunixd helper 41

Anonymous Authentication 145

Logging In With 146

Register and Enable 145

Anonymous Authentication Attributes 233

Organization Attributes

Authentication Level 234

Default Anonymous User Name 234

Valid Anonymous User List 233

- Application Server
 - configuration variables 35
 - support for 35
- Artifact Timeout 329
- Assertion Skew Factor For notBefore Time 329
- Assertion Timeout 329
- Attributes
 - Attribute Types 107
 - Dynamic Attributes 107
 - Global Attributes 108
 - Organization Attributes 107
 - Policy Attributes 108
 - User Attributes 107
- Auth Service URL 304
- Authentication
 - By Authentication Level 174
 - By Module 175
- authentication
 - methods
 - policy-based 140
- Authentication Configuration 168, 287
 - For Organizations 171
 - For Roles 172
 - For Services 173
 - For Users 174
 - User Interface 169
- Authentication Configuration Attributes 287
 - Organization Attributes
 - Authentication Configuration 287
 - Authentication Post Processing Class 289
 - Conflict Resolution Level 289
 - Login Failure URL 289
 - Login Success URL 288
- Authentication Level 255, 278
 - Anonymous Authentication 234
 - LDAP Authentication 255, 262
 - Membership Authentication 268
 - RADIUS Authentication 272
 - SafeWord Module Authentication Level 276
 - Unix Module Authentication Level 280
- Authentication Post Processing Class 253, 289
- Available Locales 315

B

- bak2am command line tool 201
 - Syntax 201
- Base DN 308
- BEA WebLogic Server
 - configuration variables 37
 - support for 29
- Bind DN 308
- Bind Password 309

C

- Certificate Authentication Attributes 237
 - Organization Attributes
 - Enable OCSP Validation 239
 - Field in Cert to Use to Access User Profile 241
 - HTTP Parameters for CRL Update 239
 - Issuer DN Attribute Used to Search LDAP for CRLs 239
 - LDAP Attribute for Profile ID 241
 - LDAP Server Principal Password 240
 - LDAP Server Principal User 240
 - LDAP Server Where Certificates are Stored 240
 - LDAP Start Search DN 240
 - Match Certificate in LDAP 238
 - Match Certificate to CRL 238
 - Other Certificate Field Used to Access User Profile 241
 - Subject DN Attribute Used to Search LDAP for Certificates 238
 - Use SSL for LDAP Access 241
- Certificate-based Authentication 146
 - Logging In With 148
 - Register and Enable 147
- Client Char Sets 315
- Client Detection Attributes 291
 - Global Attributes
 - Client Detection Class 294
 - Client Types 291
 - Default Client Type 294
 - Enable Client Detection 294
- Client Detection Class 294

- Client Types [291](#)
 - Command line tools
 - am2bak [197](#)
 - Backup procedure [199](#)
 - Syntax [197](#)
 - amadmin [187](#)
 - Syntax [188](#)
 - ampassword [203](#)
 - Running on SSL [204](#)
 - Syntax [203](#)
 - amsecuridd Helper
 - Syntax [210](#)
 - amserver [195](#)
 - Syntax [195](#)
 - bak2am [201](#)
 - Syntax [201](#)
 - VerifyArchive [207, 209](#)
 - Syntax [208](#)
 - Configurable Log Fields [299](#)
 - configuration variables
 - Application Server [35](#)
 - BEA WebLogic Server [37](#)
 - IBM WebSphere Server [39](#)
 - Identity Server [30](#)
 - Web Server [34](#)
 - Configure Later option, Java Enterprise System installer [28](#)
 - Configure Now option, Java Enterprise System installer [28](#)
 - Confirm Password [344](#)
 - Conflict Resolution Level [289](#)
 - Console See Identity Server Console
 - Containers [96](#)
 - Creating [96](#)
 - Deleting [97](#)
 - Cookie Domains [314](#)
 - Core Authentication
 - Global Attributes [243](#)
 - Default LDAP Connection Pool Size [244](#)
 - LDAP Connection Pool Size [244](#)
 - Pluggable Authentication Module Classes [244](#)
 - Supported Authentication Modules for Clients [244](#)
 - Organization Attributes [245](#)
 - Administrator [Authentication](#)
 - Configuration [247](#)
 - Alias Search Attribute Name [248](#)
 - Authentication Post Processing Class [253](#)
 - Default Authentication Level [254](#)
 - Default Authentication Locale [249](#)
 - Default Failure Login URL [253](#)
 - Default Success Login URL [252](#)
 - Email Address to Send Lockout Notification [251](#)
 - Enable Generate UserID Mode [253](#)
 - Enable Login Failure Lockout Mode [251](#)
 - Enable Persistent Cookie Mode [247](#)
 - Lockout Attribute Name [252](#)
 - Lockout Attribute Value [252](#)
 - Login Failure Lockout Count [251](#)
 - Login Failure Lockout Duration [252](#)
 - Login Failure Lockout Interval [251](#)
 - Organization [Authentication](#)
 - Configuration [250](#)
 - Organization Authentication Menu [246](#)
 - People Container For All Users [248](#)
 - Persistent Cookie Maximum Time [248](#)
 - User Naming Attribute [249](#)
 - User Profile [246](#)
 - User Profile Dynamic Creation Default Roles [247](#)
 - Warn User After N Failure [251](#)
 - Core Authentication Attributes [243](#)
 - Core Authentication Service [144](#)
 - Register and Enable [144](#)
 - Current Sessions
 - Interface [111](#)
 - Session Management
 - Terminating a Session [113](#)
 - Session Management Window [112](#)
- ## D
- Database Driver Name [299](#)
 - Database User Name [299](#)
 - Database User Password [299](#)
 - DC Node Attribute List [222](#)
 - Default Agents Container [223](#)
 - Default Anonymous User Name [234](#)
 - Default Authentication Level [254](#)

- Default Authentication Locale [249](#)
- Default Client Type [294](#)
- Default Failure Login URL [253](#)
- Default Groups Container [223](#)
- Default LDAP Connection Pool Size [244](#)
- Default People Container [223](#)
- Default Role Permissions (ACIs) [218](#)
- Default Success Login URL [252](#)
- Default User Roles [264](#)
- Default User Status [342](#)
- DEPLOY_LEVEL variable [30](#)
- deployment scenarios, Identity Server [42](#)
- DN for Root User Bind
 - LDAP Authentication [259](#)
 - Membership Authentication [266](#)
- DN to Start User Search
 - LDAP Authentication [259](#)
 - Membership Authentication [265](#)
- documentation
 - overview [20](#)
 - terminology [22](#)
 - typographic conventions [22](#)
- DSAME Console
 - Data Pane [71](#)
- DTD files
 - policy.dtd [121](#)
- Dynamic Administrative Roles ACIs [220](#)
- Dynamic Attributes
 - Administrator DN Starting View [342](#)
 - Default User Status [342](#)
 - Max Caching Time (Minutes) [337](#)
 - Max Idle Time (Minutes) [336](#)
 - Max Session Time (Minutes) [336](#)
 - User Preferred Language [342](#)
 - User Preferred Locale [342](#)
 - User Preferred Timezone [342](#)
- Dynamic Groups [217](#)

E

- Email Address [344](#)

- Email Address to Send Lockout Notification [251](#), [310](#)
- Employee Number [344](#)
- Enable Client Detection [294](#)
- Enable External Attributes Fetch [231](#)
- Enable Generate UserID Mode [253](#)
- Enable LDAP SSL [324](#)
- Enable Login Failure Lockout Mode [251](#)
- Enable OCSP Validation [239](#)
- Enable Password Reset [309](#)
- Enable Password Reset Failure Lockout [310](#)
- Enable Persistent Cookie Mode [247](#)
- Enable Personal Question [309](#)
- Enable Secure Logging [300](#)
- Enable SSL Access to LDAP Server
 - LDAP Authentication [261](#)
 - Membership Authentication [267](#)
- End User Profile Display Class [225](#)
- Event Listener Classes [230](#)

F

- Federation Management module, deploying [29](#)
- Field in Cert to Use to Access User Profile [241](#)
- Filtered Groups [218](#)
- First Name [343](#)
- Force Change Password on Next Login [310](#)
- Full Name [343](#)

G

- Global Attributes [243](#)
 - Artifact Timeout [329](#)
 - Assertion Skew Factor For notBefore Time [329](#)
 - Assertion Timeout [329](#)
 - Auth Service URL [304](#)
 - Available Locales [315](#)
 - Client Char Sets [315](#)
 - Client Detection Class [294](#)
 - Client Types [291](#)

- Configurable Log Fields [299](#)
 - Cookie Domains [314](#)
 - Database Driver Name [299](#)
 - Database User Name [299](#)
 - Database User Password [299](#)
 - DC Node Attribute List [222](#)
 - Default Agents Container [223](#)
 - Default Client Type [294](#)
 - Default Groups Container [223](#)
 - Default LDAP Connection Pool Size [244](#)
 - Default People Container [223](#)
 - Default Role Permissions (ACIs) [218](#)
 - Dynamic Administrative Roles ACIs [220](#)
 - Enable Administrative Groups [220](#)
 - Enable Client Detection [294](#)
 - Enable Compliance User Deletion [220](#)
 - Enable Domain Component Tree [219](#)
 - Enable Secure Logging [300](#)
 - LDAP Connection Pool Size [244](#)
 - Log File Location [298](#)
 - Log Signature Time [300](#)
 - Log Verification Frequency [300](#)
 - Logging Service URL [304](#)
 - Logging Type [299](#)
 - Login Service URL [314](#)
 - Logout Service URL [315](#)
 - Managed Group Type [217](#)
 - Max Log Size [298](#)
 - Maximum Number of Records [300](#)
 - Number of Files Per Archive [300](#)
 - Number of History Files [298](#)
 - Platform Locale [314](#)
 - Pluggable Authentication Module Classes [244](#)
 - Policy Service URL [304](#)
 - POST To Target URLs [333](#)
 - Profile Service URL [304](#)
 - Resource Comparator [318](#)
 - SAML Artifact Name [328](#)
 - SAML Assertion Manager Service URL [305](#)
 - SAML SOAP Service URL [305](#)
 - SAML Web Profile/Artifact Service URL [305](#)
 - SAML Web Profile/POST Service URL [305](#)
 - Search Filters for Deleted Objects [223](#)
 - Server List [313](#)
 - Session Service URL [304](#)
 - Show Containers In View Menu [217](#)
 - Show Group Containers [217](#)
 - Show People Containers [216](#)
 - Sign Assertion [328](#)
 - Sign SAML Request [328](#)
 - Sign SAML Response [328](#)
 - Site ID And Site Issuer Name [328](#)
 - Supported Authentication Modules for Clients [244](#)
 - Target Specifier [329](#)
 - Trusted Partner Sites [329](#)
 - Unix Helper Authentication Port [280](#)
 - Unix Helper Configuration Port [280](#)
 - Unix Helper Threads [280](#)
 - Unix Helper Timeout [280](#)
 - User Profile Service Class [222](#)
 - Globalization Setting Service Attributes [295](#)
 - Group Containers [98](#)
 - Creating [98](#)
 - Deleting [98](#)
 - Groups [77](#)
 - Adding to a Policy [80](#)
 - Create a Managed Group [78](#)
 - Dynamic Groups [217](#)
 - Filtered Groups [218](#)
 - Membership by Filter [77](#)
 - Membership by Subscription [77](#)
 - Static Groups [217](#)
 - Groups Default People Container [225](#)
 - Groups People Container List [225](#)
- ## H
- Header Frame [70](#)
 - Help link [71](#)
 - Home Address [344](#)
 - HTTP Basic Authentication [148](#)
 - Logging In With [150](#)
 - Register and Enable [149](#)
 - HTTP Basic Authentication Attributes [255](#)
 - Organization Attributes
 - Authentication Level [255](#)
 - HTTP Parameters for CRL Update [239](#)

I

- IBM WebSphere
 - support for [29](#)
- Identity Management [69](#)
 - Agents [95](#)
 - Deleting [96](#)
 - Containers [96](#)
 - Creating [96](#)
 - Deleting [97](#)
 - Group Containers [98](#)
 - Creating [98](#)
 - Deleting [98](#)
 - Groups [77](#)
 - Adding to a Policy [80](#)
 - Create a Managed Group [78](#)
 - Dynamic Groups [217](#)
 - Filtered Groups [218](#)
 - Membership by Filter [77](#)
 - Membership by Subscription [77](#)
 - Static Groups [217](#)
 - Identity Management Interface [73](#)
 - Identity Management View [72](#)
 - User Profile View [72](#)
 - Organizations [74](#)
 - Adding to a Policy [76](#)
 - Creating [75](#)
 - Deleting [76](#)
 - People Containers [97](#)
 - Creating [97](#)
 - Deleting [98](#)
 - Policies [95](#)
 - Properties [73](#)
 - Roles [84](#)
 - Adding to a Policy [93, 94](#)
 - Adding Users to [88](#)
 - Creating [85](#)
 - Deleting [94](#)
 - Removing Users from [92](#)
 - Services [83](#)
 - Creating a Template [83](#)
 - Registering [83](#)
 - Removing [84](#)
 - Users [81](#)
 - Adding to a Policy [82](#)
 - Adding to Services, Roles and Groups [81](#)
 - Creating [81](#)

- Deleting [82](#)
- Identity Server
 - Console [69](#)
 - installation overview [28](#)
 - related product information [24](#)
- Identity Server Console
 - Location Pane
 - Help link [71](#)
 - Location field [70](#)
 - Logout [71](#)
 - Modules [70](#)
 - Search Link [71](#)
 - Welcome [71](#)
 - Navigation Pane [71](#)
- Identity Server SDK, deploying [29](#)
- installation directory, Identity Server [28](#)
- installer, Java Enterprise System [28](#)
- instance, new Identity Server [42](#)
- Issuer DN Attribute Used to Search LDAP for CRLs [239](#)

J

- Java Enterprise System installer [28, 42](#)
- JSP Directory Name [227](#)

L

- Last Name [343](#)
- LDAP Attribute for Profile ID [241](#)
- LDAP Attribute Used to Retrieve User Profile [260, 266](#)
- LDAP Attributes Used to Search for a User to be Auth [260](#)
- LDAP Authentication Attributes [257](#)
 - Organization Attributes
 - Authentication Level [255, 262](#)
 - DN for Root User Bind [259](#)
 - DN to Start User Search [259](#)
 - Enable SSL Access to LDAP Server [261](#)

- LDAP Attribute Used to Retrieve User Profile 260
- LDAP Attributes Used to Search for a User to be Authenticated 260
- Password for Root User Bind 259, 266
- Primary LDAP Server 258
- Return User DN To Authenticate 261
- Search Scope 260
- Secondary LDAP Server 258
- User Search Filter 260
- LDAP Base DN 321
- LDAP Bind DN 321
- LDAP Bind Password 321
- LDAP Connection Pool Maximum Size 325
- LDAP Connection Pool Minimal Size 324
- LDAP Connection Pool Size 244
- LDAP Directory Authentication 150
 - Enabling Failover 152
 - Logging In With 151
 - Register and Enable 150
- LDAP Group Search Attribute 323
- LDAP Groups Search Filter 322
- LDAP Groups Search Scope 322
- LDAP Org Search Filter 321
- LDAP Org Search Scope 322
- LDAP Organization Search Attribute 323
- LDAP Roles Search Attribute 324
- LDAP Roles Search Filter 323
- LDAP Roles Search Scope 323
- LDAP Server and Port 320
- LDAP Server Principal Password 240
- LDAP Server Principal User 240
- LDAP Server Where Certificates are Stored 240
- LDAP Start Search DN 240
- LDAP Users Search Attribute 324
- LDAP Users Search Filter 322
- LDAP Users Search Scope 322
- Linux systems, base installation directory for 28
- Lockout Attribute Name 252
- Lockout Attribute Value 252
- Log File Location 298
- Log Signature Time 300
- Log Verification Frequency 300
- Logging Attributes 297
 - Global Attributes
 - Configurable Log Fields 299
 - Database Driver Name 299
 - Database User Name 299
 - Database User Password 299
 - Enable Secure Logging 300
 - Log File Location 298
 - Log Signature Time 300
 - Log Verification Frequency 300
 - Logging Type 299
 - Max Log Size 298
 - Maximum Number of Records 300
 - Number of Files Per Archive 300
 - Number of History Files 298
 - Logging Service URL 304
 - Logging Type 299
 - Login Failure Lockout Count 251
 - Login Failure Lockout Duration 252
 - Login Failure Lockout Interval 251
 - Login Failure URL 289
 - Login Service URL 314
 - Login Success URL 288
 - Logout 71
 - Logout Service URL 315

M

- Managed Group Type 217
- Managing Identity Server Objects 74
- Match Certificate in LDAP 238
- Match Certificate to CRL 238
- Max Caching Time (Minutes) 337
- Max Idle Time (Minutes) 336
- Max Log Size 298
- Max Session Time (Minutes) 336
- Maximum Entries Displayed per Page 230
- Maximum Number of Questions 309
- Maximum Number of Records 300
- Maximum Results Returned From Search 227
- Membership Authentication 152
 - Logging In With 153

- Register and Enable [152](#)
- Membership Authentication Attributes [263](#)
- Organization Attributes
 - Authentication Level [268](#)
 - Default User Roles [264](#)
 - DN for Root User Bind [266](#)
 - DN to Start User Search [265](#)
 - Enable SSL Access to LDAP Server [267](#)
 - LDAP Attribute Used to Retrieve User Profile [266](#)
 - LDAP Attributes Used to Search for a User to be Auth [266](#)
 - Minimum Password Length [264](#)
 - Primary LDAP Server [264](#)
 - Return User DN to Auth [267](#)
 - Search Scope [267](#)
 - Secondary LDAP Server [265](#)
 - User Search Filter [267](#)
 - User Status After Registration [264](#)
- methods
 - authentication
 - policy-based [140](#)
- Minimum Password Length [264](#)

N

- Naming Attributes [303](#)
 - Global Attributes
 - Auth Service URL [304](#)
 - Logging Service URL [304](#)
 - Policy Service URL [304](#)
 - Profile Service URL [304](#)
 - SAML Assertion Manager Service URL [305](#)
 - SAML SOAP Service URL [305](#)
 - SAML Web Profile/Artifact Service URL [305](#)
 - SAML Web Profile/POST Service URL [305](#)
 - Session Service URL [304](#)
- naming service
 - and policy [118](#)
- new installation, Identity Server [28](#)
- Normal Policy [119](#), [129](#), [134](#)
 - Modifying [129](#)
- NT Authentication [154](#)
 - Logging In With [155](#)

- Organization Attributes
 - NT Authentication Domain [270](#)
 - NT Authentication Host [270](#)
 - NT Module Authentication Level [270](#), [284](#)
- Register and Enable [155](#)
- NT Authentication Attributes [269](#)
- NT Authentication Domain [270](#)
- NT Authentication Host [270](#)
- NT Module Authentication Level [270](#), [284](#)
- Number of Files Per Archive [300](#)
- Number of History Files [298](#)

O

- Online Help Documents [227](#)
- operations, using amconfig [29](#)
- Organization Attributes [224](#)
 - Administrator Authentication Configuration [247](#)
 - Alias Search Attribute Name [248](#)
 - Authentication Configuration [287](#)
 - Authentication Level [255](#), [278](#)
 - Anonymous Authentication [234](#)
 - LDAP Authentication [255](#), [262](#)
 - Membership Authentication [268](#)
 - RADIUS Authentication [272](#)
 - Authentication Post Processing Class [253](#), [289](#)
 - Base DN [308](#)
 - Bind DN [308](#)
 - Bind Password [309](#)
 - Conflict Resolution Level [289](#)
 - Default Anonymous User Name [234](#)
 - Default Authentication Level [254](#)
 - Default Authentication Locale [249](#)
 - Default Failure Login URL [253](#)
 - Default Success Login URL [252](#)
 - Default User Roles [264](#)
 - DN for Root User Bind
 - LDAP Authentication [259](#)
 - Membership Authentication [266](#)
 - DN to Start User Search
 - LDAP Authentication [259](#)
 - Membership Authentication [265](#)
 - Email Address to Send Lockout Notification [251](#), [310](#)

- Enable External Attributes Fetch [231](#)
- Enable Generate UserID Mode [253](#)
- Enable LDAP SSL [324](#)
- Enable Login Failure Lockout Mode [251](#)
- Enable OCSP Validation [239](#)
- Enable Password Reset [309](#)
- Enable Password Reset Failure Lockout [310](#)
- Enable Persistent Cookie Mode [247](#)
- Enable Personal Question [309](#)
- Enable SSL Access to LDAP Server
 - LDAP Authentication [261](#)
 - Membership Authentication [267](#)
- End User Profile Display Class [225](#)
- Event Listener Classes [230](#)
- Field in Cert to Use to Access User Profile [241](#)
- Force Change Password on Next Login [310](#)
- Groups Default People Container [225](#)
- Groups People Container List [225](#)
- HTTP Parameters for CRL Update [239](#)
- Issuer DN Attribute Used to Search LDAP for CRLs [239](#)
- JSP Directory Name [227](#)
- LDAP Attribute for Profile ID [241](#)
- LDAP Attribute Used to Retrieve User Profile [260, 266](#)
- LDAP Attributes Used to Search for a User to be Auth [260](#)
 - Membership Auth [266](#)
- LDAP Base DN [321](#)
- LDAP Bind DN [321](#)
- LDAP Bind Password [321](#)
- LDAP Connection Pool Maximum Size [325](#)
- LDAP Connection Pool Minimal Size [324](#)
- LDAP Group Search Attribute [323](#)
- LDAP Groups Search Filter [322](#)
- LDAP Groups Search Scope [322](#)
- LDAP Org Search Filter [321](#)
- LDAP Org Search Scope [322](#)
- LDAP Organization Search Attribute [323](#)
- LDAP Roles Search Attribute [324](#)
- LDAP Roles Search Filter [323](#)
- LDAP Roles Search Scope [323](#)
- LDAP Server and Port [320](#)
- LDAP Server Principal Password [240](#)
- LDAP Server Principal User [240](#)
- LDAP Server Where Certificates are Stored [240](#)
- LDAP Start Search DN [240](#)
- LDAP Users Search Attribute [324](#)
- LDAP Users Search Filter [322](#)
- LDAP Users Search Scope [322](#)
- Lockout Attribute Name [252](#)
- Lockout Attribute Value [252](#)
- Login Failure Lockout Count [251](#)
- Login Failure Lockout Duration [252](#)
- Login Failure Lockout Interval [251](#)
- Login Failure URL [289](#)
- Login Success URL [288](#)
- Match Certificate in LDAP [238](#)
- Match Certificate to CRL [238](#)
- Maximum Entries Displayed per Page [230](#)
- Maximum Number of Questions [309](#)
- Maximum Results Returned From Search [227, 324](#)
- Minimum Password Length [264](#)
- NT Authentication Domain [270](#)
- NT Authentication Host [270](#)
- NT Module Authentication Level [270, 284](#)
- Online Help Documents [227](#)
- Organization Authentication Configuration [250](#)
- Organization Authentication Menu [246](#)
- Other Certificate Field Used to Access User Profile [241](#)
- Password Change Notification Option [309](#)
- Password for Root User Bind
 - LDAP Authentication [259](#)
 - Membership Authentication [266](#)
- Password Reset Failure Lockout Count [310](#)
- Password Reset Failure Lockout Duration [311](#)
- Password Reset Failure Lockout Interval [310](#)
- Password Reset Lockout Attribute Name [311](#)
- Password Reset Lockout Attribute Value [311](#)
- Password Reset Option [309](#)
- People Container For All Users [248](#)
- Persistent Cookie Maximum Time [248](#)
- Pre and Post Processing Classes [231](#)
- Primary LDAP Server [258, 264](#)
- RADIUS Server 1 [271](#)
- RADIUS Server 2 [272](#)
- RADIUS Server's Port [272](#)
- RADIUS Shared Secret [272](#)
- Required Services [228](#)
- Return User DN to Auth
 - Membership Authentication [267](#)
- Return User DN To Authenticate

- LDAP Authentication 261
- SafeWord Log File 276
- SafeWord Module Authentication Level 276
- SafeWord Server 275
- Search Filter 308
- Search Scope
 - LDAP Authentication 260
 - Membership Authentication 267
- Secondary LDAP Server 258, 265
- Secret Question 308
- SecurID ACE/Server Configuration Path 277
- SecurID Helper Authentication Port 278
- SecurID Helper Configuration Port 278
- Selected Policy Conditions 325
- Selected Policy Referrals 325
- Selected Policy Subjects 325
- Show Groups on User Profile Page 226
- Show Roles on User Profile Page 225
- Subject DN Attribute Used to Search LDAP for Certificates 238
- Subjects Result Time To Live 325
- Timeout 272
- Timeout For Search 324
- Timeout For Search (sec.) 227
- Unix Module Authentication Level
 - Unix Module Authentication Level 280
- Use SSL for LDAP Access 241
- User Creation Default Roles 226
- User Creation Notification List 229
- User Deletion Notification List 229
- User Group Self Subscription 226
- User Modification Notification List 230
- User Naming Attribute
 - Core Authentication 249
- User Profile 246
- User Profile Display Class 225
- User Profile Display Options 226
- User Profile Dynamic Creation Default Roles 247
- User Search Filter
 - LDAP Authentication 260
 - Membership Authentication 267
- User Search Key 228
- User Search Return Attribute 228
- User Status After Registration 264
- User Validation 308
- UserID and Password Validation Plugin
 - Class 231

- Valid Anonymous User List 233
- View Menu Entries 227
- Warn User After N Failure 251, 311
- Organization Authentication Configuration 250
- Organization Authentication Menu 246
- Organizations 74
 - Adding to a Policy 76
 - Creating 75
 - Deleting 76
- Other Certificate Field Used to 241
- overview
 - policy 116
 - policy agents 117
 - policy process 118
- overview, Identity Server installation 28
- owner and group, changing 44

P

- Password 343
- Password Change Notification Option 309
- password encryption key 43
- Password for Root User Bind
 - LDAP Authentication 259
 - Membership Authentication 266
- Password Reset Failure Lockout Count 310
- Password Reset Failure Lockout Duration 311
- Password Reset Failure Lockout Interval 310
- Password Reset Lockout Attribute Name 311
- Password Reset Lockout Attribute Value 311
- Password Reset Option 309
- Password Reset Service Attributes 307
 - Organization Attributes
 - Base DN 308
 - Bind DN 308
 - Bind Password 309
 - Email Address to Send Lockout Notification 310
 - Enable Password Reset 309
 - Enable Password Reset Failure Lockout 310
 - Enable Personal Question 309
 - Force Change Password on Next Login 310
 - Maximum Number of Questions 309

- Password Change Notification Option 309
 - Password Reset Failure Lockout Count 310
 - Password Reset Failure Lockout Duration 311
 - Password Reset Failure Lockout Interval 310
 - Password Reset Lockout Attribute Name 311
 - Password Reset Lockout Attribute Value 311
 - Password Reset Option 309
 - Search Filter 308
 - Secret Question 308
 - User Validation 308
 - Warn User After N Failure 311
- People Container For All Users 248
- People Containers 97
 - Creating 97
 - Deleting 98
- Persistent Cookie Maximum Time 248
- Platform Attributes 313
 - Global Attributes
 - Available Locales 315
 - Client Char Sets 315
 - Cookie Domains 314
 - Login Service URL 314
 - Logout Service URL 315
 - Platform Locale 314
 - Server List 313
- Platform Locale 314
- Pluggable Authentication Module Classes 244
- Policy 115
 - Creating for Peer and Suborganizations 128
 - Normal Policy 119
 - Adding Conditions 134
 - Adding Rules 129
 - Modifying 129
 - Referral Policy 121
 - Adding Referrals 137
 - Modifying 136
- policy
 - and naming service 118
 - DTD files
 - policy.dtd 121
 - overview 116
 - policy-based resource management (authentication) 140
 - process overview 118
- policy agents
 - overview 117
- Policy Configuration Attributes 317
 - Global Attributes
 - Resource Comparator 318
 - Organization Attributes
 - Enable LDAP SSL 324
 - LDAP Base DN 321
 - LDAP Bind DN 321
 - LDAP Bind Password 321
 - LDAP Connection Pool Maximum Size 325
 - LDAP Connection Pool Minimal Size 324
 - LDAP Group Search Attribute 323
 - LDAP Groups Search Filter 322
 - LDAP Groups Search Scope 322
 - LDAP Org Search Filter 321
 - LDAP Org Search Scope 322
 - LDAP Organization Search Attribute 323
 - LDAP Roles Search Attribute 324
 - LDAP Roles Search Filter 323
 - LDAP Roles Search Scope 323
 - LDAP Server and Port 320
 - LDAP Users Search Attribute 324
 - LDAP Users Search Filter 322
 - LDAP Users Search Scope 322
 - Maximum Results Returned From Search 324
 - Selected Policy Conditions 325
 - Selected Policy Referrals 325
 - Selected Policy Subjects 325
 - Subjects Result Time To Live 325
 - Timeout For Search 324
 - policy configuration service 138
 - Policy Service URL 304
 - policy.dtd 121
 - policy-based resource management (authentication) 140
 - POST To Target URLs 333
 - Pre and Post Processing Classes 231
 - Primary LDAP Server 258, 264
 - Profile Service URL 304
 - Properties 73
- R**
- RADIUS Authentication Attributes 271
 - Organization Attributes

- Authentication Level [272](#)
- RADIUS Server 1 [271](#)
- RADIUS Server 2 [272](#)
- RADIUS Server's Port [272](#)
- RADIUS Shared Secret [272](#)
- Timeout [272](#)
- RADIUS Server 1 [271](#)
- RADIUS Server 2 [272](#)
- RADIUS Server Authentication [156](#)
 - Logging In With [157](#)
 - Register and Enable [156](#)
- RADIUS Server's Port [272](#)
- RADIUS Shared Secret [272](#)
- reconfiguring Identity Server instance [44](#)
- Referral Policy [121](#)
 - Adding Referrals [137](#)
 - Modifying [136](#)
- Required Services [228](#)
- Resource Comparator [318](#)
- Return User DN To Auth
 - Membership Authentication [267](#)
- Return User DN to Authenticate [261](#)
- Roles [84](#)
 - Adding to a Policy [93, 94](#)
 - Adding Users to [88](#)
 - Creating [85](#)
 - Deleting [94](#)
 - Removing Users from [92](#)

S

- SafeWord Authentication [158](#)
 - Logging In With [159](#)
 - Register and Enable [159](#)
- SafeWord Authentication Attributes
 - Organization Attributes
 - SafeWord Log File [276](#)
 - SafeWord Logging Level [276](#)
 - SafeWord Module Authentication Level [276](#)
 - SafeWord Server [275](#)
 - SafeWord Server Verification Files Directory [275](#)
- SafeWord Log File [276](#)

- SafeWord Logging Level [276](#)
- SafeWord Module Authentication Level [276](#)
- SafeWord Server [275](#)
- SafeWord Server Verification Files Directory [275](#)
- SAML Artifact Name [328](#)
- SAML Assertion Manager Service URL [305](#)
- SAML Attributes [327](#)
 - Global Attributes
 - Artifact Timeout [329](#)
 - Assertion Skew Factor For notBefore Time [329](#)
 - Assertion Timeout [329](#)
 - POST To Target URLs [333](#)
 - SAML Artifact Name [328](#)
 - Sign Assertion [328](#)
 - Sign SAML Request [328](#)
 - Sign SAMLResponse [328](#)
 - Site ID And Site Issuer Name [328](#)
 - Target Specifier [329](#)
 - Trusted Partner Sites [329](#)
- SAML SOAP Service URL [305](#)
- SAML Web Profile/Artifact Service URL [305](#)
- SAML Web Profile/POST Service URL [305](#)
- Search Filter [308](#)
- Search Filters for Deleted Objects [223](#)
- Search Link [71](#)
- Search Scope
 - LDAP Authentication [260](#)
 - Membership Authentication [267](#)
- Secondary LDAP Server [258, 265](#)
- Secret Question [308](#)
- SecurID ACE/Server Configuration Path [277](#)
- SecurID Authentication [161](#)
 - Logging In With [162](#)
 - Register and Enable [162](#)
- SecurID Authentication Attributes [277](#)
 - Organization Attributes
 - Authentication Level [278](#)
 - SecurID ACE/Server Configuration Path [277](#)
 - SecurID Helper Authentication Port [278](#)
 - SecurID Helper Configuration Port [278](#)
- SecurID Helper Authentication Port [278](#)
- SecurID Helper Configuration Port [278](#)
- Selected Policy Conditions [325](#)
- Selected Policy Referrals [325](#)

- Selected Policy Subjects [325](#)
 - Server List [313](#)
 - Service Configuration
 - Service Configuration Module [108](#)
 - Service Configuration Interface [108](#)
 - Services [83](#)
 - Creating a Template [83](#)
 - Default Services Defined [102](#)
 - Certificate-based Authentication [103](#)
 - Administration [102](#)
 - Anonymous Authentication [102](#)
 - Authentication Configuration [104](#)
 - Client Detection [105](#)
 - Core Authentication [103](#)
 - Globalization Settings [105](#)
 - HTTP Basic Authentication [103](#)
 - LDAP Authentication [103](#)
 - Logging [105](#)
 - Membership Authentication [103](#)
 - Naming [105](#)
 - NT Authentication [103](#)
 - Platform [106](#)
 - Policy Configuration [106](#)
 - RADIUS Authentication [103](#)
 - SafeWord Authentication [104](#)
 - SAML [106](#)
 - SecurID Authentication [104](#)
 - Session [106](#)
 - Unix Authentication [104](#)
 - User [107](#)
 - Defined [101](#)
 - Registering [83](#)
 - Removing [84](#)
 - services
 - policy [116](#)
 - Session Attributes [335](#)
 - Dynamic Attributes
 - Max Caching Time (Minutes) [337](#)
 - Max Idle Time (Minutes) [336](#)
 - Max Session Time (Minutes) [336](#)
 - Session Service URL [304](#)
 - Show Containers In View Menu [217](#)
 - Show Group Containers [217](#)
 - Show Groups on User Profile Page [226](#)
 - Show People Containers [216](#)
 - Show Roles on User Profile Page [225](#)
 - Sign Assertion [328](#)
 - Sign SAML Request [328](#)
 - Sign SAML Response [328](#)
 - silent mode input file, amconfig script [28](#)
 - Site ID And Site Issuer Name [328](#)
 - Solaris
 - patches [24](#)
 - support [24](#)
 - Solaris systems, base installation directory for [28](#)
 - SSL
 - Configuring Identity Server For [57](#)
 - state file, Java Enterprise System installer [29](#)
 - Static Groups [217](#)
 - Subject DN Attribute Used to Search LDAP [238](#)
 - Subjects Result Time To Live [325](#)
 - support
 - Solaris [24](#)
 - Supported Authentication Modules for Clients [244](#)
 - Supported Language Locales [249](#)
- ## T
- Target Specifier [329](#)
 - Telephone Number [344](#)
 - Terminating a Session [113](#)
 - Timeout [272](#)
 - Timeout For Search [324](#)
 - Timeout For Search (sec.) [227](#)
 - Trusted Partner Sites [329](#)
- ## U
- unconfigure Identity Server instance [45](#)
 - un-install Identity Server instance [45](#)
 - Unique User IDs [346](#)
 - Unix Authentication [163](#)

- Logging In With [165, 168](#)
- Register and Enable [164](#)
- Unix Authentication Attributes [279](#)
 - Global Attributes
 - Unix Helper Authentication Port [280](#)
 - Unix Helper Configuration Port [280](#)
 - Unix Helper Threads [280](#)
 - Unix Helper Timeout [280](#)
 - Organization Attributes
 - Unix Module Authentication Level [280](#)
- Unix Helper Authentication Port [280](#)
- Unix Helper Configuration Port [280](#)
- Unix Helper Threads [280](#)
- Unix Helper Timeout [280](#)
- Use SSL for LDAP Access [241](#)
- User Attributes [341](#)
 - Service Management
 - Dynamic Attributes
 - Administrator DN Starting View [342](#)
 - Default User Status [342](#)
 - User Preferred Language [342](#)
 - User Preferred Locale [342](#)
 - User Preferred Timezone [342](#)
- User Profile Attributes [343](#)
 - Confirm Password [344](#)
 - Email Address [344](#)
 - Employee Number [344](#)
 - First Name [343](#)
 - Full Name [343](#)
 - Home Address [344](#)
 - Last Name [343](#)
 - Password [343](#)
 - Telephone Number [344](#)
 - Unique User IDs [346](#)
 - User Status [344](#)
- User Creation Default Roles [226](#)
- User Creation Notification List [229](#)
- User Deletion Notification List [229](#)
- User Group Self Subscription [226](#)
- User Modification Notification List [230](#)
- User Naming Attribute
 - Core Authentication [249](#)
- User Preferred Language [342](#)
- User Preferred Locale [342](#)
- User Preferred Timezone [342](#)
- User Profile [246](#)
- User Profile Attributes [343](#)
 - Confirm Password [344](#)
 - Email Address [344](#)
 - Employee Number [344](#)
 - First Name [343](#)
 - Full Name [343](#)
 - Home Address [344](#)
 - Last Name [343](#)
 - Password [343](#)
 - Telephone Number [344](#)
 - Unique User IDs [346](#)
 - User Status [344](#)
- User Profile Display Class [225](#)
- User Profile Display Options [226](#)
- User Profile Dynamic Creation Default Roles [247](#)
- User Search Filter
 - LDAP Authentication [260](#)
 - Membership Authentication [267](#)
- User Search Key [228](#)
- User Search Return Attribute [228](#)
- User Status [344](#)
- User Status After Registration [264](#)
- User Validation [308](#)
- UserID and Password Validation Plugin Class [231](#)
- Users [81](#)
 - Adding to a Policy [82](#)
 - Adding to Services, Roles, and Groups [81](#)
 - Creating [81](#)
 - Deleting [82](#)

V

- Valid Anonymous User List [233](#)
- VerifyArchive command line tool [207, 209](#)
 - Syntax [208](#)
- View Menu Entries [227](#)

W

- Warn User After N Failure [251, 311](#)
- Web Server
 - configuration variables [34](#)
 - support for [34](#)
- WEB_CONTAINER variable [34](#)
- WebLogic Server
 - configuration variables [37](#)
 - support for [29](#)
- WebSphere
 - configuration variables [39](#)
 - support for [29](#)
- Windows Desktop SSO Authentication [165](#)
 - Register and Enable [165](#)

