Administration Guide

Sun™ ONE Identity Server

Version 6.1

816-6773-10
December 2003

Contents

About This Guide

The *Sun™* ONE *Identity Server Administration Guide* offers information on how to customize Sun ONE Identity Server and integrate its functionality into an organization's current technical infrastructure. It also contains information about the programmatic aspects of the product and its API. This preface contains the following sections:

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 ONE servers and software. It is recommended that administrators understand the following technologies:

- Lightweight Directory Access Protocol (LDAP)
- Java™
- JavaServer Pages™ (JSP)
- HyperText Transfer Protocol (HTTP)
- HyperText Markup Language (HTML)
- eXtensible Markup Language (XML)

Because Sun ONE Directory Server is used as the data store in an Identity Server deployment, administrators should also be familiar with the documentation provided with that product. The latest Directory Server documentation can be accessed online.

Identity Server 6.1 Documentation Set

The Identity Server documentation set is separated into two core sets of manuals: the Sun ONE Identity Server 6.1 core application manuals and the Sun ONE Identity Server Policy Agents books.

Identity Server Core Documentation

The Identity Server documentation set contains the following titles:

- *Product Brief* provides an overview of the Identity Server application and its features and functions.
- *Migration Guide* provides details on how to migrate existing data and Sun ONE product deployments to the latest version of Identity Server. For instructions on installing Identity Server, see the *Sun Java Enterprise System 2003Q4 Installation Guide*.
- *Administration Guide* describes how to use the Identity Server console as well as manage user and service data via the command line.
- *Customization and API Guide* documents how to customize an Identity Server installation. It also includes instructions on how to augment the application with new services using the public APIs.
- *Deployment Guide* provides information on planning an Identity Server deployment within an existing information technology infrastructure.
- The *Release Notes* 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 ONE documentation web site. Updated documents will be marked with a revision date.

Identity Server Policy Agent Documentation Set

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 ONE 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 the documentation and welcome any comments and suggestions. Please email these comments to `docfeedback@sun.com`.

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

Below is a list of general terms 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 ONE Directory Server.
- *Application Server* refers to an installed instance of Sun ONE Application Server.
- *Web Server* refers to an installed instance of Sun ONE Web Server.
- *IdentityServer_base* is a variable place holder for the home directory where you have installed Identity Server.
- *DirectoryServer_base* is a variable place holder for the home directory where you have installed Sun ONE Directory Server.

- *ApplicationServer_base* is a variable place holder for the home directory where you have installed Sun ONE Application Server.
- *WebServer_base* is a variable place holder for the home directory where you have installed 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.

Related Information

In addition to the documentation provided with Identity Server, there are several other sets of documentation that might be helpful. Table 0-1 lists these and additional sources of information.

Table 0-1 Where to Find Related Sun ONE Resources

Information or Resource	Internet Location
Directory Server documentation	http://docs.sun.com/coll/S1_DirectoryServer_52
Web Server documentation	http://docs.sun.com/coll/S1_websvr61_en
Web Proxy Server documentation	http://docs.sun.com/prod/s1.webproxys#hic
Sun ONE Download Center	http://wwws.sun.com/software/download/
Sun ONE Technical Support	http://www.sun.com/service/sunone/software/index.html
Sun ONE Professional Services Information	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

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.

Part 1

Identity Server Console Guide

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

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

Chapter 1

Product Overview

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

Sun ONE Identity Server

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

Features of Identity Server

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

Service Configuration

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

Policy Management

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

SAML

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

Federation Management

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

Authentication

Identity Server provides a plug-in solution for user authentication. The criteria needed to authenticate a particular user is based on the authentication service configured for each organization in the Identity Server enterprise. Before being allowed access to a Identity Server session, a user must pass through authentication successfully.

Single Sign-On

Once the user is authenticated, Identity Server's API for Single Sign-On (SSO) takes over. Each time the authenticated user tries to access a protected page, the SSO API determines whether the user has the permissions required based on the user's authentication credentials. If the user is valid, access to the page is given without additional authentication. If not, the user will be prompted to authenticate again.

Policy Agents

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

Identity Management

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

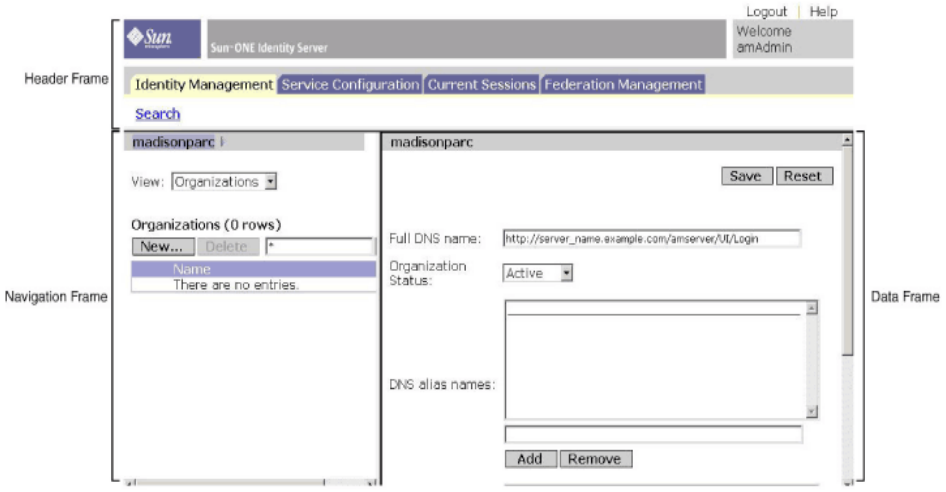
- Top-level Administrator with read and write access to all entries within the Identity Server enterprise.
- Top-level Help Desk Administrator with read access to all entries within the Identity Server enterprise and write access to user password attributes.
- Organization Administrator with read and write access to all entries within its organization.
- Organization Help Desk Administrator with read access to all entries within its organization.

- Container Administrator with read and write access to all Group Administrator with read and write access to all members of its group.

The Identity Server Console

The Identity Server console is divided into three sections: the Location frame, the Navigation frame and the Data frame. By using all three frames, the administrator is able to navigate the directory, perform user and service configurations and create policies.

Figure 1-1 The Identity Server Console



Header Frame

The Header frame runs along the top of the console. The tabs in the Header frame 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 3 of this documentation, the Attribute Reference Guide.

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

Navigation Frame

The Navigation frame 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 frame will have a corresponding *Properties* link. Selecting this link will render the entry's attributes in the Data frame to the right.) The View menu lists the directories under the selected directory object. Depending on the number of sub-directories, a paging mechanism is provided.

Data Frame

The Data frame is the right portion of the console. This is where all object attributes and their values are displayed and configured and where entries are selected for their respective group, role or organization.

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

Chapter 2

Identity Management

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

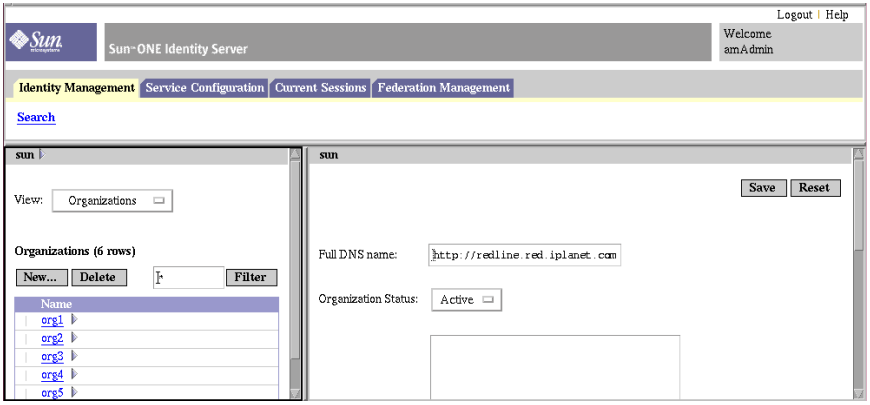
The Identity Management Interface

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.

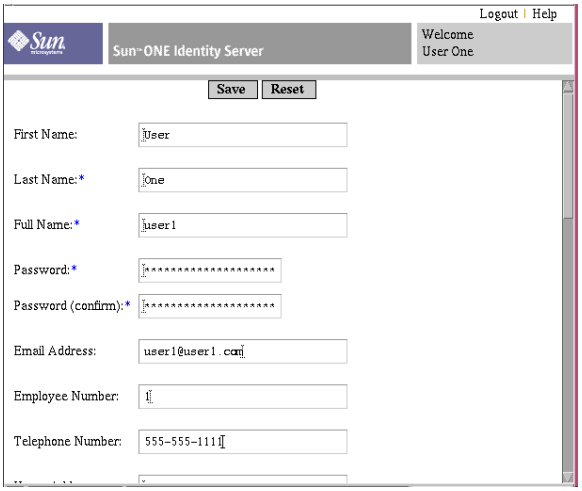
Figure 2-1 Identity Management View with Organization Properties Displayed



User Profile View

When a user who has not been assigned an administrative role authenticates to the Identity Server, the default view is 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 *Sun ONE Identity Server Customization and API Guide*.

Figure 2-2 User Profile View



Managing Identity Server Objects

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

Properties Function

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

See the *Sun ONE Identity Server Customization and API Guide* for information on how to extend an entry's properties.

Organizations

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

Create an Organization

1. Choose Organizations from the View menu in the Identity Management module.
2. Click New in the Navigation frame.

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

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

5. Enter the values, if desired, for the optional fields. The optional fields are:

   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`

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

   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 names 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.

6. Click Create.

   The new organization displays in the Navigation frame.

Delete an Organization

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

   All created organizations are displayed. To display specific organizations, enter a search string and click Filter.

2. Select the checkbox next to the name of the Organization to be deleted.

3. Click Delete.

NOTE There is no warning message when performing a delete. All entries within the organization will be deleted and you can not perform an undo.

Add an Organization to a Policy

Identity Server objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Modifying Policies" on page 87.

Groups

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

Membership By Subscription

When you specify group membership by subscription, a static group is created based on the Managed Group Type you specify. If the Managed Group Type value is `static`, group members are added to a group entry using the `groupOfNames` or `groupOfUniqueNames` object class. If the Managed Group Type value is `dynamic`, a specific LDAP filter is used to search and return only user entries that contain the `memberof` attribute. For more information, see "Managed Group Type" on page 157.

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

Create a Managed Group

1. Navigate to the organization (or group) where the group will be created.
2. Choose Groups from the View menu.
3. Click New.
4. Select the group type from within the Data frame.

   If a static subscription group is to be created, select Membership By Subscription.

   a. Enter a name for the group in the Name field. Click Next.
   b. Select the Users Can Subscribe to this Group attribute to allow users to subscribe to the group themselves.
   c. Add users to the group by selecting Add from the Member List.
   d. Enter the search criteria and click Filter. When the user list is returned, select the users you wish to add and click Submit. Adding users to the group is optional. They can be added after the group is created.
   e. Click Create.

   If a dynamic (LDAP filtered) group is to be created, select Membership By Filter.

   a. Enter a name for the group in the Name field. Click Next.
   b. Construct the LDAP search filter.
   c. The fields used to construct the filter use either an OR or AND operator. All the fields listed in the UI are used. If a field is left blank it will match all possible entries for that particular attribute.
   d. Click Create.

Delete a Managed Group

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

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 One Identity Server Migration Guide*.

Add a Group to a Policy

Identity Server objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Modifying Policies" on page 87.

Users

Users represent the identity of a person. Through the Identity Server Identity Management module, users can be created and deleted in organizations, containers and groups; added or removed from roles and/or groups; and you can assign services to the user.

Create a User

1. Navigate to the organization, container or people container where the user is to be created (or you can select the people container from the user creation page).
2. Choose Users from the View menu.
3. Click New.

   This displays the New User page in the Data frame.

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

   Information on the user profile attributes can be found in “User Attributes” on page 281.

5. Click Create.

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 frame, select the user you wish to modify and click the Properties arrow.
4. From the View menu in the Data frame, select Roles or Groups.

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

5. Select the role, or group that to which you wish to add the user, and click Save. Filtered roles and groups can not be displayed.

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.
3. In the Navigation frame, select the user you wish to modify and click the Properties arrow.
4. From the View menu in the Data frame, select Services.
5. Click Add to select the services you wish to assign to the user.
6. Click Save.

Delete a User

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

Add a User to a Policy

Identity Server objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Modifying Policies" on page 87.

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 register the service with the organization. After a service is registered, a template configured specifically for that organization must be created. For additional information, see Chapter 3, "Service Configuration"

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 *Sun ONE Identity Server Customization and API Guide*.

Register a Service

1. Navigate to the Organization where you will add services.

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

2. Choose Services from the View menu.

3. Click Register.

   The Data frame will display a list of services available to register to this organization.

4. Select the checkbox next to the services to be added.

5. Click Register. The registered services are displayed in the Navigation frame.

NOTE Only the services that are registered for the top-level organization are displayed at the role level.

Create a Template for a Service

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

   Choose Organizations from the View menu in the Identity Management module and select the organization from the Navigation frame.

2. Choose Services from the View menu.

3. Click the properties icon next to the name of the service to be activated.

   The Data frame displays the message *No Template Available For This Service. Do you want to create it?*

4. Click Create.

   A template is created for this service for the parent organization or role. The Data frame displays the default attributes and values for this service. Descriptions for the attributes for the default services are described in the "Attribute Reference Guide" on page 153.

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

Unregister a Service

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

   Choose Organizations from the View menu in Identity Management module and select the organization from the Navigation frame.

2. Choose Services from the View menu.

3. Select the checkboxes for the services to remove.

4. Click Unregister.

NOTE Services can not be unregistered at 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 "Policy Management" on page 84.

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

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

Users can view their assigned roles only if the Display User's Roles attribute is enabled in the Administration Service. For more information, see "Display User's Roles" on page 165.

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

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.

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.

Create a Filtered Role

1. In the Navigation frame, 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 frame.

   For descriptions of these roles, see "Dynamic Admin Roles ACIs" on page 160 of the Attribute Reference section.

3. Click New in the Navigation frame. The New Role template appears in the Data frame.
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 figure out where to start the user in the DIT. An administrative role notifies the console that the possessor of the role has administrative privileges; the service role notifies the console that the possessor is an end user.
7. Choose a default set of permissions to apply to the role from the Access Permission menu.

   The permissions provide access to entries within the organization. They are discussed in the section "Default Role Permissions (ACIs)" on page 158. (The default permissions shown are in no particular order.)

   Generally, the No Permissions ACI is assigned to Service roles, while Administrative roles are assigned any of the default ACIs.
8. Enter the information for the search criteria. The fields are:

   Logical Operator. Allows you to include an operator for any the fields you wish to include for the filter. `AND` returns users for all specified fields. `OR` 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=*))))
   ```

   If the filter is left blank, by default, the following role is created:

```
(objectclass = inetorgperson)
```

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

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

Create a Static Role

1. In the Navigation frame 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 frame.

   For descriptions of these roles, see "Dynamic Admin Roles ACIs" on page 160 of the Attribute Reference section.
3. Click New in the Navigation frame. The New Role template appears in the Data frame.
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 figure out where to start the user in the DIT. An administrative role notifies the console that the possessor of the role has administrative privileges; the service role notifies the console that the possessor is an end user.
7. Choose a default set of permissions to apply to the role from the Access Permission menu.

   The permissions provide access to entries within the organization. They are discussed in the section "Default Role Permissions (ACIs)" on page 158. (The default permissions shown are in no particular order.)

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

8. Click Create.

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

   The services available to the role are inherited from the parent organization for the role. You can create a service template for the role, if one does not already exist, by clicking the Edit link. If the service template already exists, the service properties are displayed and can be configured. For more information, see "Customize a Service to a Role" on page 48.

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 frame.
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:

   Logical Operator. Allows you to include an operator for any the fields you wish to include for the filter. AND returns users for all specified fields. OR 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).

   Return Users By. Allows you to specify the value returned by the search.
5. Click Filter to begin the search.
6. Choose the users from the names returned by selecting the checkbox next to the user name.
7. Click Save.

   The Users are now assigned to the role.

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

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 frame.

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

   The users are now removed from the role.

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 One Identity Server Migration Guide*.

Add a Role to a Policy

Identity Server objects are added to a policy through the policy's subject definition. When a policy is created or modified, organizations, roles, groups, and users can be defined as the subject in the policy's Subject page. Once the subject is defined, the policy will be applied to the object. For more information, see "Modifying Policies" on page 87.

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

You must register 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.

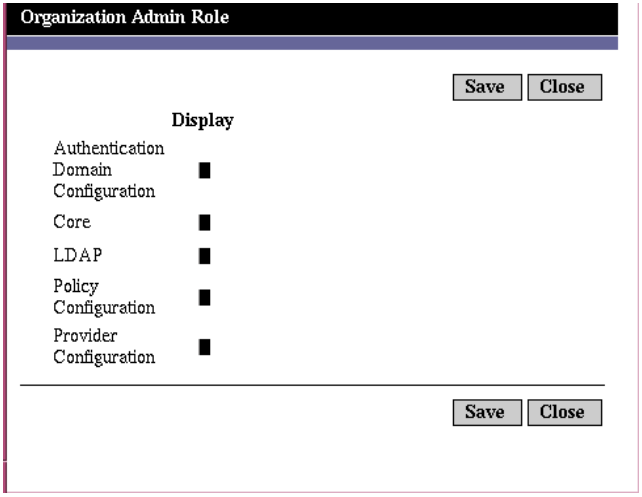
Customize Service Access

1. Click the Properties arrow for the role you wish to modify.
2. Select General from the View menu.
3. In the Role Properties page, click Edit in the Services listing.

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

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

Figure 2-3 Service Access Page



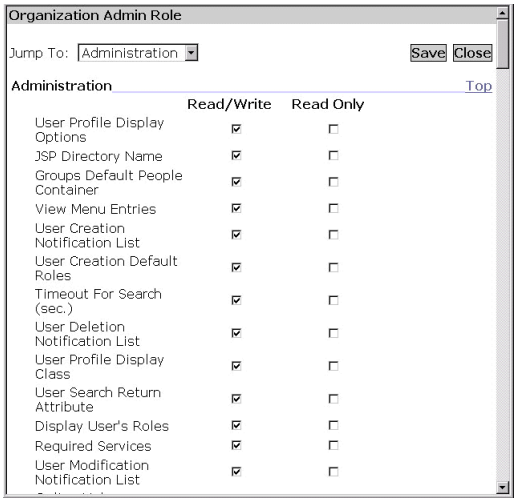
Customize Attribute Access

1. In the Role Properties page, click Edit in the Service Attribute listing. The Attribute Access page is displayed, as shown in Figure 2-4.

2. Use the Jump menu to display the attributes for a particular service.
3. Assign an access level to an attribute by selecting the Read / Write or Read Only check boxes.
4. Click Save.

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

Figure 2-4 Attribute Access Page



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

Delete a Role

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

   Choose Organizations from the View menu in Identity Management and select the organization from the Navigation frame. The Location path displays the default top-level organization and chosen organization.
2. Choose Roles from the View menu.
3. Select the checkbox next to the name of the role.
4. Click Delete.

Policies

Policies define rules to help protect an organization's web resources. Although policy creation, modification and deletion is performed through the Identity Management module, the procedures are described in "Policy Management" on page 84.

Containers

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

NOTE The display of containers is optional. To view containers you must select Display Containers in Menu in the Identity Server Administration service. For more information, see "Display Containers In Menu" on page 157.

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 frame.

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

4. Click Create.

Delete a Container

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

2. Choose Containers from the View menu.

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

4. Click Delete.

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

People Containers

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

NOTE The display of people containers is optional. To view People Containers you must select Show People Containers in the Identity Server Administration service. For more information, see "Show People Containers" on page 156.

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 frame.

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

4. Click Create.

Delete a People Container

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

2. Choose People Containers from the View menu.

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

4. Click Delete.

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

Group Containers

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

NOTE The display of group containers is optional. To view group containers you must select Show Group Containers in the Identity Server Administration service. For more information, see "Show Group Containers" on page 157.

Create a Group Container

1. Navigate to the organization or the group container which contains the group container to be created.
2. Choose group containers from the View menu.

   The default Groups was created during the organization's creation.
3. Click New.
4. Enter a value in the Name field and click Create.

   The new group container displays in the Navigation frame.

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 frame.
3. Select the checkbox next to the group container to be deleted.
4. Click Delete Selected.

Chapter 3

Service Configuration

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

Definition of a Service

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

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

IdentityServer_base`/SUNWam/dtd/`

For more information on defining a Identity Server service, see the *Sun ONE Identity Server Customization and API Guide.*

Identity Server Services

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

IdentityServer_base/SUNWamconfig/xml

or

/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 ten 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 module 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 module allows login through a personal digital certificate (PDC).

NOTE The Certificate authentication service is not supported for Application Server deployments for the 6.1 release.

Core

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

HTTP Basic

This module uses basic authentication, which is the HTTP protocol's built-in authentication support.

LDAP

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

Membership (Self-Registration)

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

NT

This module 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.

RADIUS

This module 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 ONE Application Server, you must configure Application Server's `service.policy` file. Instructions for this can be found in "Authentication Options" on page 97.

SafeWord

This module 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 ONE Application Server, you must configure Application Server's `service.policy` file. Instructions for this can be found in "Authentication Options" on page 97.

SecurID

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

Unix

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

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

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.

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.

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.

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 *Sun ONE Identity Server Customization and API 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

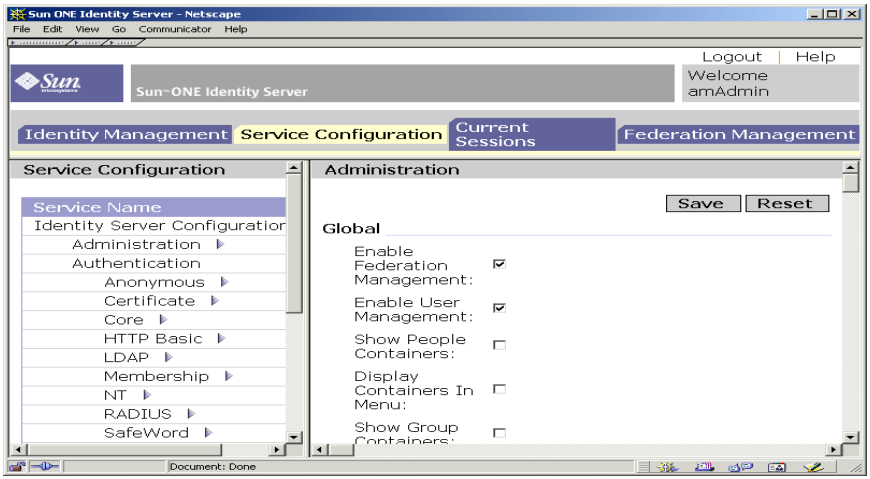
Policy attributes specify the access control actions (or privileges) associated with a service. They become a part of the rules when rules are added to a policy.

Service Configuration Interface

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

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

Figure 3-1 Service Configuration View



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

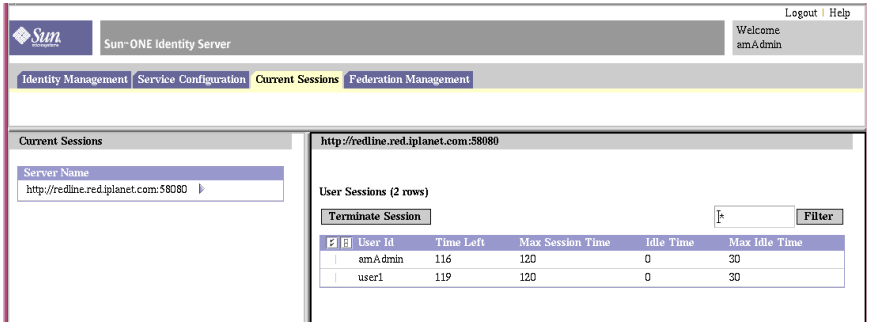
Current Sessions

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

The Current Sessions Interface

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

Figure 4-1 Current Sessions Interface



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 277 for more information.

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

Clicking the Refresh button will update the user session display.

Terminating a Session

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

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

Chapter 5

Federation Management

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

The features outlined in the LIberty Alliance Project specifications 1.0 are no longer supported. As there are virtually no 1.0 deployments, this does not have a serious impact.

This chapter contains the following sections:

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

IdentityServer_base/SUNWam/samples/liberty

Overview of Authentication Domains and Providers

The Federation Management module provides an interface for creating, modifying, and deleting authentication domains, remote providers and hosted providers. The following steps demonstrate a basic Federation Management model:

1. Create an authentication domain.

2. Create one or more hosted providers that belong to the created authentication domain.
3. Create one or more remote providers that belong to the created authentication domain. You must also include the metadata for the remote providers.
4. Establish a trusted relationship between the providers. A hosted provider can choose to trust a subset of providers, either hosted or remote, that belong to the same authentication domain.

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

Authentication Domains

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

Creating An Authentication Domain

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

   The Create Authentication Domain is displayed in the Data frame.
3. In the Create Authentication Domain window, enter the name of the Authentication Domain.
4. Enter a value for the description of the Authentication Domain.
5. Enter a value for the Writer Service URL.

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

   `http://example.com:8080/liberty/WriterServlet`
6. Enter a value for the Reader Service URL.

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

7. Choose a status of active or inactive.

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

8. Click Create.

   The new Authentication Domain displays in the Navigation frame.

Modifying An Authentication Domain

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

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

2. Modify the properties of the Authentication Domain.

3. Click Save.

Deleting An Authentication Domain

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

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

   All created Authentication Domains display in the Navigation frame.

2. Check the box next to the name of the Authentication Domain to be deleted.

3. Click Delete Selected.

NOTE There is no warning message when performing a delete.

Providers

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

Creating Remote Providers

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

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

   By default, when a Provider is created, it will be a service provider. You can optionally decide to create the remote provider as an identity provider by selecting the option described in Step 15.

2. Click New. The Create Remote Provider window is displayed.

3. Enter a value for the Provider ID.

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

4. Enter a description of the remote provider.

5. Enter the Security Key.

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

6. Enter the SOAP End Point URL.

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

7. Enter the Single Logout Service URL.

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

8. Enter the Single Logout Return URL.

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

9. Enter the Federation Termination Service URL.

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

10. Enter a value for the Federation Termination Return URL.

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

11. Define the Single Sign-On Service URL.

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

12. Enter the Name Registration Service URL.

    This field uses the Name Registration protocol that is used by a service provider to register its own Name Identifier while communicating to an identity provider. Registration occurs only after a federation session is established. This field defines the service URL used by a service provider to register a Name Identifier with an identity provider.

13. Enter the Name Registration Return URL.

    This field uses the Name Registration protocol that is used by a service provider to register its own Name Identifier while communicating to an identity provider. Registration occurs only after a federation session is established. The Name Registration Return URL is the URL to which the identity provider sends back the status of the registration.

14. Enter the Assertion Consumer URL.

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

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

16. Click Create.

    The new Provider displays in the Navigation frame.

Modifying Remote Providers

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

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

   By default the General view is displayed in the Navigation frame. Most of the fields displayed in the General view contain the data that was entered during the creation of the remote provider. The following additional field can be modified:

   Provider Succinct ID. This field uniquely identifies a service provider to an identity provider.

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

   ```
   $ echo providerID | openssl sha1
   ```

   If you modify any of the fields, click Save to save the changes.

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

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

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

   Name Registration After Federation. If enabled, this option allows for a service provider to participate in name registration after it has been federated. Name registration is a profile by which service providers specify a principal's name identifier that an identity provider will use when communicating to the service provider.

   Is Authentication Request Signed. This option, if enabled, specifies that the remote provider send signed authentication and federation requests. The identity provider will not process unsigned requests originated from the service provider.

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

   Federation Termination Profile. You can choose SOAP or HTTP/Redirect. This field specifies if the SOAP or HTTP/Redirect profile is to be used to notify of federation termination. This can be changed at any time during the life of the provider.

   Single Logout Profile. You can choose SOAP or HTTP Redirect. This field specifies if SOAP or HTTP Redirect is to be used to notify a logout event. This can be changed at any time during the life of the provider.

   Name Registration Profile. You can choose SOAP or HTTP/Redirect. This field specifies if the SOAP or HTTP/Redirect profile is to be used for name registration. This can be changed at any time during the life of the provider.

4. Click Save.

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

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

   Name Registration During SSO. If enabled, this option allows for an identity provider to participate in name registration during SSO. Name registration is a profile by which service providers specify a principal's name identifier that an identity provider will use when communicating to the service provider.

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

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

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

Creating Hosted Providers

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

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

   By default, when a Provider is created, it will be a service provider. You can optionally decide to create the remote provider as an identity provider by selecting the option described in Step 6.

2. Click New. The Create Hosted Provider window is displayed.

3. Enter a value for the Provider ID.

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

4. Enter a description for the hosted provider.
5. Enter the Alias for the provider.

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

   Federation Termination Service URL

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

   SOAP Endpoint URL

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

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

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

8. Enter the Provider URL.

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

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

    The new provider is displayed in the Navigation frame.

Modifying Hosted Providers

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

   By default the General view is displayed in the Navigation frame. Most of the fields displayed in the General view contain the data that was entered during the creation of the hosted provider. The following additional fields can be modified:

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

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

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

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

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

   Name Registration Service URL. This field uses the Name Registration protocol that is used by a service provider to register its own Name Identifier while communicating to an identity provider. Registration occurs only after a federation session is established. This field defines the service URL used by a service provider to register a Name Identifier with an identity provider.

   Name Registration Return URL. This field uses the Name Registration protocol that is used by a service provider to register its own Name Identifier while communicating to an identity provider. Registration occurs only after a federation session is established. The Name Registration Return URL is the URL to which the identity provider sends back the status of the registration.

   If you modify any of the fields, click Save.

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

   The Assertion Consumer URL field contains data that was entered during the creation of the remote provider. You can modify the following additional fields:

   Name Registration After Federation. If enabled, this option allows for a service provider to participate in name registration after it has been federated. Name registration is a profile by which service providers specify a principal's name identifier that an identity provider will use when communicating to the service provider.

   Is Authentication Request Signed. This option, if enabled, specifies that the hosted provider send signed authentication and federation requests. The identity provider will not process unsigned requests originated from the service provider.

   Federation Termination Profile. You can choose SOAP or HTTP/Redirect. This field specifies if the SOAP or HTTP/Redirect profile is to be used to notify of federation termination. This can be changed at any time during the life of the provider.

   Single Logout Profile. You can choose SOAP or HTTP Redirect. This field specifies if SOAP or HTTP Redirect is to be used to notify a logout event. This can be changed at any time during the life of the provider.

   Name Registration Profile. You can choose SOAP or HTTP/Redirect. This field specifies if the SOAP or HTTP/Redirect profile is to be used for name registration. This can be changed at any time during the life of the provider.

   Authentication Context. This field allows you to specify an authentication level for the authentication context to be used.

   If you modified any of the fields, click Save.

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

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

   Name Registration During SSO. If enabled, this option allows for an identity provider to participate in name registration during SSO. Name registration is a profile by which service providers specify a principal's name identifier that an identity provider will use when communicating to the service provider.

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

   Supported. Specifies if the identity provider supports the authentication context. The identity provider should support at least one authentication context.

   Context Reference. Defines the name of the authentication context. There are ten contexts defined in the Liberty protocol.

   Key. The query string sent to the /UI/Login (the Identity Server authentication servlet) will contain a key-value pair identifying the authentication mechanisms to be used. The possible key values are:

   - Module
   - Level
   - Role
   - Service

- User

Value. Defines the value of the key-value pair for the authentication mechanism.

Priority. Indicates the ordering determined by the identity provider for the Liberty-defined authentication contexts. If the identity provider does not support the authentication context requested by the service provider during the authentication request, it can use any other authentication context which is either at the same or higher priority level.

Click Save to save the changes.

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

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

5. Choose Trusted Providers from the View menu.

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

6. Choose Identity Server Configuration Attributes.

   The fields are as follows:

   Authentication Type. Remote / Local - Specifies if the hosted provider should contact an identity provider for authentication upon receiving an authentication request (Remote), or if authentication should be done by the hosted provider itself (Local).

   Single Signon/ Federation Profile. Specifies the profile used by the hosted provider for sending authentication requests. Identity Server provides the following protocols:

   - Browser Post - specifies a front-channel (http POST-based) protocol.

- Browser Artifact - Backchannel (non-browser) SOAP-based protocol.

Default Authentication Context. Specifies the authentication context to be used if the identity provider does not receive it as part of a service provider request. It also specifies the authentication context used by the service provider when an unknown user tries to access a protected resource. The default values are:

- Previous-Session
- Time-Sync-Token
- Smartcard
- MobileUnregistered
- Smartcard-PKI
- MobileContract
- Password
- Password-ProtectedTransport
- MobileDigitalID

- Software-PKI

Forced Authentication at Identity Provider. Indicates if the identity provider must reauthenticate (even during a live session) when an authentication request is received.

Request Identity Provider To Be Passive. If selected, this specifies that the identity provider must not interact with the principal and must interact with the user.

Organization DN. Specifies the storage location of the DN of the organization if each hosted provider chooses to manage users across different organizations leading to a hosted model.

Liberty Version URI. Specifies the version of the Liberty specification.

Name Identifier Implementation. Allows the option for a service provider to participate in name registration. Name registration is a profile by which service providers specify a principal's name identifier that an identity provider will use when communicating to the service provider.

Provider Home Page URL. Specifies the home page of the provider.

Single Signon Failure Redirect URL. Specifies the redirect URL for failed SSO.

Assertion Interval. Specifies the validity interval for the assertion issued by an identity provider. A principal will remain authenticated by the identity provider until the assertion interval expires.

Cleanup Interval. Specifies the interval of time to clear assertions that are stored in the identity provider.

Artifact Timeout. Specifies the timeout of an identity provider for assertion artifacts.

Assertion Limit. Specifies the number of assertions an identity provider can issue, or that can be stored.

7. Click Save.

Deleting Providers

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

   All created Providers display in the Navigation frame.

2. Check the boxes of the Providers you want to delete.
3. Click Delete Selected.

NOTE There is no warning message when performing a delete.

Chapter 6

Policy Management

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

This chapter contains the following sections:

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*.

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

NOTE It is acceptable to define an action without resources.

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

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

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

Referral Policy

An administrator might typically need to delegate one organization's policy definitions and decisions to another organization. (Alternately, policy decisions for a resource can be delegated to other policy products.) A *referral* policy controls this policy delegation for both policy creation and evaluation. It consists of one or more *rules* and one or more *referrals*. A rule defines the resource whose policy definition and evaluation is being referred. The referral defines the organization to which the policy definition and evaluation is being referred.

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.

There are two types of referrals bundled with Identity Server: peer organization and suborganization. They delegate to an organization on the same level and an organization on a sub-level, respectively. See "Creating Policies for Peer and Suborganizations" on page 94 for more information.

Policy Management

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

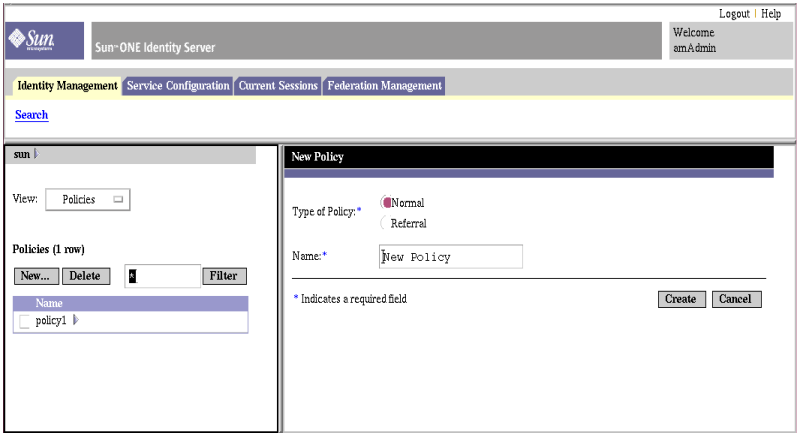
This chapter focuses on creating policies through the console.For more information on `amadmin`, see "The amadmin Command Line Tool" on page 129. For more information on the Policy API, see the "Policy Service" chapter in the *Sun ONE Identity Server Customization and API Guide.*

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

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

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

Figure 6-1 Policy View



Registering Policy Configuration Services

Registering a policy configuration service is the same as registering any type of service; it is done within the Identity Management interface. By default, the Policy Configuration service is automatically registered to the top-level organization. Any policy service you create must be registered to all organizations. Whenever you register the policy configuration service, you must enter the LDAP bind password in the template for all policies to take effect within an organization.

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 Register in the Navigation frame.

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

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

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

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

   To configure the policy service, click Create. Modify the Policy Configuration attributes. See "Policy Configuration Service Attributes" on page 259 for a description of these attributes. Click Save.

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

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

Creating Policies

Policies are created through the Identity Management interface.

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 polices for that suborganization. For more information, see “Creating Policies for Peer and Suborganizations” on page 94.

   It is not necessary to define all of the fields for normal or referral policies at this time. You may create the policy, then add rules, subjects, referrals, and so forth, later. For information on configuring normal and referral policies, see “Modifying Policies” on page 87.

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

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

7. By default, the General view is displayed.

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

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

Modifying Policies

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

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

   The policies that were created for that organization are displayed.

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

   By default, the General view is displayed.

Modify a Normal Policy

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

Adding Rules

Rules define the resource, actions and action values of the policy.

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

   The policies that were created for that organization are displayed.

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

   By default, the General view is displayed.

3. To define rules for the policy, select Rules from the View menu and click Add.

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

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

   The fields are:

   Service. Displays the 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.sunone.com`

   Currently, Policy Agents only support are `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://`.

   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 problem 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 “Adding Conditions” on page 91 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 Create to save the rule.
6. Repeat steps 1 - 5 to create additional rules.

7. All of the rules created for that policy are displayed in the table in the Rules view. Click Save to add the rules to the policy.

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

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

Adding Subjects

Subjects define the subject to which the policy will apply.

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

   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 if at least one of the subjects implies that the policy applies to the given identity. Regardless of whether or not Exclusive field is selected, the policy applies to the identity when all conditions defined in the policy are satisfied.
5. Perform a search in order to display the identities to add to the subject.

   The default (*) search pattern will display all qualified entries.
6. Select the identities that you wish to add for the subject and click Add to move them to the Selected list box. (or select Add All to add all of the identities).
7. Click Create.

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 Remove, then Save.

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

Adding Conditions

Conditions allows you to define constraints on the policy. For example, if you are defining policy for a paycheck application, you can define a condition on this action limiting access to the application only during specific hours. 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 pm, yet `http://org.example.com/finance/*.jsp` can be accessed by `org.example2.net` from 5 am to 11 pm. 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.

To add conditions to a normal policy:

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

   Click Next.

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

   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).

   Session

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

   Terminate Session. If selected, this field sets the termination of the user session if the session time exceeds the maximum allowed as defined in the Max Session Time field.

4. Once you have defined the condition, click Create.

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

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

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

Modify a Referral Policy

Through the Identity Management interface you can delegate an organization's policy definitions and decisions to another organization. (You can also delegate policy decisions for a resource to other policy products.) A *referral* policy controls this policy delegation for both policy creation and evaluation. It consists of a *rule* and the *referral* itself. If the policy service contains actions that do not require resources, referral policies cannot be created for suborganizations.

Adding Rules

Rules define the resource of the policy.

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

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

   Name. 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 are `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://`.
3. Click Create to save the rule.
4. Repeat steps 1 - 3 to create additional rules.
5. All of the rules created for that policy are displayed in the table in the Rules view. Click Save.

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

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

Adding Referrals

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

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

   Referral. Displays the current referral.

   Name. 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. Enter the organization name of the referral.
3. Click Create and Save.

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

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

Creating Policies for Peer and Suborganizations

In order to create policies for peer or suborganizations, you must first create a referral policy in the parent (or another peer) organization. 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).

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

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

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

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

2. Go to the Organization view and navigate to the suborganization `sun.com`.

3. Ensure that the policy configuration service is registered at the suborganization level, `sun.com`. For information, see “Registering Policy Configuration Services” on page 85.

4. Now that the resource is referred to `sun.com` by `isp`, normal policies can be created for the resource `http://www.sun.com`, or for any resource starting with `http://www.sun.com`.

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

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

Chapter 7

Authentication Options

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

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

Core Authentication

Identity Server provides, by default, ten different authentication services, as well as a Core authentication service. The Core authentication service provides overall configuration for the authentication service. Before registering and enabling Anonymous, Certificate-based, HTTP Basic, LDAP, Membership, NT, RADIUS, SafeWord, SecurID, and Unix authentication, the Core authentication must be registered and enabled. Chapter 19, "Core Authentication Attributes" contains a detailed listing of the Core attributes.

Registering and Enabling the Core Service

1. Navigate to the Navigation frame of the Organization for which the Core service is to be registered.
2. Choose Services from the View menu.
3. Click Add in the Navigation frame.

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

   The Core Authentication service will appear in the Navigation frame assuring the administrator that it has been registered.
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 frame.
6. Click Create.

   The Core attributes appear in the Data frame. Modify the attributes as necessary. An explanation of the Core attributes can be found in Chapter 19, "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 177). 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.

Registering and Enabling Anonymous Authentication

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

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

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

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

   The Anonymous Authentication service will appear in the Navigation frame assuring the administrator that it has been registered.
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 frame.
6. Click Create.

   The Anonymous Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 17, “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 190 must be modified to define Anonymous Authentication. This ensures that when the user logs in using `http(s)://hostname:port/DEPLOY_URI/Login?module=Anonymous&org=org_name`. To login without the Anonymous Authentication login window, use the following syntax:

`http(s)://hostname:port/DEPLOY_URI/Login?module=Anonymous&org=org_name&IDTolken1=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.

Certificate-based Authentication

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

There are a number of things that need to be accomplished before registering the Certificate-based Authentication service to an organization. First, the 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/s1websrv#hic
```

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

```
http://docs.sun.com/db/prod/s1appsrv#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.

Registering and Enabling Certificate-based Authentication

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

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

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

   A list of available services displays in the Data frame.
4. Select the checkbox for Certificate-based Authentication and click Add.

   The Certificate-based Authentication service will appear in the Navigation frame assuring the administrator that it has been registered.
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 frame.
6. Click Create.

   The Certificate-based Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 18, "Certificate Authentication Attributes" or by clicking the Help link in the upper right corner of the console.
7. Click Save.

Adding a Platform Server List for Certificate-based Authentication

In order to add you must log in to Identity Server as the Organization Administrator.

1. Select the Service Configuration module.
2. Choose the Platform service from the list of available services.
3. Add server information in the Server List attribute. For more information on the additional server attributes, see Chapter 34, "Platform Service Attributes"

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 190) 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 registered (registering the HTTP Basic module alone will not work). For more information, see "Registering and Enabling LDAP Authentication" on page 104. Once the user successfully authenticates, he/she will be able to re-authenticate without being prompted for username and password.

Registering and Enabling HTTP Basic Authentication

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

1. Navigate to the Navigation frame of the Organization for which HTTP Basic Authentication is to be registered.
2. Choose Services from the View menu.

   The Core service, if already registered, displays in the Navigation frame. If it is not already registered, it can be done concurrently with the HTTP Basic Authentication service.
3. Click Add in the Navigation frame.

   A list of available services displays in the Data frame.
4. Select the checkbox for HTTP Basic Authentication and click Add.

   The HTTP Basic Authentication service will appear in the Navigation frame assuring the administrator that it has been registered.
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 frame.
6. Click Create.

   The HTTP Basic Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 20, “HTTP Basic Authentication Attributes” or by clicking the Help link in the upper right corner of the console.
7. Click Save.

   The HTPP 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 190 must be modified to define HTTP Basic authentication. This ensures that when the user logs in using `http://hostname:port/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.

LDAP Directory Authentication

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

Registering and Enabling LDAP Authentication

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

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

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

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

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

6. Click Create.

   The LDAP Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 21, “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 Root User Bind attribute. By default, the `amldapuser` password that was entered during installation is used as the bind user.

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

8. Click Save.

   The LDAP Authentication service has been enabled.

Logging In Using LDAP Authentication

In order to log in using LDAP Authentication, the Core Authentication service attribute “Organization Authentication Modules” on page 190 must be modified to define LDAP Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=LDAP`, 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 and Port” on page 202 and “Secondary LDAP Server and Port” on page 202.

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 *Sun ONE Identity Server Customization and API 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 registered user. The user can also access the viewer interface, saved on the user profile database as authorization data and user preferences.

Registering and Enabling Membership Authentication

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

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

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

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

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

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 frame.

6. Click Create.

   The Membership Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 22, "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 190 must be modified to define Membership Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=Membership`, (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 register and enable the NT authentication service, you must obtain and install a Samba client to communicate with Identity Server on your Solaris system. For more information, see "NT Authentication Attributes" on page 213.

3. Register and enable the NT authentication service.

Registering and Enabling NT Authentication

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

1. Navigate to the Navigation frame of the Organization for which NT Authentication is to be registered.

2. Choose Services from the View menu.

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

3. Click Add in the Navigation frame.

   A list of available services displays in the Data frame.

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

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

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 frame.

6. Click Create.

   The NT Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 23, "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 190 must be modified to define NT Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=NT`, 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.

Registering and Enabling RADIUS Authentication

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

1. Navigate to the Navigation frame of the Organization for which RADIUS Authentication is to be registered.

2. Choose Services from the View menu.

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

3. Click Add in the Navigation frame.

   A list of available services displays in the Data frame.

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

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

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 frame.

6. Click Create.

   The RADIUS Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 24, "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 190 must be modified to define RADIUS Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=RADIUS`, 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.

Registering and Enabling SafeWord Authentication

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

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

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

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

   The SafeWord Authentication service will appear in the Navigation frame, assuring the administrator that it has been registered.
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 frame.
6. Click Create.

   The SafeWord Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 25, "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 190 must be modified to define SafeWord Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=SAFEWORD`, 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: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

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 161.

Registering and Enabling SecurID Authentication

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

1. Navigate to the Navigation frame of the Organization for which SecurID Authentication is to be registered.
2. Choose Services from the View menu.

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

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

   The SecurID Authentication service will appear in the Navigation frame, assuring the administrator that it has been registered.
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 frame.
6. Click Create.

   The SecurID Authentication attributes appear in the Data frame. Modify the attributes as necessary. An explanation of these attributes can be found in Chapter 26, “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 190 must be modified to define 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 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.ipadrs` (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.

The Unix authentication service is not available on the Windows platform.

Registering 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 27, “Unix Authentication Attributes”, or by clicking the Help link in the upper right corner of the console.
3. Click Save to save the new values for the attributes.

   You may log in to Identity Server as the Organization Administrator to enable Unix Authentication for an organization.
4. Navigate to the Navigation frame of the Organization for which Unix Authentication is to be registered.
5. Choose Services from the View menu.

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

   A list of available services displays in the Data frame.
7. Select the checkbox for Unix Authentication and click Add.

   The Unix Authentication service will appear in the Navigation frame, assuring the administrator that it has been registered.
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 date frame.
9. Click Create.

The Unix Authentication organization attribute appears in the Data frame. Modify the Authentication Level attribute as necessary. An explanation of this attribute can be found in Chapter 27, “Unix Authentication Attributes”, or by clicking the Help link in the upper right corner of the console.

10. Click Save.

    The Unix Authentication service has been enabled.

Logging In Using Unix Authentication

In order to log in using Unix Authentication, the Core Authentication service attribute “Organization Authentication Modules” on page 190 must be modified to define Unix Authentication. This ensures that when the user logs in using `http://hostname:port/deploy_URI/UI/Login?module=Unix`, the user will see the Unix Authentication login window. Based on the authentication type that is being used (such as service, role, user, organization), if the authentication module is configured as the default, there is no need to specify the module name in the URL.

Authentication Configuration

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

- organization
- role
- service
- user

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

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

Authentication Configuration User Interface

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

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

   The Add Module Window contains three files to define:

Module Name. 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
- HTTPBasic
- Membership
- NT
- RADIUS
- Unix

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

- REQUIRED - The authentication module is required to succeed. If it succeeds or fails, authentication continues to proceed down the authentication module list.

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

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

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

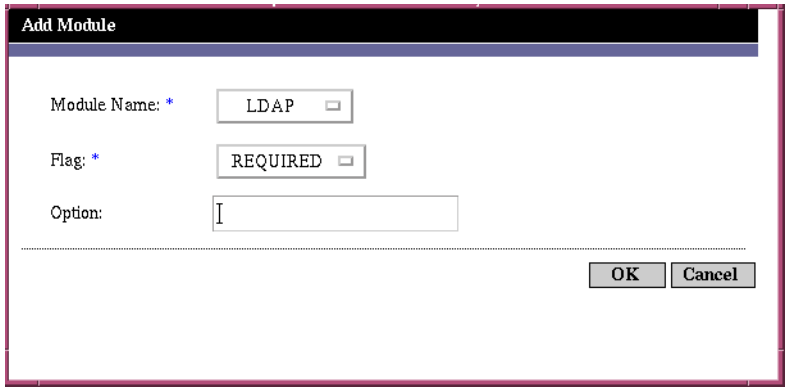
If you add multiple authentication modules and for each module the Flag is set to REQUIRED, the user must pass all authentication requirements before being granted access.

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

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

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

Figure 7-1 Add Module List Window For A User



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

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

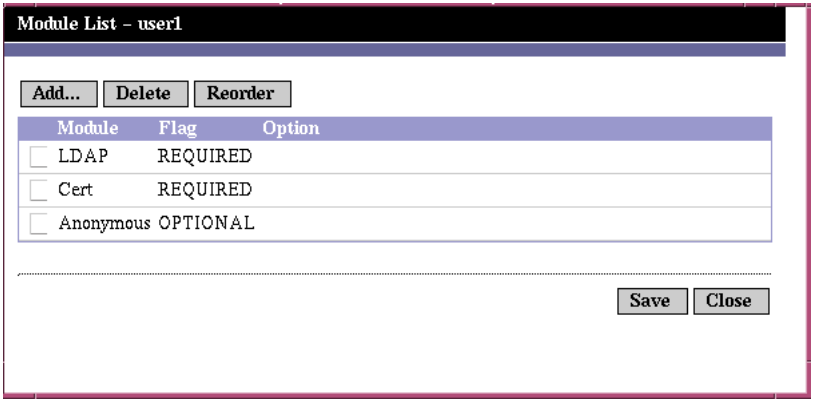
   To change the order of the authentication modules:

   a. Click the Reorder button.

   b. Select the module you wish to reorder.

   c. Use the Up and Down buttons to place it in the desired position.

Figure 7-2 Module List Window For A User



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 registering the Core Authentication service to the organization.

To configure the organization's authentication attributes:

1. Navigate to the organization for which you will configure the authentication attributes.
2. Select Services from the View menu.
3. Click the Core Properties arrow in the service listing.

   The Core authentication attributes are displayed in the Data frame.

4. Click the edit link next to the Admin Authenticator attribute. This allows you to define the authentication services for administrators only. An administrator is a user who needs access to the Identity Server console. This attribute can be used if the authentication module for administrators needs to be different from the module for end users. The default authentication module is LDAP.

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

5. Click the Edit link next to the Organization Authentication Configuration attribute. This allows you to define authentication modules for all users within the organization. The default authentication module is LDAP.

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

Authentication Configuration for Roles

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

1. Navigate to the organization for which you will configure the authentication attributes.

2. Choose Roles from the View menu.

3. Select the role for which to set the authentication configuration and click on the Properties arrow.

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

4. Select Services from the View menu in the Data frame.

5. Modify the Authentication Configuration attributes as necessary. An explanation of these attributes can be found in Chapter 28, "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 registering the Authentication Configuration service. To do so:

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

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

2. Click Add in the Navigation frame.

   A list of available services is displayed in the Data frame.

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

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

4. Click the Authentication Configuration Properties arrow.

   The Service Instance List is displayed in the in the Data frame.

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 28, “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 frame.

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

   The User Profile is displayed in the data frame.

.

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 frame.
5. Click on the Edit link next to the User Authentication Configuration attribute to define the authentication modules for the user.
6. Click Save.

Authentication By Authentication Level

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

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

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

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

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

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

Authentication By Module

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

```
http://hostname:port/deploy_URI/UI/Login?module=module_name
```

Before the authentication module can be accessed, the Core authentication service attribute Organization Authentication Modules must be modified to include the authentication module name. If the authentication module name is not included in this attribute, the “authentication module denied” page will be displayed when the user attempts to authenticate. For more information, see “Organization Authentication Modules” on page 190.

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 registering the Authentication Configuration service for identity objects, see “Authentication Configuration” on page 118.

Chapter 8

Password Reset Service

Sun™ ONE Identity Server 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

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 the Password Reset Service 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).

   The remaining attributes are optional. Descriptions of the Password Reset attributes can be found in “Password Reset Service Attributes” on page 249 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 249. 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 (minutes) attribute is greater than zero and the Password Reset Failure Lockout Mode 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 Password Reset Failure Lockout Mode 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.

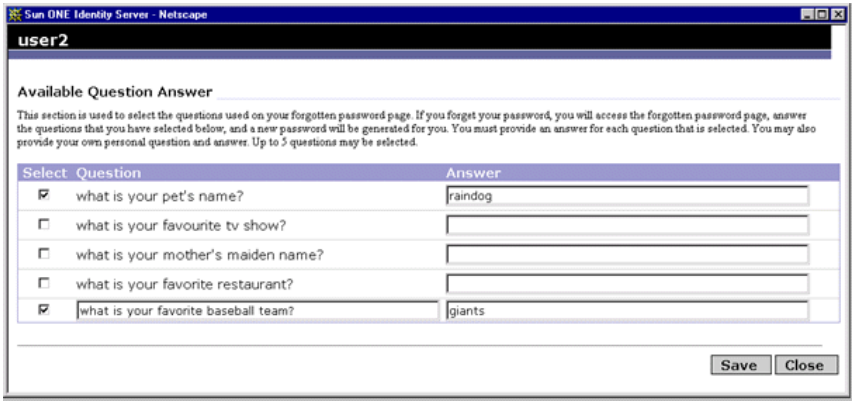
Customizing Password Reset

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 8-1 Available Questions Answer Screen with Personal Question Enabled



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 8-2 Password Questions for User Screen

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

Part 2

Command Line Reference Guide

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

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

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

```
IdentityServer_base/SUNWam/bin
```

Chapter 9

The amadmin Command Line Tool

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

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 *Sun One Identity Server Programmer'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 | --password file *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 *IdentityServer_base*`/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-st
able-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 *Sun ONE Identity Server Programmer'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 9-1 on page 133.

Code Example 9-1 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.

Creating Policies with amadmin

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

To add policies using `amadmin`, the policy XML file must be developed following the `policy.dtd`. (`policy.dtd` is described in the *Sun ONE Identity Server Customization and API Guide*) 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 Organization, LDAP groups', LDAP roles' and LDAP users' subjects; that Identity Server roles exist while creating IdentityServerRoles subjects; and that the relevant organizations exist while creating sub organization or peer organization referrals.

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.

Chapter 10

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

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

amserver Syntax

The generic syntax for the tools is:

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

amserver Commands for Solaris

create

`create` is a command that is used to create a new instance of Identity Server.The amserver script should be run as root. To create instances run amserver script `./amserver create`. Detailed steps for creating multiple server instances are described in "Using amserver for Multi-Server Installer Administration (Web Server Instances only)" on page 137. This command is only applicable for Web Server instances.

startall

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

```
IdentityServer_base/SUNWam/bin/amserver.instance_name start
```

stopall

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

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

delete

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

amserver Commands for Windows 2000

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

start

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

stop

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

NOTE `stop` and `start` may not function correctly with the new container- independent deployments. Use `stop` and `start` on the container if you experience this behavior.

restart

`restart` is a command that restarts Identity Server

amserver cannot stop or start Directory Server. You may need to restart it manually. It can only restart a Web Server instance. For other web containers, this command only restarts the authentication helpers.

Using amserver for Multi-Server Installer Administration (Web Server Instances only)

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

To install multiple instances:

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

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

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

Please enter the name of the server instance: instance1

Please enter the port number: 81

Do you want to create more server instances? y/[n]

Installing... please wait....

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

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

      IdentityServer_base/SUNWam/servers/https-*instance_name*

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

IdentityServer_base/SUNWam/servers/web-apps-*instance_name*

c. The IdentityServer_base/SUNWam/bin directory holds the instance specific version of amServer. For example:

amserver.*instance_name*

d. A copy of the Identity Server configuration file is created in IdentityServer_base/SUNWam/lib/AMConfig-*instance_name*.properties.

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

S55amserver.*instance_name*

K55amserver.*instance_name*

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

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

./amserver startall

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

IdentityServer_base/SUNWam/bin/amserver.*instance_name* start

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

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

./amserver stopall

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

IdentityServer_base/SUNWam/bin/amserver.*instance_name* stop

4. Invoke the Delete Command option by entering:

./amserver delete

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

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

   `Edit IdentityServer_base/SUNWam/lib/AMConfig-`*instan ce_name*`.properties`

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

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

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

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

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

8. Configure Identity Server to run in SSL mode. Instructions for this found in Appendix B, “Configuring Identity Server in SSL Mode” of this manual.

9. Enter the following command to start all of the Identity Server instances:

   `./amserver startall`

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

   `./amserver-`*instance* `start`

TO BE ADDED FOR 6.2!!!!!!!!!

Multiple Instances of Identity Server Do Not Contain Map Additions

If you have customized authentication screens and are also using `amserver` to create new Identity Server instances, MAP does not update Identity Server’s `services.war` (We Archive file), so the newly created instance does not contain MAP additions.

Workaround

Update the services.war file. By default, it is in the following location:

IdentityServer_base`/SUNWam/services.war`

To update the services.war file, enter the following command:

`jar -uvf services.war` *IdentityServer_base*`/SUNWam/services.war`

The uvf option will replace the old file with the new modified one. For example:

```
cd /opt/SUNWam

jar -uvf services.war index.html

rm index.html
```

The following files can be modified:

- JSPs (IdentityServer_base/SUNWam/web-apps/services/config/auth/default/*.jsp)
- javascripts (IdentityServer_base/SUNWam/web-apps/services/js/*.js)
- images (IdentityServer_base/SUNWam/web-apps/services/login_images/*.gif)
- Cascading style sheets (IdentityServer_base/SUNWam/web-apps/services/css/*.css)
- xml files (IdentityServer_base/SUNWam/web-apps/services/config/auth/default/*.xml)

Web Container Redeployment

To redeploy the .war file to the Application Server web container, enter the following commands:

```
asadmin deploy -u $IAS7_ADMIN -w $IAS7_ADMINPASSWD -H $SERVER_HOST -p $IAS7_ADMINPORT
--type web $SECURE_FLAG --contextroot

$SERVER_DEPLOY_URI --name amserver --instance $IAS7INSTANCE
${BASEDIR}/${PRODUCT_DIR}/services.war
```

To redeploy the .war file to the BEA WebLogic web container, enter the following commands:

```
java weblogic.deploy -url $SERVER_URL -component ${SERVER_DEPLOY_URI}:${WL61 _SERVER}
deploy $WL61_ADMINPASSWD

${SERVER_DEPLOY_URI}

${BASEDIR}/${PRODUCT_DIR} /services.war
```

To redeploy the .war file to the BEA WebLogic web container, see the deployment documentation at the following location:

http://www-3.ibm.com/software/webservers/studio/doc/v40/studioguide/en/html/sdsscenario1.html

Chapter 11

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

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

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.

Chapter 13

The ampassword Command Line Tool

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

- The ampassword Command Line Executable
- Running ampassword on SSL

The ampassword Command Line Executable

Identity Server contains an `ampassword` utility under `$installroot/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.

ampassword 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.

Running ampassword on SSL

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

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

   `IdentityServer_base/SUNWam/config/ums`

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

3. Change the `type` attribute to SSL.

   For example:

```
<iPlanetDataAccessLayer>

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

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

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

        <DirDN>

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

```
        </DirDN>

        <DirPassword>

                  AQIC5wM2LY4Sfcy+AQBQxghVwhBE92i78cqf

         </DirPassword>

</User> ...
```

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.

Chapter 14

The VerifyArchive Command Line Tool

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

- The VerifyArchive Command Line Executable

The VerifyArchive Command Line Executable

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

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

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

VerifyArchive Syntax

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

`VerifyArchive -l logName -p` *path* `-u` *uname* `-w` *password*

VerifyArchive Options

logName

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

path

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

uname

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

password

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

Chapter 15

The amsecuridd Helper

This chapter provides information on the `amsecuiridd` helper and contains the following section:

- The amsecuridd Helper Command Line Executable
- Running the amsecuridd helper

The amsecuridd Helper Command Line Executable

The Identity Server SecurID authentication module is implemented using the Security Dynamic ACE/Client C API and the `amsecruidd` 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 accross 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/lib/). 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 utitility, 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`.

Part 3

Attribute Reference Guide

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

- Platform Service Attributes
- Policy Configuration Service Attributes
- SAML Service Attributes
- Session Service Attributes
- User Attributes

Chapter 16

Administration Service Attributes

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

- Global Attributes
- Organization Attributes

Global Attributes

The global attributes in the Administration Service are:

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

- Admin Groups Enabled
- Compliance User Deletion Enabled
- Dynamic Admin Roles ACIs
- User Profile Service Classes
- DC Node Attribute List
- Search Filters for Deleted Objects

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.

Display Containers In Menu

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

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

Show Group Containers

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

Managed Group Type

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

- A static group explicitly lists each group member using the `groupOfNames` or `groupOfUniqueNames` object class. The group entry contains the `uniqueMember` attribute for each member of the group. Members of static groups are manually added; the user entry itself remains unchanged. Static groups are suitable for groups with few members.
- A dynamic group uses a `memberOf` attribute in the entry of each group member. Members of dynamic groups are generated through the use of an LDAP filter which searches and returns all entries which contain the `memberOf` attribute. Dynamic groups are suitable for groups that have a very large membership.

- A filtered group uses an LDAP filter to search and return members that meet the requirement of the filter. For instance, the filter can generate members with a specific uid (`uid=g*`) or email address (`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.

Domain Component Tree Enabled

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

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

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

`dc=sun,dc=com,o=internet,`*root suffix*

The DC tree may optionally have its own root suffix configured by setting `com.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.

Admin Groups Enabled

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

Compliance User Deletion Enabled

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

Dynamic Admin Roles ACIs

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

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

Container Help Desk Admin

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

Organization Help Desk Admin

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

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

Container Admin

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

Organization Policy Admin

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

People Container Admin

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

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

Group Admin

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

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

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.

Organization Attributes

The organization attributes in the administration service are:

- Groups Default People Container
- Groups People Container List
- User Profile Display Class
- Display User's Roles
- Display User's Groups
- User Group Self Subscription
- User Profile Display Options
- User Creation Default Roles

- View Menu Entries
- Maximum Results Returned From Search
- Timeout For Search (sec.)
- JSP Directory Name
- Online Help Documents
- Required Services
- User Search Key
- User Search Return Attribute
- User Creation Notification List
- User Deletion Notification List
- User Modification Notification List
- Maximum Entries Per Page
- Display Options
- Event Listener Classes
- Pre and Post Processing Classes
- External Attributes Fetch Enabled

Groups Default People Container

This field specifies the default People Container where users will be placed when they are created. There is no default value. A valid value is the DN of a people container. See the note under Groups People Container List attribute for the People Container fallback order.

Groups People Container List

This field specifies a list of People Containers from which a Group Administrator can choose when creating a new user. This list can be used if there are multiple People Containers in the directory tree. (If no People Containers are specified in this list or in the Groups Default People Container field, users are created in the default Identity Server people container, `ou=people`.) There is no default value for this field. The syntax for this attribute is as follows:

group name|*dn of people container*

NOTE When a user is created, this attribute is checked for a container in which to place the entry. If the attribute is empty, the Groups Default People Container attribute is checked for a container. If the latter attribute is empty, the entry is created under `ou=people`.

User Profile Display Class

This attribute specifies the Java class used by the Identity Server console when it displays the User Profile pages.

Display User's Roles

This option specifies whether to display a list of roles assigned to a user as part of 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`.

Display User's Groups

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`.

User Group Self Subscription

This option specifies whether users can add themselves to groups that are open to subscription. If the value is false, the user profile page allows the user's group membership to be modified only by an administrator. The default value is `false`.

NOTE This option applies only when the Display User's Groups option is selected.

User Profile Display Options

This menu specifies which service attributes will be displayed in the user profile page. An administrator can select from the following:

- `UserOnly` — Display viewable User schema attributes for services assigned to the user.

  User service attribute values are viewable by the user when the attribute contains the keyword Display. See the *Sun ONE Identity Server Customization and API Guide* for details.

- `Combined` — Display viewable User and Dynamic schema attributes for services assigned to the user.

User Creation Default Roles

This listing defines roles that will be assigned to newly created users automatically. There is no default value. An administrator can input the DN of one or more roles.

NOTE This field only takes a full Distinguished Name address, not a role name.

View Menu Entries

This field lists the Java classes of services that will be displayed in the View menu at the top of the console. The syntax is `i18N key | java class name.` (The i18N key is used for the localized name of the entry in the View menu.)

Maximum Results Returned From Search

This field defines the maximum number of results returned from a search. The default value is 100.

CAUTION Use caution when setting this attribute to large value. For sizing limits, see the *Sun ONE Directory Server Installation and Tuning Guide* at the following location:

`http://docs.sun.com/db/doc/816-6697-10`

Timeout For Search (sec.)

This field defines the amount of time (in number of seconds) that a search will continue before timing out. It is used to stop potentially long searches. After the maximum search time is reached, an error is returned. The default is 5 seconds.

JSP Directory Name

This field specifies the name of the directory that contains the `.jsp` files used to construct the console, to give an organization a different appearance (customization). The `.jsp` files need to be copied into the directory that is specified in this field.

Online Help Documents

This field lists the online help links that will be created on the main Identity Server help page. This allows other applications to add their online help links in the Identity Server page. The format for this attribute is as follows:

linki18nkey | *html page to load when clicked* | *i18n properties file*

For example:

```
IdentityServer Help | /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 19-1 on page 193 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/Identity-Server/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 19-1 on page 193 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/Identity-Server/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 19-1 on page 193 for a list of locales.

NOTE The attribute name is the same as it appears in the Directory Server schema, and not as the display name in the console.

Maximum Entries Per Page

This attribute allows you to define the maximum rows that can be displayed per page. The default is 25. For example, if a user search returns 100 rows, there will be 4 pages with 25 rows displayed in each page.

Display Options

This attribute allows you to add values to configure the display options in the Identity Server console. Enter the vaule and click Add to configure the display options. The possible values are as follows:

Table 16-1 Display Options Values

Parameter	Description and Syntax
generateUserCN	When set to true, this parameter dynamically generates the User CN when the user is created. The default is false. Syntax: `generateUserCN=[false|true]`
userAttributeNameForProfileTitle	Determines the value of the user attribute displayed on the title of the User Profile Page. uid is the default. Syntax: `userAttributeNameForProflleTitle=[uid|`*`userAttribute`*`]`
autoSelect	When set to true (default), this parameter enables Identity Server to automatically select the first item of a given identity object type in the Navigation view. Syntax: `autoselect=[true|false]`

Parameter	Description and Syntax
disableIntitialSearch	This value disables the initial Identity Server search for one or more identity object types. Disabling the initial search decreases the time to display the Identity Server console. The service attribute in the console that corresponds to this directive is Display Options, an organization attribute in the Administration Service. This console option takes precedence over any value defined in `com.iplanet.am.console.display.off`. If configuring this property in `AMConfig.properties` do not configure it using the console (and vica versa) Syntax (multiple values are delimited by a comma): `disableInitialSearch=[users|organizaitons|peopleContainers|organizationalUnits|roles|groups|policies]`
defaultUserView	This parameter sets the default view in the View menu of the User Profile page. All values are set by default. Syntax: `defaultUserView=[roles|groups|services|IplanetAMUserService|`*`service name]`*
defaultGroupView	This parameter sets the default view in the View menu of the Group Profile page. All values are set by default. Syntax: `defaultGroupView=[general|users]`
defaultRoleView	This parameter sets the default view in the View menu of the Role Profile page. All values are set by default. Syntax: `defaultRoleView=[general|users|services]`

Parameter	Description and Syntax
defaultPolicyView	This parameter sets the default view in the View menu of the Policy Profile page. All values are set by default. Syntax: `defaultPolicyView=[general|rules|subjects|referrals|conditions]`
defaultFederationHostedProviderView	This parameter sets the default view in the View menu of the Hosted Provider Profile page of the Federation Management module. All values are set by default. Syntax: `defaultFederationHostedProviderView=[general|serviceProvider|identityProvider|authenticaionDomain|trustedProviders|identityServerConfiguration]`
defaultFederationRemoteProviderView	This parameter sets the default view in the View menu of the Remote Provider Profile page of the Federation Management module. All values are set by default. Syntax: `defaultFederationRemoteProviderView=[general|serviceProvider|identityProvider|authenticaionDomain]`
rootNavMenu	This parameter sets the default view of identity objects in the root suffix navigation view. All values are set by default. Syntax: `rootNavMenu=[organizations|organizationalUnits|groupContainers|peopleContainers|roles|groups|users|policies]`

Parameter	Description and Syntax
organizationNavMenu	This parameter sets the default view of identity objects in the Organization navigation view. All values are set by default. Syntax: `organizationNavMenu=[organizations|organizationalUnits|groupContainers|peopleContainers|roles|groups|users|policies]`
groupContainerNavMenu	This parameter sets the default view of identity objects in the Group Container navigation view. All values are set by default. Syntax: `groupContainerNavMenu=[groupContainers|groups]`
peopleContainerNavMenu	This parameter sets the default view of identity objects in the People Container navigation view. All values are set by default. Syntax: `peopleContainerNavMenu=[peopleContainers|users]`
federationNavMenu	This parameter sets the default view of identity objects in the Federation Management module navigation view. All values are set by default. Syntax: `federationNavMenu=[authenticationDomains|hostedProviders|remoteProviders]`

Parameter	Description and Syntax
userProfileMenu	This parameter sets the sub-view menu entries in the User Profile page. All values are set by default. Syntax: `userProfileMenu=[roles|groups|services|iPlanetAMUserService|`*service name]*
groupProfileMenu	This parameter sets the sub-view menu entries in the Group Profile page. All values are set by default. Syntax: `groupProfileMenu=[general|users]`
roleProfileMenu	This parameter sets the sub-view menu entries in the Role Profile page. All values are set by default. Syntax: `roleProfileMenu=[general|users|services]`
policyProfileMenu	This parameter sets the sub-view menu entries in the Policy Profile page. All values are set by default. Syntax: `policyProfileMenu=[general|rules|subjects|referrals|conditions]`
federationRemoteProviderProfileMenu	This parameter sets the sub-view menu entries in the Federation Remote Provider Profile page. All values are set by default. Syntax: `federationRemoteProviderProfileMenu=[general|serviceProvider|identityProvider|authenticationDomain]`

Parameter	Description and Syntax
FederationHostedProviderProfile Menu	This parameter sets the sub-view menu entries in the Federation Hosted Provider Profile page. All values are set by default. Syntax: `federationHostedProviderProfileMenu=[general|serviceProvider|identityProvider|authenticationDomain|trustedProviders|identityServerConfiguration]`

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.iplanet.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.iplanet.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.

External Attributes Fetch Enabled

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.

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
- Case Sensitive User Name
- Default Anonymous User Name
- Authentication Level

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>
```

Case Sensitive User Name

If enabled, this option allows for case-sensitivity for user IDs. By default, this attribute is not enabled.

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>
```

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 Auth Level” on page 197 for details.

Certificate Authentication Attributes

The Certificate Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Certificate Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Certificate Authentication attributes are:

- Match Certificate in LDAP
- Attribute In Subject DN To Use To Search LDAP
- Match Certificate to CRL
- Attribute In Issuer DN To Use To Search CRL
- Enable OCSP Validation
- LDAP Server and Port
- LDAP Start Search DN
- LDAP Server Principal User
- LDAP Server Principal Password
- LDAP Attribute for Profile ID
- SSL On For LDAP Access
- Field in Cert To Use To Access User Profile
- Other Field In Cert To Use To Access User Profile
- Trusted Remote Hosts
- SSL Port Number

- Authentication Level

Match Certificate in LDAP

This option specifies whether to check if the user certificate presented at login is stored in the LDAP Server. If no match is found, the user is denied access. If a match is found and no other validation is required, the user is granted access. The default is that the Certificate Authentication service does not check for the user certificate.

NOTE A certificate stored in the Directory Server is not necessarily valid; it may be on the certificate revocation list. See "Match Certificate to CRL" on page 182. However, the web container may check the validity of the user certificate presented at login.

Attribute In Subject DN To Use To Search LDAP

This field specifies the attribute of the certificate's `SubjectDN` value that will be used to search LDAP for certificates. 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.

Attribute In Issuer DN To Use To Search CRL

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`.

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.repsonder.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.sum.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 and Port

This field specifies the name and port number of the LDAP server where the certificates are stored. The default value is the host name and port specified when Identity Server was installed. The host name and port of any LDAP Server where the certificates are stored can be used. The format is *hostname:port*.

LDAP Start Search DN

This field specifies the DN of the node where the search for the user's certificate should start. There is no default value. The field will recognize any valid DN. Multiple entries must be prefixed by the local server name.

LDAP Server Principal User

This field accepts the DN of the principal user (usually Directory Manager) for the LDAP server where the certificates are stored. There is no default value for this field which will recognize any valid DN. The principal user must be authorized to read, and search certificate information stored in the Directory Server.

LDAP Server Principal Password

This field carries the LDAP password associated with the user specified in the LDAP Server Principal User field. There is no default value for this field which will recognize the valid LDAP password for the specified principal user.

NOTE This value is stored as readable text in the directory.

LDAP Attribute for Profile ID

This field specifies the attribute in the Directory Server entry that matches the certificate whose value should be used to identify the correct user profile. There is no default value for this field which will recognize any valid attribute in a user entry (cn, sn, and so on) that can be used as the user ID.

SSL On For LDAP Access

This option specifies whether to use SSL to access the LDAP server. The default is that the Certificate Authentication service does not use SSL for LDAP access.

Field in Cert To Use To Access User Profile

This menu specifies which field in the certificate's Subject DN should be used to search for a matching user profile. For example, if you choose `email address`, the certificate authentication service will search for the user profile that matches the attribute `emailAddr` in the user certificate. The user logging in then uses the matched profile. The default field is `subject CN`. The list contains:

- email address
- subject CN
- subject DN
- subject UID
- other

Other Field In Cert To Use To Access User Profile

If the value of the Field in Cert To Use To Access User Profile attribute is set to `other`, then this field specifies the attribute that will be selected from the received certificate's `subjectDN` value. The authentication service will then search the user profile that matches the value of that attribute.

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 ONE Portal Server.

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 Sun ONE Identity Server Customization and API 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 Auth Level" on page 197 for details.

Chapter 19

Core Authentication Attributes

The Core Authentication service is the basic service for all of the default authentication services as well as any custom authentication service created with the Authentication SPI. Core authentication must be configured as a service for each organization that wishes to use any form of authentication. The Core Authentication attributes consist of global and organization attributes.The values applied to the global attributes are applied across the Sun ONE Identity Server configuration and are inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The values applied to the organization attributes under Service Configuration become the default values for the Core Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Core Authentication attributes are separated into:

- Global Attributes
- Organization Attributes

Global Attributes

The global attributes in the Core Authentication service are:

- Pluggable Auth Module Classes
- Supported Auth Modules for Clients
- LDAP Connection Pool Size
- LDAP Connection Default Pool Size

Pluggable Auth Module Classes

This field specifies the Java classes of the authentication 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 and RADIUS. Identity Server also includes a public SPI that can be used to add other authentication services. To define new services, this field must take a text string specifying the full class name (including package name) of each new authentication service.

Supported Auth Modules for Clients

This attribute specifies a list of supported authentication modules for a specific client. The format is as follows:

```
clientType | module1,module2,module3
```

This attribute is in effect when Client Detection is enabled.

LDAP Connection Pool Size

This attribute specifies the minimum and maximum connection pool to be used on a specific server and port. This attribute is for LDAP and Membership authentication services only. The format is as follows:

```
host:port:min:max
```

NOTE This connection pool is different than the SDK connection pool configured in `serverconfig.xml`.

LDAP Connection Default Pool Size

This attribute sets the default minimum and maximum connection pool to be used with all LDAP authentication module configurations. If an entry for the host and port exists in the LDAP Connection Pool Size attribute, the minimum and maximum settings will not be used from LDAP Connection Default Pool Size.

Organization Attributes

The organization attributes in the Core Authentication service are:

- Organization Authentication Modules
- User Profile
- Admin Authenticator
- User Profile Dynamic Creation Default Roles
- Persistent Cookie Mode
- Persistent Cookie Max Time (seconds)
- People Container For All Users
- Alias Search Attribute Name
- Default Auth Level
- User Naming Attribute
- Default Auth Locale
- Organization Authentication Configuration
- Login Failure Lockout Mode
- Login Failure Lockout Count
- Login Failure Lockout Interval (minutes)
- Email Address to Send Lockout Notification
- Warn User After N Failure
- Login Failure Lockout Duration (minutes)
- Lockout Attribute Name
- Lockout Attribute Value
- Default Success Login URL
- Default Failure Login URL
- Authentication PostProcessing Class
- User Name Generator Mode
- Pluggable User Name Generator Class

Organization Authentication Modules

This list specifies the authentication modules available to the organization. Each administrator can choose the type of authentication for 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

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 SSOToken for a successful authentication.

Admin Authenticator

Clicking the edit link will allow you to define the authentication service for administrators only. An administrator is a user who needs access to the Identity Server console. This attribute can be used if the authentication module for administrators needs to be different from the module for end users. The modules configured in this attribute are picked up when the Identity Server console is accessed.

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 190. There is no default value. The administrator must specify the DNs of the roles that will be assigned to the new user.

NOTE The role specified must be under the organization for which authentication is being configured.

Persistent Cookie Mode

This option determines whether users can restart the browser and still return to their authenticated session. User sessions can be retained by enabling Persistent Cookie Mode. When Persistent Cookie Mode is enabled, a user session does not expire until its persistent cookie expires, or the user explicitly logs out. The expiration time is specified in Persistent Cookie Max Time (seconds). The default value is that Persistent Cookie Mode is not enabled and the authentication service uses only memory cookies.

NOTE A persistent cookie must be explicitly requested by the client using the `iPSPCookie=yes` parameter in the login URL.

Persistent Cookie Max Time (seconds)

This field specifies the interval after which a persistent cookie expires. (Persistent Cookie Mode must be enabled by selecting its checkbox.) The interval begins when the user's session has been successfully authenticated. The default value is `2147483` (time in seconds). The field will take any integer value between `0` and `2147483`.

People Container For All Users

After successful authentication by a user, 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 193, 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 Auth Locale

This field specifies the default language subtype to be used by the authentication service. The default value is en_US. A listing of valid language subtypes can be found in Table 19-1.

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 *Sun ONE Identity Server Customization and API Guide* for more information.

Table 19-1 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 19-1 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 Sun ONE Identity Server Customization and API Guide for more information.

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 (minutes), before being locked out.

Login Failure Lockout Interval (minutes)

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 (minutes)

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 0 for `inetuserstatus`.

Default Success Login URL

This field specifies the URL to which users are redirected after successful authentication. The field will take any valid URL. The Success Login URL is set in the `LoginStatus` element in the `remote-auth.dtd`. See the *Sun ONE Identity Server Customization and API Guide* for more information.

Default Failure Login URL

This field specifies the URL to which users are redirected if authentication is unsuccessful. The field will take any valid URL. The Failure Login URL is set in the `LoginStatus` element in the `remote-auth.dtd`. See the *Sun ONE Identity Server Customization and API 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.

User Name Generator Mode

This attribute is used by the Membership authentication module. If this attribute field is enabled, the Membership module is able to generate user IDs, 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 User Name Generator Mode is enabled.

Default Auth Level

The authentication level value indicates how much to trust authentications. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application can use the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level.

The authentication level should be set within the organization's specific authentication template. The Default Auth Level value described here will apply only when no authentication level has been specified in the Authentication Level field for a specific organization's authentication template. The Default Auth Level default value is 0. (The value in this attribute is not used by Identity Server but by any external application that may chose to use it.)

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 Auth Level" on page 197 for details.

LDAP Authentication Attributes

The LDAP Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the LDAP Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The LDAP Authentication attributes are:

- Primary LDAP Server and Port
- Secondary LDAP Server and Port
- DN to Start User Search
- DN for Root User bind
- Password for Root User Bind
- Password For Root User Bind (Confirm)
- User Naming Attribute
- User Entry Search Attributes
- User Search Filter
- Search Scope
- Enable SSL to LDAP Server
- Return User DN To Auth
- LDAP Server Check Interval
- User Creation Attributes List
- Authentication Level

Primary LDAP Server and Port

This field specifies the host name and port number of the primary LDAP server specified during Identity Server installation. This is the first server contacted for LDAP authentication. The format is `hostname:port`. (If there is no port number, assume `389`.).

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 and Port

This field specifies the host name and port number of a secondary LDAP server available to the Identity Server platform. If the primary LDAP server does not respond to a request for authentication, this server would then be contacted. If 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. 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 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.

User Naming Attribute

After successful authentication by a user, the user's profile is retrieved. The value of this attribute is used to perform the search. The field specifies the LDAP attribute to use. By default, Identity Server assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`) specify the attribute name in this field.

NOTE The user search filter will be a combination of the Search Filter attribute and the User Entry Naming Attribute.

User Entry Search Attributes

This field lists the attributes to be used to form the search filter for a user that is to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber` and `mail`, the user could authenticate with any of these names.

User Search Filter

This field specifies an attribute to be used to find the user under the DN to Start User Search field. It works with the User Entry Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.

Search Scope

This menu indicates the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in the attribute "DN to Start User Search" on page 202. 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 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 Auth

When the Identity Server directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the `userId`, and no search is necessary. Normally, an authentication module returns only the `userId`, and the authentication service searches for the user in the local Identity Server LDAP. If an external LDAP directory is used, this option is typically not enabled.

LDAP Server Check Interval

This attribute is used for LDAP Server failback. It defines the number of seconds 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 Auth Level" on page 197 for details.

Membership Authentication Attributes

The Membership Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the Membership Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization. The Membership Authentication attributes are:

- Minimum Password Length
- Default User Roles
- User Status After Registration
- Primary LDAP Server and Port
- Secondary LDAP Server and Port
- DN to Start User Search
- DN for Root User bind
- Password for Root User Bind
- Password for Root User Bind (Confirm)
- User Naming Attribute
- User Entry Search Attributes
- User Search Filter
- Search Scope
- Enable SSL to LDAP Server
- Return User DN To Auth

- Authentication Level

Minimum Password Length

This field specifies the minimum number of characters required for a password set during self-registration. The default value is 8.

If this value is changed, it should also be changed in the registration and error text in the following file:

IdentitySever_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.

User Status After Registration

This menu specifies whether services are immediately made available to a user who has self-registered. The default value is `Active` and services are available to the new user. By selecting `Inactive`, the administrator chooses to make no services available to a new user.

Primary LDAP Server and Port

This field specifies the host name and port number of the primary 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 and Port

This field specifies the host name and port number of a secondary LDAP server available to the Identity Server platform. If the primary LDAP server does not respond to a request for authentication, this server would then be contacted. If 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 you use multiple entries, the entries must be prefixed by the local server name.

NOTE If multiple users match the same search, authentication will fail.

DN for Root User bind

This field specifies the DN of the user that will be used to bind to the Directory Server specified in the Primary LDAP Server and Port field as administrator. The authentication service needs to bind as this DN in order to search for a matching user DN based on the user login ID. 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.

User Naming Attribute

This field specifies the attribute used for the naming convention of user entries. By default, Identity Server assumes that user entries are identified by the `uid` attribute. If your Directory Server uses a different attribute (such as `givenname`) specify the attribute name in this field.

User Entry Search Attributes

This field lists the attributes to be used to form the search filter for a user that is to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber` and `mail`, the user could authenticate with any of these names.

User Search Filter

This field specifies an attribute to be used to find the user under the DN to Start User Search field. It works with the User Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.

Search Scope

This menu indicates the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in the attribute "DN to Start User Search" on page 209. The default value is `SUBTREE`. One of the following choices can be selected from the list:

- OBJECT — Searches only the specified node
- ONELEVEL — Searches at the level of the specified node and one level down
- SUBTREE — Search all entries at and below the specified node

Enable SSL to LDAP Server

This option enables SSL access to the Directory Server specified in the Primary and Secondary LDAP Server and Port field. By default, the box is not checked and the SSL protocol will not be used to access the Directory Server.

Return User DN To Auth

When the Identity Server directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the `userId`, and no search is necessary. Normally, an authentication module returns only the `userId`, and the authentication service searches for the user in the local Identity Server LDAP. If an external LDAP directory is used, this option is typically not enabled.

Authentication Level

The authentication level is set separately for each method of authentication. The value indicates how much to trust an authentication. Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. 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 Auth Level” on page 197 for details.

Chapter 23

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.

NT authentication is only supported on the Solaris version of Identity Server. In order to actualize the NT Authentication module, Samba Client 2.2.2 must be downloaded and installed. 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://wwws.sun.com/software/download/products/3e3af224.html.

The NT Authentication attributes are:

- NT Authentication Domain
- NT Authentication Host
- Authentication Level

NT Authentication Domain

This attribute defines the Domain name to which the user belongs.

NT Authentication Host

This attribute defines the NT authentication hostname. The hostname should be the netBIOS name, as opposed to the fully qualified domain name (FQDN). By default, the first part of the FQDN is the netBIOS name.

If the DHCP (Dynamic Host Configuration Protocol) is used, you would put a suitable entry in the HOSTS file on the Windows 2000 machine.

Name resolution will be performed based on the netBIOS name. If you do not have any server on your subnet supplying netBIOS name resolution, the mappings should be hardcoded.

For example, the hostname should be `example1` not `example1.company1.com`.

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 Auth Level" on page 197 for details.

RADIUS Authentication Attributes

The RADIUS Authentication attributes are organization attributes. The values applied to them under Service Configuration become the default values for the RADIUS Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The RADIUS Authentication attributes are:

- RADIUS Server 1
- RADIUS Server 2
- RADIUS Shared Secret
- RADIUS Shared Secret (Confirm)
- RADIUS Server's Port
- Timeout (Seconds)
- Authentication Level

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_adress ...
```

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`.

NOTE If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Auth Level. See "Default Auth Level" on page 197 for details.

Timeout (Seconds)

This field specifies the time interval in seconds to wait for the RADIUS server to respond before a timeout. The default value is 3 seconds. It will recognize any number specifying the timeout in seconds.

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 Auth Level" on page 197 for details.

SafeWord Authentication Attributes

The SafeWord Authentication Attributes are organization attributes. The values applied to them under Service Configuration become the default values for the SafeWord Authentication template. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the subtrees of the organization.

This service allows for authenticating users using Secure Computing's SafeWord or SafeWord PremierAccess authentication servers.The SafeWord Authentication attributes are:

- SafeWord Server Specification
- SafeWord System Name
- SafeWord Server Verification Files Path
- SafeWord Logging Level
- SafeWord Log Path
- Authentication Level

SafeWord Server Specification

This field specifies the SafeWord or SafeWord PremiereAccess server name and port. Port 7482 is set as the default for a SafeWord server. The default port number for a SafeWord PremierAccess server is 5030.

SafeWord System Name

This field specifies the system name configured in the SafeWord server. The default system name is STANDARD.

SafeWord Server Verification Files Path

This field specifies the directory into which the SafeWord client library places its verification files. The default is as follows:

```
/var/opt/SUNWam/auth/safeword/serverVerification
```

If a different directory is specified in this field, the directory must exist before attempting SafeWord authentication.

SafeWord Logging Level

This attribute is not used.

SafeWord Log Path

This attribute specifies the directory path and log file name for SafeWord client logging. The default path is as follows:

```
/var/opt/SUNWam/auth/safeword/safe.log
```

If a different path or filename is specified, they must exist before attempting SafeWord authentication.

If more than one organization is configured for SafeWord authentication, and different SafeWord servers are used, then different paths must be specified, or only the first organization where SafeWord authentication occurs will work. Likewise, if an organization changes SafeWord servers, the `swec.dat` file in the specified directory must be deleted before authentications to the newly configured SafeWord server will work.

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 Auth Level" on page 197 for details.

Chapter 26

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
- SecurID Helper Configuration Port
- SecurID Helper Authentication Port
- Authentication Level

NOTE In Identity Server 6.1, the SecurID Authentication service is not supported for the x86 operating system.

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 Auth Level" on page 197 for details.

Chapter 27

Unix Authentication Attributes

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

- Global Attributes
- Organization Attribute

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

Global Attributes

The global attributes in the Unix Authentication service are:

- Unix Helper Configuration Port
- Unix Helper Authentication Port
- Unix Helper Timeout (Minutes)
- Unix Helper Threads

Unix Helper Configuration Port

This attribute specifies the port to which the Unix Helper ‘listens’ upon startup for the configuration information contained in the Unix Helper Authentication Port, Unix Helper Timeout (Minutes), and Unix Helper Threads attributes. The default is 58946.

If this attribute is changed, you must also change the `unixHelper.port` entry in the `AMConfig.properties` file, and restart Identity Server.

Unix Helper Authentication Port

This attribute specifies the port to which the Unix Helper ‘listens’ for authentication requests after configuration. The default port is 57946.

Unix Helper Timeout (Minutes)

This attribute specifies the number of minutes that users have to complete authentication. If users surpass the allotted time, authentication automatically fails. The default time is set to 3 minutes.

Unix Helper Threads

This attribute specifies the maximum number of permitted simultaneous Unix authentication sessions. If the maximum is reached at a given moment, subsequent authentication attempts are not allowed until a session is freed up. The default is set to 5.

Organization Attribute

The organization attribute for the Unix Authentication service is:

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 Auth Level" on page 197 for details.

Chapter 28

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
- Login Success URL
- Login Failure URL
- Authentication Post Processing Class

Authentication Configuration

Clicking on the Edit link will display the Authentication Configuration interface. It allows you to configure the authentication modules for role-based or organization-based authentication.

The following table lists the authentication module configuration options:

Module Name	Allows you to select from the list of default authentication modules available to Identity Server.

Flag

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

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

These flags establish an enforcement criteria for the authentication module for which they are defined. There hierarchy for enforcement, with REQURIED 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.

Chapter 29

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
- Default Client Type
- Client Detection Class
- Client Detection Enabled

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, the only configured Identity Server client data available for HTML-based browsers is defined as sub-configurations of the overall schema: genericHTML and its parent HTML.

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 filed (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-UAProf-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/terms.asp?doc=WAP-248-UAProf-20011020-a.pdf

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.iplanet.services.cdm.ClientDetectionDefaultImpl`.

Client Detection Enabled

This attribute allows you to enable client detection. If client detection is enabled (selected), every request is routed thought the class specified in the Client Detection Class attribute.

By default, the client detection capability is disabled for any client type other than genericHTML. If this attribute is not selected, Identity Server assumes that the client is genericHTML and will be accessed from a HTML browser.

Chapter 30

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
- Charset Aliases
- Auto Generated Common Name Format

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
```

Chapter 31

Logging Service Attributes

The Logging Service attributes are global attributes. The values applied to them are applied across the Sun ONE Identity Server configuration and are inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The Logging Attributes are:

- Max Log Size
- Number of History Files
- Log Location
- Logging Type
- Database User Name
- Database User Password
- Database User Password (Confirm)
- Database Driver Name
- Configurable Log Fields
- Log Verification Time
- Log Signature Time
- Secure Logging
- Maximum Number of Records
- Number Of Files Per Archive
- Buffer Size
- Buffer Time

- Time Buffering

Max 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 input depending on the partition size and available disk space of the local system. The default value is 3.

Log Location

The file-based logging function needs a location where log files can be stored. This field accepts a full directory path to that location. The default location is:

`/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 Time

This attribute sets the frequency (in seconds) that the server should verify the logs to detect tampering. The default time is 3600 seconds. This parameter applies to secure logging only.

Log Signature Time

This parameter sets the frequency (in seconds) that the log will be signed. The default time is 900 seconds. This parameter applies to secure logging only.

Secure Logging

This attribute specifies whether or not to enable secure logging. By default, secure logging is off. Secure Logging enables detection of unauthorized changes or tampering of security logs.

Maximum Number of Records

This attribute sets the maximum number of records that the Java LogReader interfaces return, regardless of how many records match the read query. By default, it is set to 500. This attribute can be overridden by the caller of the Logging API through the LogQuery parameter.

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 buffered in memory before they are sent to the logging service to be logged. The default is 3600 seconds.

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.

Chapter 32

Naming Service Attributes

The Naming Service attributes are global attributes. The values applied to them are carried across the Sun ONE Identity Server configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.)

The Naming Service allows clients to find the correct service URL if the platform is running more than one Identity Server. When a naming URL is found, the naming service will decode the session of the user and dynamically replace the protocol, host, and port with the parameters from the session. This ensures that the URL returned for the service is for the host that the user session was created on. The Naming Attributes are:

- Profile Service URL
- Session Service URL
- Logging Service URL
- Policy Service URL
- Auth Service URL
- SAML Web Profile / Artifact Service URL
- SAML SOAP Service URL
- SAML Web Profile / POST Service URL
- SAML Assertion Manager Service URL
- Federation Assertion Manager Service URL
- Identity SDK Service URL

Profile Service URL

This field takes a value equal to

`%protocol://%host:%port/`*Server_DEPLOY_URI*`/profileservice`

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/FSAssertionMana
gerIF
```

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
- Secret Question
- Search Filter
- Base DN
- Bind DN
- Bind Password
- Password Reset Option
- Password Change Notification Option
- Password Reset Enabled
- Personal Question Enabled
- Number of Questions
- Password Reset Failure Lockout Count
- Password Reset Failure Lockout Interval (minutes)
- Email Address to Send Lockout Notification
- Warn User After N Failure
- Password Reset Failure Lockout Duration (minutes)

- Password Reset Failure Lockout Mode
- Password Reset Lockout Attribute Name
- Password Reset Lockout Attribute Value

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 filed 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 *Sun ONE Identity Server Customization and API 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 *Sun ONE Identity Server Customization and API Guide* for more information.

Password Reset Enabled

Selecting this attribute will enable the password reset feature.

Personal Question Enabled

Selecting this attribute will allow a user to create a unique question for password resetting.

Number of Questions

This value specifies the maximum number of questions to be asked in the password reset page.

Password Reset Failure Lockout Count

This attributes 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 (minutes)

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 (minutes)

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 Failure Lockout Mode

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 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.

Chapter 34

Platform Service Attributes

The Platform Service attributes are global attributes. The values applied to them are carried across the Sun ONE Identity Server configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The Platform Attributes are:

- Server List
- Platform Locale
- Cookie Domains
- Login Service URL
- Logout Service URL
- Available Locales
- Client Char Sets

Server List

The naming service reads this attribute at initialization time. This list contains the Identity Server session servers in a single Identity Server configuration. For example, if two Identity Servers are installed and should work as one, they must both be included in this list. If the host specified in a request for a service URL is not in this list, the naming service will reject the request. The first value in the list specifies the host name and port of the server specified during installation.At the end of the list, there is a two-byte value that uniquely identifies the server. Each server that is participating in load balancing needs to have a unique identifier. This is also used to shorten the cookie length by mapping the server URL to the server ID. For example:

protocol://*server_domain*:*port*|01

Additional servers can be added using the format protocol://*server_domain*: *port*|01|*instance_name*

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 19-1 on page 193 for a listing of all supported language subtypes.

Cookie Domains

This is the list of domains that will be returned in the cookie header when setting a cookie to the user's browser during authentication. If empty, no cookie domain will be set. In other words, the Identity Server session cookie will only be forwarded to the Identity Server itself and no other servers in the domain. If SSO is required with other servers in the domain, this attribute must be set with the cookie domain. If you had two interfaces in different domains on one Identity Server then you would need to set both cookie domains in this attribute. If a load balancer is used, the cookie domain must be that of the load balancer's domain, not the servers behind the load balancer. The default value for this field is the domain of the installed Identity Server.

Login Service URL

This field specifies the URL of the login page. The default value for this attribute is */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
```

Chapter 35

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 ONE Identity Server configuration and are inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.) The values applied to the organization attributes under Service Management become the default values for Policy configuration. The service template needs to be created after registering the service for the organization. The default values can be changed after registration by the organization's administrator. Organization attributes are not inherited by entries in the organization. The Policy Configuration attributes are separated into:

- Global Attribute
- Organization Attributes

Global Attribute

The global attribute in the Policy Configurative service is:

- Resource Comparator

Resource Comparator

This attribute specifies the resource comparator information, which is used to compare resources specified in a Policy rule definition. Resource comparison is used for both policy creation and evaluation. This attribute contains the following values:

`serviceType`	Specifies the service to which the comparator should be used.
`class`	Defines the java class that implements the resource comparison algorithm.
`wildcard`	Specifies the wildcard that can be defined in resource names
`delimiter`	Specifies the delimiter to be used in the resource name.
`caseSensitivity`	Specifies if the comparison of the two resources should consider or ignore case. `False` ignores case, `True` considers case.

Organization Attributes

The organization attributes in the Policy Configuration service are:

- LDAP Server and Port
- LDAP Base DN
- LDAP Users Base DN
- Identity Server Roles Base DN
- LDAP Bind DN
- LDAP Bind Password
- LDAP Bind Password (Confirm)
- LDAP Org Search Filter
- LDAP Org Search Scope
- LDAP Groups Search Filter
- LDAP Groups Search Scope
- LDAP Users Search Filter
- LDAP Users Search Scope

- LDAP Roles Search Filter
- LDAP Roles Search Scope
- Identity Server Roles Search Scope
- LDAP Organization Search Attribute
- LDAP Groups Search Attribute
- LDAP Users Search Attribute
- LDAP Roles Search Attribute
- Maximum Results Returned From Search
- Timeout For Search (seconds)
- LDAP SSL Enabled
- LDAP Connection Pool Minimal Size
- LDAP Connection Pool Maximum Size
- Selected Policy Subjects
- Selected Policy Conditions
- Selected Policy Referrals
- Subjects Result Time To Live
- User Alias Enabled

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:

```
machine1.example1.com|machine1.example1.com:389 machine2.example.com1:389
machine1.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 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 Org Search Filter

Specifies the search filter to be used to find organization entries. The default is `(objectclass=sunMangagedOrganization)`.

LDAP Org Search Scope

This attribute defines the scope to be used to find organization entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Groups Search Filter

Specifies the search filter to be used to find group entries. The default is `(objectclass=groupOfUniqueNames)`.

LDAP Groups Search Scope

This attribute defines the scope to be used to find group entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Users Search Filter

Specifies the search filter to be used to find user entries. The default is `(objectclass=inetorgperson)`.

LDAP Users Search Scope

This attribute defines the scope to be used to find user entries. The scope must be one of the following:

- `SCOPE_BASE`
- `SCOPE_ONE`
- `SCOPE_SUB` (default)

LDAP Roles Search Filter

Specifies the search filter to be used to find entries for roles. The default is `(&(objectclass=ldapsubentry)(objectclass=nsroledefinitions))`

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 (seconds)

This attribute specifies the amount of time before a timeout on a search occurs. If the search exceeds the specified time, the entries that have been found to that point will be returned

LDAP SSL Enabled

This attribute specifies whether or not the LDAP server is running SSL. Selected enables SSL, unselected (default) disables SSL.

LDAP Connection Pool Minimal Size

This attribute specifies the minimal size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default is 1.

LDAP Connection Pool Maximum Size

This attribute specifies the maximum size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default is 10.

Selected Policy Subjects

This attribute allows you to select a set of subject types available to be used for policy definition in the organization.

Selected Policy Conditions

This attribute allows you to select a set of conditions types available to be used for policy definition in the organization.

Selected Policy Referrals

This attribute allows you to select a set of referral types available to be used for policy definition in the organization.

Subjects Result Time To Live

This attribute specifies the amount of time (in minutes) that a cached subject result can be used to evaluate the same policy request based on an single sign-on token.

When a policy is initially evaluated for an SSO token, the subject instances in the policy are evaluated to determine whether the policy is applicable to a given user. The subject result, which is keyed by the SSO token ID, is cached in the policy. If another evaluation occurs for the same policy for the same SSO token ID within the time specified in the Subject Result Time To Live attribute, the policy framework retrieves the cached subjects result, instead of evaluating the subject instances. This significantly reduces the time for policy evaluation.

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.

Chapter 36

SAML Service Attributes

The Security Assertion Markup Language (SAML) Service attributes are global attributes. The values applied to them are carried across the Sun ONE Identity Server configuration and inherited by every configured organization. (They can not be applied directly to roles or organizations as the goal of global attributes is to customize the Identity Server application.)

For more information about the SAML Service architecture, see the *Sun ONE Identity Server Customization and API Guide*.

The SAML attributes are as follows:

- Site ID And Site Issuer Name
- Sign Request
- Sign Response
- Sign Assertion
- Artifact Name
- Target Specifier
- Artifact Timeout (seconds)
- Assertion Skew Factor For notBefore Time
- Assertion Timeout (seconds)
- Trusted Partner Sites
- POST To Target URLs

Site ID And Site Issuer Name

This attribute contains a list of entries, with each entry containing an instance ID, site ID, and site issuer name. 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 Request

This attribute specifies whether all SAML requests will be digitally signed (`XML DSIG`) before being delivered. Clicking on this option will enable this feature.

Sign Response

This attribute specifies whether all SAML responses will be digitally signed (XML DSIG) before being delivered. Clicking on this option will enable this feature.

All SAML responses used by the SAML Web Post profile will be digitally signed whether this option is enabled or not enabled.

Sign Assertion

This attribute specifies whether all SAML assertions will be digitally signed (`XML DSIG`) before being delivered. Clicking on this option will enable this feature.

Artifact Name

This attribute assigns a variable name to a SAML artifact defined in the SAML Service configuration. A SAML artifact is bounded-size data, which identifies an assertion and a source site. It is carried as part of a URL query string and conveyed by a re-direction to the destination site. The default is `SAMLart`. 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 (seconds)

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 (seconds)

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 36-1 Trusted Partner Sites Parameters

SourceID	The 20-byte sequence defined as in the SiteID and Issuer name.
target	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, target specifies the redirect to a URL defined by the SAMLUrl or POSTUrl parameters for further processing. 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. For example, if you have the following two trusted partner sites definitions: target=sun.com|SAMLUrl=http://machine1.sun.com:8080/amserver/SAMLAwareServlet and target=sun.com:8080|SAMLUrl=httyp://machine2.sun.com:80/amserver/SAMLAwareServlet and are seeking a the following page: http://somemachine.sun.com:8080/index.html the second definition will be chosen as the SAML service provider because the matching domain and port coexist in the target parameter.
SAMLUrl	Defines the URL that provides the SAML service. The servlet specified in the URL implements the Web-browser SSO with Artifact profile defined in the OASIS-SAML Bindings and Profiles specification.
POSTUrl	Defines the URL that provides the SAML service. The servlet specified in this URL implements the Web-browser SSO with POST profile defined in the OASIS-SAML Binding and Profiles specification.
issuer	Defines the creator of an assertion generated within Identity Server. The syntax is hostname:port.
SOAPUrl	Specifies the SOAP Receiver service URL.

AuthType	Defines the authentication type used in SAML. It should be one of the following: • NOAUTH • BASICAUTH • SSL • SSLWITHBASICAUTH This parameter is optional, and if not specified, the default is NOAUTH. If BASICAUTH or SSLWITHBASICAUTH is specified, the User parameter is require and the SOAPUrl should be HTTPS.
User	Defines the uid of the partner which is used to protect the partner's SOAP Receiver.
hostlist	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. 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.
AccountMapper	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: com.sun.identity.saml.plugins.DefaultAccountMapper
attributeMapper	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.
actionMapper	Specifies the class with the path to where the actionMapper is located. Applications can develop an actionMapper to obtain either an SSOToken ID or an assertion containing AuthenticationStatement from the query. The mapper is then used to retrieve the authorization decisions for the actions defined in the query. If no actionMapper is specified, DefaultActionMapper will be used.

`siteAttributeMapper`	Specifies the class with the path where the `siteAttributeMapper` is located. Applications can develop a `siteAttributeMapper` to obtain attributes to be included in the assertion during SSO. If no `siteAttributeMapper` is found, then no attributes will be included in the assertion during SSO.
`certAlias=`*aliasName*	Specifies a `certAlias` 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 `KeyInfo` 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	`sourceid`	`sourceid`
	`target`	`SOAPUrl`
	`SAMLUrl`	[`accountMapper`]
	`hostlist`	[`AuthType`]
	[`siteAttributeMapper`]	[`User`]
		[`certAlias`]
POST profile	`sourceid`	`sourceid`
	`target`	`issuer`
	`POSTUrl`	[`accountMapper`]
	[`siteAttributeMapper`]	[`certAlias`]
SOAP Request		`sourceid`
		`hostlist`
		[`attributeMapper`]
		[`actionMapper`]

Sender	Receiver
	[certAlias]
	[issuer]

POST To Target URLs

If the target URL received through SSO (either artifact profile or POST profile) by the site is listed in this attribute, the assertion or assertions that are received from SSO will be sent to the target URL by an http: FORM POST. Avoid using test URLS or any other additional URLs in a POST.

Chapter 37

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
- Timeout For Search (Seconds)

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 attributed 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)
- Max Idle Time (Minutes)
- Max Caching Time (Minutes)

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.

Chapter 38

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
- User Profile Attributes
- Unique User IDs

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
- Admin DN Starting View
- Default User Status

Default user values are set for all Identity Server registered organizations. These values can be set differently for separate organizations by registering the user service to the specific organization, creating a template and inputting a value other than the default value.

User Preferred Language

This field specifies the user's choice for the text language displayed in the Identity Server console. The default value is `en`. This value maps a set of localization keys to the user session so that 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 19-1 on page 193 can be used.

Admin DN Starting View

If this user is a Identity Server administrator, this field specifies the node that would be the starting point displayed in the Identity Server console when this user logs in. There is no default value. A valid DN for which the user has, at the least, read access can be used.

Default User Status

This option indicates the default status for any newly created user. This status is superseded by the User Entry status. Only active users can authenticate through Identity Server. The default value is `Active`. Either of the following can be selected from the pull-down menu:

- `Active` – The user can authenticate through Identity Server.
- `Inactive` – The user cannot authenticate through Identity Server, but the user profile remains stored in the directory.

The individual user status is set by registering the User service, choosing the value, applying it to a role and adding the role to the user's profile.

User Profile Attributes

The User Profile Attributes are default attributes for user profiles. These values are set in the User Profile view by an administrator or by the user when they log on. Administrators can add their own user attributes to the user profile or create a new service. For more information see *Sun ONE Identity Server Customization and API Guide*.

NOTE Identity Server does not enforce uniqueness for attributes within user entries. For example, `userA` and `userB` are both created in the same organization. For both, the email address attribute can be set `jimb@madisonparc.com`. The administrator can configure Sun ONE Directory Server's attribute uniqueness plug-in to help enforce unique attribute values. For more information, see Unique User IDs at the end of this chapter or the *Sun One Directory Server Administrator's Guide*.

First Name

This field takes the first name of the user. (The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Identity Server console.)

Last Name

This field takes the last name of the user. (The First Name value and the Last Name value identify the user in the Currently Logged In field in the upper right corner of the Identity Server console.)

Full Name

This field takes the full name of the user.

Password

This field takes the password for the name specified in the UserId field.

Password (Confirm)

Confirmation of the password.

Email Address

This field takes the email address of the user.

Employee Number

This field takes the employee number of the user.

Telephone Number

This field takes the telephone number of the user.

Home Address

This field can take the home address of the user.

User Status

This option indicates whether the user is allowed to authenticate through Identity Server. Only active users can authenticate through Identity Server. The default value is `Active`. Either of the following can be selected from the pull-down menu:

- `Active` – The user can authenticate through Identity Server.
- `Inactive` – The user cannot authenticate through Identity Server, but the user profile remains stored in the directory.

NOTE Changing the user status to `Inactive` only affects authentication through Identity Server. The Directory Server uses the `nsAccountLock` attribute to determine user account status. User accounts inactivated for Identity Server authentication can still perform tasks that do not require Identity Server. To inactivate a user account in the directory, and not just for Identity Server authentication, set the value of `nsAccountLock` to `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 *Sun ONE Identity Server Customization and API 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 on 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 19-1 on page 193 can be used.

You can use one of the following attributes in the pull-down menu:

- Ignore
- Customize
- Inherit

Success URL

This attribute specifies the URL that the user will be redirected to upon successful authentication.

Failure URL

This attribute specifies the URL that the user will be redirected to upon unsuccessful authentication.

Unique User IDs

In order to enforce uid uniqueness within the Identity Server application, the plug-in, available in Directory Server, must be configured as follows:

```
dn: cn=uid uniqueness,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: uid uniqueness
nsslapd-pluginPath: /ids908/lib/uid-plugin.so
nsslapd-pluginInitfunc: NSUniqueAttr_Init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: attribute=uid
nsslapd-pluginarg1: markerObjectClass=nsManagedDomain
nsslapd-plugin-depends-on-type: database
nsslapd-pluginId: NSUniqueAttr
```

```
nsslapd-pluginVersion: 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
```

Appendix A

Error Codes

This appendix provides a list of the error messages generated by Sun ONE 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 ONE 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 *identity object*. 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 `inactive` to `active`. 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 (`--runasdn`, `--password`, `--passwordfile`, `--schema`, `--data`, and `--addAttributes`) 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 `--runasdn` value is missing.	Provide the user DN as the value for `--runasdn`.
noservicename	4	The service name for the `--deletservice` value is missing.	Provide the service name as the value for `--deleteservice`.
nopwdforadmin	5	The password for the `--password` value is missing.	Provide the password as the value for `--password`.
nolocalename	6	The locale name was not provided. The locale will default to en_US.	See Default Auth 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 `amadmin --help`.
oprfailed	9	Operation failed.	When `amadmin` 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 `amadmin` 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.	`amadmin` 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.
parserror	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`.
parsebuilterror	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
doctyperror	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.
dbugerror	33	More than one debug option is specified.	Only one debug option should be specified.
loginFalied	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.
cannotAddusersToFileredRole	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.
cannotAdduUersToDynamicGroup	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.

Appendix B

Configuring Identity Server in SSL Mode

Using Secure Socket Layer (SSL) with simple authentication guarantees confidentiality and data integrity.

Identity Server is capable of simultaneous SSL and non-SSL communications. This means that you do not have to choose between SSL or non-SSL communications; you can use both at the same time.

The following sections describe the steps for configuring Identity Server in SSL with four different web containers:

- Configuring Identity Server With a Secure Sun ONE Web Server
- Configuring Identity Server with a Secure Sun ONE Application Server

Configuring Identity Server With a Secure Sun ONE Web Server

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

1. In the Identity Server console, click on the Properties arrow for the top-level organization (created during installation).

   The Organization Properties window will display in the Data frame.

2. Click Save to save the changes.

3. 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 4 through Step 27 describe the Sun ONE Web Server.

4. Log on to the Web Server console. The default port is 58888.
5. 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.
6. Click on the Apply button located top right corner of the screen.
7. Click Apply Settings.

   The Web Server should automatically restart. Click OK to continue.
8. Stop the select Web Server instance.
9. Click the Security Tab.
10. Click on Create Database.
11. Enter the new database password and click OK.

    Ensure that you write down the database password for later use.
12. Once the Certificate Database has been created, click on Request a Certificate.
13. Enter the data in the fields provided in the screen.

    The Key Pair Field Password field is the same as you entered in Step 11. 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.
14. Once the form is submitted, you will see a message such as:

```
--BEGIN CERTIFICATE REQUEST---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE REQUEST--
```

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

    Ensure that you get the Root CA certificate.

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

```
--BEGIN CERTIFICATE---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdflasdf

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--END CERTIFICATE---
```

17. Copy this text into your clipboard, or save the text into a file.
18. Go to the Web Server console and click on Install Certificate.
19. Click on Certificate for this Server.
20. Enter the Certificate Database password in the Key Pair File Password field.
21. 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.

22. Click Install Certificate.
23. Click Certificate for Trusted Certificate Authority.

24. Install the Root CA Certificate in the same manner described in Step 18 through Step 23.
25. Once you have completed installing both certificates, click on the Preferences tab in the Web Server console.
26. Select Add Listen Socket if you wish to have SSL enabled on a different port. Then, select Edit Listen Socket.
27. Change the security status from Disabled to Enabled, and click OK to submit the changes.

Step 28 through Step 30 describe Identity Server.

28. Open the `AMConfig.properties` file. By default, the location of this file is `/opt/SUNWam/lib`.
29. 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.
30. Save the `AMConfig.properties` file.
31. 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.
32. Enter the Certificate Database password in the text field and select Start.

Configuring Identity Server with a Secure Sun ONE Application Server

Setting up Identity Server to run on an SSL-enabled Sun ONE 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 ONE 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 ONE 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 One 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---

afajsdllwqeroisdaoi234rlkqwelkasjlasnvdknbslajowijalsdkjfalsdfla

alsfjawoeirjoi2ejowdnlkswnvnwofijwoeijfwiepwerfoiqeroijeprwpfrwl

--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 ONE 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 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 ONE 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 with WebLogic in SSL mode:

1. In the Identity Server console, click on the Properties arrow for the top-level organization (created during installation). The Organization Properties window will display in the Data frame.
2. Click Save to save the changes.
3. 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.
4. Open the `AMConfig.properties` file from the following default location:

   `/opt/SUNWam/lib`.
5. Replace all of the protocol occurrences of `http://` to `https://` and change the port number to an SSL-enabled port number.
6. Save the `AMConfig.properties` file.
7. Restart the Application Server.

SYMBOLS

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W