Technical Note: Sun Java System Access Manager ACI Guide

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

Part No: 820–1058–10
March 12, 2007

070312@17039

Technical Note: Sun Java System Access Manager ACI Guide

March 12, 2007

This technical note describes the ACIs (access control instructions) configured for the Sun Java™ System Access Manager 7 2005Q4 in both Realm Mode and Legacy Mode, including:

Introduction

This technical note describes the ACIs configured for the Sun Java Access Manager 7 2005Q4 in Realm and Legacy Modes of installation, in terms of the ACIs defined for Sun Java Access Manager 6 2005Q1 (6.3). The intent of this technical note is to describe the changes that have taken place, as far as ACIs are concerned, in Sun Java Access Manager 7 2005Q4 in comparison with the previous release of the product, Sun Java System Access Manager 6 2005Q1 (6.3), especially when Sun Java Access Manager 7 2005Q4 is configured to run in the Realm Mode of operation.

When Sun Java Access Manager 7 2005Q4 is configured in the Realm Mode of operation:

- The number of ACIs used are considerably lower than when configured in the Legacy Mode of operation. It's essentially a subset of the ACIs defined in Access Manager 7 2005Q4 Legacy Mode and Access Manager 6 2005Q1 (6.3). Hence, the performance overhead is reduced, due to fewer ACIs. The reason for using fewer ACIs is due to the adoption of a new access control model.
- The anonymous ACIs are deleted, to avoid any anonymous access.

The ACIs recorded in `install.ldif` for a new directory instance or `installExisting.ldif` for an existing directory instance files are created during the installation of Access Manager in the following directory, depending on your platform:

- Solaris systems: `/etc/opt/SUNWam/config/ldif`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config/ldif`
- Windows systems: *AccessManager-base*`\identity\config\ldif`

AccessManager-base is the base installation directory: `/opt` on Solaris systems and `/opt/sun` on Linux and HP-UX systems.

On Windows systems, *AccessManager-base* is *javaes-install-directory*`\AccessManager`. For example: `C:\Program Files\Sun\AccessManager`

Note – In this document, the terms `ORG_ROOT_SUFFIX` and `ROOT_SUFFIX` are the same and have the same value. Regard references in this document to those terms to be the same node in the directory DIT.

Access Manager 6 2005Q1 (6.3) and Access Manager 7 2005Q4 Legacy Mode

Overview

Legacy Mode is based on the Access Manager 6 2005Q1 (6.3) architecture. This legacy Access Manager architecture uses the LDAP directory information tree (DIT) that comes with Sun Java System Directory Server. In Legacy Mode, both user information and access control information are stored in LDAP organizations. Here, the delegation model is based on LDAP Roles, and not LDAP Groups. The ACIs are typically based on administrative roles, and they set by the Access Manager SDK at the time of role-creation. The relevant roles are:

- Top-level Admin role
- Top-level Help Desk Admin role
- Top-level Policy Admin role
- Organization Admin role
- Organization Help Desk Admin role
- Organization Policy Admin role

- People Container Admin role (People Admin)
- Group Admin role
- Container Admin role
- Deny Write Access role (for anonymous access)

In addition, there are the following types of ACIs:

- User
- Miscellaneous

Access Manager 7 2005Q4, when configured in Legacy Mode of operation, still uses the Directory Server ACI model to provide delegation to be backward-compatible.

Access Manager 7 2005Q4 Realm Mode

Overview

Realm-based architecture provides an independent tree structure to store the Organization Configuration data and User Management data.

To avoid the ACI-related performance issues and to make delegation easy to understand, in a Realm Mode installation of Access Manager 7 2005Q4, access control for the Identity Repository (IdRepo) framework and Service Management is based on the new Policy Management delegation model.

In Access Manager 7 2005Q4, the Realm Mode Policy infrastructure is created in a Realm instead of in an Organization.

Policy Management delegation uses the existing Policy Authorization mechanism, thus replacing the ACIs to determine the accessibilities of Realms and Policies. Policies are used to control the Realm and Policy delegations.

When a Realm gets created, a Policy for this Service is created for the access privileges of the Realm. The Subjects defined in the Policy, determine who is able to manage the Realm and Policies, and in what manner. Based on the Policy Conditions defined, restrictions are applied on the accessibilities of the users to the Realms and Polices.

The new Policy delegation model has introduced the concepts of the Realm Admin and Realm Policy Admin:

- The Realm Admin of a Realm has all the permissions to manage the Realm.
- The Realm Policy Admin has all the permissions to manage the Policies within the scope of the Realm.

At the time of creating a Realm in the Access Manager Console, the user needs to specify which Subjects will be used as the Realm Admin and which Subjects will be used as the Realm Policy Admin. Optionally, the user can specify some Conditions to further restrict the management of

the Realm and the Policies. Default delegation Policies are described in a delegation service. (For more information, see the `/etc/opt/SUNWam/config/request/defaultDelegationPolicies.xml` file.)

SM (Service Management) in Access Manager 7 2005Q4 Realm Mode, enforces Policies and Privileges for Realm access control. A Privilege is an Access Control mechanism for the resources within Access Manager, for example: service-configuration data and user data.

For the Identity Repository (IdRepo) framework, delegation is provided for pre-defined roles like Top-level Admin Role, Organization Admin Role and Help Desk Admin Role.

The new delegation model in Access Manager 7 2005Q4 Realm Mode serves the purpose of Access Manager being datastore-agnostic. The new Access Control model/delegation is managed from the Access Manager Console. Assuming a fresh Directory Server instance, that Access Manager 7 2005Q4 is installed into, there are equal number of ACIs loaded into the Directory Server instance, both in Legacy Mode and Realm Mode. The ACIs are loaded from either `install.ldif` or `installExisting.ldif` as described in the "Introduction" on page 3.

There is a performance tuning script to eliminate the unnecessary or unused ACIs installed in Access Manager. For more information, see under "Running the `amtune-directory` Script to Remove Unnecessary ACIs in Realm Mode" on page 21.

ACI Descriptions

Top-Level Admin Role ACIs

ACI 1:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetattr="*") (version 3.0; acl "S1IS Top-level admin rights"; allow (all)
roledn = "ldap:///cn=Top-level Admin Role,ROOT_SUFFIX"; )
```

Members of this specific role (cn=Top-level Admin Role) have all rights to all entries of the targeted resource ROOT_SUFFIX. The Top-Level Admin Role members can delete/read/modify/write to or from all entries under the top node. ROOT_SUFFIX is the root node.

ACI 2:

```
aci: (target="ldap:///cn=amldapuser,ou=DSAME Users,ORG_ROOT_SUFFIX")
(targetattr = "*") (version 3.0; acl "S1IS special ldap auth user modify right";
deny (write) roledn !="ldap:///cn=Top-level Admin Role,ROOT_SUFFIX";)
```

Members of this specific role (cn=Top-level Admin Role) can modify/write all entries of the targeted resource, (cn=amldapuser). In other words:

- modify/write access to the targeted entry (cn=amldapuser) is granted for the user who binds using a DN that belongs to the Top-Level Admin Role
- modify/write access to the targeted entry (cn=amldapuser) is denied if the user is not bound using a DN that belongs to the Top-Level Admin role

Top-Level Help Desk Admin Role ACIs

ACI 1:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)))
(targetattr= "*") (version 3.0; acl "S1IS Top-level Help Desk Admin Role access allow";
allow (read,search) roledn = "ldap:///cn=Top-level Help Desk Admin Role,ROOT_SUFFIX";)
```

Members with Top-level Help Desk Admin role:

- have permissions only to read or search all the entries under the default organization (root suffix node)
- do not have read or search permissions to the entries of Top-Level Admin Role members.

ACI 2:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)))
(targetattr= "userPassword")
(version 3.0; acl "S1IS Top-level Help Desk Admin Role access allow"; allow (write)
roledn ="ldap:///cn=Top-level Help Desk Admin Role,ROOT_SUFFIX";)
```

Members with Top-Level Help Desk Admin role:

- have write permission only to userPassword attribute for all members under the root suffix node/default organization

- do not have any write permission to the `userPassword` entry of Top-Level Admin Role members

Top-Level Policy Admin Role ACIs

ACI 1:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX))))
(targetattr = "*") (version 3.0; acl "S1IS Top-level Policy Admin Role access allow";
allow (read,search) roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

Members with Top-level Policy Admin role:

- have permissions only to read or search all the entries under the default organization (root suffix node)
- do not have any read or search permissions to the entries of Top-Level Admin Role members

ACI 2:

```
aci: (target="ldap:///ou=iPlanetAMAuthService,ou=services,*ROOT_SUFFIX")
(targetattr = "*")
(version 3.0; acl "S1IS Top-level Policy Admin Role access Auth Service deny";
deny(add,write,delete)
roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

Members with Top-Level Policy Admin role do not have permissions to add, write, or delete all the entries under the authentication service. This authentication service `iPlanetAMAuthService` is in the services node of the default organization (root suffix node). This ACI will also be enforced in the sub-organizations created under the default organization.

ACI 3:

```
aci: (target="ldap:///ou=services,*ROOT_SUFFIX")(targetattr = "*")
(version 3.0; acl "S1IS Top-level Policy Admin Role access allow"; allow (all)
roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

Members with Top-Level Policy Admin role have `all` permissions to read, modify, search, add, write, or delete to all the entries of all services under the default organization (root suffix node). But based on the ACI #2 above, this Top-Level Policy Admin does not have add, write, or delete permissions for authentication service. This ACI will also be enforced in the sub-organizations created under the default organization.

ACI 4:

```
aci:(target="ldap:///ROOT_SUFFIX")
(targetfilter="(objectclass=ORG_OBJECT_CLASS)")
(targetattr = "sunRegisteredServiceName") (version 3.0;
acl "S1IS Top-level Policy Admin Role access allow"; allow (read,write,search)
roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

Members with Top-Level Policy Admin role have permissions to read, write, or search the attribute `sunRegisteredServicename` of all entries with the object class that matches the `ORG_OBJECT_CLASS`.

For example:

```
aci: (target="ldap:///dc=iplanet,dc=com")
(targetfilter="(objectclass=sunmanagedorganization)")
(targetattr = "sunRegisteredServiceName") (version 3.0;
acl "S1IS Top-level Policy Admin Role access allow"; allow (read,write,search)
roledn = "ldap:///cn=Top-level Policy Admin Role,dc=iplanet,dc=com";)
```

Organization Admin Role ACIs

ACI 1:

```
aci: (target="ldap:///($dn),ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX))))
(targetattr != "nsroledn")(version 3.0;
acl "S1IS Organization Admin Role access allow all";
 allow (all) roledn = "ldap:///cn=Organization Admin Role,[$dn],ORG_ROOT_SUFFIX";)
```

This ACI gives `all` permissions to the members who belong to the Organization Admin Role. Members of Organization Admin Role have 'all' permissions to all the entries and attributes for that organization on the organization entry. But the `'all'` access is not applied to the `nsroledn` attribute where the values are `Top-level Admin Role`, `Top-level Help Desk Admin Role`, `Top-level Policy Admin Role`.

In other words, members of Organization Admin Role cannot read, write, delete, modify, or searchthe directory entries of Top-level Admin, Top-level Help Desk Admin, and Top-level Policy Admin. But members of Organization Admin Role have permission to modify the `nsroledn` attribute in their own profiles; however, they cannot assign the following values to the `nsroledn` attribute:

- Top-level Admin Role
- Top-level Help Desk Admin Role
- Top-level Policy Admin Role

ACI 2:

```
aci: (target="ldap:///cn=Organization Admin Role,($dn),ORG_ROOT_SUFFIX")
(targetattr="*")(version 3.0; acl "S1IS Organization Admin Role access deny";
deny (write,add,delete,compare,proxy)
 roledn = "ldap:///cn=Organization Admin Role,($dn),ORG_ROOT_SUFFIX";)
```

Members of Organization Admin Role are denied write, add, delete, compare, or proxy permissions to all the attributes for that organization admin role entry.

- ACI #1 allows all modification of everything under the sub-tree in which the role exists, except being able to edit users with the top level admin and top level help desk admin roles.
- ACI #2 prevents organization admins from modifying their attributes. ACI #2 is needed so that Org Admin role can give roles to users that are strictly defined only under this sub-tree.

Organization Help Desk Admin Role ACIs

ACI 1:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Organization Admin Role,ORG_ROOT_SUFFIX))))(targetattr = "*")
(version 3.0; acl "S1IS Organization Help Desk Admin Role access allow";
allow (read,search)
roledn = "ldap:///cn=Organization Help Desk Admin Role,ORG_ROOT_SUFFIX";)
```

Members of Organization Help Desk Admin Role:

- have read and search rights to all entries under the root suffix
- do not have any rights to read or search the members who belong to Top-level Help Desk Admin Role, Top-level Policy Admin Role, and Organization Admin Role.

ACI 2:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Organization Admin Role,ORG_ROOT_SUFFIX))))
(targetattr = "userPassword")
(version 3.0; acl "S1IS Organization Help Desk Admin Role access allow";
allow (write) roledn = "ldap:///cn=Organization Help Desk Admin Role,ORG_ROOT_SUFFIX";)
```

Members of Organization Help Desk Admin Role:

- have write permissions to the `userPassword` attribute for all users under the root suffix.

- do not have write permissions to `userPassword` attribute for the members who belong to Top-level Help Desk Admin Role, Top-level Policy Admin Role, and Organization Admin Role.

Container Admin Role ACIs

ACI 1:

```
aci: (target="ldap:///($dn),ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX))))
(targetattr != "nsroledn")(version 3.0; acl "S1IS Container Admin Role access allow";
allow (all) roledn = "ldap:///cn=Container Admin Role,[$dn],ORG_ROOT_SUFFIX";)
```

This ACI gives `'all'` permissions to the members who belong to the Container Admin Role. Therefore, members of Container Admin Role have `'all'` permissions to all the entries and attributes for that sub-organization on the sub-organization entry. But the `'all'` access is not applicable to the `nsroledn`attribute, if the values for `nsroledn` are one or more of the following:

- Top-level Admin Role
- Top-level Help Desk Admin Role
- Top-level Policy Admin Role

In other words, members of Container Admin Role cannot read, write, delete, modify, or search the directory entries of members belonging to the above-listed roles. However, members of Container Admin Role have permissions to modify the `nsroledn` attribute in their own profiles.

ACI 2:

```
aci: (target="ldap:///cn=Container Admin Role,($dn),ORG_ROOT_SUFFIX")
(targetattr="*")(version 3.0; acl "S1IS Container Admin Role access deny";
deny (write,add,delete,compare,proxy)
roledn = "ldap:///cn=Container Admin Role,($dn),ORG_ROOT_SUFFIX";)
```

This ACI is for Container Admin Role. Members of Container Admin Role are denied write, add, delete, compare, and proxy permissions to all the attributes for that container/sub-organization admin role entry.

Deny Write Access Role ACIs

ACI 1:

```
aci: (targetattr = "*")
(version 3.0; acl "S1IS Deny write to anonymous user"; deny (add,write,delete)
roledn ="ldap:///cn=Deny Write Access,ROOT_SUFFIX";)
```

Members of the Deny Write Access role (that is, anonymous users) do not have add, write, or delete rights to all entries under the root suffix. Anonymous users are allowed only to search and read entries.

User ACIs

ACI 1:

```
aci: (targetattr = "objectclass  || inetuserstatus || iplanet-am-user-login-status
|| iplanet-am-web-agent-access-allow-list || iplanet-am-domain-url-access-allow
|| iplanet-am-web-agent-access-deny-list || iplanet-am-user-account-life
|| iplanet-am-session-max-session-time || iplanet-am-session-max-idle-time
|| iplanet-am-session-get-valid-sessions || iplanet-am-session-destroy-sessions
|| iplanet-am-session-add-session-listener-on-all-sessions
|| iplanet-am-user-admin-start-dn || iplanet-am-auth-post-login-process-class")
(targetfilter=(!(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)))
(version 3.0; acl "S1IS User status self modification denied";
deny (write) userdn="ldap:///self";)
```

This ACI specifically prevents users from writing or modifying certain attributes (mentioned in the target attribute of the ACI) to their own directory entry. Of course, if these entries needed to be modified, an Admin user would be able to do it.

ACI 2:

```
aci: (targetattr != "iplanet-am-static-group-dn  || uid || nsroledn || aci
|| nsLookThroughLimit || nsSizeLimit || nsTimeLimit || nsIdleTimeout || memberOf
|| iplanet-am-web-agent-access-allow-list || iplanet-am-domain-url-access-allow
|| iplanet-am-web-agent-access-deny-list")
(version 3.0; acl "S1IS Allow self entry modification except for nsroledn, aci,
and resource limit attributes"; allow (write)userdn ="ldap:///self";)
```

This ACI specifically prevents users from writing or modifying certain attributes to their own directory entry. But the Organization Admin Role ACIs defined override this ACI and allows self modification of the `nsroledn` attribute, so that administrators can assign themselves certain service roles and lesser or equal privileged admin roles. This is because the current ACIs prevent the organization admin from assigning the top-level admin roles.

ACI 3:

```
aci: (targetattr != "aci  || nsLookThroughLimit || nsSizeLimit  || nsTimeLimit
|| nsIdleTimeout  || iplanet-am-domain-url-access-allow") (version 3.0;
acl "S1IS Allow self entry read search except for nsroledn, aci, resource limit
and web agent policy attributes"; allow (read,search)userdn ="ldap:///self";)
```

This ACI specifically allows users to read or search certain attributes from their own directory entry. But this ACI does not allow the following target attributes to be read by the users in their own directory entries: `aci`, `nsLookThroughLimit`, `nsSizeLimit`, `nsTimeLimit`, `nsIdleTimeout`, and `iplanet-am-domain-url-access-allow`.

ACI 4:

```
aci: (targetattr = "*")(version 3.0;
acl "S1IS Deny deleting self"; deny (delete) userdn  ="ldap:///self";)
```

This ACI specifically prevents users from deleting all attributes from their own directory entries.

Miscellaneous ACIs

ACI 1:

```
aci: (target="ldap:///cn=schema")(targetattr="*")
(version 3.0; acl "S1IS Proxy user rights"; allow (proxy)
userdn = "ldap:///cn=puser,ou=DSAME Users,ORG_ROOT_SUFFIX"; )
```

This ACI states that the DN `cn=puser` has proxy rights to access the target directory entry that contains all the schema information for the server (that is `cn=schema`). It has the rights of Directory Manager entry (`cn=Directory Manager`) to do this. (Only Directory Manager has write permission on the schema and no other user has write permission on the schema.)

In other words, the proxy user DN (`cn=puser`) gains access to the `cn=schema` subtree using the same access permissions as the Directory Manager. With this ACI in place, the `puser` can bind to the directory and send an LDAP command such as `ldapsearch` or `ldapmodify` that requires the access rights of the Directory Manager.

ACI 2:

```
aci: (target="ldap:///ROOT_SUFFIX")(targetattr="*")
(version 3.0; acl "S1IS Proxy user rights"; allow (proxy)
userdn = "ldap:///cn=puser,ou=DSAME Users,ORG_ROOT_SUFFIX"; )
```

This ACI states that the DN `cn=puser` has proxy rights to access the target directory entry which is the top organization or root node. It has the rights of Directory Manager entry (`cn=Directory Manager`) to do this. In other words, the proxy user DN (`cn=puser`) gains access to the top organization or root node using the same access permissions as the Directory Manager. With this ACI in place, the `puser` can bind to the directory and send an LDAP command such as `ldapsearch` or `ldapmodify` that requires the access rights of the Directory Manager.

ACI 3:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetattr="*")(version 3.0; acl "S1IS special ldap  auth user rights";
allow (read,search) userdn = "ldap:///cn=amldapuser,ou=DSAME Users,ORG_ROOT_SUFFIX"; )
```

This ACI states that the DN `cn=amldapuser` has only read and search rights to all entries under the target directory entry as well the target directory entry which is the top organization or root node. In other words, the `amldapuser` DN (`cn=amldapuser`) has read and search rights to the targeted entry. `amldapuser` is the bind DN user for LDAP Authentication, Membership, and Policy services. This user has read and search access to all Directory Server entries.

ACI 4:

```
aci: (target="ldap:///ROOT_SUFFIX") (targetattr="*")
(version 3.0; acl "S1IS special dsame  user rights for all under the root suffix";
allow (all) userdn = "ldap:///cn=dsameuser,ou=DSAME Users,ORG_ROOT_SUFFIX"; )
```

This ACI states that the DN `cn=dsameuser` has all rights to access all entries under the target directory entry as well the target directory entry which is the top organization or root node. In other words, the `dsameuser` DN (`cn=dsameuser`) has all rights (read, write, search, delete, compare, and selfwrite) to the targeted entry, except proxy rights. `dsameuser` retrieves the LDAP configuration (for users, organizations, policies, services, agents, etc.) for the Access Manager SDK. The Directory Server administrator (by default `uid=admin,ou=Administrators, ou=TopologyManagement,o=NetscapeRoot`) has all rights except proxy rights.

ACI 5:

```
aci: (targetattr="iplanet-am-saml-user  || iplanet-am-saml-password")
(targetfilter="(objectclass=iplanet-am-saml-service)")(version 3.0;
acl "S1IS Right to modify saml  user and password"; deny (all)
(roledn != "ldap:///cn=Top-level Admin Role,ROOT_SUFFIX")
AND (userdn  != "ldap:///cn=dsameuser,ou=DSAME Users,ORG_ROOT_SUFFIX")
AND (userdn != "ldap:///cn=puser,ou=DSAME Users,ORG_ROOT_SUFFIX"); )
```

Only special users (such as `dsameuser`, `proxyuser`, or top-level admin) can configure the SAML service at the global level. SAML service attributes and values are added as key/value pair for the trusted partners Trusted Partner Sites in the console using the edit button and the passwords are not encrypted. Liberty and SAML does not want all users to see the values in clear text. This ACI denies access to SAML Service for all users but gives permission to members who belong to the Top-Level Admin role and `puser` and `dsameuser`.

ACI 6:

```
aci: (target="ldap:///ou=services,ROOT_SUFFIX")
(targetfilter=(!(objectclass=sunServiceComponent)))
(targetattr = "*")(version 3.0; acl "S1IS Services anonymous access";
allow (read, search, compare) userdn = "ldap:///anyone";)
```

This ACI allows anyone anonymous read, search, and compare access to the Service Schema, which is defined under the `ou=services` node of the tree. But this ACI does not allow anyone read, search, or compare access to the Service Configuration entries (Deny if `objectclass=sunServiceComponent`. That is, deny access to Service Configuration).

ACI 7:

```
aci: (target="ldap:///ou=iPlanetAMAdminConsoleService,*,ROOT_SUFFIX")
(targetattr = "*")(version 3.0; acl "S1IS iPlanetAMAdminConsoleService
anonymous access"; allow (read, search, compare) userdn  = "ldap:///anyone";)
```

This ACI allows anonymous read, search, and compare access to all the attributes under `ou=iPlanetAMAdminConsoleService` node of the tree. In an Access Manager 6 2005Q1 (6.3) and Access Manager 7 2005Q4 Legacy Mode installation, the console service (`iPlanetAMConsoleService`) can be under any Organization, and it is not restricted to be only under the root suffix. This ACI facilitates the privilege of reading this service for any Organization.

Important: Consider the potential performance impact of evaluation of this ACI.

ACI 8:

```
aci: (target="ldap:///cn=Top-level Admin Role,ROOT_SUFFIX")
(targetattr="*")(version 3.0; acl "S1IS Top-level admin delete right denied";
deny (delete) userdn = "ldap:///anyone"; )
```

Any user or users with anonymous access cannot delete the members of Top-Level Admin Role.

ACI 9:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(entrydn=ORG_ROOT_SUFFIX))(targetattr="*")
(version 3.0; acl "S1IS Default Organization delete right denied";
deny (delete) userdn = "ldap:///anyone"; )
```

Any user or users with anonymous access cannot delete the top level default organization.

Dynamic ACIs

These ACIs are created at runtime when a new Organization, People Container, Group is created.

- "Organization Policy Admin Role ACIs" on page 16
- "People Container Admin Role ACIs" on page 16
- "Group Admin Role ACIs" on page 17

Organization Policy Admin Role ACIs

ACI 1 example:

```
aci=(target="ldap:///o=suborg,dc=iplanet,dc=com")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,dc=iplanet,dc=com)
(nsroledn=cn=Top-level Help Desk Admin Role,dc=iplanet,dc=com)
(nsroledn=cn=Organization Admin Role,o=suborg,dc=iplanet,dc=com))))
(targetattr = "*")(version 3.0; acl "Organization Policy Admin access allow";
allow (read,search)
roledn = "ldap:///cn=Organization Policy Admin Role,o=suborg,dc=iplanet,dc=com";)
aci=(target="ldap:///ou=services,*o=suborg,dc=iplanet,dc=com")(targetattr = "*")
(version 3.0; acl "Organization Policy Admin Role access allow"; allow (all)
roledn = "ldap:///cn=Organization Policy Admin Role,o=suborg,dc=iplanet,dc=com";)
```

ACI 2 example:

```
aci=(target="ldap:///ou=iPlanetAMAuthService,ou=services,
*o=suborg,dc=iplanet,dc=com") (targetattr = "*")
(version 3.0; acl "Organization Policy Admin Role access Auth Service deny";
deny (add,write,delete)
roledn = "ldap:///cn=Organization Policy Admin Role,o=suborg,dc=iplanet,dc=com";
```

ACI 3 example:

```
aci=(target="ldap:///o=suborg,dc=iplanet,dc=com")
(targetfilter="(objectclass=sunmanagedorganization)")
(targetattr = "sunRegisteredServiceName")
(version 3.0; acl "Organization Policy Admin Role access allow";
allow (read,write,search)
roledn = "ldap:///cn=Organization Policy Admin Role,o=suborg,dc=iplanet,dc=com";)
```

People Container Admin Role ACIs

ACI 1:

```
aci: (target="ldap:///ou=People,ORG_ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Organization Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Container Admin Role,ORG_ROOT_SUFFIX))))
(targetattr != "iplanet-am-web-agent-access-allow-list
|| iplanet-am-domain-url-access-allow
|| iplanet-am-web-agent-access-deny-list || nsroledn")
(version 3.0; acl "S1IS Group and people container admin role";
allow (all) roledn ="ldap:///cn=ou=People_NM_ORG_ROOT_SUFFIX,ORG_ROOT_SUFFIX";)
```

Members of Group container role and People container role have all rights to all entries under the node `ou=People` of the root suffix. But they do not have any rights for the members who belong to Top-level Help Desk Admin Role, Top-level Policy Admin Role, Container Admin Role and Organization Admin Role. In addition members of Group container role and People container role do not have any rights to access the following attributes:

- `iplanet-am-web-agent-access-allow-list`
- `iplanet-am-domain-url-access-allow`
- `iplanet-am-web-agent-access-deny-list`
- `nsroledn`

Group Admin Role ACIs

ACI 1 example:

```
aci=(target="ldap:///ou=People,dc=iplanet,dc=com") (targetattr="nsroledn")
(targattrfilters="add=nsroledn:(!(nsroledn=*)),del=nsroledn:(!(nsroledn=*))")
(version 3.0; acl "Group admin's  right to add user to people container"; allow (add)
roledn ="ldap:///cn=cn=blach_ou=Groups_dc=iplanet_dc=com,dc=iplanet,dc=com";)
```

ACI 2 example:

```
aci=(target="ldap:///cn=blach,ou=Groups, dc=iplanet,dc=com")
(targetattr = "*") (version 3.0; acl "Group and people container admin role";
allow (all)
roledn = "ldap:///cn=cn=blach_ou=Groups_dc=iplanet_dc=com,dc=iplanet,dc=com";)
```

ACI 3 example:

```
aci=(target="ldap:///dc=iplanet,dc=com")
(targetfilter=(!(|(!(|(memberof=*cn=blach,ou=Groups, dc=iplanet,dc=com)
(iplanet-am-static-group-dn=*cn=blach,ou=Groups,dc=iplanet,dc=com)))
(|(nsroledn=cn=Top-level Admin Role,dc=iplanet,dc=com)
(nsroledn=cn=Top-level Help Desk Admin Role,dc=iplanet,dc=com)
(nsroledn=cn=Top-level Policy Admin Role,dc=iplanet,dc=com)
(nsroledn=cn=Organization Admin Role,dc=iplanet,dc=com)
(nsroledn=cn=Container Admin Role,dc=iplanet,dc=com)
(nsroledn=cn=Organization Policy Admin Role,dc=iplanet,dc=com)))))
(targetattr  != "iplanet-am-web-agent-access-allow-list
||iplanet-am-web-agent-access-not-enforced-list || iplanet-am-domain-url-access-allow
|| iplanet-am-web-agent-access-deny-list || nsroledn")
(version 3.0; acl  "Group admin's right to the members"; allow (read,write,search)
roledn  = "ldap:///cn=cn=blach_ou=Groups_dc=iplanet_dc=com,dc=iplanet,dc=com";)
```

Elimination of ACIs During Installation

The following ACIs, which were defined by Directory Server, are deleted from the Directory Server during installation of Access Manager.

ACI 1:

```
aci: (targetattr != "userPassword")
(version 3.0; acl "Anonymous access";
allow (read, search, compare)userdn = "ldap:///anyone";)
```

All users have anonymous access to the directory for search, compare, and read operations, except for the following attribute:

`userPassword`

ACI 2:

```
aci:(targetattr != "userPassword || passwordHistory")
(version 3.0; acl "Anonymous access";
allow (read, search, compare)userdn = "ldap:///anyone";)
```

All users have anonymous access to the directory for search, compare, and read operations, except for the following attributes:

- `userPassword`
- `passwordHistory`

ACI 3:

```
aci:(targetattr != "userPassword || passwordHistory || passwordExpirationTime
|| passwordExpWarned || passwordRetryCount || retryCountResetTime || accountUnlockTime
|| passwordAllowChangeTime ") (version 3.0; acl "Anonymous access";
allow (read, search, compare)userdn = "ldap:///anyone";)
```

All users have anonymous access to the directory for search, compare, and read operations, except for the following attributes:

- `userPassword`
- `passwordHistory`
- `passwordExpirationTime`
- `passwordExpWarned`
- `passwordRetryCount`
- `retryCountResetTime`
- `accountUnlockTime`
- `passwordAllowChangeTime`

ACI 4:

```
aci: (targetattr != "nsroledn || aci || nsLookThroughLimit || nsSizeLimit
|| nsTimeLimit || nsIdleTimeout || passwordPolicySubentry ")
(version 3.0; acl "S1IS Allow self entry modification except for nsroledn,
aci, resource limit attributes,and passwordPolicySubentry";
allow (write)userdn ="ldap:///self";)
```

This ACI specifically prevents all users with 'self' access to the Directory Server from writing to certain attributes. Access Manager deletes this self-access ACI during installation, to allow self-access for some Administrative functions, for instance to allow the Organization Admins to modify their own profiles. Since the current ACIs do prevent Organization Admins from assigning the Top-level Admin roles, they should be allowed to assign thenselves other administrative (and service) roles, which can only be lesser in privilege to their current capabilities. The deletion of this ACI helps achieve the requirement for the Organization Admin to be able to modify the `nsroledn` attribute in his profile.

Custom Tuning of ACIs in Legacy Mode

Organizations

The creation of the following roles and the related ACIs, every time an organization is created, can be eliminated:

- `Organization Admin Role`
- `Organization Help Desk Admin Role`
- `Policy Admin Role`

Eliminate the roles and the related ACIs by making a change to the DAI service in the `/etc/opt/SUNWam/config/ums/ums.xml` file.

You can selectively remove only one of these roles, instead of all of them:

```
<AttributeValuePair>
      <Attribute name="childNode" />
      <Value>PeopleContainer</Value>
      <Value>GroupContainer</Value>
      <Value>DefaultOrgRole</Value>
      <Value>DPOrgAdminRole</Value>
      <Value>DPOrgHelpDeskAdminRole</Value>
      <Value>DPOrgPolicyAdminRole</Value>
 </AttributeValuePair>
```

The above are lines 143-151 in the `ums.xml` file.

It is not possible to eliminate the creation of this role: `People Admin Role`.

Every time an organization is created, a default People container is created and along with the People container, this role is also created. If you do not need this role, you may delete this role from the Access Manager Console. That will clean up all the ACIs related to this role as well.

Organizational Unit or Containers

When a Container is created, the following roles are created by default:

- `Container Admin Role`
- `Container Help-Desk Admin Role`
- `People Admin Role` (for the default People container that is created)

The creation of the following roles and the related ACIs, every time an organization is created, can be eliminated:

- `Container Admin Role`
- `Container Help Desk Admin Role`

Eliminate the roles and the related ACIs by making the following changes to the DAI service in the `/etc/opt/SUNWam/config/ums/ums.xml` file.

You can selectively remove only one of these roles, instead of all of them:

```
<AttributeValuePair>
       <Attribute name="childNode" />
      <Value>PeopleContainer</Value>
      <Value>GroupContainer</Value>
      <Value>DPOrgUnitAdminRole</Value>
      <Value>DPOrgUnitHelpDeskAdminRole</Value>
 </AttributeValuePair>
```

The above are lines 170-175 in the `/etc/opt/SUNWam/config/ums/ums.xml` file.

It is not possible to eliminate the creation of this role: `People Admin Role`.

Every time an organization is created, a default People container is created and along with the People container, this role is also created. If you do not need this role, you may delete this role from the Access Manager Console. That will clean up all the ACIs related to this role as well.

Groups

To prevent the creation of the Group Admin Role and related ACIs every time a group is created, do the following in the Access Manager Console:

1. Choose the Admin Console Service from the Services Configuration tab.
2. Select Group Admin permission from the list of Dynamic Administrative role ACIs in the global configuration.
3. Delete this permission by clicking Remove.
4. Save the configuration change.

The roles and relates ACIs will no longer be created when a group is created.

Note – None of the new groups will have this facility. The permission and role creation is deleted permanently.

Running the amtune-directory Script to Remove Unnecessary ACIs in Realm Mode

Overview

If Access Manager 7 2005Q4 is installed in Realm Mode, delegation privileges are used to determine access permissions, and therefore some Directory Server ACIs are not needed. Access Manager 7 2005Q4 patch 5 allows you to remove the unnecessary ACIs by running the `amtune-directory` script, which is generated by the `amtune-prepareDSTuner` script. This script read a list of ACIs from the `remacis.ldif` file and then calls the `ldapmodify` utility to remove them.

The Access Manager tuning scripts are available in the following directory, depending on your platform:

- Solaris systems: *AccessManager-base*`/SUNWam/bin/amtune`
- Linux and HP-UX systems: *AccessManager-base*`/identity/bin/amtune`
- Windows systems: *AccessManager-base*`\identity\bin\amtune`

AccessManager-base is the base installation directory: `/opt` on Solaris systems and `/opt/sun` on Linux and HP-UX systems.

On Windows systems, *AccessManager-base* is *javaes-install-directory*\AccessManager. For example: C:\Program Files\Sun\AccessManager

Access Manager 7 2005Q4 patch 5 allows you to run the tuning scripts with either a password file or the password string as a command-line argument.

For more information about the Access Manager tuning scripts, see the *Sun Java System Access Manager 7 2005Q4 Performance Tuning Guide*.

Removing ACIs on Solaris, Linux, and HP-UX Systems

To remove unneeded ACIs on Solaris, Linux, and HP-UX systems in Realm Mode:

1. On the Access Manager server, login as or become superuser (root).
2. To ensure that Access Manager is in Realm Mode, check the AM_REALM parameter in the amsamplesilent file . The parameter should be set as follows:

   ```
   AM_REALM="enabled"
   ```

   The amsamplesilent file is located in the following directory, depending on your platform:

   - Solaris systems: *AccessManager-base*/SUNWam/bin
   - Linux and HP-UX systems: *AccessManager-base*/identity/bin

   AccessManager-base is the base installation directory: /opt on Solaris systems and /opt/sun on Linux and HP-UX systems.
3. Run the amtune-prepareDSTuner script to create the amtune-directory.tar file.
4. Copy the amtune-directory.tar file to a temporary location on the Directory Server machine and untar the file in the temporary location.
5. Because he amtune-directory script tunes Directory Server, it is recommended that you first run the script in REVIEW mode. In the amtune-directory script, set REVIEW mode as follows:

   ```
   AMTUNE_MODE="REVIEW"
   ```
6. Run the amtune-directory script in REVIEW mode and review the recommended tuning settings for Directory Server in the tuning log file.
7. If the changes in the debug log file are acceptable for your deployment, modify the amtune-directory script to run in CHANGE mode by setting AMTUNE_MODE as follows:

   ```
   AMTUNE_MODE="CHANGE"
   ```
8. Backup the Directory Server data.
9. Run the amtune-directory script to remove the ACIs.
10. Check the tuning log file for the results of the run.

Removing ACIs on Windows Systems

On Windows systems, the Access Manager tuning scripts are written in Perl and require Active Perl 5.8.

To remove unneeded ACIs on Windows systems in Realm Mode:

1. On the Access Manager server, login as an administrator.
2. To ensure that Access Manager is in Realm Mode, check the AM_REALM parameter in the AMConfigurator.properties file . The parameter should be set as follows:

   AM_REALM="enabled"

   The AMConfigurator.properties file is located in the *AccessManager-base*\identity\bin directory.

   On Windows systems, *AccessManager-base* is *javaes-install-directory*\AccessManager. For example: C:\Program Files\Sun\AccessManager
3. In the amtune-env.pl file, set $BASEDIR to the Access Manager installation directory.
4. Run the amtune-prepareDSTuner.pl script to generate the required tuning scripts and files.
5. Copy the amtune-utils.pl, amtune-directory.pl, remacis.ldif, and amtune-samplepassordfile files from the previous step to a temporary directory on the Directory Server machine.
6. Because he amtune-directory.pl script tunes Directory Server, it is recommended that you first run the script in REVIEW mode. In the amtune-directory.pl script on the Directory Server machine, set REVIEW mode as follows:

   AMTUNE_MODE="REVIEW"
7. On the Directory Server machine, run the amtune-directory.pl script in REVIEW mode and review the recommended tuning settings for Directory Server in the tuning log file.
8. If the changes in the debug log file are acceptable for your deployment, modify the amtune-directory.pl script to run in CHANGE mode by setting AMTUNE_MODE as follows:

   AMTUNE_MODE="CHANGE"
9. Backup the Directory Server data.
10. On the Directory Server machine, run the amtune-directory script to remove the ACIs.
11. Check the tuning log file for the results of the run.

ACIs That are Removed by the amtune-directory Script

The following ACIs in the remacis.ldif file are removed by the amtune-directory script when Access Manager is installed in Realm Mode:

ACI 1:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(entrydn=ORG_ROOT_SUFFIX))(targetattr="*")
(version 3.0; acl "S1IS Default Organization delete right denied";
deny (delete) userdn = "ldap:///anyone"; )
```

ACI 2:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)))(targetattr = "*")
(version 3.0; acl "S1IS Top-level Help Desk Admin Role access allow";
allow (read,search) roledn = "ldap:///cn=Top-level Help Desk Admin Role,ROOT_SUFFIX";)
```

ACI 3:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)))
(targetattr = "userPassword") (version 3.0;
acl "S1IS Top-level Help Desk Admin Role access allow"; allow (write)
roledn = "ldap:///cn=Top-level Help Desk Admin Role,ROOT_SUFFIX";)
```

ACI 4:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX))))(targetattr = "*")
(version 3.0; acl "S1IS Top-level Policy Admin Role access allow";
allow (read,search) roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

ACI 5:

```
aci: (target="ldap:///ou=iPlanetAMAuthService,ou=services,*ROOT_SUFFIX")
(targetattr = "*") (version 3.0;
acl "S1IS Top-level Policy Admin Role access Auth Servi ce deny";
deny (add,write,delete) roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

ACI 6:

```
aci: (target="ldap:///ou=services,*ROOT_SUFFIX")
(targetattr = "*") (version 3.0; acl "S1IS Top-level Policy Admin Role access allow";
allow (all) roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

ACI 7:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter="(objectclass=ORG_OBJECT_CLASS)")
(targetattr = "sunRegisteredServiceName") (version 3.0;
acl "S1IS Top-level Policy Admin Role access allow"; allow (read,write,search)
roledn = "ldap:///cn=Top-level Policy Admin Role,ROOT_SUFFIX";)
```

ACI 8:

```
aci: (targetattr != "aci || nsLookThroughLimit || nsSizeLimit || nsTimeLimit
|| nsIdleTimeout || iplanet-am-domain-url-access-allow") (version 3.0;
acl "S1IS Allow self entry read search except for nsroledn, aci, resource limit
and web agent policy attributes"; allow (read,search)userdn ="ldap:///self";)
```

ACI 9:

```
aci: (target="ldap:///ou=iPlanetAMAdminConsoleService,*,ROOT_SUFFIX")
(targetattr = "*")(version 3.0;
acl "S1IS iPlanetAMAdminConsoleService anonymous access";
allow (read, search, compare) userdn = "ldap:///anyone";)
```

ACI 10:

```
aci: (target="ldap:///($dn),ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX))))
(targetattr != "nsroledn")(version 3.0;
acl "S1IS Organization Admin Role access allow all";
allow (all) roledn = "ldap:///cn=Organization Admin Role,[$dn],ORG_ROOT_SUFFIX";)
```

ACI 11:

```
aci: (target="ldap:///cn=Organization Admin Role,($dn),ORG_ROOT_SUFFIX")
(targetattr="*")(version 3.0; acl "S1IS Organization Admin Role access deny";
deny (write,add,delete,compare,proxy) roledn = "ldap:///cn=Organization Admin Role,
($dn),ORG_ROOT_SUFFIX";)
```

ACI 12:

```
aci: (target="ldap:///($dn),ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX))))
(targetattr != "nsroledn")(version 3.0; acl "S1IS Container Admin Role access allow";
allow (all) roledn = "ldap:///cn=Container Admin Role,[$dn],ORG_ROOT_SUFFIX";)
```

ACI 13.

```
aci: (target="ldap:///cn=Container Admin Role,($dn),ORG_ROOT_SUFFIX")
(targetattr="*")(version 3.0; acl "S1IS Container Admin Role access deny";
deny (write,add,delete,compare,proxy)
roledn = "ldap:///cn=Container Admin Role,($dn),ORG_ROOT_SUFFIX";)
```

ACI 14:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetattr!="nsroledn")(version 3.0;
acl "S1IS Group admin's right to the users he creates";
allow (all) userattr = "iplanet-am-modifiable-by#ROLEDN";)
```

ACI 15:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Organization Admin Role,ORG_ROOT_SUFFIX))))(targetattr = "*")
(version 3.0; acl "S1IS Organization Help Desk Admin Role access allow";
allow (read,search)
roledn = "ldap:///cn=Organization Help Desk Admin Role,ORG_ROOT_SUFFIX";)
```

ACI 16:

```
aci: (target="ldap:///ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Organization Admin Role,ORG_ROOT_SUFFIX))))
(targetattr = "userPassword") (version 3.0;
acl "S1IS Organization Help Desk Admin Role access allow";
allow (write) roledn = "ldap:///cn=Organization Help Desk Admin Role,ORG_ROOT_SUFFIX";)
```

ACI 17:

```
aci: (target="ldap:///ou=People,ORG_ROOT_SUFFIX")
(targetfilter=(!(|(nsroledn=cn=Top-level Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Help Desk Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Top-level Policy Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Organization Admin Role,ROOT_SUFFIX)
(nsroledn=cn=Container Admin Role,ORG_ROOT_SUFFIX))))
(targetattr != "iplanet-am-web-agent-access-allow-list
|| iplanet-am-domain-url-access-allow || iplanet-am-web-agent-access-deny-list
|| nsroledn") (version 3.0; acl "S1IS Group and people container admin role";
allow (all) roledn = "ldap:///cn=ou=People_NM_ORG_ROOT_SUFFIX,ORG_ROOT_SUFFIX";)
```

Accessing Sun Resources Online

The docs.sun.com web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

Third-Party Web Site References

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

Note – 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 or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 820-1058.

Revision History

Release Date	Description of Changes
March 12, 2007	Initial publication