

## Chapter 8

# Access Control Scheme

Each of the SRC components has an entry in the directory under *ou = components*, *o = operators*, *o = umc*. Service providers can establish a multilayered access control scheme for operators. For instance, a network operator might be able to write new objects only under the folder *o = network*. The operator entries are subordinates of *o = operators*, *o = umc*.

Topics in this chapter include:

- Directory Configuration on page 67
- Directories on page 68
- User Class on page 68
- Permissions on page 68
- Access Controls on page 69
- Directory-Specific Access Control Implementation on page 78

## Directory Configuration

---

During configuration of the directory, the following entries for components and operators are created:

- bind DN for SSP: *cn = ssp, ou = components, o = operators, o = umc*
- bind DN for RADIUS: *cn = radius, ou = components, o = operators, o = umc*
- bind DN for POM: *cn = pom, ou = components, o = operators, o = umc*
- bind DN for directory eventing: *cn = des, ou = components, o = operators, o = umc*
- bind DN for object state machine: *cn = osm, ou = components, o = operators, o = umc*
- bind DN for system management: *cn = sysman, ou = components, o = operators, o = umc*
- bind DN for SDX operators: *cn = ssc-operator, o = operators, o = umc*

- bind DN for network operators: *cn = network-operators, o = operators, o = umc*
- bind DN for service operators: *cn = service-operator, o = operators, o = umc*
- bind DN for subscriber operators: *cn = subscriber-operator, o = operators, o = umc*

## Directories

---

Directories specify the access rights for certain users to particular information in the directory, whereas other users might not receive any rights to that information. The access rights are defined through access control lists. Using access control lists, you can define permissions to the following targets:

- Entire directory content
- Particular subtree in the directory
- Objects that match a given search filter
- Specific object in the directory

The objects that are part of the target can be protected on an entry level and on an attribute level.

## User Class

---

The access control lists specify the user class from which the items are protected. The user class can be one of the following:

- Specific user
- Members of a specific group
- All entries of a subtree
- Users that match a given search filter
- All users
- This entry (for self-administration)

## Permissions

---

You must set permissions for the target. The following permissions are available:

- Add
- Search
- Compare

- Filter match
- Modify (write)
- Read
- Remove (delete)
- Rename

You can grant or deny these permissions. Deny takes precedence over grant.

## Access Controls

---

### Access Controls for the Entire Tree

A client who accesses the directory without binding to it does not have any access rights. All clients who bind with the credentials of an SRC component or an operator are members of the SSC-component-operator group and by default have the following access rights:

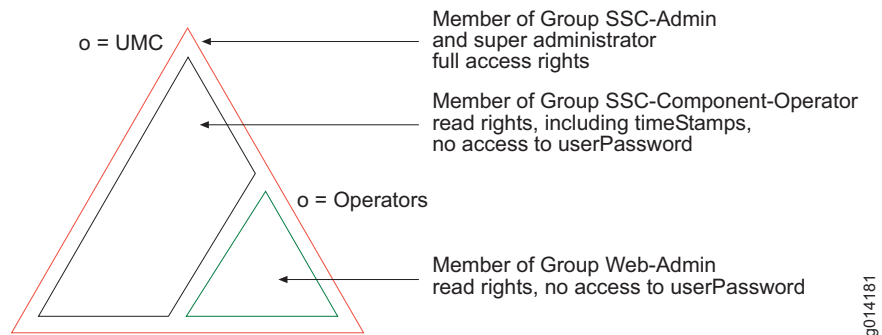
- No access to the subtree  $o = \text{Operators}, o = \text{umc}$
- Read access to the remaining directory tree, including the operational attributes `creationTimeStamp` and `modifyTimeStamp`
- No read and compare rights for any `userPassword` values

Clients binding with the Apache DN or a member of the WebAdmin group do have read and search permissions in the subtree  $o = \text{Operators}, o = \text{umc}$ :

- Read access for all user attributes
- No read and no filter match permissions for the attribute `userPassword`

Members of the WebAdmin group are allowed to administer the SAE through the SAE Web Administration pages.

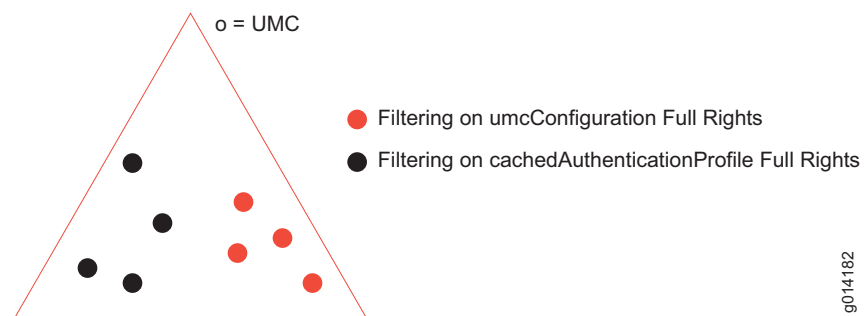
The members of the SSC\_Admin group and the super-administrator have access rights to the entire tree.

**Figure 8: Access Rights for the UMC Tree**

### **Access Controls Against Objects from Type `cachedAuthenticationProfile` and `UmcConfiguration`**

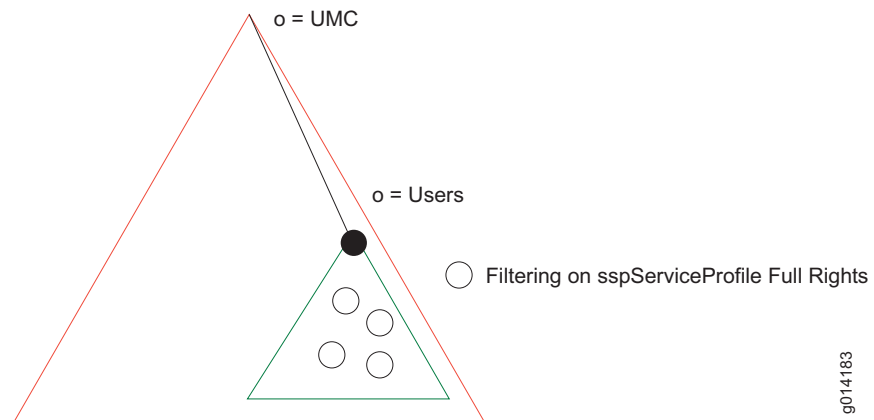
The SAE binds as *cn = ssp, ou = components, o = operators, o = umc* against the directory and needs to have full access rights for the entries from the type object class `cachedAuthenticationProfile` and `umcConfiguration`.

It is easier to implement the cached entries through the targets of the two subtrees (*o = AuthCache, o = umc* and *o = UserProfilesCache, o = umc*).

**Figure 9: Access Rights Against `cachedAuthenticationProfile` and `umcConfiguration` Objects**

### **Access Controls Against `sspServiceProfile`**

In addition to the previously discussed access rights, the SAE requires full access against objects from the tree `sspServiceProfile`.

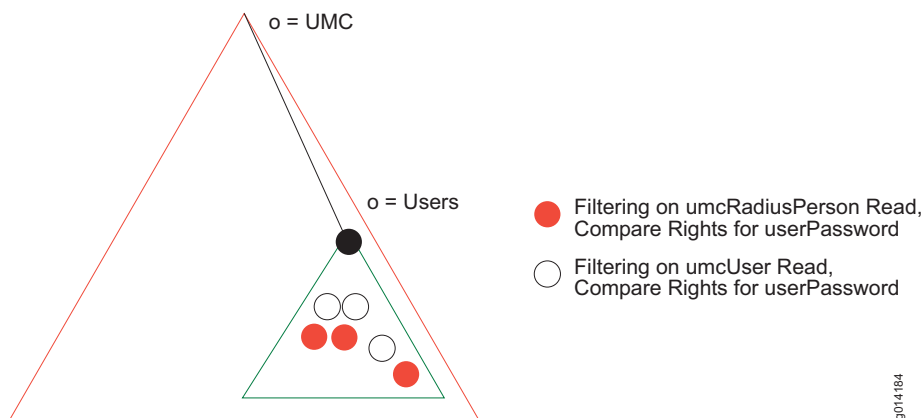
**Figure 10: Access Controls Against sspServiceProfiles in the User Subtree**

g014183

### Access Controls Against umcRadius Person and umcUser

The SAE requires read access to the userPassword attribute for entries from type umcRadiusPerson and umcUser.

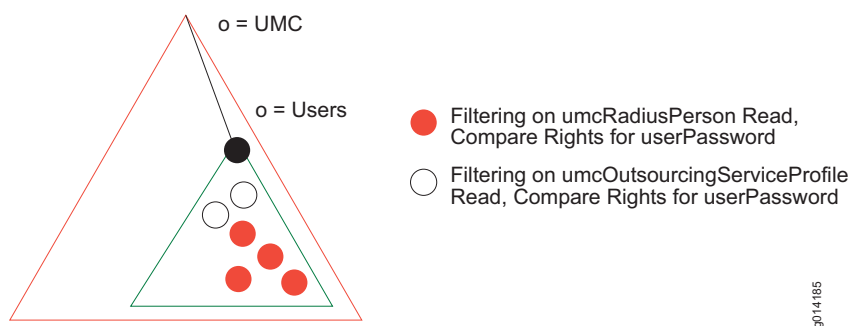
**Figure 11: Access Rights Against umcRadiusPerson and umcUser**



### Access Controls Against RADIUS Profiles

RADIUS requires read access to the userPassword attribute in entries from umcRadiusPerson to authenticate requests of a subscriber, and from umcOutsourcingServiceProfile to determine the tunnel parameter for a Layer 2 Tunneling Protocol (L2TP) outsourcing scenario. The RADIUS server binds with the credentials of *cn = radius*, *ou = components*, *o = operators*, *o = umc*.

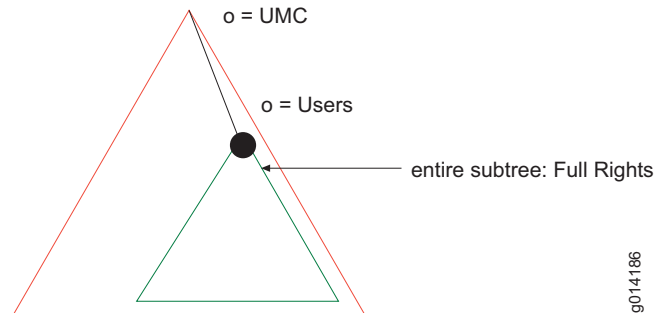
**Figure 12: Access Rights Against umcRadiusPerson and umcOutsourcingServiceProfile Objects**



### Access Controls Against the Policy Subtree

The policy management component uses the credentials of  $cn = pom$ ,  $ou = components$ ,  $o = operators$ ,  $o = umc$  and requires the following set of access rights for the policy subtree. It needs to perform add, delete, and modify operations on all policy and policyFolder objects in the  $o = Policies$ ,  $o = umc$  subtree.

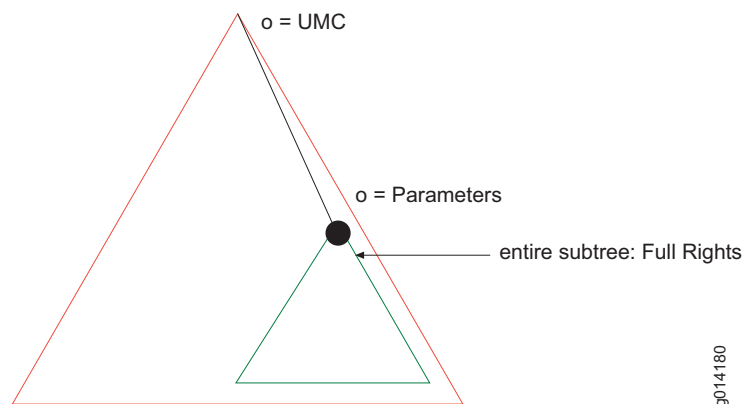
**Figure 13: Policy Rights Against All Objects in the  $o=Policies,o=umc$  Tree**



### Access Controls Against the Parameter Subtree

The policy management component requires the following set of access controls for the parameter subtree. It needs to perform add, delete and modify operations on all objects in the  $o = Parameter$ ,  $o = umc$  subtree.

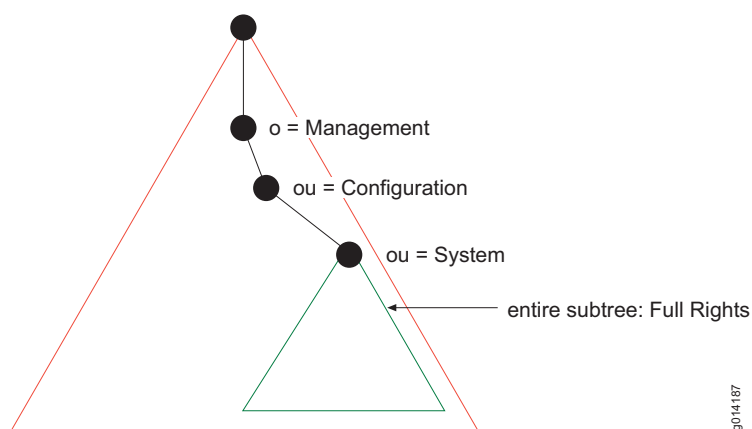
**Figure 14: Access Rights Against All Objects in the Tree  $o=Parameters, o=umc$**



## Access Controls for System Management

The system management component binds as *cn = sysman*, *ou = components*, *o = operators*, *o = umc* and requires full access rights for the subtree *ou = SystemManagement*, *o = Configuration*, *o = Management*, *o = umc*.

**Figure 15: Access Rights for System Management**

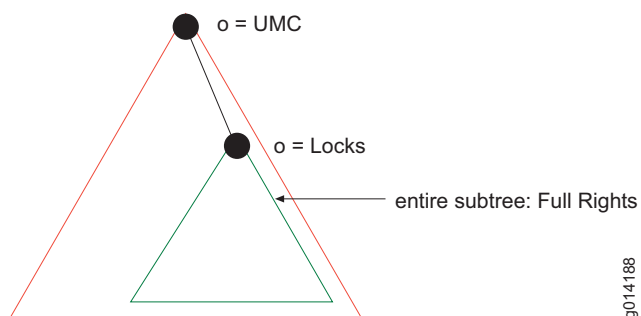


g014187

## Access Controls Against the Lock Subtree

The object state manager component requires full access rights to the subtree *o = Locks*, *o = umc*. This component uses the credentials of *cn = osm*, *ou = components*, *o = operators*, *o = umc* to bind against the directory.

**Figure 16: Access Rights Against the Entire o=Locks,o=umc Subtree**



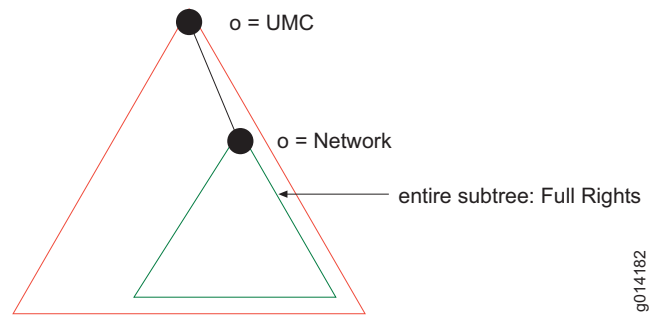
g014188



### Access Controls Against the Network Subtree

The network operator is allowed to administer only objects within the subtree  $o = \text{Network}$ ,  $o = \text{umc}$  and bind against the directory using the credentials of  $cn = \text{network-operator}$ ,  $o = \text{operators}$ ,  $o = \text{umc}$ .

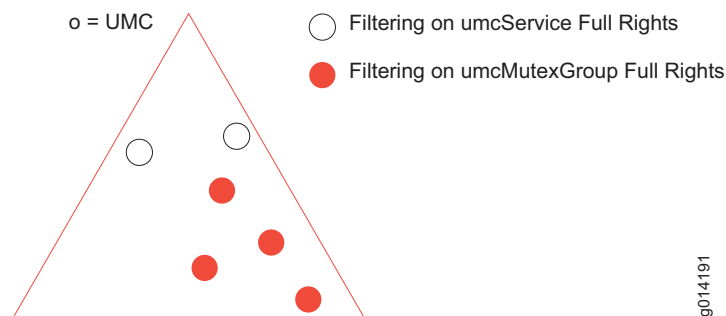
**Figure 17: Access Rights Against the Entire  $o = \text{Network}, o = \text{umc}$  Subtree**



### Access Controls Against Services and Mutex Group Objects

The service operator requires full access rights for umcService objects, as well as for umcMutexGroup objects. These objects are subordinates of the entries  $o = \text{Services}$ ,  $o = \text{umc}$  and  $o = \text{Scopes}$ ,  $o = \text{umc}$ . The service-operator binds with the DN  $cn = \text{service-operator}$ ,  $o = \text{operators}$ ,  $o = \text{umc}$  against the directory.

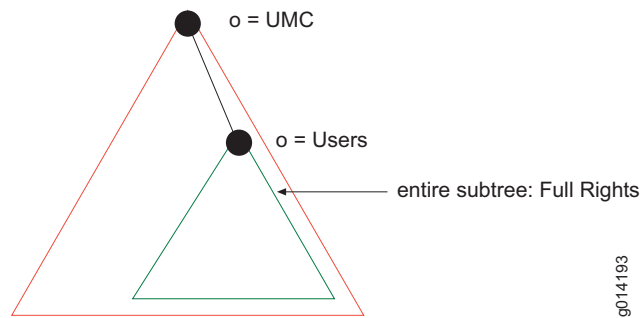
**Figure 18: Access Rights Against umcService and umcMutexGroup Objects**



### Access Controls Against the User Subtree

Subscriber operators are responsible for the entire *o = users*, *o = umc* subtree and require full access rights. The subscriber operator uses the credentials of the entry *cn = subscriber-operator*, *o = operators*, *o = umc*.

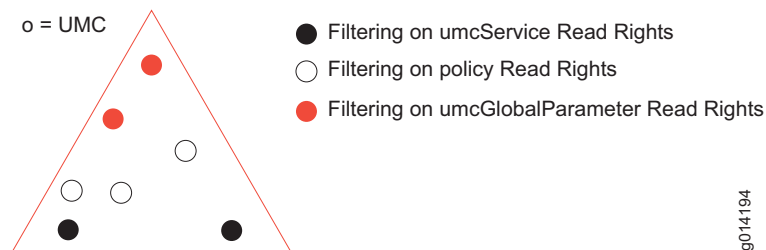
**Figure 19: Access Rights Against the Entire *o=users*, *o=umc* Subtree**



### Access Controls Against Service, Policy, and Global Parameter Objects

All enterprise managers require read and search rights against objects from the type *umcService*, *policy*, and *umcGlobalParameter*. Those managers bind with their credentials against the directory.

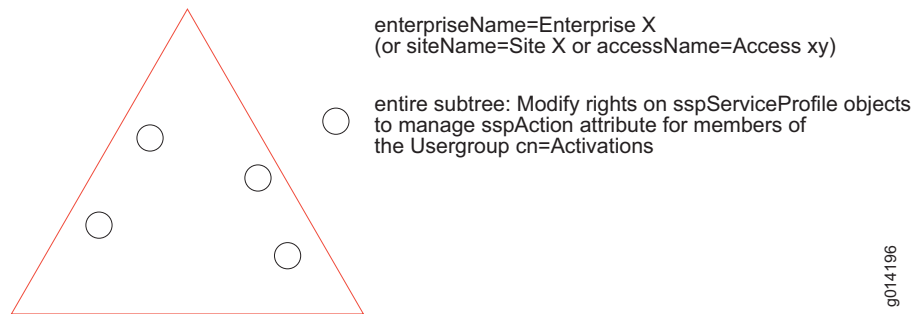
**Figure 20: Access Rights Against *umcService*, *Policy*, and *umcGlobalParameter* Objects**



### Activation Access Rights

Operators who are members of the user group `cn = Activations` need to be able to change the attribute `sspAction` to activate or deactivate SSP services in an enterprise, site, or access scope. Figure 21 shows these modify rights.

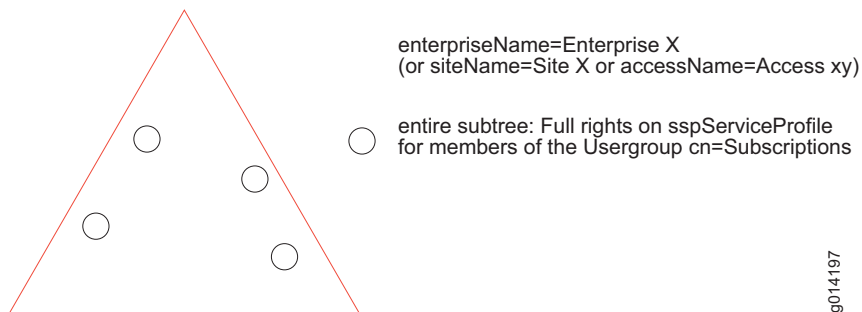
**Figure 21: Modify Rights for Activation Managers**



### Subscription Access Rights

Subscription operators are members of the user group `cn = Subscriptions` and are able to subscribe and unsubscribe to and from SSP services in their specific scope (that is, enterprise, site, or access). This is the creation and deletion of objects from the type `sspServiceProfile`. As a result, subscription operators require full access rights to the objects shown in Figure 22.

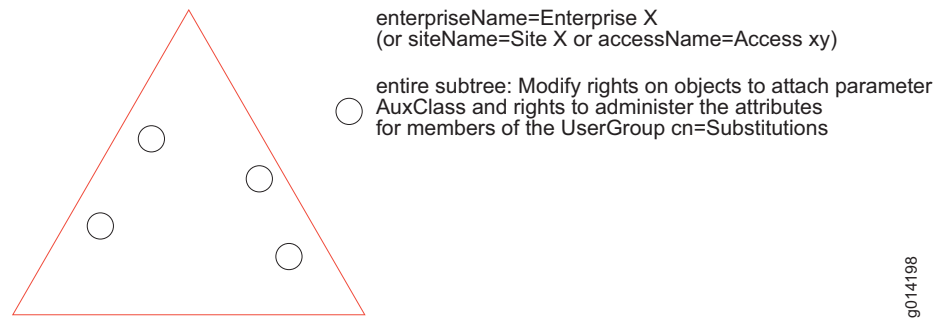
**Figure 22: Access Rights for Subscription Managers**



## Substitution Access Rights

Members of the substitutions user group get the required access rights that grant to attached auxiliary object classes, to objects and modify the attribute type belonging to the auxiliaryclass parameterAuxClass.

**Figure 23: Access Rights for Substitution Managers**

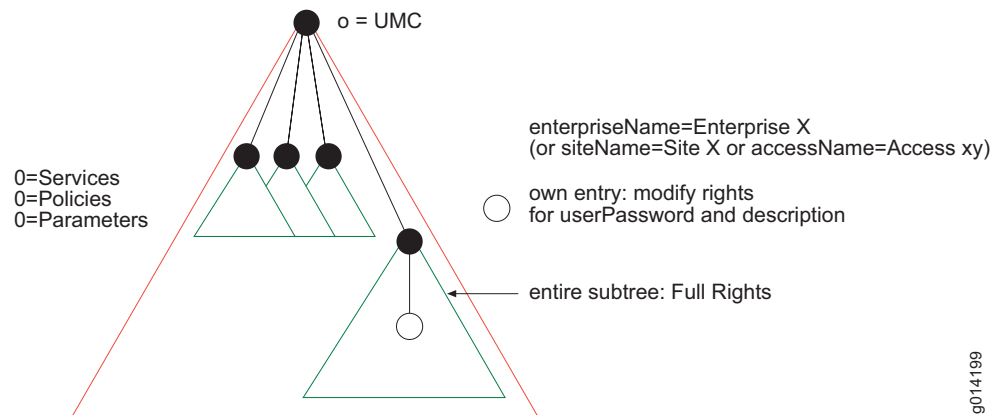


## Common Access Rights for All Managers

All enterprise managers (that is, members of the previously mentioned user groups) have the following common rights:

- Read access to the service subtree (*o = services, o = umc*)
- Read access to the policy subtree (*o = policies, o = umc*)
- Read access to the global parameter subtree (*o = parameters, o = umc*)
- Read access to the scope of the manager; that is, enterprise, site, or access read access
- Modify rights to change the user password and description value of its entry

**Figure 24: Access Rights for All Managers**



## Directory-Specific Access Control Implementation

---

The supported directories (for example, DirX, and Sun ONE) have complex mechanisms for controlling access, depending on the user bound to the directory.

DirX stores the access control lists in subentries that conform to the X.500 standard. You create the access control subentries by using the DirX client `dirxcp`. These access control subentries are replicated in a shadowing scenario.

Sun ONE stores the access control lists in the directory. Sun ONE extends the standard object class `top` by the optional attribute `aci`, which is used to store the access control lists. This means that the access control information (ACI) can be added through LDAP. The `aci` values are replicated to the slave directory.

### DirX Directory Server

DirX access control information is stored in subentries that are from the type *subentry* and *acceSDXontrolSubentry*. These subentries include the information about the target (that is, what is controlled), precedence (that is, higher precedence overwrites lower precedence), and the access control information (that is, prescriptive ACI) that includes the user class (that is, who is affected by the control parameters) and the permissions on entry and attribute level.

Access control subentries can contain many prescriptive ACIs with a list of one or more items to be protected, such as entries and sets of operation or user attributes.

The `UMCdirxa` package includes a TCL file, called *acldefs.tcl*, which defines the following variables for the permissions:

- DAER—Deny read access on entry level
- AER—Grant read access on entry level
- AEM—Grant full access on entry level
- AEME—Grant modify access on entry level
- DAAR—Deny read access on attribute level
- AAR—Grant read rights on attribute level
- AAM—Grant modify rights on attribute level

The `UMCdirxa` package includes the file *access.cp*, which sets the access controls for the SRC software.

Figure 25 shows a TCL script with an explanation of the various parts.

**Figure 25: Creation of an Access Control Subentry Example in DirX**

```
create /o=UMC/CN=SSP-AccessControl-Subentry \ 1
{OCL=SUBE;ACS} \
{SS={SF={OR={ITEM=authProfile}; \ 2
{ITEM=umcConf} } }}\
3 PACI={ID=SSP: Full rights on Cached Profiles and Configuration;
PR=254, 4
5 AL={BL={L=SIMPLE}},
UF={UC={N={DN={/o=UMC/o=Operators/ou=Components/cn=ssp}}},
7 UP={PI={E=TRUE}, GAD=$AEM}; 8 6
{PI={AUATV=TRUE}, GAD=$AAM} } }
9 10
```

1. DN of subentry
2. Target (entire area)
3. (one or more) Identifier(s) of Prescriptive ACI
4. Precedence [0-255]
5. Authentication-level simple-bind
6. User-class: SSP component
7. First protected items (all entries)
8. Grant and denials for all entries: Full Rights
9. Second protected items (all user-attributes)
10. Grant and denials for all user-attributes: Modify Rights

g014955

## Sun ONE Directory Server

Access control information is stored in the aci attribute of each directory entry. Because the access control information is stored in the directory, it can be managed by means of LDIF files.

ACIs take the following form:

```
aci: (<target>) (version 3.0;aci "<name>"; <permission> <bind rule> ;)
```

where

<target> defines the object, attribute, or filter that you are using to define what resource to control access to. The target can be a distinguished name, one or more attributes, and/or a single LDAP filter.

version 3.0 is a required string that identifies the ACI version.

aci "<name>" is a name for the ACI. <name> can be any string that identifies the ACI. The ACI name is required.

<permission> defines the actual access rights and whether they are to be allowed or denied.

<bind rules> identify the circumstances under which the directory login must occur for the ACI to take effect.

The UMCiDSa package includes the LDIF file *access.ldif*, which implements the SDX access control scheme.

Figure 26 shows the LDIF file for implementing the same kind of access level as previously depicted with a Sun ONE directory.

**Figure 26: Creation of Access Control List Example in Sun ONE**

```
dn: o=UMC
changetype: modify
add: aci
aci: (target="ldap:///o=UMC") (targetattr="*" 1)
    (targetfilter="( | (objectClass=cachedAuthenticationProfile) 2
    (objectClass=umcConfiguration) )" )
    (version 3.0; acl "SSP: enable admin of cahced profiles and configuration"; 3
4 allow (all) userdn = "ldap:///cn=ssp,ou=Components,o=Operators,o=UMC"
5 and (authmethod = "Simple");)
```

1. All user-attributes implicitly included
2. Target (entire area)
3. (one or more) Identifier(s) of Prescriptive ACI
4. Grant write access, which includes the rights auth, compare, read, and search
5. User-class: SSP component

g014957

