Using LDAP with Java CAPS

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

Part No: 820–3399–10
June 2008

080618@20490

Contents

◆ ◆ ◆ CHAPTER 1

Using LDAP with Java CAPS

The topics listed here provide information about how to use the Lightweight Directory Access Protocol (LDAP) with Sun Java™ Composite Application Platform Suite (Java CAPS).

If you have any questions or problems, see the Java CAPS web site at `http://goldstar.stc.com/support`.

LDAP Overview

The Lightweight Directory Access Protocol (LDAP) is a standard that enables clients to query and update data in directory services.

An LDAP directory includes a series of *entries*. An entry is a collection of *attributes*, plus a *Distinguished Name* that uniquely identifies the entry.

In the following example, the first line specifies the DN. The succeeding lines specify the attributes.

```
dn: cn=all, ou=Roles, dc=company, dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: all
ou: Roles
```

The components of a DN are ordered hierarchically from most specific to least specific. Thus, the last component in the DN identifies the root entry of the directory.

Each attribute contains a type and one or more values. For example, the attribute `ou: Roles` has a type of `ou` (organizational unit) and a value of `Roles`. An *object class* is an attribute that specifies the required and optional attributes for an entry. You can find definitions of many object classes in RFC 2256.

The preceding example is represented in the LDAP Data Interchange Format (LDIF). The entry could also be represented graphically.

When searching an LDAP directory, you use a *search filter* to specify the search criteria. You can use an asterisk as a wildcard character. For example:

```
(cn=John S*)
```

Using an LDAP Server for Repository User Management

You can configure the Java CAPS Repository to use an LDAP server for user management.

When a user attempts to log into the Repository, the user name and password are checked against the user name and password that are stored in the LDAP server. In addition, the list of roles for the user is retrieved from the server to authorize the user's access to various objects in the Repository.

The following LDAP servers are supported:

- Sun Java™ System Directory Server versions 5.1, 5.2, and 6.*x*
- Microsoft's Active Directory (the version delivered with Windows Server 2003)
- OpenLDAP Directory Server 2.*x*

First, you must configure the LDAP server. See the appropriate section:

- "Configuring the Sun Java™ System Directory Server" on page 7
- "Configuring the Active Directory Service" on page 8
- "Configuring the OpenLDAP Directory Server" on page 8

Then, you configure the Repository so that it can locate the LDAP server and find the appropriate information (such as the portion of the directory that contains users). See "Configuring the Repository" on page 10.

If you want to encrypt communications between the Repository and the LDAP server, see "SSL Support" on page 13.

Managing Java CAPS Users provides basic information about Repository user management.

Configuring the Sun Java™ System Directory Server

Sun Java System Directory Server versions 5.1 and 5.2 include the following main components:

- Directory Server
- Administration Server
- Directory Server console

The Directory Server console enables you to perform most administrative tasks. The console contains four top-level tabs: Tasks, Configuration, Directory, and Status. The Directory tab displays the directory entries as a tree. You can browse, display, and edit all of the entries and attributes from this tab.

You can also perform administrative tasks manually by editing configuration files or by using command-line utilities.

Sun Java System Directory Server version 6.*x* provides the following ways for you to manage the entries in a directory:

- Directory Service Control Center (DSCC)
- Directory Editor
- `ldapmodify` and `ldapdelete` command-line utilities

DSCC is integrated into the Sun Java™ Web Console. DSCC contains five top-level tabs: Common Tasks, Directory Servers, Proxy Servers, Server Groups, and Settings.

If you click the Directory Servers tab, the name of a server, and the Entry Management tab, then you reach the page that enables you to browse, add, and modify entries. The Directory Information Tree (DIT) appears on the left.

You can also use the Common Tasks tab to create a new entry or browse data.

Note – For detailed information about how to perform the following steps, see the documentation provided with Sun Java System Directory Server.

▼ To Configure the Sun Java System Directory Server

1. **Create the `admin` user and the `Administrator` user under the `People` directory.**
2. **Create the roles `all`, `administration`, and `management` under the top node.**
3. **Assign the roles that you created to the `admin` user and the `Administrator` user.**
4. **Go to "Configuring the Repository" on page 10.**

Configuring the Active Directory Service

Active Directory is a key part of Windows 2003. It provides a wide variety of manageability, security, and interoperability features. The main administration tool is a snap-in called Active Directory Users and Computers.

Active Directory does not support the concept of roles. Therefore, you must simulate the Java CAPS roles in Active Directory using the concept of *groups*.

Rather than creating the groups within the `Users` directory, you create the groups in a new organizational unit called `CAPSRoles`.

Note – For detailed information about how to perform the following steps, see the documentation provided with Active Directory.

▼ To Configure the Active Directory Service

1. **Start the Active Directory Users and Computers administration tool.**
2. **Right-click the root node and select New > Organizational Unit.**

   The New Object - Organization Unit dialog box appears.
3. **In the Name field, enter a value (for example, `CAPSRoles`).**
4. **Click OK.**
5. **Under the organizational unit, create the following groups: `all`, `administration`, and `management`. To create a group, you right-click the organizational unit and select New > Group. Use the default values for Group scope and Group type.**

   After you add the groups, they appear under the organizational unit.
6. **Add the `admin` user and the `Administrator` user as members of all the groups that you created by double-clicking each group and selecting `admin` and `Administrator` from the dialog box.**
7. **Go to "Configuring the Repository" on page 10.**

Configuring the OpenLDAP Directory Server

The OpenLDAP Project provides an open source implementation of the LDAP protocol. The LDAP server runs as a stand-alone daemon called `slapd`. The main configuration file is called `slapd.conf`. This file contains global, backend-specific, and database-specific information. You can use various approaches to add entries to the database, such as using the `slapadd` program. To search the database, use the `ldapsearch` program.

For more information, see `http://www.openldap.org`.

Note – For detailed information about how to perform the following steps, see the documentation provided with OpenLDAP Directory Server.

▼ To Configure the OpenLDAP Directory Server

1. **Create the `admin` user and the `Administrator` user under the node where the users are located.**

2. **If you do not have a node for roles in your schema, then create a node for the Java CAPS-specific roles that you will create in the following step. For example:**

```
dn: ou=CAPSRoles, dc=sun, dc=com
objectClass: top
objectClass: organizationalUnit
ou: CAPSRoles
```

3. **Create the roles `all`, `administration`, and `management` under the node where the roles are located. Add the `admin` user and the `Administrator` user as unique members of each role. For example:**

```
dn: cn=all, ou=CAPSRoles, dc=sun, dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: all
ou: CAPSRoles
uniqueMember: uid=admin, ou=People, dc=sun, dc=com
uniqueMember: uid=Administrator, ou=People, dc=sun, dc=com

dn: cn=administration, ou=CAPSRoles, dc=sun, dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: administration
ou: CAPSRoles
uniqueMember: uid=admin, ou=People, dc=sun, dc=com
uniqueMember: uid=Administrator, ou=People, dc=sun, dc=com

dn: cn=management, ou=CAPSRoles, dc=sun, dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: management
ou: CAPSRoles
uniqueMember: uid=admin, ou=People, dc=sun, dc=com
uniqueMember: uid=Administrator, ou=People, dc=sun, dc=com
```

4 **Add other users to one or more roles, as necessary. For example:**

```
dn: cn=all, ou=CAPSRoles, dc=sun, dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: all
ou: CAPSRoles
uniqueMember: uid=admin, ou=People, dc=sun, dc=com
uniqueMember: uid=Administrator, ou=People, dc=sun, dc=com
uniqueMember: uid=userA, ou=People, dc=sun, dc=com
uniqueMember: uid=userB, ou=People, dc=sun, dc=com

dn: cn=administration, ou=CAPSRoles, dc=sun, dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: administration
ou: CAPSRoles
uniqueMember: uid=admin, ou=People, dc=sun, dc=com
uniqueMember: uid=Administrator, ou=People, dc=sun, dc=com
uniqueMember: uid=userB, ou=People, dc=sun, dc=com

dn: cn=management, ou=CAPSRoles, dc=sun, dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: management
ou: CAPSRoles
uniqueMember: uid=admin, ou=People, dc=sun, dc=com
uniqueMember: uid=Administrator, ou=People, dc=sun, dc=com
```

5 **Go to "Configuring the Repository" on page 10.**

Configuring the Repository

To use an LDAP server for Repository user management, you must add a `<Realm>` element to the Repository's `server.xml` file, which is located in the `JavaCAPS-install-dir/repository/repository/server/conf` directory.

The `server.xml` file contains a default `<Realm>` element that specifies a flat file implementation of the user database. The flat file implementation uses the `tomcat-users.xml` file in the `JavaCAPS-install-dir/repository/repository/data/files` directory.

The following table describes the attributes used by the LDAP versions of the `<Realm>` element. For a detailed description of all the possible attributes, see the Tomcat documentation for the `org.apache.catalina.realm.JNDIRealm` class.

Attribute	Description
className	Always use the following value: `org.apache.catalina.realm.JNDIRealm`
connectionURL	Identifies the location of the LDAP server. Includes the LDAP server name and the port that the LDAP server listens on for requests.
roleBase	The base entry for the role search. If this attribute is not specified, then the search base is the top-level directory context.
roleName	The attribute in a role entry containing the name of the role.
roleSearch	The LDAP search filter for selecting role entries. It optionally includes pattern replacements `{0}` for the Distinguished Name and/or `{1}` for the user name of the authenticated user. In certain cases of an authenticated user (for example, Administrator), option `{0}` should be selected.
roleSubtree	By default, the Roles portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`.
userBase	The entry that is the base of the subtree containing users. If this attribute is not specified, then the search base is the top-level context.
userPattern	A pattern for the Distinguished Name (DN) of the user's directory entry, following the syntax supported by the `java.text.MessageFormat` class with `{0}` indicating where the actual user name should be inserted.
userRoleName	The name of an attribute in the user's directory entry containing zero or more values for the names of roles assigned to this user. In addition, you can use the `roleName` attribute to specify the name of an attribute to be retrieved from individual role entries found by searching the directory. If `userRoleName` is not specified, then all roles for a user derive from the role search.
userRoleNamePattern	A pattern for the Distinguished Name (DN) of the role's directory entry, following the syntax supported by the `java.text.MessageFormat` class with `{0}` indicating the actual role name. This pattern is used to parse the DN to get the actual role name for authorization purposes in Java CAPS, where the actual user name should be inserted.
userSearch	The LDAP search filter to use for selecting the user entry after substituting the user name in `{0}`.
userSubtree	By default, the Users portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`.

▼ To Configure the Repository

1 **Open the `server.xml` file in the `JavaCAPS-install-dir/repository/repository/server/conf` directory.**

2 **Remove or comment out the default** `<Realm>` **element.**

3 **If you are using Sun Java System Directory Server, add the following** `<Realm>` **element inside the** `<Engine>` **tag. Change the default values as necessary. The preceding table describes the attributes.**

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
    connectionURL="ldap://localhost:489"
    userBase="cn=People,dc=sun,dc=com"
    userSearch="(uid={0})"
    userSubtree="true"
    userRoleName="nsroledn"
    userRoleNamePattern="cn={0},dc=sun,dc=com"
    roleSubtree="true"
/>
```

4 **If you are using Active Directory, add the following** `<Realm>` **element inside the** `<Engine>` **tag. Change the default values as necessary. The preceding table describes the attributes.**

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
    connectionURL="ldap://localhost:389"
    userBase="cn=Users,dc=sun,dc=com"
    userSearch="(cn={0})"
    userSubtree="true"
    roleBase="ou=CAPSRoles,dc=sun,dc=com"
    roleName="cn"
    roleSearch="(member={0})"
    roleSubtree="true"
/>
```

5 **If you are using OpenLDAP Directory Server, add the following** `<Realm>` **element inside the** `<Engine>` **tag. Change the default values as necessary. The preceding table describes the attributes.**

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
    connectionURL="ldap://localhost:389"
    userBase="ou=People,dc=sun,dc=com"
    userSearch="(uid={0})"
    userSubtree="true"
    roleBase="ou=CAPSRoles,dc=sun,dc=com"
    roleName="cn"
    roleSearch="(uniquemember={0})"
    roleSubtree="true"
/>
```

6 **If your LDAP server is not configured for anonymous read access, add the `connectionName` and `connectionPassword` attributes to the `<Realm>` element. Set the first attribute to the DN of the `Administrator` user. Set the second attribute to the user's encrypted password. Refer to the following examples.**

Sun Java System Directory Server:

```
connectionName="cn=Directory Manager"
connectionPassword="E451KDVb0OPcH+GN46OZcg=="
```

Active Directory:

```
connectionName="Administrator@sun.com"
connectionPassword="geEiVIbtO+DcH+GN46OZcg=="
```

OpenLDAP Directory Server:

```
connectionName="cn=Manager,dc=sun,dc=com"
connectionPassword="l/ZRt1cfNKc="
```

To encrypt the password, use the `encrypt` utility in the `JavaCAPS-install-dir/repository/repository/util` directory. The file extension of the utility depends on your platform. This utility takes the unencrypted password as an argument. For example:

```
C:\JavaCAPS6\repository\repository\util>encrypt mypwd
LCUApSkYpuE
```

7 **Save and close the `server.xml` file.**

8 **Start the LDAP server.**

9 **Shut down and restart the Repository.**

SSL Support

By default, communications between the Repository and the LDAP server are unencrypted.

To encrypt communications between the Repository and the LDAP server, make the following additions and modifications to the procedures described earlier in this topic.

Configuring SSL on the LDAP Server

Ensure that the LDAP server is configured to use the Secure Sockets Layer (SSL). For detailed instructions, see the documentation provided with the LDAP server.

In preparation for the next step, export the LDAP server's certificate to a file.

Importing the LDAP Server's Certificate

You must add the LDAP server's certificate to the Repository's list of trusted certificates. The list is located in a file called `cacerts`.

In the following procedure, you use the `keytool` program. This program is included with the Java SDK.

▼ To Import the LDAP Server's Certificate

1. **Navigate to the `JDK-install-dir/jre/bin` directory.**

   Use the JDK that was specified during the installation of the Repository.

2. **Run the following command:**

   ```
   keytool -import -trustcacerts -alias alias -file certificate_filename
   -keystore cacerts_filename
   ```

   For the `-alias` option, you can assign any value.

   For the `-file` option, specify the fully qualified name of the LDAP server's certificate. For example:

   ```
   C:\mycertificate.cer
   ```

   For the `-keystore` option, specify the fully qualified name of the `cacerts` file. The `cacerts` file is located in the `JDK-install-dir/jre/lib/security` directory. For example:

   ```
   C:\Java\jdk1.6.0_06\jre\lib\security\cacerts
   ```

3. **When prompted, enter the keystore password. The default password is `changeit`.**

4. **When prompted to trust this certificate, enter `yes`.**

   The following message appears:

   ```
   Certificate was added to keystore
   ```

Modifying the LDAP Server URL

In the `<Realm>` element of the `server.xml` file, modify the URL of the LDAP server as follows:

- Set the protocol to `ldaps`.
- Set the port number to the port number that the LDAP server listens on for SSL requests. Typically, this number is 636.

For example:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
    connectionURL="ldaps://myldapserver:636"
    ...
```

Using an LDAP Server for Sun JMS IQ Manager User Management

You can configure a Sun JMS IQ Manager to use an LDAP server for user management.

A *realm* is a collection of users, groups, and roles that are used in enforcing security policies. The JMS IQ Manager supports multiple LDAP realms running at the same time.

When you perform the following steps, access to the JMS IQ Manager is granted only when the connection has a valid user name and password.

The following LDAP servers are supported:

- Sun Java System Directory Server versions 5.1, 5.2, and 6.*x*
- Microsoft's Active Directory (the version delivered with Windows Server 2003)
- OpenLDAP Directory Server 2.*x*

Managing Java CAPS Users provides basic information about Sun JMS IQ Manager user management.

Configuring the LDAP Server

In the following procedure, you create users and roles in the LDAP server.

▼ To Configure the LDAP server

1. **Create one or more JMS IQ Manager users.**
2. **Create one or more of the following roles:**

Role	Description
application	Enables clients to access the JMS IQ Manager.
asadmin	Enables use of the JMS control utility (`stcmsctrlutil`) or Enterprise Manager, and enables clients to access the JMS IQ Manager.

3. **Assign the roles to your users as needed.**

Configuring the Sun JMS IQ Manager

You must configure the JMS IQ Manager so that it can locate the LDAP server and find the appropriate information.

You can enable more than one LDAP server. In addition, you can specify the default realm.

▼ To Configure the Sun JMS IQ Manager

1. **If the application server is not running, then start the application server.**
2. **Log in to the Configuration Agent. The format of the URL is** `http://hostname:port-number/configagent`**. Set the hostname to the TCP/IP host name of the computer where the application server is installed. Set the port number to the administration port number of the application server. For example:**

   ```
   http://localhost:4848/configagent
   ```

3. **In the left pane, click the JMS IQ Manager node (for example,** `IQ_Manager_18007`**).**
4. **Click the Access Control tab.**
5. **Ensure that the check box to the right of the Require Authentication label is selected.**
6. **If you want to enable Sun Java System Directory Server, then select the check box to the right of the Enable Sun Java System Directory Server label and click Show Properties.**

   The following table describes the properties that appear. The default values are intended to match the standard schema of Sun Java System Directory Server. Review the default value for each property. If necessary, modify the default value.

Property	Description
Naming Provider URL	The URL of the Java Naming and Directory Interface (JNDI) service provider. The default value is `ldap://IP_address:589`.
Naming Initial Factory	The fully qualified name of the factory class that creates the initial context. The initial context is the starting point for JNDI naming operations. The default value is `com.sun.jndi.ldap.LdapCtxFactory`.

Property	Description
Naming Security Authentication	The security level to use in JNDI naming operations. The default value is `simple`.
Naming Security Principal	The security principal used for connecting to the LDAP server.
Naming Security Credentials	The password of the naming security principal. The default value is `STC`. The value is encrypted when you save and then view it again.
Group DN Attribute Name in Group	The name of the Distinguished Name attribute in group entries. The default value is `entrydn`.
Group Name Field in Group DN	The name of the group name field in group Distinguished Names. The default value is `cn`.
Groups of User Filter Under Groups Parent DN	The LDAP search filter used to retrieve all of a user's groups. This property follows the syntax supported by the `java.text.MessageFormat` class with `{1}` indicating where the user's Distinguished Name should be inserted. The default value is `uniquemember={1}`.
Groups Parent DN	The parent Distinguished Name of the group entries. In other words, this property specifies the root entry of the groups portion of the LDAP directory.
Role Name Attribute Name in User	The name of the role name attribute in user entries. The default value is `nsroledn`.
Role Name Field in Role DN	The name of the role name field in role Distinguished Names. The default value is `cn`.
Roles Parent DN	The parent Distinguished Name of the role entries. In other words, this property specifies the root entry of the roles portion of the LDAP directory.
Search Groups Sub Tree	By default, the groups portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.

Property	Description
Search Roles Sub Tree	By default, the roles portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.
Search Users Sub Tree	By default, the users portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.
User DN Attribute Name in User	The name of the Distinguished Name attribute in user entries. The default value is `entrydn`.
User ID Attribute Name in User	The name of the user ID attribute in user entries. The default value is `uid`.
Users Parent DN	The parent Distinguished Name of the user entries. In other words, this property specifies the root entry of the users portion of the LDAP directory.

7 If you want to enable Active Directory, then select the check box to the right of the Enable Microsoft Active Directory Server label and click Show Properties.

The following table describes the properties that appear. The default values are intended to match the standard schema of Active Directory. Review the default value for each property. If necessary, modify the default value.

Property	Description
Naming Provider URL	The URL of the Java Naming and Directory Interface (JNDI) service provider. The default value is `ldap://IP_address:389`.
Naming Initial Factory	The fully qualified name of the factory class that creates the initial context. The initial context is the starting point for JNDI naming operations. The default value is `com.sun.jndi.ldap.LdapCtxFactory`.
Naming Security Authentication	The security level to use in JNDI naming operations. The default value is `simple`.

Property	Description
Naming Security Principal	The security principal used for connecting to the LDAP server.
Naming Security Credentials	The password of the naming security principal. The default value is `STC`. The value is encrypted when you save and then view it again.
Users Parent DN	The parent Distinguished Name of the user entries. In other words, this property specifies the root entry of the users portion of the LDAP directory.
User DN Attribute Name in User	The name of the Distinguished Name attribute in user entries. The default value is `distinguishedName`.
User ID Attribute Name in User	The name of the user ID (that is, the login ID) attribute in user entries. The default value is `sAMAccountName`.
Roles Parent DN	The parent Distinguished Name of the role entries. In other words, this property specifies the root entry of the roles portion of the LDAP directory.
Role DN Attribute Name in Role	The name of the Distinguished Name attribute in role entries. The default value is `cn`.
Roles of User Filter Under Roles Parent DN	The LDAP search filter used to retrieve all of a user's roles. This property follows the syntax supported by the `java.text.MessageFormat` class with `{1}` indicating where the user's Distinguished Name should be inserted. The default value is `(&(member={1})(objectclass=group))`.
Groups Parent DN	The parent Distinguished Name of the group entries. In other words, this property specifies the root entry of the groups portion of the LDAP directory.
Group DN Attribute Name in Group	The name of the Distinguished Name attribute in group entries. The default value is `distinguishedName`.
Group Name Field in Group DN	The name of the group name field in group Distinguished Names. The default value is `cn`.

Property	Description
Groups of User Filter Under Groups Parent DN	The LDAP search filter used to retrieve all of a user's groups. This property follows the syntax supported by the `java.text.MessageFormat` class with `{1}` indicating where the user's Distinguished Name should be inserted. The default value is `(&(member={1})(objectclass=group))`.
Search Groups Sub Tree	By default, the groups portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.
Search Users Sub Tree	By default, the users portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.
Search Roles Sub Tree	By default, the roles portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.

8 If you want to enable OpenLDAP Directory Server, then select the check box to the right of the Enable Generic LDAP Server label and click Show Properties.

The following table describes the properties that appear. Review the default value for each property. If necessary, modify the default value.

Property	Description
Naming Provider URL	The URL of the Java Naming and Directory Interface (JNDI) service provider. The default value is `ldap://IP_address:489`.
Naming Initial Factory	The fully qualified name of the factory class that creates the initial context. The initial context is the starting point for JNDI naming operations. The default value is `com.sun.jndi.ldap.LdapCtxFactory`.

Property	Description
Naming Security Authentication	The security level to use in JNDI naming operations. The default value is `simple`.
Users Parent DN	The parent Distinguished Name of the user entries. In other words, this property specifies the root entry of the users portion of the LDAP directory.
User ID Attribute Name in User	The name of the user ID attribute in user entries. The default value is `uid`.
Roles Parent DN	The parent Distinguished Name of the role entries. In other words, this property specifies the root entry of the roles portion of the LDAP directory.
Role Name Attribute Name in Role	The name of the role name attribute in user entries. The default value is `cn`.
Roles of User Filter Under Roles Parent DN	The LDAP search filter used to retrieve all of a user's roles. This property follows the syntax supported by the `java.text.MessageFormat` class with `{1}` indicating where the user's Distinguished Name should be inserted. The default value is `uniquemember={1}`.
Group Name Field in Group DN	The name of the group name field in group Distinguished Names. The default value is `cn`.
Groups Parent DN	The parent Distinguished Name of the group entries. In other words, this property specifies the root entry of the groups portion of the LDAP directory.
Groups of User Filter Under Groups Parent DN	The LDAP search filter used to retrieve all of a user's groups. This property follows the syntax supported by the `java.text.MessageFormat` class with `{1}` indicating where the user's Distinguished Name should be inserted. The default value is `uniquemember={1}`.
Search Groups Sub Tree	By default, the groups portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.

Property	Description
Search Users Sub Tree	By default, the users portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.
Search Roles Sub Tree	By default, the roles portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to `true`. The default value is `false`.

9 **If you want to change the default realm, then select the realm from the Default Realm drop-down list.**

10 **Click Save.**

Using an LDAP Server for Enterprise Manager User Management

You can configure Enterprise Manager to use an LDAP server for user management.

The following LDAP servers are supported:

- Sun Java System Directory Server versions 5.1, 5.2, and 6.*x*
- Microsoft's Active Directory (the version delivered with Windows Server 2003)
- OpenLDAP Directory Server 2.*x*

First, you configure the LDAP server. Then you configure the Enterprise Manager server so that it can locate the LDAP server and find the appropriate information (for example, the portion of the directory that contains users).

Managing Java CAPS Users provides basic information about Enterprise Manager user management.

Configuring the Sun Java System Directory Server

Sun Java System Directory Server versions 5.1 and 5.2 include the following main components:

- Directory Server
- Administration Server
- Directory Server console

The Directory Server console enables you to perform most administrative tasks. The console contains four top-level tabs: Tasks, Configuration, Directory, and Status. The Directory tab displays the directory entries as a tree. You can browse, display, and edit all of the entries and attributes from this tab.

You can also perform administrative tasks manually by editing configuration files or by using command-line utilities.

Sun Java System Directory Server version 6.*x* provides the following ways for you to manage the entries in a directory:

- Directory Service Control Center (DSCC)
- Directory Editor
- `ldapmodify` and `ldapdelete` command-line utilities

DSCC is integrated into the Sun Java™ Web Console. DSCC contains five top-level tabs: Common Tasks, Directory Servers, Proxy Servers, Server Groups, and Settings.

If you click the Directory Servers tab, the name of a server, and the Entry Management tab, then you reach the page that enables you to browse, add, and modify entries. The Directory Information Tree (DIT) appears on the left.

You can also use the Common Tasks tab to create a new entry or browse data.

Note – For detailed information about how to perform the following steps, see the documentation provided with Sun Java System Directory Server.

▼ To Configure the Sun Java System Directory Server

1. **Create the `admin` user and the `Administrator` user under the `People` directory.**
2. **Create the following roles under the top node:**
   - Deployment
   - User Management
   - Read-Only Monitor
   - Controlling Monitor
   - JMS Read-Only Monitor
   - JMS Read-Write Monitor
   - Manager
3. **Assign the roles that you created to the `admin` user and the `Administrator` user.**
4. **Go to "Configuring the Enterprise Manager Server" on page 25.**

Configuring the Active Directory Service

Active Directory is a key part of Windows 2000. It provides a wide variety of manageability, security, and interoperability features. The main administration tool is a snap-in called Active Directory Users and Computers.

Active Directory does not support the concept of roles. Therefore, you must simulate the Enterprise Manager roles in Active Directory using the concept of *groups*.

Note – For detailed information about how to perform the following steps, see the documentation provided with Active Directory.

▼ To Configure the Active Directory Service

1. **Start the Active Directory Users and Computers administration tool.**
2. **Right-click the root node and select New > Organizational Unit.**

   The New Object - Organization Unit dialog box appears.
3. **In the Name field, enter a value (for example,** `EntMgrRoles`**).**
4. **Click OK.**
5. **Under the organizational unit, create the following groups:**

   - Deployment
   - User Management
   - Read-Only Monitor
   - Controlling Monitor
   - JMS Read-Only Monitor
   - JMS Read-Write Monitor
   - Manager

   After you add the groups, they appear under the organizational unit.
6. **Add the** `admin` **user and the** `Administrator` **user as members of all the groups that you created by double-clicking each group and selecting** `admin` **and** `Administrator` **from the dialog box.**
7. **Go to "Configuring the Enterprise Manager Server" on page 25.**

Configuring the OpenLDAP Directory Server

The OpenLDAP Project provides an open source implementation of the LDAP protocol. The LDAP server runs as a stand-alone daemon called `slapd`. The main configuration file is called `slapd.conf`. This file contains global, backend-specific, and database-specific information. You can use various approaches to add entries to the database, such as using the `slapadd` program. To search the database, use the `ldapsearch` program.

For more information, see `http://www.openldap.org`.

Note – For detailed information about how to perform the following steps, see the documentation provided with OpenLDAP Directory Server.

▼ To Configure the OpenLDAP Directory Server

1. **Create the `admin` user and the `Administrator` user under the node where the users are located.**
2. **If you do not have a node for roles in your schema, then create a node for the Enterprise Manager roles that you will create in the following step.**
3. **Create the following roles under the node where the roles are located:**
   - Deployment
   - User Management
   - Read-Only Monitor
   - Controlling Monitor
   - JMS Read-Only Monitor
   - JMS Read-Write Monitor
   - Manager
4. **Add the `admin` user and the `Administrator` user as unique members of each role.**
5. **Add other users to one or more roles, as necessary.**
6. **Go to "Configuring the Enterprise Manager Server" on page 25.**

Configuring the Enterprise Manager Server

Once you have configured the LDAP server, you configure the Enterprise Manager server so that it can locate the LDAP server and find the appropriate information.

You must edit the following Enterprise Manager files: `web.xml` and `ldap.properties`.

▼ To Configure the Enterprise Manager Server

1. **Shut down the server component of Enterprise Manager.**

2. **Open the** `web.xml` **file in the** `JavaCAPS-install-dir/emanager/server/webapps/sentinel/WEB-INF` **directory.**

3. **Locate the following lines:**

```
<param-name>com.stc.emanager.sentinel.authHandler</param-name>
<param-value>com.stc.cas.auth.provider.tomcat.TomcatPasswordHandler</param-value>
```

4. **Change the parameter value to:**

```
<param-value>com.stc.cas.auth.provider.ldap.LDAPHandler</param-value>
```

5. **Save the** `web.xml file`**.**

6. **Open the** `ldap.properties` **file in the** `JavaCAPS-install-dir/emanager/server/webapps/sentinel/WEB-INF/classes` **directory.**

7. **The following table describes all of the properties that appear in the** `ldap.properties` **file. Edit the properties in the section for your LDAP server, and ensure that the properties are not commented out.**

Property	Description
com.stc.sentinel.auth.ldap.serverType	The type of LDAP server.
com.stc.sentinel.auth.ldap.serverUrl	The URL of the LDAP server.
com.stc.sentinel.auth.ldap.searchFilter	The name of the user ID attribute in user entries.
com.stc.sentinel.auth.ldap.searchBase	The root entry of the portion of the LDAP directory where Enterprise Manager will search for users.
com.stc.sentinel.auth.ldap.searchScope	This property is not currently used.
com.stc.sentinel.auth.ldap.bindDN	The security principal used for connecting to the LDAP server.
com.stc.sentinel.auth.ldap.bindPassword	The password of the security principal.

Property	Description
com.stc.sentinel.auth.ldap.referral	The LDAP referral policy. The default value is `follow`, which indicates that LDAP referrals will be automatically followed. Note that referrals must be enabled in the LDAP server. The other valid values are `throw` (for referral exceptions) and `ignore`. This property is optional. This property appears only in the Active Directory and OpenLDAP sets of properties.
com.stc.sentinel.auth.ldap.roleAttribute	The name of the role name attribute in user entries.
com.stc.sentinel.auth.ldap.roleBaseDN	The root entry of the portion of the LDAP directory where Enterprise Manager will search for roles. This property appears only in the OpenLDAP set of properties.
com.stc.sentinel.auth.ldap.rolePattern	Enables you to configure pattern matching for role names. You can place the Enterprise Manager users in a separate line of business from other users in the LDAP directory. This property appears only in the Active Directory set of properties.

8 **Save the `ldap.properties` file.**

9 **Start the server component of Enterprise Manager.**

Specifying an Application Configuration Property Dynamically

To specify application configuration properties, you can use the static approach or the dynamic approach.

Using the static approach, you specify a property value at design time in the NetBeans IDE. The property value is included in the application file. If the value needs to be changed after deployment, then you must change the value in the NetBeans IDE, rebuild the application file, and redeploy the application file.

Using the dynamic approach, you specify an LDAP URL at design time. The URL must point to an attribute in an LDAP server. When you deploy the application file, the actual value is retrieved from the LDAP server. You can change the value in the LDAP server after deployment without performing the steps of the static approach. However, you do need to disable and then reenable the application file in order for the change to take effect.

You can use this feature for properties that accept string values (including passwords), numeric values, or boolean values.

Note – Another approach to updating property values does not require the use of LDAP. In the `asadmin` tool, run the `extract-caps-application-configuration` command. The configuration properties of the specified application file are extracted to a properties file. Update the value of one or more properties, and then run the `import-caps-configuration` command. Restart the application.

Enabling the Application Server to Access the LDAP Server

In this task, you edit properties that specify how the application server can access the LDAP server.

▼ To Enable the Application Server to Access the LDAP Server

1. **Start the `asadmin` tool included with Sun Java System Application Server.**

2. **Run the `export-caps-ldap-configuration` command. You must specify the directory where you want to store the `LDAP.properties` file.**

   ```
   asadmin> export-caps-ldap-configuration --capsconfigdir c:\temp
   ```

   The `LDAP.properties` file is generated.

3. **Using a text editor, open the `LDAP.properties` file.**

4. **Set values for the following properties, which specify how to access the LDAP server.**

   - `host`
   - `port`
   - `sslport`
   - `password`
   - `loginDN`

   The `ldapVersion` property is optional. You can set this property to any numeric value.

5. **Save the `LDAP.properties` file.**

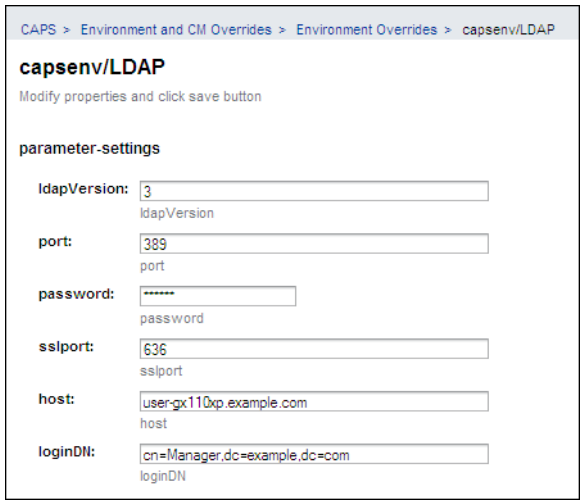
6. **Run the `import-caps-configuration` command. You must specify the directory that contains the `LDAP.properties` file.**

   ```
   asadmin> import-caps-configuration c:\temp
   ```

7 **Start the Admin Console included with Sun Java System Application Server.**

8 **In the left pane, expand the** `CAPS` **node, the** `Environment` and `CM` `Overrides` **node, and the** `Environment` `Overrides` **node. Select the** `capsenv/LDAP` **node.**

The property fields appear in the right pane. You can now update the properties from the Admin Console. Or you can update the `LDAP.properties` file and run the `import-caps-configuration` command again.



Specifying an LDAP URL for a Property

Here are two examples of LDAP URLs that might be used in Java CAPS:

```
ldap://uid=BatchFTP_TargetFileName,ou=Batch_Adapter,dc=Adapters,dc=sun,dc=com?cn
ldap://uid=BatchFTP_Password,ou=Batch_Adapter,dc=Adapters,dc=sun,dc=com?cn
```

The correct path to the property value in the LDAP server depends on the directory structure.

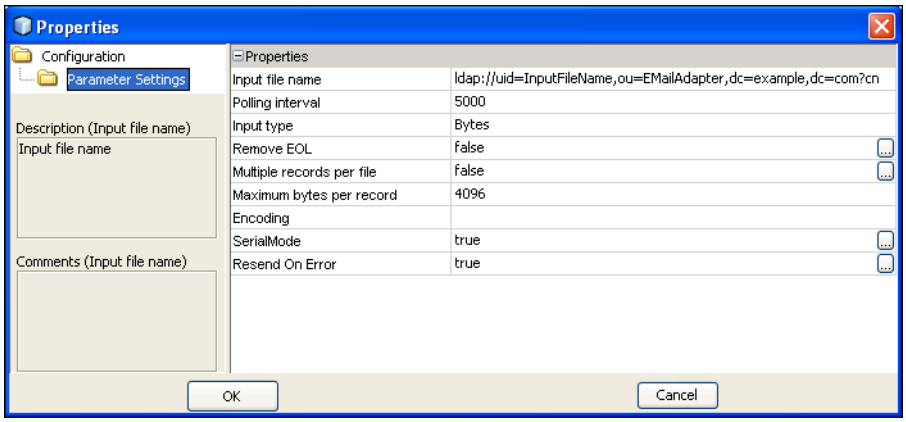
Do not include the backslash character (\) in an LDAP URL.

RFC 2255 defines the format of LDAP URLs. You can view the RFC at `http://www.ietf.org/rfc.html`.

▼ To Specify an LDAP URL for a Property

1 **In the NetBeans IDE, access the Properties dialog box that includes the property.**

2 **Enter an LDAP URL that points to the corresponding attribute in the LDAP server.**
In the following screen capture, the Input File Name property is set to an LDAP URL.



3 **Go to the LDAP server and enter the actual value.**

4 **When you deploy the application file, ensure that the LDAP server is running. If the LDAP server is not running, then the deployment will not succeed.**

Index