

Configuring Oracle9iAS Portal for LDAP Authentication

*An Oracle White Paper
December 2000*

Configuring Oracle9iAS Portal for LDAP Authentication

EXECUTIVE OVERVIEW

The use of Lightweight Directory Access Protocol (LDAP) directories is gaining popularity in corporate data centers. LDAP directories provide the benefit of centralized user administration, resulting in lower administrative support costs and efficient user account management. Oracle9iAS Portal integrates with the Oracle Internet Directory, Oracle's LDAP v3-compliant directory, to leverage a pre-existing user repository for authentication.

INTRODUCTION

There is currently great interest among corporate IT departments in deploying LDAP directories. LDAP is a standards-based protocol that permits hosting of user accounts, allowing other products and applications to leverage the directory information so that user accounts can be managed by a central administrator.

Oracle9iAS Portal works with Oracle Internet Directory, so that pre-existing users can readily become Oracle9iAS Portal users.

Oracle recognizes the value proposition of this argument and has thus developed Oracle Internet Directory (OID), an LDAP v3-compliant directory. Oracle9iAS Portal can be integrated with OID to support centralized administration of user accounts. The design goal in developing this integration capability was to address the need to use a pre-existing LDAP Directory Information Tree (DIT), which hosts user accounts, and authenticates those user accounts as valid Portal accounts. The idea is to leverage an existing schema in the directory, rather than requiring a particular schema to be represented in the tree. Since OID is a standards-compliant LDAP directory implementation, many other applications can leverage this directory information in the enterprise.

Although the sample scripts delivered with the Portal represent a rudimentary and simplistic LDAP schema, the configuration options of the Portal allow it to be adapted to more complex directory information trees.

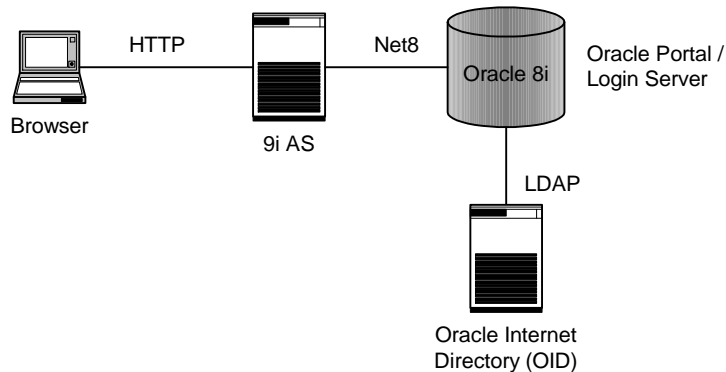
LOGIN SERVER EXTERNAL AUTHENTICATION

Technically, Oracle9iAS Portal does not communicate with LDAP at all. Oracle9iAS Portal delegates authentication of users to the Login Server, another component of Oracle 9i. By default, the Login Server provides an authentication service to Oracle Web-based applications by verifying user credentials against a

local repository stored in the Login Server's product schema. The Login Server also supports an external authentication mode, which allows an implementation of an external authentication module. That module must be written to a specification that allows an alternative authentication routine to be employed, and defines interfaces for changing a user's password and for resetting a user's password.

LDAP External Authentication

Oracle9iAS Portal ships an external authentication module for LDAP. The figure below depicts the Login Server in this configuration. In the diagram, the Portal and Login Server are separate schemas in a common database instance, which is the default configuration on a standard installation. By configuring the Login Server appropriately, it can use an external LDAP directory as its authentication repository. Replacement logic for the authentication routine checks the user's credentials against the LDAP directory to authenticate them.



In this figure, Oracle Internet Directory is the LDAP directory in use. OID is an LDAP v3 standards-compliant directory server with which Oracle9iAS Portal interoperates out-of-the-box. Only a simple configuration procedure, described later in this paper, is required.

Note the various protocols employed in this architecture. The browser communicates with Oracle 9i Application Server using HTTP or HTTPS. The Oracle 9i Application Server communicates with the Oracle 8i database through the mod_plsql module using the Net8 protocol. Within the database, the Login Server PL/SQL code makes an external procedure call to a dynamically linked library, which makes LDAP API calls to OID to do the authentication check.

Requirements on the LDAP Schema

To configure the Login Server for use with OID:

- Users must be represented in the Directory Information Tree (DIT) with an object class that allows for binding, with a userPassword attribute, and a distinguished name (DN).

- The object class used to represent users must have a unique attribute containing the value to be used for the single sign-on user name, which is the global identification of the user. This value must be unique across the searchable range of the DIT.
- The values used for the single sign-on user name must not exceed 30 characters.
- All directory nodes representing user accounts must be under the node specified as the search root.
- Currently, a password change can be accomplished by an `ldap_modify_s` on the `userPassword` attribute of the node representing the user. The directory access control policy should allow a user to perform this change on his own entry.
- If using the reset password feature, the DN specified for the bind DN should be an account which has administrative privileges to modify the `userPassword` attribute for other user's entries. If not using this feature, the bind DN may be an account that can search the DIT from the search base, down.

Software Components

The external authentication module for LDAP is implemented by the `SSOXLDPK.PKB` file. This is an implementation of the external authentication specification - `WSSO_AUTH_EXTERNAL`.

The `SSOXLDPK.PKB` file contains a few procedures that are defined as 'external' and implemented in an external library named 'auth_ext', which is implemented in C. This procedure is provided in the `SSOXLDPK.DLL` file for NT, the `ssoxldap.so` file for Solaris, and similarly named files for other platforms.

The 'create library ...' step described in the configuration instructions performs the linkage between the PL/SQL reference and the library implementation.

HOW TO CONFIGURE FOR LDAP

Let's begin by going through the default, simple installation of the LDAP authentication module. Then we will address how to adapt the Login Server to a pre-existing LDAP directory schema.

Default Installation

After successfully installing the Portal and the Login Server, you can reconfigure the Login Server for LDAP authentication mode as described in this section.

Before you begin, ensure that you have an LDAP directory that is properly set up and that you have administrative access to it. For the purposes of this tutorial, we'll assume the following:

```
ldap hostname: ldap.us.oracle.com
```

```
ldap port: 389
admin dn entry: "cn=orcladmin"
admin password: "welcome"
```

Step 1: Copy the LDAP Callout Library

Copy the appropriate library file used for the LDAP API callouts from the \$PORTAL_HOME/portal30/admin/plsql/sso directory of the product installation into the appropriate place on the Login Server machine:

Example for NT

```
copy %PORTAL_HOME%\portal30\admin\plsql\sso\ssoxldap.dll
%ORACLE_HOME%\bin\ssoxldap.dll
```

Example for Solaris

```
cp $PORTAL_HOME/portal30/admin/plsql/sso/ssoxldap.so
$ORACLE_HOME/lib/ssoxldap.so
```

Step 2: Create the Library Linkage

Log into the Login Server schema through SQL*Plus and run the following command to create the external library:

```
create or replace library auth_ext as 'library_file_name';
commit;
```

Example for NT

```
create or replace library auth_ext as
'C:\Oracle\Ora81\bin\ssoxldap.dll';
commit;
```

Example for Solaris

```
create or replace library auth_ext as
'/u01/app/oracle/product/816/lib/ssoxldap.so';
commit;
```

Step 3: Verify that External Procedure Calls are Enabled

Since the interface between the Login Server and the LDAP directory is implemented through an external procedure call that makes LDAP API calls, the database that the Login Server resides in must have a configured TNS listener for external procedure (PLSExtProc) calls.

The listener.ora file of the database ORACLE_HOME must have an EXTPROC0 entry, such as:

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = pncarna_us)(PORT = 1521))
      )
    )
  )
  (DESCRIPTION =
    (PROTOCOL_STACK =
      (PRESENTATION = GIOP)
      (SESSION = RAW)
    )
  )
```

```

    )
    (ADDRESS = (PROTOCOL = TCP)(HOST = pencarna_us)(PORT = 2481))
  )
)

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = D:\Oracle\Ora81)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = PENCARNA)
      (ORACLE_HOME = D:\Oracle\Ora81)
      (SID_NAME = PENCARNA)
    )
  )
)

```

Similarly, the TNSNAMES.ORA file should have an EXTPROC_CONNECTION_DATA entry, such as:

```

EXTPROC_CONNECTION_DATA.US.ORACLE.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )
)

```

To test whether the PLSExtProc listener is running, you can run:

```
C:\>set ORACLE_HOME=D:\Oracle\Ora81
```

```
C:\>lsnrctl status
```

```
LSNRCTL for 32-bit Windows: Version 8.1.6.0.0 - Production on 10-NOV-2000 09:28:02
```

```
(c) Copyright 1998, 1999, Oracle Corporation. All rights reserved.
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)))
STATUS of the LISTENER
```

```

-----
Alias                LISTENER
Version              TNSLSNR for 32-bit Windows: Version
8.1.6.0.0 - Production
Start Date           10-NOV-2000 09:26:57
Uptime               0 days 0 hr. 1 min. 4 sec
Trace Level          off
Security             OFF
SNMP                 OFF
Listener Parameter File D:\Oracle\Ora81\network\admin\listener.ora
Listener Log File    D:\Oracle\Ora81\network\log\listener.log
Services Summary...
  PENCARNA           has 1 service handler(s)
  PENCARNA           has 2 service handler(s)
  PLSExtProc         has 1 service handler(s)
The command completed successfully

```

The PLSExtProc should be shown in the Services Summary, along with a service handler. This shows that the listener is running and ready to receive requests.

Next, to check whether the environment is properly set to access the appropriate TNS names file, perform a `tnsping` on the `EXTPROC_CONNECTION_DATA`, as follows:

```
C:\>tnsping extproc_connection_data
```

```
TNS Ping Utility for 32-bit Windows: Version 8.1.6.0.0 - Production on
10-NOV-2000 09:32:10
```

```
(c) Copyright 1997 Oracle Corporation. All rights reserved.
```

```
Attempting to contact (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC))
OK (100 msec)
```

```
C:\>
```

Step 4: Switch the Login Server to External LDAP Authentication Mode

Log into the Login Server schema through SQL*Plus (assuming the default - `portal30_sso`) and run the `ssoldap` script. This script, which is interactive, prompts the user for values:

```
sqlplus portal30_sso/portal30_sso
@ssoldap
Configuring Login Server to use LDAP.
Enter host, port, search base, unique attribute,
bind DN for searching and bind password for searching.
```

EXAMPLE:

```
Host: ldap.us.oracle.com
Port: 389
Search Base: cn=Login Server (portal30_sso)
Unique Attribute: cn
Bind DN: cn=orcladmin
Bind Password: welcome
```

```
Enter value for Host: ldap.us.oracle.com
```

```
Enter value for Port: 389
```

```
Enter value for Search_Base: cn=Login Server (portal30_sso)
```

The `Search_Base` should be set to a node on the Directory Information Tree (DIT) that has all of the users under it. If you are using the LDIF file created by the `ssoldif.sql` script, this base can be specified as:

```
cn=Login Server (<sschema>)
```

```
Enter value for Unique_Attribute: cn
```

This attribute should contain the user's single sign-on user name. In addition, searching for the user name in this attribute - starting from the `search_base`, and using a subtree search - should result in a single hit, to obtain the DN. If using the `ssoldif.sql` script, we use the 'cn' attribute of the *person* object class.

```
Enter value for Bind_DN: cn=orcladmin
```

This value represents the bind DN for a user account which is used for searching the DIT for the user's DN. Pick an account that has search access to the relevant hierarchy in the DIT.

```
Enter value for Bind_Password: welcome
```

This is the password necessary for a bind for the above account.

The Login Server is now configured for LDAP authentication.

Step 5: Migrate the User Accounts to LDAP

Step 5a: Populate the LDAP Directory

Now we need to setup the LDAP server to accommodate the login accounts. This particular example assumes that the LDAP directory does not have a pre-existing user repository, so we populate it with some entries from the Login Server's local repository.

Before proceeding, make sure your init.ora file allows you to write to a file system. The script you will run migrates the accounts that have privileges in the Portal installation to the LDAP directory so that the privileges are preserved. Run the ssoldif.sql script from the Login Server schema. This script creates an LDIF file containing the user accounts that are in the Login Server at the time of execution.

The init.ora directive, which indicates the file location where the LDIF file can be written, is specified by the UTL_FILE_DIR directive. The init.ora file should contain something like:

```
UTL_FILE_DIR = D:\Oracle\Ora81\admin\udump
```

Step 5b: Create an LDIF File for Current Users

Assuming the directive has been set in the init.ora file, run these steps to create an LDIF file containing the current users defined in the Login Server's local repository.

```
cd PORTAL_HOME$/portal30/admin/plsql/sso
sqlplus portal30_sso/portal30_sso
SQL> @ssoldif
Generating 'users.ldif' file for existing Portal users.
Enter the desired file location.
NOTE: The file location must be specified in the appropriate
parameter in the init.ora file.

EXAMPLE:
File Location: /u01/app/oracle/product/816/admin/w816dev6/udump

Enter value for file_location: D:\oracle\admin\pencarna\udump

PL/SQL procedure successfully completed.

No errors.
SQL>quit
```

Step 5c: Add the Entries To the LDAP Directory

Get the file that was created in Step 5b (users.ldif), and add the entries to the LDAP directory. This example uses Oracle Internet Directory's ldapadd command line utility:

```
ldapadd -h ldap_server.us.oracle.com -p 389 -D 'cn=orcladmin'
-w welcome -f users.ldif
```

Once these users are successfully added, you are ready to log into the Portal through the Login Server, authenticating against this LDAP directory!

Step 5d: Create Users

Now that the Portal/Login Server is configured to use an external authentication repository, such as LDAP, you must create users (sometimes called provisioning users) externally to the Portal and Login Server's administrative interface. Use a tool compatible with LDAP for the user administration. If you are using the Oracle Internet Directory, you can use Oracle Directory Manager for this purpose. There are also a number of third party tools on the market that operate with OID for user provisioning. Alternatively, custom-built tools that are LDAP-enabled may also be developed to suit the specific purposes of user provisioning and administration. The 8.1.7 version of the Oracle database includes the DBMS_LDAP package, which allows PL/SQL applications to implement LDAP functionality.

LDIF File

When Oracle9iAS Portal is installed, there are a couple of accounts created that are defined as administrative accounts. When LDAP authentication is enabled, you must migrate these accounts into the LDAP directory so that these administrative users can log on and grant other appropriate privileges to the other users defined in the LDAP directory.

The `ssoldif.sql` script is provided for this purpose. This script generates a file like this (this example shows the case for a Portal schema named 'portal30'):

```
dn: cn=Login Server (portal30_sso)
cn: Login Server (portal30_sso)
description: Central Authentication Authority
objectClass: top
objectClass: applicationProcess

dn: cn=PORTAL30_SSO, cn=Login Server (portal30_sso)
sn: PORTAL30_SSO
cn: PORTAL30_SSO
userPassword: portal30_sso
objectClass: top
objectClass: person

dn: cn=PORTAL30_SSO_ADMIN, cn=Login Server (portal30_sso)
sn: PORTAL30_SSO_ADMIN
cn: PORTAL30_SSO_ADMIN
userPassword: portal30_sso_admin
objectClass: top
objectClass: person

dn: cn=PORTAL30, cn=Login Server (portal30_sso)
sn: PORTAL30
cn: PORTAL30
userPassword: portal30
objectClass: top
objectClass: person

dn: cn=PORTAL30_ADMIN, cn=Login Server (portal30_sso)
```

```
sn: PORTAL30_ADMIN
cn: PORTAL30_ADMIN
userPassword: portal30_admin
objectClass: top
objectClass: person

dn: cn=PUBLIC, cn=Login Server (portal30_sso)
sn: PUBLIC
cn: PUBLIC
userPassword: public
objectClass: top
objectClass: person
```

As this example shows, the default object class used to represent the user entries is the *person* object class. The unique attribute containing the value representing the single sign-on user name is the *cn* attribute. All the user entries are children of the node 'cn=Login Server (portal30_sso)' which is of the *applicationProcess* object class.

If you already have a set of entries defining users, and some Directory Information Tree (DIT) organization is already defined, this script can be modified to produce the necessary format. Or you can manually create entries in the LDAP directory of the appropriate object class so that the users *<portal_schema>* and *<portal_schema>_ADMIN* can log on.

Note that if using a different object class than *person*, make sure that there is a unique attribute in the object class that assures a unique DN. This attribute should contain the single sign-on user name. The name of this attribute should be specified when running the *ssoldap.sql* script.

Adding User Entries

In LDAP mode, you must add users directly to the LDAP directory. You can do this through command line tools or graphical administration tools.

Using Command Line Tools

To add users from the command line, use the *ldapadd* command line tool that comes with Oracle 8i client. This command, which is in the *\$ORACLE_HOME/bin* directory, submits an LDIF file to the LDAP directory, to add to the directory information tree. This command will succeed only if the user specified in the *-D* parameter of the command has appropriate authorization on the LDAP server.

If you are using an Oracle Internet Directory installation with the default setup, this command would add the users specified in the *newuser.ldif* file to the directory:

```
ldapadd -h ldap_server.us.oracle.com -p 389 -D 'cn=orcladmin'
-w welcome -f newuser.ldif
```

Please see the Oracle Internet Directory Administrator's Guide for further usage information.

An example newuser.ldif file is as follows:

```
dn:cn=MAYA, cn=Login Server (portal30_sso)
sn:Encarnacion
cn:MAYA
userPassword:paolo
objectclass:person
objectclass:top
```

Using Administrative Graphical Tools

The Oracle Directory Manager is an example of an appropriate graphical administration tool. Log in to the directory using an account with the necessary authorization, e.g., 'cn=orcladmin', and add an entry to the tree of the appropriate object class at the appropriate node. An 'appropriate node', is one that is somewhere below the search root specified in the ssoldap.sql script.

See the Oracle Directory Manager documentation for usage information.

Adapting the Login Server to an Existing DIT

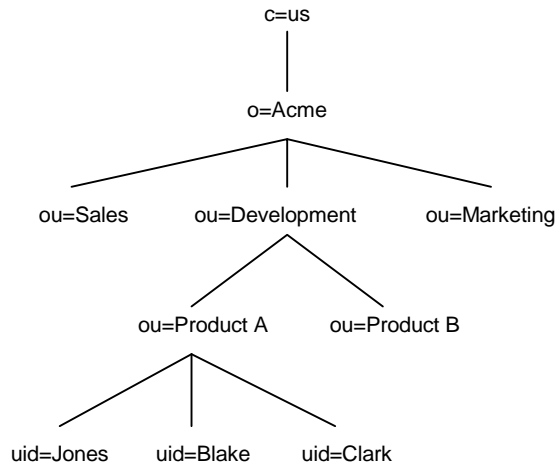
If the LDAP directory already hosts user accounts in a directory information tree, the Login Server and Portal can leverage the existing user accounts as long as

1. the appropriate information into the ssoldap.sql script, and
2. the administrative user accounts are migrated over to the LDAP directory

An Example Directory Structure

Let's say that a particular company already has an LDAP directory, and they are using the inetOrgPerson object class to represent each user.

Let's take the following DIT as an example:



In this example, the 'Acme' company has organized their directory by organizational function, and users are listed under the product area in which they function. It is assumed that there are users listed under Sales and Marketing as

well. So, since all users can be found beneath the node identified by “o=Acme, c=us”, this should be identified as the search base.

We assume that the object class representing Jones, Blake and the other users, is the inetOrgPerson class. In this class, uid is typically used as the relative distinguished name (RDN). Furthermore, there is the requirement that the uid be unique across the DIT under the search base, so that when the Login Server searches the tree for uid=Clark, for example, it should find only one node – in this example, the node identified by:

```
"uid=Clark, ou=Product A, ou=Development, o=Acme, c=us"
```

Configuring the Login Server Appropriately

To hook up the Login Server to this LDAP directory and allow these users to log on to the Portal with their LDAP accounts, one would run the script `ssoldap.sql` with the following parameters:

```
Host: ldap.acme.com  
Port: 389  
Search Base: o=Acme, c=us  
Unique Attribute: uid  
Bind DN: cn=orcladmin  
Bind Password: welcome
```

Of course, the LDAP host name and port, as well as the bind DN and password, are dependent on the particular LDAP installation.

Specifying the Appropriate Bind DN

The decision on what bind DN should be used for the Login Server requires an understanding of how it is used. The bind DN is used for searching the tree for the user’s DN, as well as for resetting a user’s password when the user has forgotten it. This is referred to as the ‘reset password’ feature.

The reset password feature is an optional feature that is not installed in the Login Server by default. If the administrator decides to enable this feature, the bind DN specified in `ssoldap.sql` should be for an account which has the access control privileges necessary to modify any user’s `userPassword` attribute.

However, if the reset password feature is not used, the bind DN need only be an account which is able to search the DIT from the search base and below.

Adding the Administration Accounts

After configuring the Login Server for the LDAP directory, thereby allowing the LDAP users to log into Oracle9iAS Portal, the final step required to complete the adaptation is to create the administrator accounts defined in the Portal in the LDAP directory. This allows the predefined administrator accounts to log on and transfer access privileges to the real users defined in the LDAP server.

Since this example has an LDAP schema that differs from the default schema generated by the `ssoldif.sql` file, running `ssoldif.sql` will not satisfy this requirement.

In this case, the easiest thing to do is just create user entries in the LDAP directory as if provisioning any new user, and create an account for PORTAL30 – or whatever the Portal schema name is for this instance. The administrator should then log on as PORTAL30 and assign administrative privileges to other users using the Portal Administrative screens. The easiest way to grant someone full administrative privileges in the Portal is to use the Users portlet and add the user to the DBA group.

LDAP OPERATIONS

The following sections describe the actions performed by the procedures in the external authentication module for LDAP. They are described here to help you understand how the configuration values provided in the configuration step are used by the module.

Authentication Flow

Here's what happens on an authentication request: When a user provides a user name and password in the login page or portlet, the Login Server calls the external authentication module for LDAP, which attempts to bind to the LDAP directory with the bind DN specified in the `ssoldap.sql` script.

If the module is unable to bind to the directory, an error message indicating a setup problem is displayed. If the bind is successful, the login server then searches the DIT, starting from the search base, for the node with `uid=username`, where `uid` was specified as the unique attribute and `username` is the name provided by the user in the login form.

If no node is found, it is considered an authentication failure. If a node is found (which would have to be a single node), the Login Server obtains the DN for this node and attempts to bind to the directory as this DN, with the password provided by the user in the login form. If the bind attempt is successful, the Login Server considers it a successful authentication. Otherwise, it is an authentication failure.

Change Password Flow

The external authentication specification also supports a change password feature, which is implemented by the LDAP authentication module.

When a user wants to change his/her password, the Login Server provides a change password form. The user provides his/her current password and the new password, entered twice to confirm its proper entry.

The Login Server obtains the user's DN by performing a search, as described in the authentication flow, after doing a bind with the bind DN. Once the user's DN is obtained, the Login Server binds to the directory with that DN, using the user's current password, and then does an `ldap_modify_s` of the `userPassword` attribute on

that node. It is assumed that the access control policy on the directory allows a user to modify his own password.

USER LIST SYNCHRONIZATION

When using LDAP authentication – or any other external repository, for that matter – the list of users for authentication is held on the external repository.

However, there is also a list of users held on the Login Server, which is used to associate privileges to the user. Ideally, this list is maintained transparently and automatically. In fact, if a user account is created on LDAP and a user attempts to log in, the login will succeed, and an entry is automatically created on the Login Server for that user, after a successful login. Similarly, once the user accesses the Portal, an entry on the Portal is created – also for the purpose of associating Portal privileges and group memberships to the user.

Since the Portal uses the user list on the Login Server to determine the users to which privileges can be granted, and from which to select group members, it is sometimes necessary to have that list populated with the user entry before the user logs on. If this is necessary, you must a script to synchronize the Login Server list with the centralized authentication list held in LDAP. Alternatively, the user provisioning system could be designed to add a shadow entry to the Login Server for each user entry added to the directory.

Future versions of the Login Server will plan to provide better synchronization between these systems.

TROUBLESHOOTING

If you are unable to log in after configuring the Login Server for LDAP authentication, you will most likely see one of the error messages addressed in this section. This section indicates the error messages, the possible causes, and remedies.

Setup Errors

If incorrect inputs were provided when executing the SSOLDAP.SQL script, this error message is most likely to be displayed:

There is an error in the setup of the external authentication mechanism.
Please contact the administrator to make sure the external repository is setup appropriately. (WWC-41655)

You can verify the values you provided by logging into the Login Server schema and selecting the values from the configuration file:

```
select * from wssso_ext_configuration_info$;
```

Verify the information displayed and rerun the ssoldap.sql file if necessary. Note that in the listing created by the select statement above, the bind DN and bind

passwords are obfuscated. If you feel that these may be the problem, just run `ssoldap.sql` again. You can also revert the authentication to the default –local mode –by running the script `ssolocal.sql`.

Setup Error or Invalid Credentials

If the unique attribute specified in `ssoldap.sql` is wrong, or if the user entered an incorrect password when attempting to log in, the following error message is displayed:

Authentication failed. Please try again. (WWC-41419)

If the wrong value was entered for the unique attribute, just run `ssoldap.sql` again, specifying the correct value.

Unexpected Errors

Sometimes you may get an error message that says simply:

Unexpected errors (WWC-41400)

This is usually due to one of the following problems.

Library Linkage Not Set

The create library step needs to be run. Enter the following command:

```
create or replace library auth_ext as 'library_file_name'  
/  
commit;
```

where `library_file_name` is the full path and name of the library file described in Step 2, on page 5.

Example for NT

```
create or replace library auth_ext as  
'c:\oracle\ora81\bin\ssoxldap.dll'
```

External Library File Not In Right Place

The `ssoxldap.dll` or `ssoxldap.so` file is not in the proper location. Verify that the appropriate library file is in the correct place.

EXTPROC Not Functioning

Verify this by doing a `tnsping` on the `extproc` listener:

```
C:\>set oracle_home=D:\oracle\ora81  
C:\>tnsping extproc_connection_data
```

TNS Ping Utility for 32-bit Windows: Version 8.1.6.0.0 -
Production on 09-NOV-2000 18:37:51

(c) Copyright 1997 Oracle Corporation. All rights reserved.

Attempting to contact (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0))
OK (70 msec)

If 'lsnrctl status' indicates that the extproc listener is running, but the tns ping fails, then the ORACLE_HOME or TNS_ADMIN environment variables are probably not pointing to the proper place. The tnsnames.ora and listener.ora files used in this extproc configuration are on the database ORACLE_HOME – not the 9i AS ORACLE_HOME.



Using LDAP for Authentication with Oracle9i/AS Portal
December 2000

Author: Paul Encarnación

Contributors: Michael Mesaros, David Saslav, Bill Lankenau

Editor: Marcie Caccamo

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle Corporation provides the software
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2000 Oracle Corporation
All rights reserved.