Oracle9*i*AS Single Sign-On

Administrator's Guide

Release 3.0.9

May 2001

Part No. A88732-01

Oracle9*i*AS Single Sign-On Administrator's Guide, Release 3.0.9

Part No. A88732-01

Primary Author: Ted Burroughs

Contributing Authors: Susan Barton, Cynthia Kibbe, David Mathews, Frank Rovitto

Contributors: Gaurav Bhatia, Ratna Bhavsar, Kamalendu Biswas, Paul Encarnacion, John Heimann, Mike Hwa, Arun Swaminathan

Contents

2 Administrative Basics

3 User Management

4 Application Management

5 Customizing Single Sign-On

6 Deployment Considerations

Index

List of Figures

List of Tables

Send Us Your Comments

Oracle9*i*AS Single Sign-On Administrator's Guide, Release 3.0.9

Part No. A88732-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227 Attn: Server Technologies Documentation Manager
- Postal service:

  Oracle Corporation
  Server Technologies Documentation
  500 Oracle Parkway, Mailstop 4op11
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

The *Oracle9i*AS Single Sign-On Administrator's Guide provides conceptual information and instructions for using Oracle9*i*AS Single Sign-On to sign on to multiple applications from Oracle9*i*AS Portal.

This preface contains these topics:

- Audience
- Organization
- Related Documentation
- Conventions
- Documentation Accessibility

Audience

The *Oracle9i*AS Single Sign-On Administrator's Guide is intended for:

- Administrators integrating Oracle9*i*AS Single Sign-On with applications
- Users logging into applications using Oracle9*i*AS Single Sign-On
- Anyone interested in Oracle9*i*AS Single Sign-On conceptual information

Organization

This document contains the following chapters:

Chapter 1, "Concepts and Architecture" provides a high-level overview of Oracle9*i*AS Single Sign-On for users and administrators in an enterprise.

Chapter 2, "Administrative Basics" describes the Login Server and Oracle9*i*AS Single Sign-On, and how you can perform various tasks using them.

Chapter 3, "User Management" describes how to administer Oracle9*i*AS Single Sign-On user accounts.

Chapter 4, "Application Management" explains how to use Oracle9*i*AS Single Sign-On for partner and external applications.

Chapter 5, "Customizing Single Sign-On" explains how to customize pages to match the look and feel of your own portal or product and configure LDAP authentication.

Chapter 6, "Deployment Considerations" discusses deployment considerations.

Related Documentation

For additional information, see the *Oracle9iAS Single Sign-On Application Developer's Guide*, and the online help and related documentation for Oracle9*i*AS Portal.

In North America, printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

`http://www.oraclebookshop.com/`

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

```
http://technet.oracle.com/membership/index.htm
```

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

```
http://technet.oracle.com/docs/index.htm
```

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.
UPPERCASE monospace (fixed-width font)	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.

Convention	Meaning	Example
`lowercase monospace (fixed-width font)`	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter `sqlplus` to open SQL*Plus. The password is specified in the `orapwd` file. Back up the datafiles and control files in the `/disk1/oracle/dbs` directory. The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table. Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true.` Connect as `oe` user. The `JRepUtil` class implements these methods.
`lowercase monospace (fixed-width font) italic`	Lowercase monospace italic font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run `U`*`old_release`*`.SQL` where *`old_release`* refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (`*`digits`*` [ , `*`precision`*` ])`
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`

|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	`{ENABLE | DISABLE}`
		`[COMPRESS | NOCOMPRESS]`

Convention	Meaning	Example

...	Horizontal ellipsis points indicate either:
	■ That we have omitted parts of the code that are not directly related to the example
	■ That you can repeat a portion of the code
		CREATE TABLE ... AS *subquery*;
		SELECT *col1*, *col2*, ... , *coln* FROM employees;

. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	

Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2);
		acct CONSTANT NUMBER(4) := 3;

Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*

UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees;
		SELECT * FROM USER_TABLES;
		DROP TABLE hr.employees;

lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		SELECT last_name, employee_id FROM employees;
		sqlplus hr/hr
		CREATE USER mjones IDENTIFIED BY ty3MU9;

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

1

Concepts and Architecture

This chapter provides a high-level overview of Oracle9*i*AS Single Sign-On.

This chapter contains these topics:

Introduction to Oracle9*i*AS Single Sign-On

Oracle9*i*AS Single Sign-On is a component of Oracle9*i*AS Portal that provides a framework for secure single sign-on; allowing users to log in to multiple Web-based applications, such as expense reporting, e-mail, and benefits information, using a single username and password.

Oracle9*i*AS Single Sign-On provides the following benefits:

- Reduced administrative costs associated with supporting multiple accounts and passwords for each user
- Convenient login experience, because users do not need to maintain a separate username and password for each application they access
- Increased security, because when the password is only required once, users are less likely to use simple, easy-to-remember passwords or write them down

Oracle9*i*AS Single Sign-On Components

This section describes the following Oracle9*i*AS Single Sign-On components:

- Login Server
- Partner Applications
- External Applications
- Oracle9iAS Single Sign-On Software Development Kit

Login Server

The core of the Oracle9*i*AS Single Sign-On technology is the Login Server. For partner applications, which rely on the Login Server for authentication, the Login Server authenticates users and passes their identities to the applications. For external applications, which do not use the Login Server authentication mechanism, the Login Server provides single sign-on through a centralized password repository.

When a user first tries to access a Oracle9*i*AS Single Sign-On enabled application, the Login Server:

- Authenticates the user by means of username and password
- Stores an encrypted login cookie on the authenticated client
- Passes the client’s identity to the Oracle9*i*AS Single Sign-On enabled application

During subsequent user logins, the login cookie provides the Login Server with the user's identity and indicates that authentication has already been performed. If there is no login cookie, the Login Server presents the user with a login page.

To guard against eavesdropping, the Login Server can send the login cookie to the client browser over an encrypted SSL channel.

The login cookie expires with the session, either at the end of a time interval specified by the administrator, or when the user exits the browser. The login cookie is never written to disk.

Note: To log out of a partner application and log in as another user, the user must also log out of the Login Server session; otherwise the authentication request returns the partner application to the logged in state of the previous user.

Partner Applications

Partner applications are integrated with the Login Server. They support a single sign-on mechanism that enables them to accept a user's username and password as validated by the Login Server. Partner applications are generally written using the Oracle9*i*AS Single Sign-On SDK or Login Server specification.

A partner application delegates its authentication to the Login Server. If a partner application is registered with the Login Server, users can log into it using the single sign-on mechanism.

A partner application is responsible for determining whether a user authenticated by Oracle9*i*AS Single Sign-On has the requisite privileges for using the partner application. It also controls user access within the application.

An example of a partner application is the Oracle9*i*AS Portal itself.

External Applications

External applications retain their own authentication logic and do not delegate authentication to the Login Server. External applications therefore require application-specific usernames and passwords for providing access.

External applications typically use HTML login forms to accept the username and password. The username for the application can be different from the Oracle9*i*AS Single Sign-On username, in which case Login Server provides the required mapping.

Examples of external applications that use HTML login forms include Oracle Mobile and Yahoo! Mail. A unique username and password may be required to access each external application.

The Login Server provides usernames and passwords to external applications on behalf of the user. Users have the option of mapping external application authentication information to one or more usernames and passwords that are stored in the Login Server's password store. To map a password, the user selects the Remember My Login Information For This Application checkbox for the external application.

> **See Also:** "Specifying External Application Login Information" in Chapter 4, "Application Management" for information about the Remember My Login Information For This Application checkbox.

A single Login Server account can be mapped to several external application usernames and passwords. When the user attempts to log in, the Login Server automatically checks the mappings and sends the user's authentication information to the appropriate external application.

> **Note:** Only a system administrator can add external applications accessed by the Login Server.

> **See Also:** "Adding Partner and External Applications" in Chapter 4, "Application Management" for information about adding external applications.

Oracle9*i*AS Single Sign-On Software Development Kit

The Oracle9*i*AS Single Sign-On Software Development Kit (SDK) enables the following functionality:

- Applications can communicate with the Login Server and accept a user’s identity as validated by the Login Server
- Administrators can manage the application’s association with the Login Server

> **See Also:** The *Oracle9iAS Single Sign-On Application Developer’s Guide* for information about the Oracle9*i*AS Single Sign-On Development Kit.

Oracle9*i*AS Single Sign-On Authentication Methods

Oracle9*i*AS Single Sign-On uses the following authentication methods:

- Local User Authentication
- External Repository Authentication

Local User Authentication

Local user authentication uses a lookup table within the Login Server schema on the Oracle database associated with Oracle9*i*AS Portal. The table contains usernames, passwords, and Login Server privilege levels for the users. The incoming password is one-way hashed and compared to the entry in the table.

External Repository Authentication

External repository authentication typically relies on an LDAP-compliant directory, specifically Oracle Internet Directory. In this case, the Login Server binds to Oracle Internet Directory, then looks up the user credentials stored in the directory. External repository authentication includes LDAP and other methods that can be custom developed.

Oracle9*i*AS Single Sign-On Processes

Before a user can access a Oracle9*i*AS Single Sign-On enabled application, the Login Server must authenticate the user.

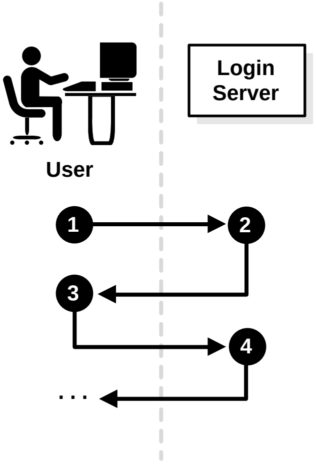
This section describes the following Oracle9*i*AS Single Sign-On processes:

- Authenticating to the Login Server
- Accessing a Partner Application
- Partner Application Development Requirements
- Accessing an External Application

Authenticating to the Login Server

Figure 1–1 illustrates the Login Server authentication process.

Figure 1–1 Authenticating to the Login Server



1. The Login Server checks for a login cookie. If one is present, the Login Server identifies the user from the encrypted information in the login cookie.

2. If a login cookie is not present, the Login Server prompts the user for the user’s credentials.

3. The user provides a username and password.

4. The Login Server authenticates the user by passing the provided name and password to the configured authentication routine—either the local routine or one provided by an external authentication module for an external repository. If the authentication is successful, the Login Server establishes a login cookie on the client browser to facilitate Oracle9*i*AS Single Sign-On for future authentication requests.

See Also: "Login Server" for information about the login cookie

Accessing a Partner Application

Figure 1–2 illustrates the process that occurs when a user seeks access to a partner application.

Figure 1–2 Accessing a Partner Application

1. The user seeks access to the partner application directly.

2. If this is the first time during a session that the user is accessing the partner application, the partner application transparently redirects the user to the Login Server to obtain authentication credentials.

3. The Login Server authenticates the user as described in "Authenticating to the Login Server".

4. The Login Server transparently redirects the user to the partner application by using a URL with an encrypted parameter containing the user's identity.

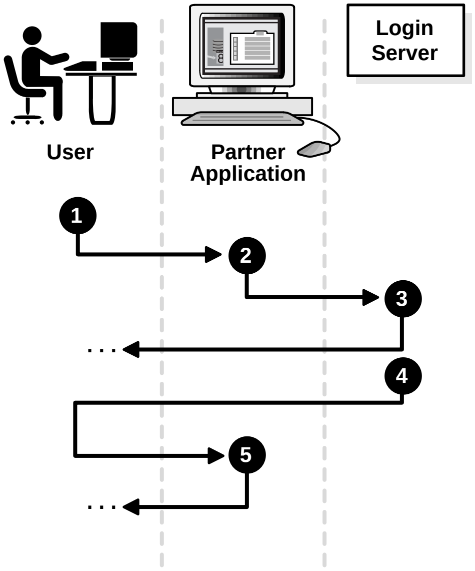
5. The partner application:

—Decrypts the parameter

—Identifies the user

—Establishes its own session management (for example, determining what, if any, access privileges to grant to the user)

—Sets a partner application cookie so that subsequent user access does not require a redirect to the Login Server



In Step 2 of this process, the partner application redirects the user to the Login Server only if the application requires it, based on the requested URL. Some URLs may be public and no redirection to the Login Server is required. When necessary, the partner application protects itself from unauthenticated access by its own session management.

If, during the same session, the user again seeks access to the same or to a different partner application, the Login Server does not prompt the user for a username and password. Instead, the Login Server obtains the information from the login cookie that is already on the client browser.

Partner Application Development Requirements

To implement an authentication check:

1. If the URL is publicly accessible, no authorization check is implemented.
2. Protected URLs check for an application session cookie for authorization.
3. If an application session cookie does not exist, the browser redirects the user to the Login Server.

To implement a sign-on URL:

1. The URL must establish an application session cookie using the identity information sent by the Login Server.
2. The browser redirects the user to the requested URL.

Accessing an External Application

You can access an external application through Oracle9*i*AS Portal. In this scenario, Oracle9*i*AS Portal functions as a partner application.

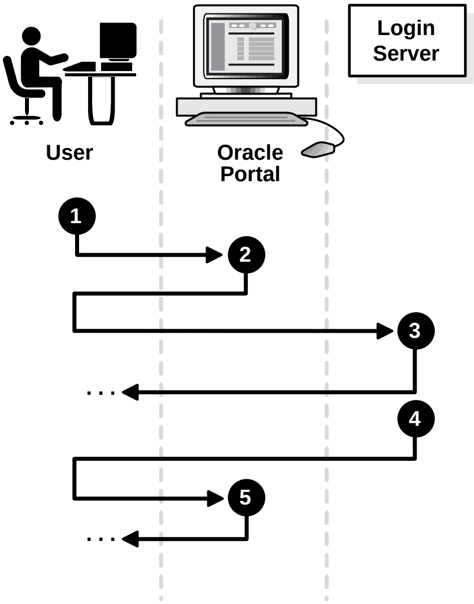
This section contains these topics:

- Authenticating to Oracle9iAS Portal
- Authenticating to an External Application for the First Time
- Authenticating to an External Application After the First Time

Authenticating to Oracle9*i*AS Portal

When a user seeks access to an external application through Oracle9*i*AS Portal, the Login Server authenticates the user to Oracle9*i*AS Portal by the process described in Figure 1–3.

Figure 1–3 Authenticating to the Oracle9iAS Portal



1. The user seeks access to the Oracle9*i*AS Portal site.

2. If this is the first time during a session that the user is accessing Oracle9*i*AS Portal, Oracle9*i*AS Portal transparently directs the user to the Login Server to obtain authentication credentials.

3. The Login Server authenticates the user as described in "Authenticating to the Login Server".

4. The Login Server transparently directs the user to Oracle9*i*AS Portal. It does this by using a URL with an encrypted parameter containing the user’s identity.

5. Oracle9*i*AS Portal:

—Decrypts the parameter

—Identifies the user

—Establishes its own session management

—Presents the user with links to the external applications

If, during the same session, the user again seeks access to Oracle9*i*AS Portal, the Login Server does not prompt the user for a username and password. Instead, it obtains that information from the login cookie on the client browser.

Authenticating to an External Application for the First Time

Oracle9*i*AS Single Sign-On uses the process described in Figure 1–4 if the user:

- Has authenticated to the Oracle9*i*AS Portal
- Is accessing an external application for the first time through Oracle9*i*AS Portal

Figure 1–4 Authenticating to an External Application for the First Time

1. Oracle9*i*AS Portal presents links to external applications to the user. The links invoke a routine on the Login Server.

2. A user clicks one of the links.

3. Clicking a link begins the external application login procedure. The procedure checks the Login Server password store for the user’s credentials for the requested external application. If it finds that the user has no such credentials, the Login Server prompts the user for them.

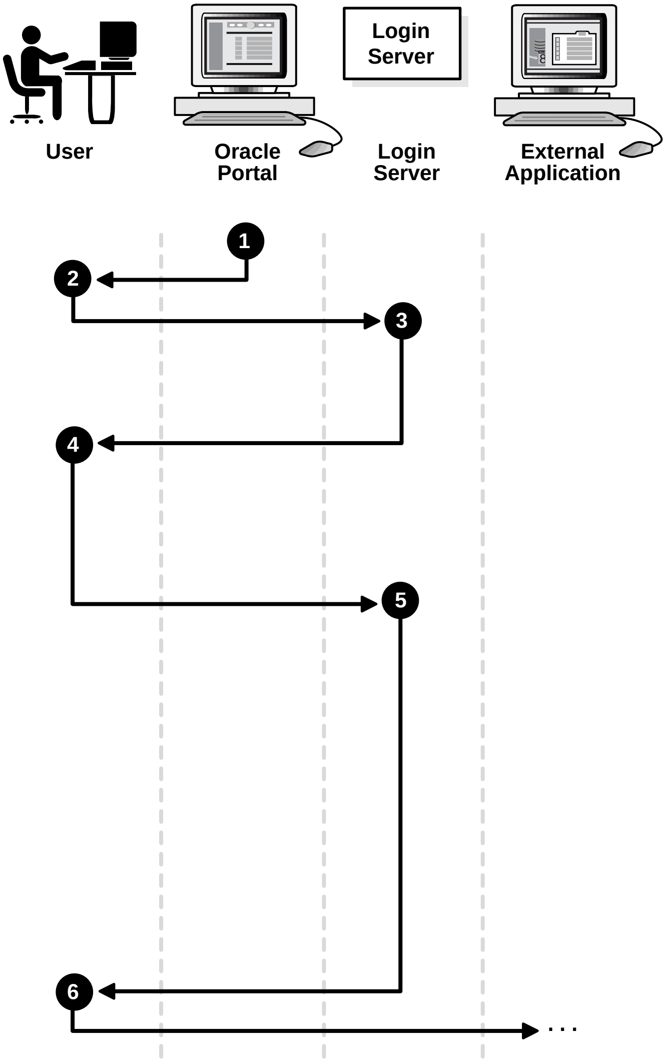
4. The user enters the username and password. The user can also indicate whether to save these credentials in the Login Server password store.

5. If the user chooses to save the credentials in the Login Server password store, the Login Server saves them. The Login Server performs the following tasks:

—Constructs a login form using the user's credentials for submission to the external application login processing routine. This routine has been preconfigured by the Login Server administrator and is associated with the requested application.

—Sends the form to the client browser, with a directive to post it immediately to the external application

6. The client posts the form to the external application and logs in.



Authenticating to an External Application After the First Time

Oracle9*i*AS Single Sign-On uses the process described in Figure 1–5 if the user:

- Has authenticated to the Oracle9*i*AS Portal
- Has a username and password stored in the Login Server password store. (If the username and password are not stored in the Login Server password store, Oracle9*i*AS Single Sign-On follows the process described in "Authenticating to an External Application for the First Time".)
- Is accessing an external application after the first time

Figure 1–5 Authenticating to an External Application After the First Time

1. Oracle9*i*AS Portal presents to the user links to external applications. These links invoke a routine on the Login Server.

2. A user clicks one of the links.

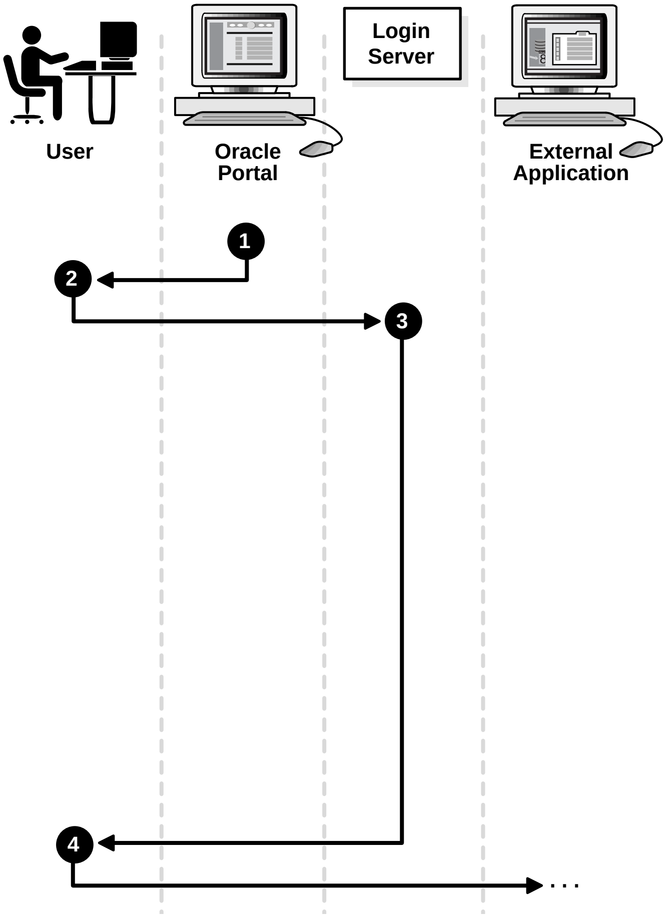
3. Clicking a link begins the external application login procedure. The procedure checks the password store for any credentials the user has for the requested external application.

The Login Server:

—Constructs a login form using the user's credentials for submission to the external application's login processing routine. The routine has been preconfigured by the Login Server administrator and is associated with the requested application.

—Sends the form to the client browser, with a directive to post it immediately to the external application

4. The client posts the form to the external application and logs in.



2

Administrative Basics

This chapter describes the Login Server and Oracle9*i*AS Single Sign-On, and how to you can perform various tasks using them.

This chapter contains the following topics:

- User Accounts
- Administrator Roles
- Granting Login Server Administrator Privileges
- Logging In Using Oracle9iAS Single Sign-On
- Configuring the Login Server

User Accounts

This section contains the following topics:

- Oracle9iAS Single Sign-On User Accounts
- Oracle9iAS Portal User Accounts
- Oracle Database User Accounts

Oracle9*i*AS Single Sign-On User Accounts

An Oracle9*i*AS Single Sign-On user account is used to access multiple applications, including Oracle9*i*AS Portal, with a single username and password. Once you have entered the Oracle9*i*AS Single Sign-On username and password for one application, you can access other applications without having to log in again.

Note: The username ADMIN is reserved for a specific Login Server administrative purpose. See "Troubleshooting" in Chapter 6, "Deployment Considerations" for information about the ADMIN user.

Login Server administrators create Oracle9*i*AS Single Sign-On user accounts using the Users portlet, which by default is on the Administer tab of the Oracle9*i*AS Portal home page.

Note: To access the Administer tab of the Oracle9*i*AS Portal, you must also be a Oracle9*i*AS Portal administrator.

Oracle9*i*AS Portal User Accounts

An Oracle9*i*AS Portal user account establishes user details, preferences, and privileges within Oracle9*i*AS Portal.

Oracle9*i*AS Portal user accounts do not have any privileges on the database itself. However, because Oracle9*i*AS Portal pages are displayed by executing procedures in the database, an Oracle9*i*AS Portal user account must have execute privileges on those procedures. To do this, each Oracle9*i*AS Portal user account must be associated with an Oracle database schema that is authorized to display Oracle9*i*AS Portal pages.

Oracle9*i*AS Portal user accounts are created automatically when one of the following occurs:

- An Oracle9*i*AS Portal administrator first attempts to edit the Oracle9*i*AS Portal user settings of an Oracle9*i*AS Single Sign-On user account
- A user first attempts to log in to Oracle9*i*AS Portal using an Oracle9*i*AS Single Sign-On account

Oracle9*i*AS Portal administrators can edit Oracle9*i*AS Portal user accounts using the Users portlet, which by default is on the Administer tab of the Oracle9*i*AS Portal home page.

See Also: "Creating User Accounts"

Oracle Database User Accounts

An Oracle database user account, which follows rules set by the database's schema, is used to store database objects, applications, and components, and to determine a user's database privileges. Database accounts are required because Oracle9*i*AS Portal and Login Server are implemented using an underlying Oracle database.

Database administrators can create Oracle database schemas using SQL commands or the Schemas portlet on the Administer Database tab of the Oracle9*i*AS Portal home page.

Default Oracle9*i*AS Portal User Accounts and Schemas

When you install Oracle9*i*AS Portal, several user accounts and schemas are created by default, as described in Table 2–1 and Table 2–2.

Warning: For security reasons, change all user account passwords after initial login. By default, the password is set to the username. Change the password by logging on to the Login Server and editing the appropriate user accounts.

Table 2–1 Users Created by Default

User	Description
`PUBLIC`	Account created for public users, for use in unauthenticated sessions.This is the account that all sessions are associated with prior to authentication.

Table 2–1 Users Created by Default

User	Description
SCHEMA	Account created for the Database Administrator (DBA) with the highest privileges in Oracle9*i*AS Portal and the Login Server.
*SCHEMA*_ADMIN	Account created for the Oracle9*i*AS Portal administrator. This account is similar to the *SCHEMA* account, however, it does not have privileges that provide access to database administration features, such as creating and managing schemas and other database objects.
*SCHEMA*_SSO	Account created for the Login Server administrator. Since the Login Server is implemented with significant reuse of Oracle9*i*AS Portal infrastructure code, this user account is created as a result of this reuse.
*SCHEMA*_SSO_ADMIN	Another account created for the Login Server administrator. This user has the same set of privileges as the *SCHEMA*_SSO user.

Table 2–2 Database Schemas Created by Default

Schema	Description
SCHEMA	The schema in which Oracle9*i*AS Portal is installed
*SCHEMA*_PUBLIC	The schema associated with Oracle9*i*AS Portal users by default to execute the procedures that display Oracle9*i*AS Portal pages
*SCHEMA*_DEMO	The schema that contains the demo applications shipped with Oracle9*i*AS Portal
*SCHEMA*_SSO	The schema in which the Login Server is installed
*SCHEMA*_SSO_PUBLIC	The schema used to execute the procedures that display Login Server pages
*SCHEMA*_SSO_PS	The schema used by Oracle9*i*AS Portal to access the Login Server password store

Administrator Roles

This section contains the following topics:

- Oracle9iAS Portal Administrator Role
- Login Server Administrator Role

Oracle9*i*AS Portal Administrator Role

In the Oracle9*i*AS Portal environment, the Oracle9*i*AS Portal administrator role allows access to the Administer tab of the Oracle9*i*AS Portal home page.

If the administrator has also been granted privileges for the Login Server administrator role, the administrator can access the Login Server administration menus.

Login Server Administrator Role

The Login Server administrator role allows access to the Login Server page and its configuration settings. This allows the Login Server administrator to:

- Create and edit any Oracle9*i*AS Single Sign-On user account
- Configure the Login Server (for example, establish password restrictions and logout behavior)
- Administer partner applications (Web-based applications that have delegated authentication processing to the Login Server)
- Administer external applications (Web-based applications that perform their own username and password authentication through HTML forms)

Note: Within the Oracle9*i*AS Portal environment, the database or Oracle9*i*AS Portal administrator is typically a Login Server administrator as well.

See Also: "Creating User Accounts" and "Deleting User Accounts" in Chapter 3, "User Management"

Granting Login Server Administrator Privileges

To grant Login Server administrator privileges, you must be:

- A Login Server administrator to grant Login Server privileges to another user
- An Oracle9*i*AS Portal administrator to access the Administer tab of the Oracle9*i*AS Portal home page

Note: Oracle Corporation recommends that you restrict Login Server administrator privileges to database and Oracle9*i*AS Portal administrators only.

To grant Login Server administrator privileges to a new Oracle9*i*AS Single Sign-On user account, see "Creating User Accounts" in Chapter 3, "User Management".

Perform the following steps to grant Login Server administrator privileges to an existing Oracle9*i*AS Single Sign-On user account.

1. Navigate to the Oracle9*i*AS Portal home page.
2. In the User portlet, select the username of the Oracle9*i*AS Single Sign-On user account that you want to grant administrator privileges to from the provided list.

   By default, the User portlet is located on the Administer tab of the Oracle9*i*AS Portal home page.
3. Click Edit.

   The Edit User page displays.
4. In the Administrator’s password field, enter your password to confirm that you have the authority to change user account information.
5. In the Login Server Privilege Level field, select Full Administrator.

   Note: To revoke Login Server administrator privileges, select End User.
6. Click OK.

   See Also: "Login Server Administrator Role"

Logging In Using Oracle9*i*AS Single Sign-On

The Oracle9*i*AS Single Sign-On login page is used to log in to Oracle9*i*AS Portal.

After a you log in, you can access all of the pages, content areas, and applications available to public users, as well as pages, content areas, and applications that have been created for administrative purposes and made available through the Oracle9*i*AS Portal security mechanism.

Note: If you try to log in with an incorrect password too many times, the account becomes inaccessible for a certain period of time, depending upon the configuration.

See Also: "Administering Passwords" in Chapter 3, "User Management"

Table 2–3 describes the fields in the Oracle9*i*AS Single Sign-On login page.

Table 2–3 Oracle9iAS Single Sign-On Login Page

Field	Description
User Name	Enter your username. Usernames are always case-insensitive; that is, Portal30_admin is the same as PORTAL30_ADMIN.
Password	Enter your password. Depending upon the installation options, passwords are either case-sensitive or case-insensitive.

- Click Login to log in to Oracle9*i*AS Portal.
- Click Cancel to exit the page without logging in.

Configuring the Login Server

This section contains the following topics:

- Edit Login Server Page
- User Lockout
- Login Server Configuration Procedure

Edit Login Server Page

The Edit Login Server page is used to configure the Login Server.

Table 2–4 describes the fields in the Edit Login Server page.

Table 2–4 Edit Login Server Page

Field	Description
Password Policy	

Table 2–4 Edit Login Server Page

Field	Description
Password Life	Enter the number of days that the user's Oracle9*i*AS Single Sign-On password will remain valid. This value determines when a user's password must be changed or reset.
Number of days before password expiration to show warning	Enter how many days before a password expiration warning message displays to users.
Minimum Password Length	Enter the minimum number of characters that users can specify when choosing passwords.
Password Case Sensitivity	Displays whether the passwords are case-sensitive. This setting is established during installation of the Login Server and cannot be modified.
Do not allow password to be the same as user name	Select to prevent users from choosing a password that is the same as their username, such as `janedoe/janedoe`.
Do not allow new password to be the same as current password	Select to prevent users from choosing the same password when renewing an expired password.
Require password to contain at least one numeric digit	Select to require users to include at least one numeric character (0-9) when choosing a password.
Require password to contain at least one character	Select to require users to include at least one alphabetic character (A-Z) when choosing a password.
Account Lock Policy	
Number of login failures allowed from any IP address	Enter the number of failed login attempts that can made from any IP address before the user account is temporarily locked out. The duration of the lockout is set in the Global Lockout Duration field. **See Also:** "User Lockout"
Number of login failures allowed from one IP address	Enter the number of failed login attempts that can made from a single IP address before login attempts from that IP address are temporarily disabled. **See Also:** "User Lockout"
Global lockout duration	Enter the number of days that users are prevented from logging into the Login Server after exceeding the number of login failures allowed from any IP address. **See Also:** "User Lockout"

Table 2–4 Edit Login Server Page

Field	Description
Lockout duration for one IP address	Enter the number of minutes that users are prevented from logging into the Login Server after exceeding the number of login failures allowed from one IP address. **See Also:** "User Lockout"
Single Sign-On session duration	Enter the number of hours a user can be logged into the Login Server without timing out and having to log in again.
Verify IP addresses for requests made to the Login Server	Select to verify that the IP address of the browser is same as the IP address in the authentication request to the Login Server.
Logout Behavior	
Logout closes both the Login Server application and Single Sign-On sessions	Select to end the sessions for the Login Server application as well as the session for Oracle9*i*AS Single Sign-On when the user clicks the Login Server's logout link.
Logout closes only the Login Server application session	Select to end only the Login Server application session, keeping the Oracle9*i*AS Single Sign-On session active.
Authentication Mechanism	Displays the authentication method used by the Login Server
Territory Selection	
Enable Users to Choose Territory	Select to allow users to specify territory, which determines localization settings such as date, currency, and decimal formats, when logging in.

User Lockout

A user lockout occurs when the user submits an invalid username and password combination more times than is permitted by the Login Server. In a lockout situation, the Login Server prevents the user from accessing the Login Server even if the user submits the correct username and password combination because the incorrect combination has been submitted more times than is permitted by the Login Server.

The types of user lockout are:

- IP Lockout
- Global Lockout

IP Lockout

An IP lockout occurs when a user is not permitted to access the Login Server from a single workstation because the user has submitted the incorrect password from that single workstation more times than is permitted by the Login Server.

Global Lockout

A global lockout occurs when a user is not permitted to access the Login Server from any workstation because the user has submitted an incorrect password from more than one workstation for more times than is permitted by the Login Server. A global lockout remains in effect for a longer duration than local lockout because a global lockout is more likely to occur in response to a determined attacker.

Login Server Configuration Procedure

The Login Server allows users to log in to Oracle9*i*AS Portal and to any partner or external application using a single username and password.

To configure the Login Server:

- You must have Full Administrator privileges on the Login Server to change any of its settings.
- You must be an authenticated Oracle9*i*AS Portal administrator to access the Administer tab of the Oracle9*i*AS Portal home page.

Perform the following steps to configure the Login Server:

1. Select the Administer tab on the Oracle9*i*AS Portal home page.
2. In the Login Server Administration portlet, select Edit Login Server Configuration.
3. The Edit Login Server page displays.
4. In the Password Policy section, choose options that set the rules for selecting a valid password.
5. In the Account Lock Policy section, choose options that set the rules for locking out users from the Login Server after unsuccessful login attempts.
6. In the Logout Behavior section, choose whether users log out of both the Login Server and the Oracle9*i*AS Single Sign-On session after clicking the Logout link, or whether they log out of the Login Server only.
7. Click OK.

3

User Management

This chapter describes how to administer Oracle9*i*AS Single Sign-On user accounts and passwords.

This chapter contains the following topics:

- Usernames and Passwords
- Creating User Accounts
- Editing User Accounts
- Deleting User Accounts
- Administering Passwords
- Exporting and Importing User Accounts

Usernames and Passwords

Rules for specifying usernames are as follows:

- The username must be unique and is limited to 50 characters; short names are easier to remember
- The following characters are not allowed in the username: tilde (~), semi-colon (;), and single quote (’).
- Usernames are not case-sensitive. For example, the Login Server evaluates the following values identically: SCOTT, Scott, and scott.
- Usernames can be created in multibyte character sets.

Rules for specifying passwords are as follows:

- You can establish restrictions on what can be used as a password. For example, you can restrict passwords to a minimum number of characters, or to include at least one numeric character.
- Passwords should be easy to remember but not obvious to others.
- New users should change their password the first time they log in.

Creating User Accounts

In order to log in to Oracle9*i*AS Portal and access non-public information and features, a user must have both a Oracle9*i*AS Single Sign-On user account and an Oracle9*i*AS Portal user account.

You only have to create a user’s Oracle9*i*AS Single Sign-On user account, because an Oracle9*i*AS Portal account is automatically created when you first edit a user's Oracle9*i*AS Portal settings, or when the user first logs in to Oracle9*i*AS Portal using the Oracle9*i*AS Single Sign-On account.

To create an Oracle9*i*AS Single Sign-On user account:

- You must be a Login Server administrator.
- You must be an Oracle9*i*AS Portal administrator to access the Administer tab of the Oracle9*i*AS Portal home page.

The Create User page is used to create a Oracle9*i*AS Single Sign-On user account for a new user.

Table 3–1 describes the fields on the Create User page.

Table 3–1 Create User Page

Fields	Description
User Details	
User Name	Enter a username for the account. **See Also:** "Usernames and Passwords" for rules for specifying usernames
Password	Enter a password for the account. The user uses this password to confirm that the user is authorized to log in using the account. **See Also:** "Usernames and Passwords" for rules for specifying passwords You can establish restrictions on what can be used as a password. For example you can restrict passwords to contain a minimum number of characters or to include at least one numeric character. **See Also:** "Configuring the Login Server" in Chapter 2, "Administrative Basics" for information about establishing password restrictions You should advise new users to change their password the first time they log in.
Confirm Password	Enter the password again to confirm that you entered it correctly in the Password field.
E-mail Address	Enter the user's e-mail address.
Account Activation and Termination	
Activate Account On	Enter the date when the user can start using the account. Use the format specified to the right of the field.
Terminate Account On	Enter the date when the user will no longer be able to use the account. Use the format specified to the right of the field. **Note:** If you want the account to be available indefinitely, leave this field blank.
Login Server Privileges	
Login Server Privilege Level	Select which privileges to grant the user on the Login Server: End User: No administrative privileges Full Administrator: Login Server administrator privileges

Perform the following steps to create a Oracle9*i*AS Single Sign-On user account.

1. Navigate to the Oracle9*i*AS Portal home page.
2. In the User portlet, click Create New Users.

   By default, the User portlet is located on the Administer tab of the Oracle9*i*AS Portal home page.

   The Create User page displays.
3. In the User Name field, enter a unique username for the account.

   See Also: "Usernames and Passwords" for information about username specification rules
4. In the Password field, enter a password for the account.

   See Also: "Usernames and Passwords" for information about password specification rules
5. Enter the same password in the Confirm Password field to confirm that you entered it correctly.
6. Optionally, in the E-Mail Address field, enter the user's e-mail address.
7. In the Activate Account On field, enter the date when the user can start logging on using the account. Use the format specified to the right of the field.
8. Optionally, in the Terminate Account On field, enter the date when the user will no longer be able to log in using the account. Use the format specified to the right of the field.

   Note: For the account to be available indefinitely, leave the Terminate Account On field blank.
9. In the Login Server Privileges list, select which privileges to grant the user on the Login Server, Full Administrator or End User.
10. Click Create.

Note: When you click Create, a link is displayed at the top of the page that you can click to edit the Oracle9*i*AS Single Sign-On user account. You can also create additional user accounts, or click Close to return to exit the Create User page

11. Click Close.

See Also:

"Editing User Accounts" for information about editing Oracle9*i*AS Single Sign-On user accounts

"Deleting User Accounts" for information about deleting Oracle9*i*AS Single Sign-On user accounts

Editing User Accounts

The Edit User page is used to specify the properties of Oracle9*i*AS Single Sign-On user accounts, such as passwords, account termination dates, and Login Server privileges.

Table 3–2 describes the fields in the Edit User page.

Note: You can also use this page to delete a user account.

Table 3–2 Edit User Page

Field	Description
User Details	
User Name	Edit the account's username. **See Also**: "Usernames and Passwords" for information about specifying usernames.
Administrator's Password	Enter your password to confirm that you have the authority to reset user account passwords. **Note:** You only need to enter your password if you are resetting a user's password.

Table 3–2 Edit User Page

Field	Description
Password	Enter a new password for the account. The user uses this password to confirm that he or she is authorized to log in using the account. **See Also**: "Usernames and Passwords" for information about specifying passwords
Confirm Password	Enter the new password again to confirm that you entered it correctly in the Password field.
E-mail Address	Enter the user's e-mail address.
Account Activation and Termination	
Activate Account On	Edit the date when the user can start using the account. Use the format specified to the right of the field.
Terminate Account On	Edit the date when the user will no longer be able to use the account. Use the format specified to the right of the field. **Note:** If you want the account to be available indefinitely, leave this field blank.
Login Server Privileges	
Login Server Privilege Level	Select which privileges to grant the user on the Login Server. End User: The user has no administrative privileges on the Login Server Full Administrator: The user is a Login Server administrator and has full administrative privileges on the Login Server.

Perform the following steps to edit an Oracle9*i*AS Single Sign-On user account.

1. Navigate to the Oracle9*i*AS Portal home page.
2. In the User portlet, select the username of the Oracle9*i*AS Single Sign-On user account that you want to edit from the provided list.

   By default, the User portlet is located on the Administer tab of the Oracle9*i*AS Portal home page.
3. Click Edit.

   The Edit User page displays.
4. Edit the appropriate fields, as described in Table 3–2.

5. Click Close.

See Also:

- "Administering Passwords"
- "Exporting and Importing User Accounts"
- "Administrator Roles"
- "Deleting User Accounts"
- Oracle9*i*AS Portal online documentation for information about editing Oracle9*i*AS Portal user accounts

Deleting User Accounts

This section describes how to delete Oracle9*i*AS Single Sign-On user accounts.

When you delete a user's Oracle9*i*AS Single Sign-On account, the user can no longer log in to applications through Oracle9*i*AS Portal. The user can still log in to external applications but must use the username and password for that particular external application.

Deleting an Oracle9*i*AS Portal user account does not delete the corresponding Oracle9*i*AS Single Sign-On user account. The user can therefore still log in to other applications using the Oracle9*i*AS Single Sign-On user account. Also, if the user attempts to log in to Oracle9*i*AS Portal using the Oracle9*i*AS Single Sign-On user account, a new Oracle9*i*AS Portal user account is automatically created. To prevent a user from logging in to Oracle9*i*AS Portal, ensure that the user is not an authorized Oracle9*i*AS Single Sign-On user.

To delete a user account:

- You must be an Oracle9*i*AS Portal administrator to access the Administer tab of the Oracle9*i*AS Portal home page and delete an Oracle9*i*AS Portal user account.
- You must be a Login Server administrator to delete a Oracle9*i*AS Single Sign-On user account.

See Also:

"Creating User Accounts" for information about creating Oracle9*i*AS Single Sign-On user accounts

"Editing User Accounts" for information about editing Oracle9*i*AS Single Sign-On user accounts

Oracle9*i*AS Portal online documentation for information about deleting Oracle9*i*AS Portal user accounts

Perform the following steps to delete a Oracle9*i*AS Single Sign-On user account using Oracle9*i*AS Portal:

1. Navigate to the Oracle9*i*AS Portal home page.
2. In the User portlet, enter the username of the user account that you want to delete in the Name field or select it from the provided list.

   By default, the User portlet is located on the Administer tab of the Oracle9*i*AS Portal home page.
3. Click Delete.
4. A confirmation dialog is displayed. Click Yes.
5. You should also delete the user's Oracle9*i*AS Portal user account. If you do not delete the account, and you create another Oracle9*i*AS Single Sign-On with the same username, the user will automatically have the same Oracle9*i*AS Portal privileges as the old user account.

Exporting and Importing User Accounts

You can export Oracle9*i*AS Single Sign-On user accounts from a source Login Server to a target Login Server using the following scripts provided with Oracle9*i*AS Portal:

- For UNIX:

  `ssoexp.csh` and `ssoimp.csh`

- For Windows NT:

  `ssoexp.cmd` and `ssoimp.cmd`

Before you can import applications or content areas into an instance of Oracle9*i*AS Portal, you must first import the Oracle9*i*AS Single Sign-On user accounts used by those applications and content areas to the Login Server used by that instance of Oracle9*i*AS Portal.

Exporting User Accounts

Perform the following steps to export Oracle9*i*AS Single Sign-On user accounts.

1. Start a command line prompt.
2. Change to the `src/wwu` directory of the directory in which Oracle9*i*AS Portal is installed.
3. For UNIX systems, enter the following:

   ```
   ssoexp.csh -s sso_schema [-p sso_password] [-d dump_file_name] [-c connect_
   string]
   ```

   For Windows NT systems, enter the following:

   ```
   ssoexp.cmd -s sso_schema [-p sso_password] [-d dump_file_name] [-c connect_
   string]
   ```

   where:

sso_schema	is the database schema in which the source Login Server is installed. Example: PORTAL30_SSO **Note**: You must provide a value for this parameter.
sso_password	is the password for the above schema. The default filename is *sso_schema*
dump_file_name	is the file name you want to give the dump file created by the export script. The default filename is *sso.dmp*
connect_string	is the connect string for the database in which the source Login Server is installed. You must provide the connect string only if you are performing the export from a different database.

   Example:

   ```
   ssoexp.csh -s portal30_sso -p portal30_sso -d export_sso.dmp -c orcl
   ```

4. Press Enter or Return.

A dump file with the filename you specified is created that contains all of the required data for the Oracle9*i*AS Single Sign-On user accounts in the source Login Server.

You can now use the dump file to import Oracle9*i*AS Single Sign-On user accounts into the target Login Server.

Importing User Accounts

Perform the following steps to import Oracle9*i*AS Single Sign-On user accounts.

1. Ensure that the Oracle9*i*AS Single Sign-On user accounts have been exported, and that the dump file is located in the `src/wwu` directory of the directory in which Oracle9*i*AS Portal is installed.
2. Start a command line prompt.
3. Change to the `src/wwu` directory.
4. For UNIX systems, enter the following:

```
ssoimp.csh -s sso_schema [-p sso_password] [-o
from_sso_schema] [-d dump_file_name] [-m merge_mode] [-u
db_user_mode] [-c connect_string]
```

For Windows NT systems, enter the following:

```
ssoimp.cmd -s sso_schema [-p sso_password] [-o
from_sso_schema] [-d dump_file_name] [-m merge_mode] [-u
db_user_mode] [-c connect_string]
```

where:

sso_schema is the database schema in which the target Login Server is installed.

Example: `PORTAL30_SSO`

Note: You must provide a value for this parameter.

sso_password is the password for the above schema

The default filename is *sso_schema*

from_sso_schema is the database schema in which the source Login Server is installed.

The default filename is *sso_schema*

dump_file_name is the name of the dump file you want to use to import Oracle9*i*AS Single Sign-On user accounts.

The default filename is `sso.dmp`

merge_mode is the mode used to determine what happens if an Oracle9*i*AS Single Sign-On user account with the same username already exists on the target Login Server.

`reuse` mode:

> If an Oracle9*i*AS Single Sign-On user account with the same username already exists in the target Login Server, keep the existing user

`check` mode:

> Does not actually import any Oracle9*i*AS Single Sign-On user accounts, but produces a list of duplicate usernames and their roles, so that you can decide what to do about duplications before performing the import

db_user_mode is the mode used to determine which database schema to use for Oracle9*i*AS Single Sign-On user accounts.

`public-user` mode:

> Resets every Oracle9*i*AS Single Sign-On user account to use the Oracle9*i*AS Portal public schema.

`database-user` mode:

> Uses the database schema specified for each Oracle9*i*AS Single Sign-On user account if it exists in the target database. If the specified database schema does not exist in the target schema, it is reset to the Oracle9*i*AS Portal public schema.

The default filename is `database_user`

connect_string is the connect string for the database in which the target Login Server is installed. You must provide the connect string only if you are performing the import from a different database.

Example:

```
ssoimp.csh -s newportal30_sso -p newportal30_sso -o portal30_sso -d export_
sso.dmp -m reuse -u public_user -c orcl
```

5. Press Enter or Return.

The passwords of all the Oracle9*i*AS Single Sign-On user accounts imported into the target Login Server are reset to the username of the account. You should advise users to change their passwords as soon as possible after the import.

Warning: Advise users to change their passwords immediately after the import.

Note: To create a log file for the export or import scripts, redirect the screen output to a file, as in the following example:

```
ssoexp.csh -s portal30_sso -p portal30_sso -d export_sso.dmp
-c orcl | tee export.log
```

Administering Passwords

For security purposes, the Login Server administrator specifies password expiration dates. Passwords must also be reset immediately if they are compromised or forgotten.

Changing a password in the Login Server affects access to all of the Oracle9*i*AS Single Sign-On applications, not just Oracle9*i*AS Portal. If a user's password is not changed before its expiration date, the user cannot log in until the Login Server administrator resets it for the user.

To administer passwords:

- You must be a Login Server administrator to reset a user's password.
- You must be an Oracle9*i*AS Portal administrator to access the Administer tab of the Oracle9*i*AS Portal home page.

This section contains the following topics:

- Change Password Page
- Resetting the Administrator's Password
- Resetting User Passwords
- Installing the Password Reset Feature
- Reset Password Page Example
- WWSSO_ALERT Package Body Example

Change Password Page

The Change Password page is used to change passwords.

Table 3–3 describes the fields in the Change Password page.

Table 3–3 Change Password Page

Field	Description
User Name	Displays the username.
Old Password	Enter the password that you currently use to log in.
New Password	Enter a new password. **See Also:** "Usernames and Passwords" for information about specifying passwords
Confirm New Password	Enter the new password again to confirm that you entered it correctly in the New Password field.

Resetting the Administrator's Password

Perform the following steps to change the Login Server administrator password.

1. In the top right corner of your home page, click Account Info.

   The Edit Account Information page displays.

2. In the top right corner of the Edit Account Information page, click Change Password.

   The Change Password page displays.

3. In the Old Password field, enter the password that you currently use to log in.

4. In the New Password field, enter the new password.

   See Also: "Usernames and Passwords" for information about specifying passwords

5. Enter the same password in the Confirm New Password field to confirm that you entered it correctly.

6. Click OK to return to the Edit Account Information page.

7. Click OK to return to your home page.

   The next time you log in, use the new password.

Resetting User Passwords

Perform the following steps to reset a user's password.

1. Navigate to the Oracle9*i*AS Portal home page.
2. In the User portlet, select the username of the user account for which you want to reset the password from the provided list.

   By default, the User portlet is located on the Administer tab of the Oracle9*i*AS Portal home page.
3. Click Edit.
4. In the Administrator's Password field, enter your password to confirm that you have the authority to reset user account passwords.
5. In the Password field, enter the new password for the user.
6. Enter the same password in the Confirm Password field to verify that you entered it correctly.

   You should advise new users to change their password the first time they log in.

   Note: You can establish restrictions on what can be used as a password. For example, you can restrict passwords to a minimum number of characters or to include at least one numeric character.

   See Also: "Configuring the Login Server" in Chapter 2, "Administrative Basics", for information about establishing restrictions on passwords
7. Click OK.

Installing the Password Reset Feature

Sometimes users forget their passwords and must have them reset. The Login Server offers a feature that resets a user's password to a random value and then notifies the user of the new password.

This feature can present a security risk, because the user is not authenticated when requesting a reset password for a particular user account. For this reason, the password reset feature is not enabled by default and must be installed.

Perform the following steps to install the password reset feature.

1. On the database where the Login Server is installed, log in to SQL*Plus as the Login Server schema, as in the following example:

   ```
   sqlplus portal30_sso/portal30_sso
   ```

2. Enter the following:

   ```
   @ssoreset
   ```

   The `ssoreset` script creates the `WWSSO_APP_ACCOUNT` package in the Login Server schema and grants execute privileges on the `WWSSO_APP_ACCOUNT` package to `PUBLIC.`

   `WWSSO_APP_ACCOUNT` contains a single procedure, `reset_password`, that resets a password to a random value.

3. After resetting the password, the `reset_password` procedure calls the `WWSSO_ALERT.password_reset_notification` procedure.

   The `WWSSO_ALERT.password_reset_notification` procedure informs the user of the new password. If you do not enable the password reset feature, the `WWSSO_ALERT.password_reset_notification` procedure does not inform the user of the change.

   > **Note:** By default, implementation of the `WWSSO_ALERT` package body, created during the installation of the Login Server, does not alert the user when the password is reset. You must replace the `WWSSO_ALERT` package body with an implementation that sends the user the new password; for example, through e-mail, using `UTL_SMTP` or workflow. If you do not replace the `WWSSO_ALERT` package body, the password is reset to an unknown value, and the user still cannot log in.

   The `WWSSO_ALERT` package specification is as follows:

   ```
   CREATE OR REPLACE PACKAGE wwsso_alert
   IS
     /* General failure exception. This will be used
     * by the UI to alert the user that the notification
     * failed
     */
     NOTIFICATION_FAILURE EXCEPTION;
     PROCEDURE password_reset_notification
     (
   ```

```
  p_user VARCHAR2,
  p_password VARCHAR2,
  p_email VARCHAR2 DEFAULT NULL
  );
END wwsso_alert;
```

See Also: "WWSSO_ALERT Package Body Example" for an example of a package body that sends the newly assigned password through e-mail.

4. Create a page that calls the `reset_password` procedure to allows users to reset their passwords.

Reset Password Page Example

The following is an example of how to design a page for resetting a user's password.

```
<HTML>
 <HEAD>
 <TITLE="Reset password">
 </HEAD>
 <BODY>
  <H1>Reset password</H1>
  <FORM ACTION="http://server.domain[:port]/pls/dad/
  schema.WWSSO_APP_ACCOUNT.RESET_PASSWORD">
  <B>User Name: </B>
  <INPUT TYPE="TEXT" NAME="p_user">
  <BR><BR>
  <INPUT TYPE="HIDDEN" NAME="p_back_url"
  VALUE="http://server.domain[:port]/pls/dad/schema.home">
  <INPUT TYPE="HIDDEN" NAME="p_error_url"
  VALUE="http://server.domain[:port]/pls/dad/schema.error">
  <INPUT TYPE="SUBMIT" VALUE="Reset Password">
  <FORM>
 </BODY>
</HTML>
```

Note: After the password for a username is reset using the reset_password procedure, the page must pass at least a username (p_user) and the URL of a page to which to return (p_back_url). The page may also pass the URL of a page to display if any errors are encountered (p_error_url).

See Also: "Installing the Password Reset Feature"

WWSSO_ALERT Package Body Example

The following is an example of how you might implement the WWSSO_ALERT package body for informing a user of the new password after resetting it.

```
set define ON
set verify OFF

CREATE or REPLACE PACKAGE BODY wwsso_alert
IS

   PROCEDURE send_mail
   (
    p_sender IN VARCHAR2,
     p_recipient IN VARCHAR2,
     p_message IN VARCHAR2
   )
   IS
     mailhost VARCHAR2(80) := '&smtp_server';
     mail_conn utl_smtp.connection;
   BEGIN
     mail_conn := utl_smtp.open_connection(mailhost, 25);
     utl_smtp.helo(mail_conn, mailhost);
     utl_smtp.mail(mail_conn, p_sender);
     utl_smtp.rcpt(mail_conn, p_recipient);
     utl_smtp.data(mail_conn, p_message);
     utl_smtp.quit(mail_conn);
   END;

   PROCEDURE password_reset_notification
  (
     p_user VARCHAR2,
     p_password VARCHAR2,
     p_email VARCHAR2 DEFAULT NULL
   )
   IS
   BEGIN
     send_mail
     (
       p_sender => '&password_administrator',
       p_recipient => p_email,
       p_message => p_user || 'Your new password is ' || p_password
     );
   EXCEPTION
     when OTHERS then
     raise NOTIFICATION_FAILURE;
   END;

END wwsso_alert;
 /

show errors PACKAGE BODY wwsso_alert
```

4

Application Management

This chapter explains how to use Oracle9*i*AS Single Sign-On to log in to partner and external applications.

This chapter contains the following topics:

- Administering Partner Applications
- Administering External Applications
- Adding Partner and External Applications
- Editing Partner and External Applications
- Adding External Applications to the External Applications Portlet
- Specifying External Application Login Information

Administering Partner Applications

The Administer Partner Applications page is used to add, edit, or delete a partner application. Partner applications delegate authentication services to the Login Server. The user logs in to a partner application by providing the username and password for the Login Server when required. The Oracle9*i*AS Single Sign-On feature of the Login Server ensures that, regardless of how many partner applications are accessed through the Login Server, the user has to provide a username and password only once to the Login Server for that Login Server session.

Table 4–1 described the fields in the Administer Partner Applications page.

Table 4–1 Administer Partner Applications Page

Field	Description
Add Partner Application	Click this link to create a new partner application.
Edit/Delete Partner Applications	Displays existing partner applications. You can: ■ Click the Edit link next to a partner application to view or edit it. ■ Click the Delete link next to a partner application to delete it. ■ Click an application name to go to the application’s home URL

Administering External Applications

The Administer External Applications page is used to add, edit, or delete an external application. External applications are Web-based applications that perform their own username and password authentication through HTML login forms. External applications that use HTML login forms include Yahoo! Mail and Oracle Mobile.

After you add an external application to the Login Server, users can provide their username and password for the application to the Login Server's password store. To store a password, the user selects the Remember My Login Information For This Application checkbox when logging into the external application through the Login Server. Once the username and password are stored, the user is not prompted for them when logging into the application.

Table 4–2 describes the fields in the External Applications page.

Table 4–2 External Applications Page

Field	Description
Add External Application	Click this link to create a new external application.
Edit/Delete External Applications	Displays existing external applications. You can: ■ Click the Edit link next to an external application to view or edit it. ■ Click the Delete link next to an external application to delete it. ■ Click an application name to test the login

Adding Partner and External Applications

Partner applications delegate authentication services to the Login Server. A user logs into a partner application using the same username and password that was used for the Login Server. After a user is logged on to the Login Server, the partner application does not request further authentication from the user.

External applications, by comparison, are not fully integrated into Oracle9*i*AS Portal. Instead, they perform their own authentication. When they are registered with the Login Server, they are visible in Oracle9*i*AS Portal as links in the External Applications portlet. Adding an external application enables the Login Server to submit an HTML login form with the user's credentials to the application.

To add a partner or external application:

- You must have Full Administrator privileges on the Login Server to change any of its settings.
- You must be an Oracle9*i*AS Portal administrator to access the Administer tab of the Oracle9*i*AS Portal home page.

This section contains the following topics:

- Adding a Partner Application
- Adding an External Application

Adding a Partner Application

Partner applications are added in the Create Partner Application page.

Table 4–3 describes the fields in the Create Partner Application page.

Table 4–3 Create Partner Application Page

Field	Description
Partner Application Login	
Name	Enter a unique name for the partner application.
Home URL	Enter the URL of the application’s home page.
Success URL	Enter the URL to the routine responsible for establishing the partner application's session and session cookies. This routine should redirect the browser to the URL that the user originally requested. The URL must point to a procedure that processes the user identification information from the Login Server. Include the `http://` prefix in the URL, as in the following example: `http://server.domain.com:5000/pls/DAD/portal.wwsec_app_priv.process_signon`
Valid Login Timeframes	
Start Date	Enter the date when users will first be able to access the partner application through the Login Server. Use the format shown next to the field label.
End Date	Enter the end date when users will last be able to access the partner application through the Login Server. Use the format shown next to the field label. **Note:** If you leave this field blank, users can log into the partner application using the Login Server indefinitely.
Application Administrator	
Administrator E-mail	Enter the e-mail address for the administrator responsible for this partner application.
Administrator Information	Enter any additional information you want to include about the administrator responsible for this partner application.

Perform the following steps to add a partner application using Oracle9*i*AS Portal.

1. Navigate to the Oracle9*i*AS Portal home page.
2. Click the Administer tab.
3. In the Login Server Administration portlet, click Administer Partner Applications.

4. Click Add Partner Application.

   The Create Partner Application page displays.

5. In the Partner Application Login section, enter the partner application's name, the URL to the application's home page, and a Success URL. The Success URL points to a Web page where the browser should be redirected after a successful login. It must correspond to the procedure that processes the user identification information from the Login Server.

6. In the Valid Login Timeframe section, enter the dates when users can log in to the application through the Login Server. If you leave the End Date field blank, users can log into the application indefinitely.

7. In the Application Administrator section, enter the e-mail address and other information for the application's contact person or administrator.

8. Click OK. The new partner application appears in the Edit/Delete Partner Application list on the Partner Application page.

Adding an External Application

External applications are added in the Create External Application page.

Table 4–4 describes the fields in the Create External Application page.

Table 4–4 Create External Application Page

Field	Description
External Application Login	
Application Name	Enter a name that you want to use to identify the external application. This is the default name for the external application.
Login URL	Enter the URL to which the external application credentials are submitted for authentication. For example, the login URL for Yahoo!Mail is: `http://login.yahoo.com/config/login?6p4f5s403j3h0`
Username/ID Field Name	Enter the name that identifies the username or user ID field of the external application's login form. You can find this name by viewing the HTML source for the login form. **Note**: This field is not applicable if you are using Basic authentication.

Table 4–4 Create External Application Page

Field	Description
Password Field Name	Enter the name that identifies the password field of the external application's login form. You can find this name by viewing the HTML source for the login form. **Note**: This field is not applicable if you are using Basic authentication.
Authentication Method	
Type of Authentication Used	Select the type of credential submission method to use for the external application. This specifies how message data is sent by the browser. POST: Submits the login credentials to the login URL of the external application using the HTTP POST method GET: Submits the login credentials to the login URL of the external application using the HTTP GET method BASIC AUTHENTICATION: Submits the login credentials in the application URL, which is protected by HTTP Basic Authentication
Additional Fields	
Field Name	Enter the name of any additional fields on the external application HTML login form that may require user input in order to log into the application **Note**: This field is not applicable if you are using Basic authentication.
Field Value	Enter a default value for a corresponding Field Name value, if applicable. **Note**: This field is not applicable if you are using Basic authentication.

Warning: If the external application uses the Basic authentication method, the username and password may be stored in clear text in the browser cache and browser URL history.

Perform the following steps to add an external application from Oracle9*i*AS Portal.

1. Navigate to the Oracle9*i*AS Portal home page.

2. Click the Administer tab.
3. In the Login Server Administration portlet, click Administer External Applications.
4. Click Add External Application.

   The Create External Application page displays.
5. In the External Application Login section, enter the name of the external application and the URL to which the application's HTML login form is submitted or the protected URL to access if you are using Basic authentication.
6. If the application uses HTTP POST or HTTP GET authentication, in the User Name/ID Field Name, enter the name that identifies the username or user ID field of the external application's HTML login form. You can find the name by viewing the HTML source for the external application's login form.

   If the application uses the Basic authentication method, the User Name/ID Field Name should be empty.
7. If the application uses HTTP POST or HTTP GET authentication, in the Password Field Name, enter the name that identifies the password field of the external application. You must view the HTML source for the login form for this information as well.

   If the application uses the Basic authentication method, the Password Field Name should be empty.
8. In the Additional Fields section, enter the name and default values for any additional fields on the external application HTML login form that may require user input to log into the application.

   If the application uses the Basic authentication method, these fields should be empty.
9. Select the related Display to User checkbox to allow the default value of an Additional Field to be changed by the user on the external application HTML login form.
10. Click OK. The new external application appears in the Edit/Delete External Application list on the External Application page.
11. Optionally, in the Edit/Delete Partner Application list, click an application name to test the log in.

The following example shows the source for the values that are used for the External Application Login section for Yahoo! Mail.

```
<form method=post action="http://login.yahoo.com/config/login?6p4f5s403j3h0" autocomplete=off name=a>
...
<td><input name=login size=20 maxlength=32></td>
....
<td><input name=passwd type=password size=20 maxlength=32></td>
...
<input type=checkbox name=".persistent" value="Y" >Remember my ID & password
...
</form>
```

The source provides values for the following:

- Login URL: `http://login.yahoo.com/config/login?6p4f5s403j3h0`
- Username/ID Field Name: `login`
- Password Field Name: `passwd`
- Type of Authentication Used: `POST`
- Field Name: `.persistent Y`
- Field Value: `[off]`

Editing Partner and External Applications

This section contains the following topics:

- Editing a Partner Application
- Editing an External Application

Editing a Partner Application

The Edit Partner Application page is used to edit configuration information for external applications.

The Edit Partner Application page contains all of the fields that are in the Create Partner Application page, plus three additional display fields in the Partner Application Login section. The additional display fields are described in Table 4–5.

Table 4–5 Display Fields in the Edit Partner Application Page

Field	Description
ID	Displays only when you are editing a partner application. The ID value is automatically set when a partner application is added. It is used by the Login Server to identify the partner application.

Table 4–5 Display Fields in the Edit Partner Application Page

Field	Description
Token	Displays only when you are editing a partner application. The token is automatically set when a partner application is added. It is used by the Login Server to identify the partner application. The partner application must use the application token to identify itself to the Login Server when requesting authentication.
Encryption Key	Displays only when you are editing a partner application. The encryption key is automatically set when a partner application is added. When a user tries to log in using Oracle9*i*AS Single Sign-On, the Login Server generates a cookie that indicates a user's identity and whether the user has been authenticated. This key is used to encrypt the login cookie.

Perform the following steps to edit a partner application from Oracle9*i*AS Portal.

1. Click the home link to navigate to the Oracle9*i*AS Portal home page.
2. Click the Administer tab.
3. In the Services portlet, click Login Server Administration.
4. Click Administer Partner Applications.
5. Click the Edit link next to the Application Name.

   The Edit Partner Application page displays.
6. Edit the appropriate field values, as described in Table 4–3.
7. Click Apply to store changes for the current screen and redisplay the screen with updated values, or click OK to store all changes and return to the previous screen.

Editing an External Application

The Edit External Application page is used to edit configuration information for external applications.

Perform the following steps to edit an external application from Oracle9*i*AS Portal.

1. Click the home link to navigate to the Oracle9*i*AS Portal home page.
2. Click the Administer tab.

3. In the Services portlet, click Login Server Administration.
4. Click Administer External Applications.
5. Click Edit link next to the Application Name.

   The Edit External Application page displays.
6. Edit the appropriate field values, as described in Table 4–4.
7. Click Apply to store changes for the current screen and redisplay the screen with updated values, or click OK to store all changes and return to the previous screen.

Adding External Applications to the External Applications Portlet

The Edit External Applications Portlet Settings page is used to add external applications to the External Applications portlet.

Table 4–6 describes the fields in the Edit External Applications Portlet Settings page.

Table 4–6 Edit External Applications Portlet Settings Page

Field	Description
External Applications Portlet Banner	
Banner	Enter a new name to customize the title of the External Applications portlet banner
Select External Applications	
Display	Select this checkbox to display the application name in the External Applications portlet
Change Stored Password	Click this icon to display the Edit External Applications Login Information page to change your username, password, or additional field information, as applicable.
Application Name	Displays the name of the external application.
Preferred Name	Enter a name for the application for display in the External Applications portlet (optional)

Perform the following steps to access the Edit External Applications Portlet Settings page.

1. In the external applications portlet banner, click Customize.

By default, the external applications portlet is located on the Administer tab of the Oracle9*i*AS Portal home page and is called External Applications

The Edit External Applications Portlet Settings page displays.

2. Change the values as described in Table 4–6.
3. Click OK to save changes, or click Reset to Defaults to revert to the original values.

Specifying External Application Login Information

The External Application Login page is used to specify your username and password for the application. If you select the Remember My Login Information For This Application checkbox, Oracle9*i*AS Portal automatically logs on for you each time you launch the application.

> **Note:** If you change your password in the external application, be sure to update your password on this page as well. Otherwise, Oracle9*i*AS Portal cannot log in for you and the external application's error message is displayed.

Table 4–7 described the fields in the External Application Login page.

Table 4–7 External Application Login Page

Name	Description
Application Name	Displays the name of the application you are logging on to.
User Name/ID	Enter your username for this application.
Password	Enter your password for this application.
Remember My Login Information For This Application	Select to keep this information permanently within Oracle9*i*AS Portal. The next time you launch the application, Oracle9*i*AS Portal automatically logs you in without displaying a login screen.

Perform the following steps to access the External Application Login page.

1. In the External Applications portlet, click the name of the external application.

Note: By default, the external application portlet is located on the Administer tab of the Oracle9*i*AS Portal home page, and the banner name is External Applications.

Note: If the name of the external application is not displayed in the External Application portlet, it must be added using the Edit External Applications Portlet Settings page.

See Also: "Adding External Applications to the External Applications Portlet" for information about adding an external application to the external Applications portlet.

The External Application Login page displays.

2. Enter your username and password.
3. Click Login to log in to the application or Close to cancel.

5

Customizing Single Sign-On

This section explains how to customize the Login and Change Password pages to match the look and feel of your portal or product. It also describes how to configure the Login Server for LDAP authentication.

This section contains the following topics:

- Customizing the Login and Change Password Pages
- Configuring the Login Server for LDAP Authentication

Customizing the Login and Change Password Pages

The Single Sign-On login page and Change Password page can be customized to match the look and feel of your portal or product.

Any type of Web page can be customized: PL/SQL stored procedures, CGI scripts, or Java Server Pages. Although there is no restriction on the type of page, customized pages must support certain parameters and error codes in order to function properly.

This section contains the following topics:

- Customizing the Single Sign-On Login Page
- Customizing the Change Password Page
- Installing Customized Login and Change Password Pages

Customizing the Single Sign-On Login Page

This section contains the following sections:

- Changing the Login Page
- Login Page Parameters
- Login Page Error Codes.

Changing the Login Page

The authentication model contains logic that calls a PL/SQL stored procedure to create the Single Sign-On login page. The Login Server recognizes the user's request to bring up the login screen and makes a PL/SQL call to create the screen. The screen submits a form to the appropriate Login Server routine, which is the same as the original calling routine, and the user's credentials are processed.

The process is as follows:

1. The application calls `WWSSO_APP_ADMIN.LS_LOGIN` to authenticate the user.
2. If the user does not yet have an Oracle9*i*AS Single Sign-On session, `LS_LOGIN` calls `WWSSO_LOGIN.DRAW_LOGIN_PAGE` to display the standard login page.
3. `DRAW_LOGIN_PAGE` submits a form to `WWSSO_APP_ADMIN.LS_LOGIN` to process the credentials.
4. If the user is authenticated, `LS_LOGIN` redirects to the application's success URL, which then redirects to the requested application page.

The customized solution provides the option of redirecting to a separate URL to create the login page, instead of making a PL/SQL call to WWSSO_LOGIN.DRAW_LOGIN_PAGE. The URL can point to a Java Server Page, a CGI script, or other type of page. The page should process the name of the routine to submit the login form to, such as WWSSO_APP_ADMIN.LS_LOGIN, and submit the form appropriately.

The flow of logic is as follows:

1. The Oracle9*i*AS Portal calls WWSSO_APP_ADMIN.LS_LOGIN to authenticate the user.
2. If a URL is specified in the LOGIN_URL column for displaying the login page, LS_LOGIN redirects to that URL.

   If a URL is not specified in the LOGIN_URL column, LS_LOGIN calls WWSSO_LOGIN.DRAW_LOGIN_PAGE to draw the standard login page.
3. The login page submits a form to WWSSO_APP_ADMIN.LS_LOGIN to process the credentials.
4. If the user is authenticated, LS_LOGIN redirects to the requested application page. The Login Server uses the LOGIN_URL column of the WWSSO_LS_CONFIGURATION_INFO$ table to store the URL for the customized login page.

Login Page Parameters

The URL for the Login page must accept the parameters listed in Table 5–1.

Table 5–1 Login Page Parameters

Parameter	Description
site2pstoretoken	Contains the authentication request token for login processing.
ssousername	Contains the username.
p_error_code	Contains the error code, in the form of a VARCHAR2, if an error occurred during authentication.
p_cancel_url	Contains the URL to redirect to if the user clicks Cancel, if such a button exists on the login page.
p_submit_url	Contains the URL that the login page must submit the form to, WWSSO_APP_ADMIN.LS_LOGIN.
subscribername	Reserved for future use. **Note**: This field is required on the login page.

The customized login page must conform to the `wwsso_app_admin.ls_login` procedure in the same manner as the standard login page; passing the parameters listed in Table 5–2 to the `p_submit_url routine`:

Table 5–2 Customized Login Page Parameters

Parameter	Description
`site2pstoretoken`	Contains the redirect URL information for login processing.
`ssousername`	Contains the username.
`p_error_code`	Contains the error code, in the form of a VARCHAR2, if an error occurred during authentication.
`password`	Contains the password entered by the user.
`subscribername`	Reserved for future use. **Note**: This field is required on the login page.

The customized login page must have at least two fields: a text field with the parameter name `ssousername` and a password field with the parameter name `password`. The values are submitted to the `p_submit_url` routine. The login page must also submit the `site2pstoretoken` value as a hidden parameter.

In addition to submitting these parameters, the login page is responsible for displaying appropriate error messages, as specified by the `p_error_code` parameter, redirecting to `p_cancel` if the user clicks Cancel and populating the `ssousername` text field with the given parameter value in the case of a login error.

If the customized login page requires additional fields, you can include them. Ensure that additional fields are appropriately wrapped to conform to the above convention for integration with the Login Server.

Login Page Error Codes

The customized login page must process the error codes listed in Table 5–3.

Table 5–3 Customized Login Page Error Codes

Value of p_error_code	Corresponding error
`acct_ip_lock_err`	The user has committed too many login failures from this IP address and has been locked out.
`acct_lock_err`	The user has committed too many login failures from any IP address and has been globally locked out.
`null_uname_pwd_err`	The user did not type in a username.

Table 5–3 Customized Login Page Error Codes

Value of p_error_code	Corresponding error
`no_papp_err`	The partner application configuration is missing or expired.
`ssl_not_used_err`	SSL is not being used.
`ls_config_not_found_err`	The login server configuration is missing.
`cookies_disabled_err`	The user's browser is not accepting cookies.
`auth_fail_exception`	Authentication has failed.
`account_deactivated_err`	The user's account has been terminated.
`value_error_exception`	An invalid value was specified in `site2pstoretoken`.
`null_password_err`	The user did not type in a password.
`ext_auth_unknown_err`	There was an unknown error in accessing the external authentication mechanism.
`ext_auth_setup_err`	There was an error in the setup of the external authentication mechanism.
`sso_cookie_expired_err`	The login cookie has expired. The user needs to log in again.
`unexpected_exception`	An unexpected error occurred during authentication.

Customizing the Change Password Page

This section contains the following sections:

- Changing the Change Password Page
- Change Password Page Parameters
- Change Password Page Error Codes

Changing the Change Password Page

The Change Password page is created by the PL/SQL routine `WWSSO_APP_USER_MGR.CHANGE_PASSWORD`. This routine renders the screen and commits the form through an API to the database.

The process is as follows:

1. Oracle9*i*AS Portal calls the `WWSSO_APP_USER_MGR.CHANGE_PASSWORD` routine.

2. CHANGE_PASSWORD displays the Change Password page, which displays the username and has fields for the old password, the new password, and the password confirmation. It also has OK and Cancel buttons.
3. CHANGE_PASSWORD processes the new password.
4. CHANGE_PASSWORD saves the new password and redirects to the appropriate application page.

To accommodate a customized Change Password page, the logic for the Change Password page has been modified as follows:

1. Oracle9*i*AS Portal calls WWSSO_APP_USER_MGR.CHANGE_PASSWORD to display the Change Password page.
2. If a separate URL is to display the Change Password page, CHANGE_PASSWORD redirects to that URL.

   If no separate URL is specified, CHANGE_PASSWORD calls WWSSO_APP_USER_MGR.DRAW_CHANGE_PASSWORD_PAGE to display the standard Change Password page.
3. The Change Password page submits a form to WWSSO_APP_USER_MGR.SAVE_NEW_PASSWORD to process and save the new password.
4. If there are no errors, SAVE_NEW_PASSWORD saves the new password and redirects to the appropriate Oracle9*i*AS Portal page.

The LOGIN_URL column of the WWSSO_LS_CONFIGURATION_INFO$ table stores the URL for the customized Change Password page. The CHANGE_PASSWORD routine queries the value of the LOGIN_URL column to determine how to proceed. This column contains URLs for the Login and Change Password pages, separated by a space.

The Change Password page is also displayed immediately following a user login if the user’s password has expired or will be expiring soon. If the password has expired, the Change Password page appears with the appropriate message and the following process occurs:

1. WWSSO_APP_ADMIN.LS_LOGIN calls WWSSO_APP_USER_MGR.CHANGE_PASSWORD to display the Change Password page.
2. If a separate URL is to display the Change Password page, CHANGE_PASSWORD redirects to that URL.

   If a separate URL is not specified, CHANGE_PASSWORD calls WWSSO_APP_USER_MGR.DRAW_CHANGE_PASSWORD_PAGE and displays the standard Change Password page.

3. The Change Password page submits a form to WWSSO_APP_USER_MGR.SAVE_NEW_PASSWORD to process and save the new password.
4. If there are no errors and the user clicks OK, SAVE_NEW_PASSWORD saves the new password and returns control to WWSSO_APP_ADMIN.LS_LOGIN to perform the necessary login steps.
5. If there are errors or if the user clicks Cancel, SAVE_NEW_PASSWORD calls CHANGE_PASSWORD and redisplays the Change Password page. This process repeats until the user changes the password successfully.

Note: Clicking Cancel should not allow a user to continue if the password has expired. However, if the password is set to expire after a certain number of days and the user clicks Cancel, the login resumes and the user's password remains unchanged.

If the user's password is about to expire, the Change Password page appears with the appropriate message and the following process occurs:

1. WWSSO_APP_ADMIN.LS_LOGIN calls WWSSO_APP_USER_MGR.CHANGE_PASSWORD to display the Change Password page.
2. If a separate URL is to display the Change Password page, CHANGE_PASSWORD redirects to the separate URL.

   If no separate URL is specified, CHANGE_PASSWORD calls WWSSO_APP_USER_MGR.DRAW_CHANGE_PASSWORD_PAGE to display the standard Change Password page.
3. The Change Password page submits a form to WWSSO_APP_USER_MGR.SAVE_NEW_PASSWORD to process and save the new password.
4. If there are no errors and the user clicks OK, SAVE_NEW_PASSWORD saves the new password and returns control to WWSSO_APP_ADMIN.LS_LOGIN to perform the necessary login steps.
5. If there are errors, SAVE_NEW_PASSWORD calls CHANGE_PASSWORD and redisplays the Change Password page.
6. If the user clicks Cancel, SAVE_NEW_PASSWORD does not save the new password but returns control to WWSSO_APP_ADMIN.LS_LOGIN to perform the login steps using the current password.

Change Password Page Parameters

The URL for the Change Password page must accept the parameters listed in Table 5–4.

Table 5–4 Change Password Page Parameters

Parameter	Description
`p_username`	Contains the username to be displayed somewhere on the page.
`p_error_code`	Contains the error code, in the form of a VARCHAR2, if an error occurred in the prior attempt to change password.
`p_submit_url`	Contains the URL that the Change Password form must submit to.
`p_done_url`	Contains the URL of the appropriate Oracle9*i*AS Portal page to return to after the password is saved.
`p_pwd_is_exp`	Contains the flag value indicating whether the password has expired or is about to expire.
`site2pstoretoken`	Contains the `site2pstoretoken` that is required by the `LS_LOGIN` routine if the password has expired or is about to expire.

The customized Change Password page must pass the parameters listed in Table 5–5 to the `p_submit_url` routine.

Table 5–5 Customized Change Password Page Parameters

Parameter	Description
`p_password`	Contains the user's original password (if a password has or is about to expire).
`p_old_password`	Contains the user’s old password.
`p_new_password`	Contains the user’s new password.
`p_new_password_confirm`	Contains the confirmation of the user’s new password.
`p_done_url`	Contains the URL of the appropriate Oracle9*i*AS Portal page to return to after the password is saved.
`p_pwd_is_exp`	Contains the flag value indicating whether the password has expired or is about to expire.
`site2pstoretoken`	Contains the redirect URL information for login processing.

Table 5–5 Customized Change Password Page Parameters

p_password	Contains the password entered by the user.

The Change Password page must have at least three password fields with the following parameter names:

- p_old_password
- p_new_password
- p_new_password_confirm.

The Change Password page should submit these fields to the p_submit_url parameter.

The Change Password page should also submit the p_done_url parameter, as a hidden parameter, to the p_submit_url parameter, and should appropriately display any error messages according to the value of p_error_code.

It must also submit the following parameters, as hidden parameters, to the standard HTML login form presented to the external application from the Login Server:

- p_pwd_is_exp
- site2pstoretoken
- p_password

Change Password Page Error Codes

The customized Change Password page must process the error codes listed in Table 5–6.

Table 5–6 Change Password Page Error Codes

Value of p_error_code	Corresponding Error
null_old_pwd_err	The user did not type in an old password.
null_new_pwd_err	The user did not type in a new password.
confirm_pwd_fail_txt	The user typed in a new password confirmation that did not match the new password.
auth_fail_err	The user typed in an invalid old password.
pwd_rule_err	The user typed in a new password that does not meet the login server’s password requirements.

Table 5–6 Change Password Page Error Codes

Value of p_error_code	Corresponding Error
`invalid_auth_mode_err`	The change password operation is not supported by the current authentication mechanism.
`ext_not_supported_err`	The external repository is not supported.
`ext_change_pwd_err`	The change password operation was unsuccessful on the external repository.
`pwd_expired_err`	The password has expired.
`pwd_needs_change_err`	The password is about to expire. The user is allowed to log in.

The reset password page can encounter errors, which generates one of the following error codes, which is passed to the `p_error_url page` in the `p_error_code` argument.

Table 5–7 Reset Password Page Error Codes

Value of p_error_code	Corresponding Error
`pwd_reset_err`	The reset password failed in local authentication mode.
`notification_failure_err`	The reset password notification failed. If using external authentication, the password is reset at this point. In local mode, the password reset is rolled back.
`ext_auth_not_supported_err`	The Login Server is configured for external authentication, and it does not support the reset password.
`ext_auth_reset_password_err`	The Login Server is configured for external authentication, and the reset password failed.
`ext_auth_setup_err`	The Login Server is configured for external authentication, but it is not set up correctly.

Installing Customized Login and Change Password Pages

The `WWSSO_LS_CONFIGURATION_INFO$` table in the Login Server schema contains the `LOGIN_URL` column, which is used to enable customized Login and Change Password pages.

The LOGIN_URL column contains two values separated by a space. The first value specifies the URL for the Login page, and the second value specifies the URL for the Change Password page.

By default, the LOGIN_URL column contains the values UNUSED UNUSED, which specifies that the Login and Change Password pages use the standard Login Server pages.

Perform the following steps to install customized Single Sign-On Login and Change Password pages.

1. On the database where the Login Server is installed, log in to the Single Sign-On schema using SQL*Plus, as in the following example:

   ```
   sqlplus portal30_sso/portal30_sso
   ```

2. Update the LOGIN_URL column.

   To replace just the Login page with the customized page, update the first value in the LOGIN_URL column, as in the following example:

   ```
   UPDATE WWSSO_LS_CONFIGURATION_INFO$
   SET LOGIN_URL='http://server.domain[:port]/login.jsp UNUSED';
   ```

   To replace just the Change Password page with a customized page, update the second value in the LOGIN_URL column, as in the following example:

   ```
   UPDATE WWSSO_LS_CONFIGURATION_INFO$
   SET LOGIN_URL='UNUSED http://server.domain[:port]/change_password.jsp';
   ```

   To replace both pages, update both values in the LOGIN_URL column, as in the following example:

   ```
   UPDATE WWSSO_LS_CONFIGURATION_INFO$
   SET LOGIN_URL='http://server.domain[:port]/login.jsp
   http://server.domain[:port]/change_password.jsp';
   ```

3. To revert to using the standard pages, restore the original values, as in the following example:

   ```
   UPDATE WWSSO_LS_CONFIGURATION_INFO$
   SET LOGIN_URL='UNUSED UNUSED';
   ```

Configuring the Login Server for LDAP Authentication

If the users for whom you are granting access to Oracle9*i*AS Portal and other Oracle9*i*AS Single Sign-On applications are already listed in an LDAP directory, you can use the LDAP directory to authenticate users, instead of creating each user again in the Login Server.

The Login Server provides the `ssoldap.sql`, `ssooid.sql`, and `ssoldif.sql` scripts to configure the Login Server for LDAP user authentication.

If you use LDAP for user authentication, you cannot create new Oracle9*i*AS Single Sign-On users through the Login Server. Instead, you create new users with the LDAP directory using tools such as Oracle Directory Manager.

You must be the administrator of the LDAP directory to configure the Login Server for LDAP user authentication.

To create an LDIF file using the `ssoldif.sql` script, the initialization parameter file must be set up to allow you to write files to a directory.

This section contains the following topics:

- Configuring the Login Server for LDAP User Authentication
- LDIF File Example

Configuring the Login Server for LDAP User Authentication

You can configure the Login Server for LDAP user authentication using either the `DBMS_LDAP` package implemented in `ssoxoid.pkb`, or by using the external procedure listener, implemented in `ssoxldap.pkb`. Oracle Corporation recommends that you use the `ssoxoid.pkb` package if you are using Oracle 8*i* Release 3 (8.1.7) or later.

> **Note:** In order to configure an LDAP directory for authentication, it must have a `UserPassword` attribute populated for each user.

To complete the configuration, you run the `ssoldap.sql` or `ssooid.sql` scripts and the `ssoldif` script and copy the information in the generated LDIF file to the LDAP server.

This section contains these topics:

- Configuring the Login Server for LDAP Using ssoxoid.pkb
- Configuring the Login Server for LDAP Using ssoxldap.pkb
- Entering the LDAP Configuration Parameters
- Generating and Loading the User List

Configuring the Login Server for LDAP Using ssoxoid.pkb

Perform the following steps to configure the Login Server for LDAP user authentication using the DBMS_LDAP package ssoxoid.pkb.

> **Note:** If you are using Oracle 8*i* Release 3 (8.1.7), the database should be in dedicated mode.

1. Navigate to the Oracle home of the database where the Login Server is installed.
2. Navigate to the rdbms/admin directory.
3. Log on to SQL*Plus as SYS:

```
sqlplus sys/change_on_install
```

4. Enter the following:

```
@catldap.sql
```

The catldap.sql script installs the required LDAP packages.

5. Exit SQL*Plus.
6. Navigate to the Oracle9*i*AS Portal src/sso directory to exit.
7. On the database where the Login Server is installed, log on to SQL*Plus as the Login Server schema, as in the following example:

```
sqlplus portal30_sso/portal30_sso
```

8. Enter the following command to install ssoxoid.pkb:

```
@ssooid
```

9. Provide the LDAP configuration parameters as described in "Entering the LDAP Configuration Parameters".

Configuring the Login Server for LDAP Using ssoxldap.pkb

Perform the following steps to configure the Login Server for LDAP user authentication using the external listener procedure `ssoxldap.pkb`:

1. Copy the appropriate library file from the Oracle9*i*AS Portal `src/sso` directory to the appropriate directory on the Login Server:

   - If the Login Server is installed on a Windows NT machine, copy the `ssoxldap.dll` library file from the Oracle9*i*AS Portal `src\sso` directory to the `%ORACLE_HOME%\bin` directory on the Login Server machine.

   - If the Login Server is installed on a UNIX machine, copy the `ssoxldap.so` library file from the Oracle9*i*AS Portal `src/sso` to the `$ORACLE_HOME/lib` directory on the Login Server machine.

2. On the database where the Login Server is installed, log in to SQL*Plus as the Login Server schema, as in the following example:

   ```
   sqlplus portal30_sso/portal30_sso
   ```

3. Enter the following:

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

   where *`library_file_name`* is the full path and file name of the library file in Step 1.

   If the Login Server is installed on a Windows NT machine, enter the following:

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

   where *`oracle\ora81\bin`* is the path to the `%oracle_home%\bin`.

   If the Login Server is installed on a UNIX machine, enter the following:

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

   where *`u01/app/oracle/product/816prod/lib`* is the path to the `$oracle_home/bin`.

4. Enter the following command to install ssoxldap.pkb:

   ```
   @ssoldap
   ```

5. Provide the LDAP configuration parameters as described in "Entering the LDAP Configuration Parameters".

Entering the LDAP Configuration Parameters

The ssooid or ssoldap script prompts you to enter configuration information. The prompts are described in Table 5–8.

> **Note:** If you are using a multithreaded database, you must specify TCP instead of IPC in the tnsnames.ora and listener.ora files.

Table 5–8 Required Values to Configure the Login Server for LDAP User Authentication

Prompt	Description
Enter value for Host	Enter the name of the server on which the LDAP directory is installed, as in the following example: `ldap_server.mycompany.com`
Enter value for Port	Enter the port number used to access the server on which the LDAP directory is installed, as in the following example: `389`
Enter value for Search_Base	Enter the node in the LDAP Directory Information Tree (DIT) under which all your user entries are located, as in the following example: `cn=Login Server (portal30_sso)`
Enter value for Unique_Attribute	Enter the name of the attribute that contains Single Sign-On usernames. This attribute should be able to uniquely identify user entries in the LDAP directory, as in the following example: `cn`

Table 5–8 Required Values to Configure the Login Server for LDAP User Authentication

Prompt	Description
Enter value for Bind_DN	Enter the username of an account that has privileges to search the part of the LDAP DIT that contains your user entries, as in the following example: `cn=ldapadmin`
Enter value for Bind_Password	Enter the password for the above account, as in the following example: `welcome`

Generating and Loading the User List

To enable users to log in to Oracle9*i*AS Portal with default user accounts using LDAP authentication, you must migrate the Oracle9*i*AS Single Sign-On accounts created during Oracle9*i*AS Portal installation to the LDAP directory.

You should still be logged on to SQL*Plus as the Login Server schema.

1. Enter the following:

   ```
   @ssoldif
   ```

   This creates an LDIF file, `users.ldif`, which contains the Oracle9*i*AS Single Sign-On user accounts created during the installation of Oracle9*i*AS Portal. For an example of what the LDIF file might look like, see "LDIF File Example".

2. Quit SQL*Plus.

3. Add the information in the generated LDIF file to the LDAP directory.

   For example, to add the LDIF file to the Oracle Internet Directory LDAP server, you can use the following `ldapadd` command provided with the Oracle database:

   ```
   ldapadd -h <Host> -p <Port> -D <Bind_DN> -w <Bind_Password> -f users.ldif
   ```

Note:

- Descriptions of parameters are included in the list of prompts and responses in Table 5–8.
- The username you enter for the bind distinguished name should have sufficient privileges to be able to add entries to the directory.

See Also: *Oracle9iAS Internet Directory Administrator's Guide*

The Login Server is now fully configured to authenticate users with the LDAP directory.

Note: The Login Server is certified against Oracle Internet Directory and conforms to the LDAP specification. Other LDAP directories can be synchronized with Oracle Internet Directory.

When a user is added to the LDAP directory, Oracle9*i*AS Portal automatically creates a profile for the user when first logging in.

To delete a user, remove both the user's Oracle9*i*AS Portal profile and the user's LDAP entry. If the LDAP entry alone is removed, the user will not be able to login to the Oracle9*i*AS Portal, but the profile information will remain.

To remove LDAP integration with the Login Server, use SQL*Plus to run the `ORACLE_HOME/portal30/admin/plsql/sso/ssolocal.sql` script when you are logged in as the Login Server schema owner.

Note: For more information about configuring the Login Server and Oracle9*i*AS Portal with LDAP, see the following on Oracle Technology Network:
`http://technet.oracle.com/products/iportal/pdf/conf_ldap.pdf`

LDIF File Example

The following is an example of the LDIF file created by the ssoldif.sql script when configuring the Login Server for LDAP user authentication.

The example shows the LDIF file that would be created if Oracle9*i*AS Portal was installed in a 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
```

The `ssoldif.sql` script produces the above code example by default. If you already have a set of user entries and a Directory Information Tree (DIT) organization defined in the LDAP directory, you can modify the script to produce the necessary format. You can also manually create entries in the LDAP directory of the appropriate object class so that users *`portal_schema`* and *`portal_schema_`*`ADMIN` can log in.

6

Deployment Considerations

This chapter discusses deployment considerations for Oracle9*i*AS Single Sign-On and Login Server.

This chapter contains these topics:

- Troubleshooting
- Auditing
- Security

Troubleshooting

This section contains the following topics:

- User ADMIN
- Demo Certificate
- Oracle9iAS Portal Troubleshooting

User ADMIN

The user ADMIN must have Full Administrator privileges and must be created before you switch authentication methods from Local User Authentication to External Repository Authentication. The user ADMIN, which is always authenticated by the local repository, can perform Login Server administrative tasks even if the external repository is not operational.

ADMIN can only log in to Login Server for administrative purposes and cannot access any partner or external applications.

Demo Certificate

The demo certificate that ships with the 9*i* Application Server cannot be used to configure the Oracle9*i*AS Portal for SSL. In order to enable SSL on Oracle9*i*AS Portal, you must obtain a valid certificate from a supported certificate vendor.

Oracle9*i*AS Portal currently supports the Verisign, GTE CyberTrust, Entrust, Thawte, and Netscape certificate providers.

Oracle9*i*AS Portal Troubleshooting

For help with troubleshooting Oracle9*i*AS Portal technical problems, refer to the Oracle Portal Troubleshooting Guide in the Download / Install / Configure section in the Oracle9*i*AS Portal page on the Oracle Technology Network:

`http://technet.oracle.com/products/iportal`

Auditing

Login Server logs user activity, such as login successes, login failures, and password changes the `wwwsso_audit_log_view` audit log.

The following entries are logged in the audit table:

```
ACTION_LOGIN_OK
ACTION_LOGIN_FAILED
ACTION_CHNG_PASSWD_OK
ACTION_CHANGE_PASSWD_FAILED
ACTION_RESET_PASSWD_OK
ACTION_RESET_PASSWD_FAILED
ACTION_CLEAR_AUDIT_LOG
ACTION_PURGE_AUDIT_LOG
```

To view the log table, use SQL*Plus to log into the schema as follows:

```
sqlplus>  ???
sqlplus> select * from wwsso_audit_log_view;
```

The Login Server Administrator must occasionally purge the log table from the audit log to save disk space. To purge the log, run the `purgelog.sql` script from the Login Server schema.

> **Note:** User lockout (global or IP) functionality depends upon the audit log. If the audit log is fully purged, all users of the login server will be unlocked.

Security

This section contains the following topics:

- Secure Login
- Password Policy
- Account Lock Policy
- Unlocking a User

Secure Login

For a secure login, Login Server administrators must observe the following precautions:

1. The Login Server should have SSL to protect password transmission from the user's browser to the Login Server.
2. The schema password for the Login Server should be changed from the default value and should not be either the default schema name or `portal30_sso.`

Also, the administrator must change the password for all default Login Server administrator accounts.

> **Note:** When you change the schema password for Login Server, you must also change the password in the corresponding DAD.

3. The administrator URL for `mod_plsql` must be protected either by disabling from `mod_plsql` or performing an authentication check, for example:

   `http://foo.com/pls/admin_/`

4. The Login Server database must be in a trusted environment so that it is accessible only by the administrator. All Oracle Net connections to the Login Server database must be protected if they do not come from trusted zones.
5. When the Login Server is LDAP enabled, the connection from the LDAP Server to the Login Server must be within a protected channel. SSL can be used to protect the channel.

Password Policy

The Login Server supports the following password rules. The Login Server administrator can enable or disable them from the Login Server Administration menu.

1. The password expires after certain time interval.
2. The user is prompted to change the password before the password expiries
3. The password must have a minimum length
4. The password cannot be the same as the user name.
5. A new password cannot be the same as an existing password.
6. The password must contain at least one numeric digit.
7. The password must contain at least one character.

Account Lock Policy

Login Server supports two kinds of user lockout:

- IP Lockout:

When a user unsuccessfully tries to log in too many times from a single machine, the Login Server blocks that user from logging in from that machine's IP address for a certain amount of time.

- Global Lockout:

  If a user is IP locked out from more than one machine, then that user is locked out for all machine IP addresses for a certain amount of time.

The IP lockout configuration is as follows:

1. Number of login failures allowed from any IP address is per day
2. Lockout duration for one IP address in minutes.

The Global lockout configuration is as follows:

1. Number of login failures allowed from one IP address
2. Global lockout duration in days

Unlocking a User

To unlock a user, the Login Server must do the following:

1. Log in to the Oracle9*i*AS Single Sign-On schema using SQL*Plus
2. Run the script `ssounlck.sql` to unlock the user.

> **Note:** The `ssounlck.sql` script prompts the administrator for the user name to be unlocked.

Index