CHAPTER 2

Administering User Security

This chapter provides instructions for administering user security in the Oracle GlassFish Server environment by using the `asadmin` command-line utility. GlassFish Server enforces its authentication and authorization policies upon realms, users, and groups. This chapter assumes that you are familiar with security features such as authentication, authorization, and certificates. If you are not, see Chapter 1, "Administering System Security."

The following topics are addressed here:

- "Administering Authentication Realms" on page 51
- "Administering File Users" on page 60

Instructions for accomplishing these tasks by using the Administration Console are contained in the Administration Console online help.

Administering Authentication Realms

The following topics are addressed here:

- "Overview of Authentication Realms" on page 52
- "To Create an Authentication Realm" on page 53
- "To List Authentication Realms" on page 54
- "To Update an Authentication Realm" on page 54
- "To Delete an Authentication Realm" on page 55
- "To Configure a JDBC or Digest Authentication Realm" on page 55
- "To Configure LDAP Authentication with OID" on page 57
- "To configure LDAP Authentication with OVD" on page 58
- "To Enable LDAP Authentication on the GlassFish Server DAS" on page 59

Overview of Authentication Realms

An *authentication realm*, also called a security policy domain or security domain, is a scope over which the GlassFish Server defines and enforces a common security policy. GlassFish Server is preconfigured with the file, certificate, and administration realms. In addition, you can set up LDAP, JDBC, digest, Oracle Solaris, or custom realms. An application can specify which realm to use in its deployment descriptor. If the application does not specify a realm, GlassFish Server uses its default realm (`file`).

File realm

GlassFish Server stores user credentials locally in a file named `keyfile`. The file realm is the initial default realm.

Administration realm

The administration realm is also a file realm and stores administrator user credentials locally in a file named `admin-keyfile`.

Certificate realm

GlassFish Server stores user credentials in a certificate database. When using the certificate realm, the server uses certificates with the HTTPS protocol to authenticate web clients.

LDAP realm

GlassFish Server can get user credentials from a Lightweight Directory Access Protocol (LDAP) server such as Oracle Virtual Directory (OVD), Oracle Internet Directory (OID), and Oracle Directory Server Enterprise Edition. LDAP is a protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet.

See "To Configure LDAP Authentication with OID" on page 57 for instructions on configuring GlassFish Server to work with an OVD/OID LDAP provider.

JDBC realm

GlassFish Server gets user credentials from a database. The server uses the database information and the enabled JDBC realm option in the configuration file.

Digest realm

Digest Authentication authenticates a user based on a user name and a password. However, the authentication is performed by transmitting the password in an encrypted form.

Oracle Solaris realm

GlassFish Server gets user credentials from the Oracle Solaris operating system. This realm is supported on the Oracle Solaris 9 and Oracle Solaris 10 operating systems. Consult your Oracle Solaris documentation for information about managing users and groups in the Oracle Solaris realm.

PAM realm

A Pluggable Authentication Module (PAM) realm allows applications deployed on GlassFish Server to authenticate users against a native Unix (Solaris/Linux/Mac OS) users list. PAM realms use the class name `com.sun.enterprise.security.auth.realm.pam.PamRealm` and the JAAS Context `pamRealm`.

This realm is supported on all Unix Operating Systems, including the Oracle Solaris 9 and Oracle Solaris 10 operating systems

Custom realm — You can create other repositories for user credentials, such as a relational database or third-party components. For more information about custom realms, see the Administration Console online help. For instructions on creating a custom realm, see "Creating a Custom Realm" in *Oracle GlassFish Server 3.1 Application Development Guide*.

The GlassFish Server authentication service can govern users in multiple realms.

▼ To Create an Authentication Realm

Use the `create-auth-realm` subcommand in remote mode to create an authentication realm.

1 **Ensure that the server is running.**

Remote subcommands require a running server.

2 **Create a realm by using the `create-auth-realm(1)` subcommand.**

Information about properties for this subcommand is included in this help page.

Example 2–1 Creating a Realm

This example creates a realm named `db`.

```
asadmin> create-auth-realm --classname com.iplanet.ias.security.
auth.realm.DB.Database --property defaultuser=admin:Password=admin db
Command create-auth-realm executed successfully.
```

See Also You can also view the full syntax and options of the subcommand by typing `asadmin help create-auth-realm` at the command line.

For information on creating a custom realm, see "Creating a Custom Realm" in *Oracle GlassFish Server 3.1 Application Development Guide*.

▼ To List Authentication Realms

Use the `list-auth-realms` subcommand in remote mode to list the existing authentication realms.

1. **Ensure that the server is running.**

   Remote subcommands require a running server.

2. **List realms by using the `list-auth-realms(1)` subcommand.**

Example 2–2 Listing Realms

This example lists the authentication realms on `localhost`.

```
asadmin> list-auth-realms
db
certificate
file
admin-realm
Command list-auth-realms executed successfully.
```

See Also You can also view the full syntax and options of the subcommand by typing `asadmin help list-auth-realms` at the command line.

▼ To Update an Authentication Realm

Use the `set` subcommand to modify an existing authentication realm.

Note – A custom realm does not require server restart.

1. **List realms by using the `list-auth-realms(1)` subcommand.**

2. **Modify the values for the specified thread pool by using the `set(1)` subcommand.**

   The thread pool is identified by its dotted name.

3. **To apply your changes, restart GlassFish Server.**

   See "To Restart a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.

▼ To Delete an Authentication Realm

Use the delete-auth-realm subcommand in remote mode to delete an existing authentication realm.

1 **Ensure that the server is running.**

Remote subcommands require a running server.

2 **List realms by using the list-auth-realms(1) subcommand.**

3 **If necessary, notify users that the realm is being deleted.**

4 **Delete the realm by using the delete-auth-realm(1) subcommand.**

5 **To apply your changes, restart GlassFish Server. See "To Restart a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.**

Example 2–3 Deleting a Realm

This example deletes an authentication realm named db.

```
asadmin> delete-auth-realm db
Command delete-auth-realm executed successfully.
```

See Also You can also view the full syntax and options of the subcommand by typing asadmin help delete-auth-realm at the command line.

▼ To Configure a JDBC or Digest Authentication Realm

GlassFish Server enables you to specify a user's credentials (user name and password) in the JDBC realm instead of in the connection pool. Using the jdbc type realm instead of the connection pool prevents other applications from browsing the database tables for user credentials.

Note – By default, storage of passwords as clear text is not supported in the JDBC realm. Under normal circumstances, passwords should not be stored as clear text.

1 **Create the database tables in which to store user credentials for the realm.**

How you create the database tables depends on the database that you are using.

2 **Add user credentials to the database tables that you created.**

How you add user credentials to the database tables depends on the database that you are using.

3 **Create a JDBC connection pool for the database.**

See "To Create a JDBC Connection Pool" in *Oracle GlassFish Server 3.1 Administration Guide*.

4 **Create a JDBC resource for the database.**

"To Create a JDBC Resource" in *Oracle GlassFish Server 3.1 Administration Guide*.

5 **Create a realm.**

For instructions, see "To Create an Authentication Realm" on page 53.

Note – The JAAS context should be `jdbcDigestRealm` for digest authentication or `jdbcRealm` for other authentication types.

6 **Modify the deployment descriptor to specify the `jdbc` realm.**

Modify the deployment descriptor that is associated with your application.

- **For an enterprise application in an Enterprise Archive (EAR) file, modify the `sun-application.xml` file.**
- **For a web application in a Web Application Archive (WAR) file, modify the `web.xml` file.**
- **For an enterprise bean in an EJB JAR file, modify the `sun-ejb-jar.xml` file.**

For more information about how to specify a realm, see "How to Configure a Realm" in *Oracle GlassFish Server 3.1 Application Development Guide*.

7 **Assign security roles to users in the realm.**

To assign a security role to a user, add a `security-role-mapping` element to the deployment descriptor that you modified.

8 **Verify that the database is running.**

If needed, see "To Start the Database" in *Oracle GlassFish Server 3.1 Administration Guide*.

9 **To apply the authentication, restart the server.**

See "To Restart a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.

Example 2–4 Assigning a Security Role

This example shows a `security-role-mapping` element that assigns the security role `Employee` to user `Calvin`

```
<security-role-mapping>
    <role-name>Employee</role-name>
    <principal-name>Calvin</principal-name>
  </security-role-mapping>
```

▼ To Configure LDAP Authentication with OID

This procedure explains how to configure GlassFish Server to use LDAP authentication with Oracle Internet Directory (OID).

1. **Install Oracle Enterprise Manager 11g and the latest Enterprise Manager patches, if they are not installed already.**

   Instructions for installing Oracle Enterprise Manager are provided in the Oracle Enterprise Manager documentation set.

2. **Install the Oracle Identity Management Suite (IDM) 11g and Patch Set 2 or later, if they are not installed already.**

   Instructions for installing the Oracle Identity Management suite are provided in the *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

3. **Configure SSL for Oracle Internet Directory (OID), if it is not configured already. Configure the OID instance in the server authentication mode and with the protocol version set to SSLv3**

   Instructions for configuring SSL for OID are provided in the SSL chapter of the *Oracle Internet Directory Administrator's Guide*.

4. **Using Oracle Wallet Manager, export an SSL self-signed certificate you want to use with GlassFish Server.**

   Instructions for using Oracle Wallet Manager to create and export SSL certificates are provided in the Configure Oracle Internet Directory for SSL section of the SSL chapter in the *Oracle Internet Directory Administrator's Guide*.

5. **On the GlassFish Server side, use the `keytool` command import the certificate you exported with Oracle Wallet Manager.**

   The `keytool` command is available in the `$JAVA_HOME/bin` directory. Use the following syntax:

   ```
   keytool -importcert -alias "alias-name" -keystore domain-dir/config/cacerts.jks
   -file cert-name
   ```

   where the variables are defined as follows:

alias-name Name of an alias to use for the certificate

domain-dir Name of the domain for which the certificate is used

cert-name Path to the certificate that you exported with Oracle Wallet Manager.

For example, to import a certificate named `oi.cer` for a GlassFish Server domain in `/glassfishv3/glassfish/domains/domain1`, using an alias called "OID self-signed certificate," you would use the following command:

```
keytool -importcert -alias "OID self signed certificate" -keystore \
/glassfishv3/glassfish/domains/domain1/config/cacerts.jks -file oid.cer
```

6 **Restart the GlassFish Server domain.**

See "To Restart a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.

7 **Use the Oracle Enterprise Manager `ldapmodify` command to enable Anonymous Bind for OID.**

For example:

```
ldapmodify -D cn=orcladmin -q -p portNum -h hostname -f ldifFile
```

In this example, the LDIF file might contain the following:

```
dn: cn=oid1,cn=osdldapd,cn=subconfigsubentry
changetype: modify
replace: orclAnonymousBindsFlag
orclAnonymousBindsFlag: 1
```

To disable all anonymous binds, you would use a similar LDIF file with the last line changed to:

```
orclAnonymousBindsFlag: 0
```

See Managing Anonymous Binds in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for complete instructions on the `ldapmodify` command.

▼ To configure LDAP Authentication with OVD

This procedure explains how to configure GlassFish Server to use LDAP authentication with Oracle Virtual Directory (OVD).

1 **Create the OVD adapter, as described in the Creating and Configuring Oracle Virtual Directory Adapters (`http://download.oracle.com/docs/cd/E12839_01/oid.1111/e10046/basic_adapters.htm#BABCBGJA`) chapter of the Administrator's Guide for Oracle Virtual Directory (`http://download.oracle.com/docs/cd/E12839_01/oid.1111/e10046/toc.htm`).**

2 **Configure SSL for Oracle Virtual Directory (OVD), if it is not configured already. For instructions on configuring SSL for OVD, see the section "Enable SSL for Oracle Virtual Directory Using Fusion**

Middleware Control" in SSL Configuration in Oracle Fusion Middleware (`http://download.oracle.com/docs/cd/E12839_01/core.1111/e10105/sslconfig.htm#ASADM1800`).

Also, configure the SSL for the OVD listener in server authentication mode.

3 **Export the certificate from JKS keystore you want to use with GlassFish Server. See Exporting a Keystore Using Fusion Middleware Control (`http://download.oracle.com/docs/cd/E16764_01/core.1111/e10105/wallets.htm#CIHECAIB`) for information.**

4 **On the GlassFish Server side, use the `keytool` command to import the certificate you exported from the JKS keystore.**

The `keytool` command is available in the `$JAVA_HOME/bin` directory. Use the following syntax:

```
keytool -importcert -alias "alias-name" -keystore domain-dir/config/cacerts.jks
-file cert-name
```

where the variables are defined as follows:

alias-name	Name of an alias to use for the certificate
domain-dir	Name of the domain for which the certificate is used
cert-name	Path to the certificate that you exported from the keystore.

For example, to import a certificate named `ovd.cer` for a GlassFish Server domain in `/glassfishv3/glassfish/domains/domain1`, using an alias called "OVD self-signed certificate," you would use the following command:

```
keytool -importcert -alias "OVD self signed certificate" -keystore \
/glassfishv3/glassfish/domains/domain1/config/cacerts.jks -file ovd.cer
```

5 **Restart the GlassFish Server domain.**

See "To Restart a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.

▼ To Enable LDAP Authentication on the GlassFish Server DAS

This procedure explains how to enable LDAP authentication for logins to the GlassFish Server Domain Administration Server (DAS). Logging in to the DAS is typically only performed by GlassFish Server administrators who want to use the GlassFish Server Administration Console or `asadmin` command. See "To Configure LDAP Authentication with OID" on page 57 for instructions on enabling general LDAP authentication for GlassFish Server.

Before You Begin Ensure that you have followed the configuration instructions in "To Configure LDAP Authentication with OID" on page 57

- **Use the `asadmin` `configure-ldap-for-admin` subcommand to enable user authentication to the GlassFish Server DAS.**

  Use the following syntax:

  ```
  asadmin configure-ldap-for-admin --basedn "dn-list" --url [ldap|ldaps]://ldap-url
  --ldap-group group-name
  ```

  where the variables are defined as follows:

dn-list	basedn parameters
ldap-url	URL and port number for the LDAP server; can use standard (`ldap`) or secure (`ldaps`) protocol
group-name	LDAP group name for allowed users, as defined on the LDAP server.

  For example:

  ```
  asadmin configure-ldap-for-admin --basedn "dc=red,dc=iplanet,dc=com" \
  --url ldap://interopoel54-1:3060 --ldap-group sqestaticgroup

  asadmin configure-ldap-for-admin --basedn "dc=red,dc=iplanet,dc=com" \
  --url ldaps://interopoel54-1:7501 --ldap-group sqestaticgroup
  ```

See Also See `configure-ldap-for-admin(1)` for more information about the `configure-ldap-for-admin subcommand`.

Administering File Users

A *user* is an individual (or application program) identity that is defined in GlassFish Server. A user who has been authenticated is sometimes called a *principal*.

As the administrator, you are responsible for integrating users into the GlassFish Server environment so that their credentials are securely established and they are provided with access to the applications and services that they are entitled to use.

The following topics are addressed here:

- "To Create a File User" on page 61
- "To List File Users" on page 61
- "To List File Groups" on page 62
- "To Update a File User" on page 63
- "To Delete a File User" on page 63

To Create a File User

Use the `create-file-user` subcommand in remote mode to create a new user by adding a new entry to the `keyfile`. The entry includes the user name, password, and any groups for the user. Multiple groups can be specified by separating the groups with colons (:).

Creating a new `file` realm user is a dynamic event and does not require server restart.

1 **Ensure that the server is running.**

Remote subcommands require a running server.

2 **If the user will belong to a particular group, see the current groups by using the `list-file-groups(1)` subcommand.**

3 **Create a file user by using the `create-file-user(1)` subcommand.**

Example 2–5 Creating a User

This example create user `Jennifer` on the default realm `file` (no groups are specified).

The `asadmin --passwordfile` option specifies the name of a file that contains the password entries in a specific format. The entry for a password must have the `AS_ADMIN_` prefix followed by the password name in uppercase letters, an equals sign, and the password. See `asadmin(1M)` for more information.

```
asadmin> create-file-user --user admin
--passwordfile=c:\tmp\asadminpassword.txt Jennifer
Command create-file-user executed successfully.
```

See Also You can also view the full syntax and options of the subcommand by typing `asadmin help create-file-user` at the command line.

To List File Users

Use the `list-file-users` subcommand in remote mode to list the users that are in the `keyfile`.

1 **Ensure that the server is running.**

Remote subcommands require a running server.

2 **List users by using the `list-file-users(1)` subcommand.**

Example 2–6 Listing File Users

This example lists file users on the default `file` realm file.

```
asadmin> list-file-users
Jennifer
Command list-file-users executed successfully.
```

See Also You can also view the full syntax and options of the subcommand by typing `asadmin help list-file-users` at the command line.

▼ To List File Groups

A *group* is a category of users classified by common traits, such as job title or customer profile. For example, users of an e-commerce application might belong to the `customer` group, and the big spenders might also belong to the `preferred` group. Categorizing users into groups makes it easier to control the access of large numbers of users. A group is defined for an entire server and realm. A user can be associated with multiple groups of users.

A group is different from a role in that a role defines a function in an application, while a group is a set of users who are related in some way. For example, in the personnel application there might be groups such as `full-time`, `part-time`, and `on-leave`. Users in these groups are all employees (the `employee` role). In addition, each user has its own designation that defines an additional level of employment.

Use the `list-file-groups` subcommand in remote mode to list groups for a file user, or all file groups if the `--name` option is not specified.

1 **Ensure that the server is running.**

Remote subcommands require a running server.

2 **List file groups by using the `list-file-groups(1)` subcommand.**

Example 2–7 Listing Groups for a User

This example lists the groups for user `joesmith`.

```
asadmin> list-file-groups --name joesmith
staff
manager
Command list-file-groups executed successfully
```

▼ To Update a File User

Use the update-file-user subcommand in remote mode to modify the information in the keyfile for a specified user.

1 **Ensure that the server is running.**

Remote subcommands require a running server.

2 **Update the user information by using the update-file-user(1) subcommand.**

3 **To apply your changes, restart GlassFish Server.**

See "To Restart a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.

Example 2–8 Updating a User

The following subcommand updates the groups for user Jennifer.

```
asadmin> update-file-user --passwordfile c:\tmp\asadminpassword.txt --groups
staff:manager:engineer Jennifer
Command update-file-user executed successfully.
```

See Also You can also view the full syntax and options of the subcommand by typing asadmin help update-file-user at the command line.

▼ To Delete a File User

Use the delete-file-user subcommand in remote mode to remove a user entry from the keyfile by specifying the user name. You cannot delete yourself, that is, the user you are logged in as cannot be deleted during your session.

1 **Ensure that the server is running.**

Remote subcommands require a running server.

2 **List users by using the list-file-users(1) subcommand.**

3 **Delete the user by using the delete-file-user(1) subcommand.**

Example 2–9 Deleting a User

This example deletes user Jennifer from the default file realm.

```
asadmin> delete-file-user Jennifer
Command delete-file-user executed successfully.
```

See Also You can also view the full syntax and options of the subcommand by typing `asadmin help delete-file-user` at the command line.