Oracle® Analytics

Managing Security for Oracle Analytics Server

5.9.0
F24229-11
September 2021

Oracle Analytics Managing Security for Oracle Analytics Server, 5.9.0

F24229-11

Primary Author: Stefanie Rhone

Contributors: Oracle Business Intelligence development, product management, and quality assurance teams.

Contents

3 Use Alternative Authentication Providers

Preface

Learn how to secure Oracle Analytics Server.

Audience

This guide is intended for system administrators who are responsible for setting up and managing Oracle Analytics Server security.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Get Started with Oracle Analytics Server Security

This chapter contains overview concepts, a terminology list, and a workflow to help you configure security.

Topics:

Typical Workflow to Set Up Security

Use this workflow to understand how to set up security in a new Oracle Analytics Server instance.

Task	Description	More Information
Decide if you want to use the default embedded WebLogic LDAP Server for authentication to create users and groups	Oracle doesn't recommend using WebLogic LDAP Server in an environment with more than 1,000 users. If you need a production environment with high-availability and scalability, then use a directory service such as Oracle Internet Directory or a third-party directory service. Use the WebLogic Server Administration Console to create users and groups and assign users to groups. You can't use the Oracle Analytics Server Console to create and manage users and groups.	Create a New User in the Embedded WebLogic LDAP Server Create a New Group in the Embedded WebLogic LDAP Server Assign a User to a Group in the Embedded WebLogic LDAP Server
Decide if you want to use an alternative authentication provider such as Oracle Internet Directory to create users and groups	Configure Oracle Internet Directory as the authentication provider. Use your authentication provider tools to create users and groups and assign users to groups. You can't use the Oracle Analytics Server Console to create and manage users and groups.	High-Level Steps for Configuring an Alternative Authentication Provider

Task	Description	More Information
Set up application roles	Review the application roles provided with the installation and decide if you need to create additional roles. Use the Oracle Analytics Server Console to add application roles.	Predefined Application Roles Add Your Own Application Roles
Customize the permission sets assigned to the application roles	Add or remove permissions as needed. Use the grant or revoke permissions script to add or remove application role permissions.	Grant or Revoke Permission Assignments
Assign application roles to users and groups	Add application roles to users and groups as needed. Use the Oracle Analytics Server Console to assign application roles to users and groups.	Assign Application Roles to Users Assign Application Roles to Multiple Users Through Roles
Fine-tune privileges in the BI repository and Presentation Services	Add and remove the privileges that users and groups have in the Oracle BI Repository and in the Classic Home Page. Use the Oracle BI Administration Tool and the Oracle Analytics Server Classic Administration Page to add and remove these privileges.	Managing Metadata Repository Privileges Using the Oracle BI Administration Managing Presentation Services Privileges Using Application Roles
Decide if you want to deploy single sign-on (SSO) authentication	Configure SSO authentication.	Enabling SSO Authentication
Decide if you want to deploy secure socket layer (SSL)	Configure Oracle Analytics Server components to communicate over SSL.	Configuring SSL in Oracle Business Intelligence

Overview of Security in Oracle Analytics Server

Oracle Analytics Server is tightly integrated with the Oracle Fusion Middleware Security architecture and delegates core security functionality to components of that architecture. Specifically, any Oracle Analytics Server installation makes use of the following types of security providers:

- An authentication provider that knows how to access information about the users and groups accessible to Oracle Analytics Server and is responsible for authenticating users.
- A policy store provider that provides access to application roles and application policies, which forms a core part of the security policy and determines what users can and cannot see and do in Oracle Analytics Server.
- A credential store provider that is responsible for storing and providing access to credentials required by Oracle Analytics Server.

By default, an Oracle Analytics Server installation is configured with an authentication provider that uses the Oracle WebLogic Server embedded LDAP server for user and

group information. The Oracle Analytics Server default policy store provider and credential store provider store credentials, application roles, and application policies in a database.

After installing Oracle Analytics Server you can reconfigure the domain to use alternative security providers, if desired. For example, you might want to reconfigure your installation to use an Oracle Internet Directory, Oracle Virtual Directory, Microsoft Active Directory, or another LDAP server for authentication. You might also decide to reconfigure your installation to use Oracle Internet Directory, rather than a database, to store credentials, application roles, and application policies.

About Authentication

You manage users and groups within the authentication provider.

> **Note:**
>
> Use your authentication provider tools to create users and groups and assign users to groups. You can't use the Oracle Analytics Server Console to create and manage users and groups.

Each Oracle Analytics Server installation has an associated Oracle WebLogic Server domain. Oracle Analytics Server delegates user authentication to the authentication providers configured for that domain.

The default authentication provider accesses user and group information that is stored in the LDAP server that is embedded in the Oracle WebLogic Server domain for Oracle Analytics Server. You can use the Oracle WebLogic Server Administration Console to create and manage users and groups in the embedded LDAP server.

You might choose to configure an authentication provider for an alternative directory. You can use the Oracle WebLogic Server Administration Console to view the users and groups in the directory. However, you must continue to use the appropriate tools to make any modifications to the directory. For example, if you reconfigure Oracle Analytics Server to use Oracle Internet Directory (OID), you can view users and groups in Oracle WebLogic Server Administration Console but you must manage them using the OID Console. Refer to the BI certification matrix for information on supported LDAP directories.

About Authorization

Authorization is about ensuring users can do and see what they are authorized to do and see.

After a user has been authenticated, the next critical aspect of security is ensuring that the user can do and see what they are authorized to do and see. Authorization for Oracle Analytics Server is controlled by a security policy defined in terms of application roles.

Topics:

About Application Roles

Application roles define the security policy for users.

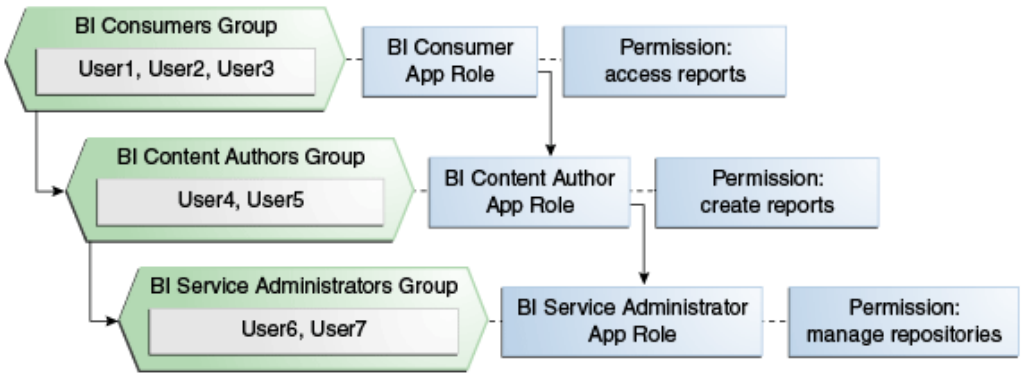
Instead of defining the security policy in terms of users in groups in a directory server, Oracle Analytics Server uses a role-based access control model. Security is defined in terms of application roles that are assigned to directory server groups and users. For example, application roles BIServiceAdministrator, BI Consumer, and BIContentAuthor.

Application roles represent a functional role that a user has given the user the privileges required to perform that role. For example, the Sales Analyst application role might grant a user access to view, edit, and create reports on a company's sales pipeline.

This indirection between application roles and directory server users and groups allows the administrator to define the application roles and policies without creating additional users or groups in the corporate LDAP server. Instead, the administrator defines application roles that meet the authorization requirements and assigns those roles to preexisting users and groups in the corporate LDAP server.

In addition, the indirection afforded by application roles allows moving artifacts between development, test, and production environments. No change to the security policy is needed as a result of the environment moves, and all that is required is to assign the application roles to the users and groups available in the target environment.

For example, the diagram below shows a set of groups, users, application roles, permissions, and inheritance.



The diagram shows the following:

- The group named BI Consumers Group contains User1, User2, and User3. Users in the BI Consumers Group are assigned the application role BI Consumer, which enables the users to view reports.
- The group named BI Content Authors Group contains User4 and User5. Users in the BI Content Authors Group are assigned the application role BI Content Author, which enables the users to create reports.

- The group named BI Service Administrators Group contains User6 and User7. Users in the BI Service Administrators Group are assigned the application role BI Service Administrator, which enables the users to manage repositories.

About the Security Policy

The security policy is split across Presentation Services, the metadata repository, and the policy store.

Presentation Services

Presentation Services defines the specific catalog objects and functionality that users can access with specific application roles. Access to functionality is defined in the Managing Privileges page and access to catalog objects is defined in the Permission dialog.

Metadata Repository

The repository defines the metadata items in the repository that user can access with assignment to specific application roles. You can define the security policy using the Oracle BI Administration Tool.

Policy Store

The Policy Store defines the BI Server and Publisher functionality that user can access with specific application roles. In the default Oracle Analytics Server configuration, the policy store is managed using the grant and revoke scripts or by using Oracle WebLogic Scripting Tool (WLST).

About Users, Groups, and Application Roles

When you install and configure Oracle Analytics Server, you select an application (BAR file) to install into your initial service instance. The application you select determines your instance's initial security policy.

The imported security policy includes the application role definitions, the application role memberships, permission set definitions, permission definitions, permission set grants, permission grants, and the Presentation Services and repository security policy.

You can use the application roles and permission grants provided by the application you chose during install or you can modify them as needed. If a development team creates an Oracle Analytics Server application, then they don't have to use the default application roles and permissions and can define and name the application roles and permission grants specific to their applications.

Terminology

The following terms are used throughout this guide:

Application Policy
Oracle Analytics Server permissions are granted by its application roles. In the default security configuration, each role conveys a predefined set of permissions. An application policy is a collection of Java EE and JAAS policies that are applicable to a specific application. The application policy is the mechanism that defines the permissions each application role grants. Permission grants are managed in the application policy corresponding to an application role.

Application Role
Represents a role a user has when using Oracle Analytics Server. Is also the container used by Oracle Analytics Server to grant permissions to members of a role. Application roles are managed in the Oracle Analytics Server console.

Authentication
The process of verifying identity by confirming the credentials presented during log in.

Authentication Provider
A security provider used to access user and group information and responsible for authenticating users. Oracle Analytics Server default authentication provider is Oracle WebLogic Server embedded directory server and is named DefaultAuthenticator.

Authorization
The process of granting an authenticated user access to a resource in accordance to their assigned privileges.

Catalog Groups
Catalog groups are not supported in Oracle Analytics Server.

Catalog Permissions
These rights grant access to objects that are stored in the Oracle BI Presentation Catalog. The rights are stored in the catalog and managed by Presentation Services.

Catalog Privileges
These rights grant access to features of the Oracle BI Presentation Catalog. The rights are stored in the catalog and managed by Oracle BI Presentation Services. These privileges are either granted or denied.

Credential Store
An Oracle Analytics Server credential store is a file used to securely store system credentials used by the software components. This file is automatically replicated across all machines in the installation.

Credential Store Provider
The credential store is used to store and manage credentials securely that are used internally between Oracle Analytics Server components. For example, SSL certificates are stored here.

Encryption
A process that enables confidential communication by converting plain text information (data) to unreadable text which can be read-only with the use of a key. Secure Sockets Layer (SSL) enables secure communication over TCP/IP networks, such as web applications communicating through the Internet.

Identity Store
An *identity store* contains user name, password, and group membership information. In Oracle Analytics Server, the identity store is typically a directory server and is what an authentication provider accesses during the authentication process. For example, when a user name and password combination is entered at log in, the authentication provider searches the identity store to verify the credentials provided. Oracle Analytics Server can be re-configured to use alternative identity stores.

Impersonation
Impersonation is a feature used by Oracle Analytics Server components to establish a session on behalf of a user without employing the user's password. For example, impersonation is used when Oracle BI Scheduler executes an Agent.

Oracle WebLogic Server Domain
A logically related group of Oracle WebLogic Server resources that includes an instance known as the Administration Server. Domain resources are configured and managed in the Oracle WebLogic Server Administration Console.

Permission Set
Represents a set of permissions.

Policy Store Provider
The policy store is the repository of system and application-specific policies. It holds the mapping definitions between the default Oracle Analytics Server application roles, permissions, users and groups all configured as part of installation. Oracle Analytics Server permissions are granted by assigning users and groups from the identity store to application roles and permission grants located in the policy store.

Policy Store
Contains the definition of application roles, application policies, and the members assigned such as users, groups, and application roles to application roles. The default policy store is a file that is automatically replicated across all machines in an Oracle Analytics Server installation. A policy store can be database-based or LDAP-based.

Secure Sockets Layer (SSL)
Provides secure communication links. Depending upon the options selected, SSL might provide a combination of encryption, authentication, and repudiation. For HTTP based links the secured protocol is known as HTTPS.

Security Policy
The security policy defines the collective group of access rights to Oracle Analytics Server resources that an individual user or a particular application role have been granted. Where the access rights are controlled is determined by which Oracle Analytics Server component is responsible for managing the resource being requested. A user's security policy is the combination of permission and privilege grants governed by the following elements:

- Oracle BI Presentation Catalog:

  Defines which Oracle BI Presentation Catalog objects and Oracle BI Presentation Services functionality can be accessed by users. Access to this functionality is managed in Oracle Analytics Server user interface. These permissions and privileges can be granted to individual users or by membership in corresponding application roles.

- Repository File:

  Defines access to the specified metadata within the repository file. Access to this functionality is managed in the Oracle BI Administration Tool. These permissions and privileges can be granted to individual users or by membership in corresponding application roles.

- Policy Store:

  Defines which Oracle Analytics Server and Publisher functionality can be accessed. You use the grant and revoke scripts to manage access to functionality by application role.

Security Realm
During deployment an Oracle WebLogic Server domain is created and Oracle Analytics Server is deployed into that domain. Security for an Oracle WebLogic Server domain is managed in its *security realm*. A security realm acts as a scoping mechanism. Each security realm consists of a set of configured security providers, users, groups, security roles, and security policies. Only one security realm can be active for the domain. Oracle Analytics Server authentication is performed by the authentication provider configured for the default security realm for the WebLogic Server domain in which it is installed. Oracle WebLogic Server Administration Console is the Administration Tool for managing an Oracle WebLogic Server domain.

Single Sign-On
A method of authorization enabling a user to authenticate once and gain access to multiple software application during a single browser session.

Users and Groups
A *user* is an entity that can be authenticated. A user can be a person, such as an application user, or a software entity, such as a client application. Every user is given a unique identifier within in the identity store.

2

Set Up Security With Users, Groups, and Application Roles

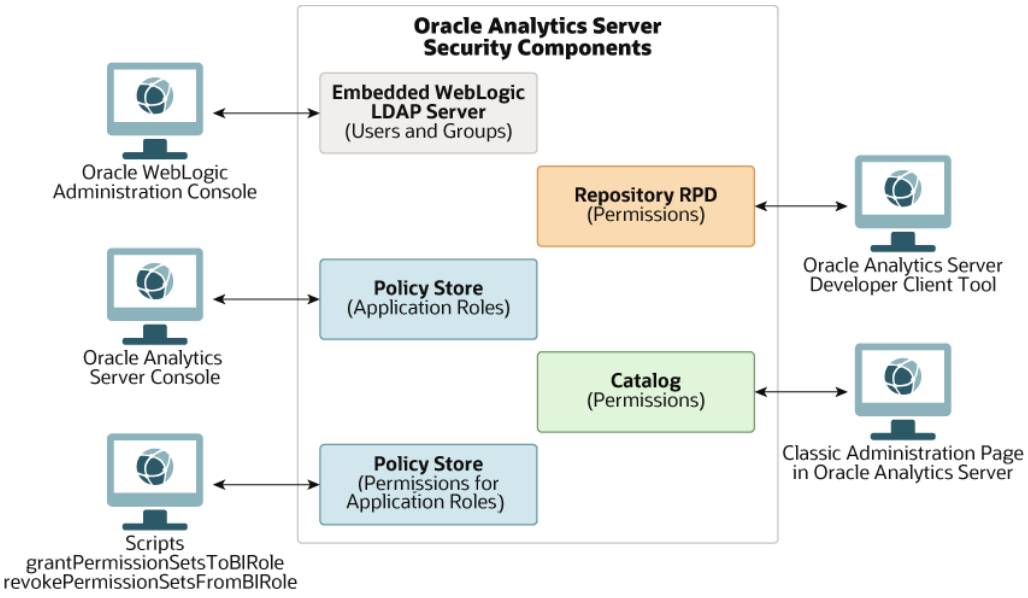
This topic explain how to deploy Oracle Analytics Server security using the embedded WebLogic LDAP Server and the default application.

You can also use the information in this topic to modify the security settings for an application archive (BAR file) exported from another installation and imported into Oracle Analytics Server.

Topics:

Security Configuration Tools

This diagram shows the tools that you'll use to configure security in an installation that uses the embedded WebLogic LDAP Server.



Manage Users and Groups in the Embedded WebLogic LDAP Server

This section explains how to manage users and groups in the Embedded WebLogic LDAP Server.

Topics:

Use the Oracle WebLogic Server Administration Console

You use Oracle WebLogic Server Administration Console to manage the WebLogic LDAP Server that enables you to authenticate users and groups.

Oracle WebLogic Server is automatically installed and serves as the default administration server. The Oracle WebLogic Server Administration Console is browser-based and is used, among other things, to manage the embedded directory server.

When you configure Oracle Analytics Server, the initial security configuration uses the embedded WebLogic LDAP directory, the default authenticator, as the Identity Store. The Oracle Analytics Server installation adds specific BI users and groups into the LDAP directory. The installation does not add default BI groups into the LDAP directory. If your application expects LDAP groups such as the BIConsumers, BIContentAuthors, and BIServiceAdministrators to exist in the Identity Store, you need

to add these groups manually or configure the domain to use a different Identity Store, where these groups are already provisioned after the initial configuration has finished.

You can launch the Oracle WebLogic Server Administration Console by entering its URL into a web browser. The default URL takes the following form: `http://hostname:port_number/console`. The port number is the same port number as used for the Administration server. The default port number is 9500. See *Oracle WebLogic Server Administration Console Online Help*.

The user name and password were supplied during the installation of Oracle Analytics Server. If these values have since been changed, then use the current administrative user name and password combination.

If you use an alternative authentication provider such as Oracle Internet Directory instead of the default the WebLogic LDAP Server, then you must use the alternative authentication provider administration application, for example, an administration console to manage users and groups.

1. Display the Oracle WebLogic Server login page by entering its URL into a web browser.

   For example, `http://hostname:9500/console`.

2. Log in using the Oracle Analytics Server administrative user and password credentials.

Create a New User in the Embedded WebLogic LDAP Server

You typically create a separate user for each business user in your Oracle Analytics Server environment.

For example, you might plan to deploy 30 report consumers, 3 report authors, and 1 administrator. In this case, you would use Oracle WebLogic Server Administration Console to create 34 users, which you would then assign to appropriate groups.

All users who are able to log in are given a basic level of operational permissions conferred by the built-in Authenticated User application role. The author of the application that is imported into your instance might have designed the security policy so that all authenticated users are members of an application role that grants privileges in the application.

DefaultAuthenticator is the name for the default authentication provider.

1. Log in to the Oracle WebLogic Server Administration Console.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane, and then click the realm you are configuring, for example, **myrealm**.
3. Select **Users and Groups** tab, then **Users**. Click **New**.
4. In Create a New User, in **Name**, type the name of the user.
5. Optional: In **Description**, provide additional information about the user.
6. From the **Provider** list, select the authentication provider that corresponds to the identity store where the user information is contained.
7. In **Password**, type a password for the user that is at least 8 characters long.
8. In **Confirm Password**, retype the user password.
9. Click **OK**.

Create a New Group in the Embedded WebLogic LDAP Server

You can create a separate group for each functional type of business user in your Oracle Analytics Server environment.

A typical deployment might require three groups: *BIConsumers*, *BIContentAuthors*, and *BIServiceAdministrators*. You could create groups with those names and configure the group to use with Oracle Analytics Server, or you might create your own custom groups.

DefaultAuthenticator is the default authentication provider.

1. Launch Oracle WebLogic Server Administration Console.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.
3. Click the **Users and Groups** tab, and then click **Groups**.
4. Click **New**.
5. In **Create a New Group**, in the **Name** field, type a group names that is unique.
6. Optional: In the **Description** field, type a brief note about the composition of the group.
7. From the **Provider** list, select the authentication provider that corresponds to the identity store where the group information is contained.
8. Click **OK**

Assign a User to a Group in the Embedded WebLogic LDAP Server

You typically assign each user to an appropriate group.

For example, a typical deployment might require user IDs created for report consumers to be assigned to a group named BIConsumers. In this case, you could either assign the users to the default group named BIConsumers, or you could assign the users to your own custom group that you have created.

1. Launch Oracle WebLogic Server Administration Console.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring, for example, **myrealm**.
3. Select **Users and Groups** tab, then **Users**.
4. In the **Users** table select the user you want to add to a group.
5. Select the **Groups** tab.
6. Select a group or groups from the **Available** list.
7. Click **Save**.

Delete a User

When a user is no longer required you must completely remove their user ID from the system to prevent an identical, newly-created user from inheriting the old user's

access permissions. This situation can occur because authentication and access permissions are associated with user ID.

You delete a user by removing the user from the policy store, the Oracle Presentation Catalog, the metadata repository, and the identity store. If you've assigned the user to any application roles, you must update the application roles to remove all references to that user.

If you're using an identity store other than Oracle WebLogic Server LDAP, follow the appropriate instructions for your identity store.

1. Delete the user from the policy store.
2. Delete the user from the Oracle BI Presentation Catalog, and the metadata repository using the `deleteusers` command.
3. Log in to the Oracle WebLogic Server Administration Console.
4. Select **Security Realms**, and select the realm containing the user, for example, **myrealm**.
5. Click **Users and Groups** tab, then click **Users**.
6. Select a user, click **Delete**.
7. In Delete Users, click **Yes**.
8. Click **OK**.

Change a User Password in the Embedded WebLogic LDAP Server

You can change a user's password.

If you change the password of the system user, you also need to change it in the credential store.

1. In Oracle WebLogic Server Administration Console, select **Security Realms**, and click the realm you're configuring, for example, *myrealm*.
2. Select the **Users and Groups** tab, and then click **Users**.
3. In the Users table, select the user receiving the changed password.
4. In the user's Settings page, select the **Passwords** tab.
5. Type the password in the **New Password** and **Confirm Password** fields.
6. Click **Save**.

Manage Application Roles

Administrators create, modify, and assign application roles to determine what users can see and do in Oracle Analytics Server.

Topics:

About Application Roles

An application role comprises a set of privileges that determine what users can see and do after signing in to Oracle Analytics Server. It's your job as an administrator to assign people to one or more application roles.

There are two types of application role:

Type of Application Role	Description
Predefined	Include a fixed set of privileges.
User-defined	Created by administrators. Include one or more predefined application roles. See Add Your Own Application Roles.

Predefined Application Roles

Oracle Analytics Server provides several predefined application roles to get you started. In many cases, these predefined application roles are all that you need.

Predefined Application Roles in Oracle Analytics Server	Description	Default Members
BI Service Administrator	Allows users to administer Oracle Analytics Server and delegate privileges to others using the Console.	Administrator who created the service
DV Content Author	Allows users to create visualization projects, load data for data visualizations, and explore data visualizations.	BI Service Administrator
BI Content Author	Allows users to create analyses, dashboards, and pixel-perfect reports in Oracle Analytics Server and share them with others.	BI Service Administrator DV Content Author
DV Consumer	Allows users to explore data visualizations.	DV Content Author

Predefined Application Roles in Oracle Analytics Server	Description	Default Members
BI Consumer	Allows users to view and run reports in Oracle Analytics Server (projects, analyses, dashboards, pixel-perfect reports). Use this application role to control who has access to the service.	DV Consumer BI Content Author
BI Data Model Author	Not used	N/A
BI Data Load Author	Not used	N/A

You can't delete predefined application roles or remove default memberships.

Application roles can have users, roles, or other application roles as members. This means that a user who is a member of one application role might indirectly be a member of other application roles.

For example, any member of the BI Service Administrator application role inherits membership of other application roles, such as BI Data Model Author and BI Consumer. This means that any user that is a member of BI Service Administrator can do everything that these other application roles allow. So you don't need to add a new user (for example, John) to all these application roles. You can simply add the user to the BI Service Administrator application role.

Get Started with Application Roles

Administrators configure what users see and do in Oracle Analytics Server from the **Users and Roles** page in the Console. This page presents user information in 3 different views:

Users and Roles Page	Description
Users tab	Shows users from the identity domain associated with your instance. You can't add or remove user accounts through the Users tab in Oracle Analytics Server. To add or remove user accounts, use embedded WebLogic LDAP Server.
Roles tab	Shows roles from the identity system associated with your instance. You can't add or remove roles (groups of users) through the Roles tab in Oracle Analytics Server. To add or remove roles, use embedded WebLogic LDAP Server. From the Roles tab you can also see who belongs to each role.
Application Roles tab	Shows application roles for Oracle Analytics Server together with any custom application roles you define. From the Application Roles tab you can assign application roles to multiple users, roles, and other application roles. You can also create application roles of your own and assign privileges to them through other application roles.

Add Members to Application Roles

Application roles determine what people are allowed to see and do in Oracle Analytics Server. It's the administrator's job to assign appropriate application roles to all users and to manage the privileges of each application role.

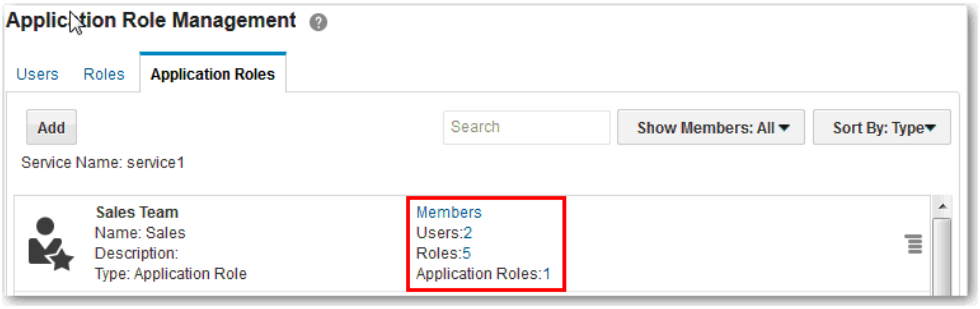
Remember:

- Members inherit the privileges of an application role.
- Application roles inherit privileges from their parent (application roles).

You select members for an application role or change parent privileges using the Console.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click the **Application Roles** tab.
4. To display all available application roles, leave the **Search** field blank and **Show Members: All**.

   To filter the list by name, enter all or part of an application role name in the **Search** filter and press Enter. The search is case-insensitive, and searches both name and display name.
5. Look in the **Members** area to see who belongs to each application role:



   The number of users, roles, and application roles that are members displays on the page. Click a number, such as **5** in this image, to see those members in more detail (either users, roles or application roles).
6. To add new members or remove members from an application role:
   a. Click **Members**.
   b. Select either users, roles, or application roles from the **Type** box and click **Search** to show the current members.
   c. Use the shuttle controls to move members between the **Available** and **All Selected** list.

      Some application roles aren't eligible to be members and these are grayed. For example, you can't select a parent application role to be a member.

Users marked ‘absent’ no longer have an account in your identity domain. To remove absent users, use the shuttle control to move the user from the **All selected users** list to the **Available users** list.

d. Click **OK**.

7. To see whether an application role, such as Sales Analyst, inherits privileges from other application roles:

a. Click the action menu.

b. Select **Manage Application Roles**.

Inherited privileges are displayed in the Selected Application Roles pane.

8. To add or remove privileges:

a. Click **Search** to display all available application roles.

Alternatively, enter all or part of an application role name and click **Search**.

b. Use the shuttle controls to move application roles between the **Available Application Roles** list and the **Selected Application Roles** list.

You can’t select application roles that are grayed out. Application roles are grayed out so you can’t create a circular membership tree.

c. Click **OK**.

Why Is the Administrator Application Role Important?

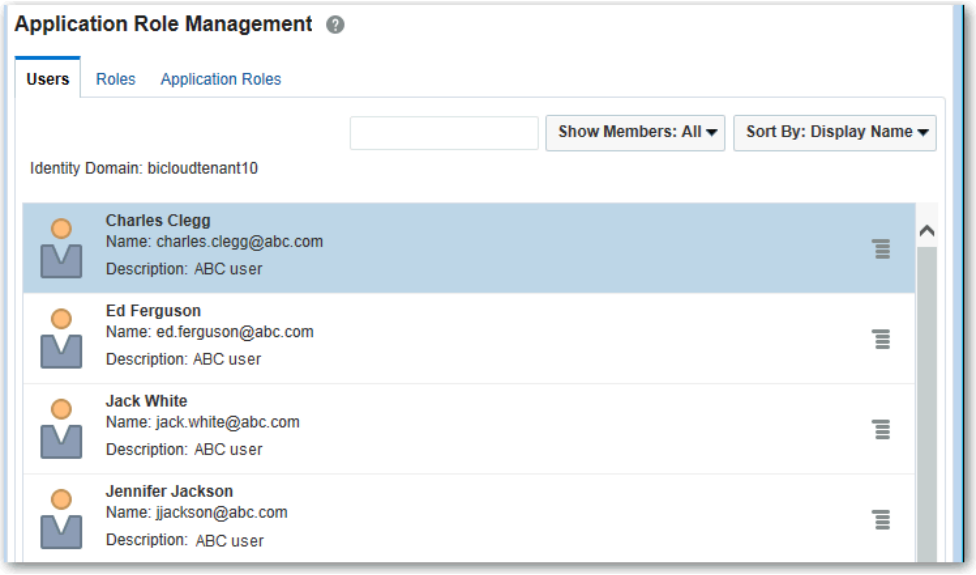
You need the BI Administrator application role to access administrative options in the Console.

There must always be at least one person in your organization with the BI Administrator application role. This ensures there is always someone who can delegate permissions to others. If you remove yourself from the BI Administrator role you’ll see a warning message.

Assign Application Roles to Users

The Users page lists all the users who can sign in to Oracle Analytics Server. The list of names comes directly from the identity domain associated with your instance. It’s the administrator’s job to assign users to appropriate application roles.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click the **Users** tab.



4. To show everyone, leave the **Search** field blank and click **Show Members: All**.

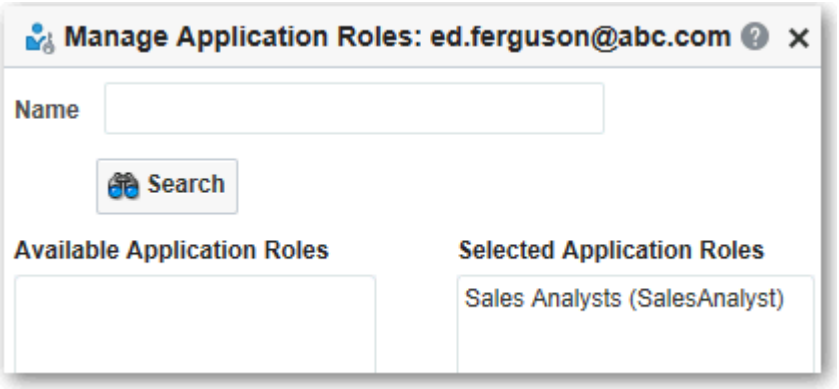
   To filter the list by name, enter all or part of a user name in the **Search** filter and press enter. The search is case-insensitive, and searches both name and display name.

5. To see what application roles are assigned to a user:

   a. Select the user.

   b. Click the action menu and select **Manage Application Roles**.

   The user's current application role assignments are displayed in the **Selected Application Roles** pane.



   For example, this image shows a user called Ed Ferguson assigned with the Sales Analysts application role.

6. To assign additional application roles or remove current assignments:

   a. Show available application roles. Click **Search** to display all the application roles.

      Alternatively, filter the list by **Name** and click **Search**.

b. Use the shuttle controls to move application roles between the **Available Application Roles** list and the **Selected Application Roles** list.

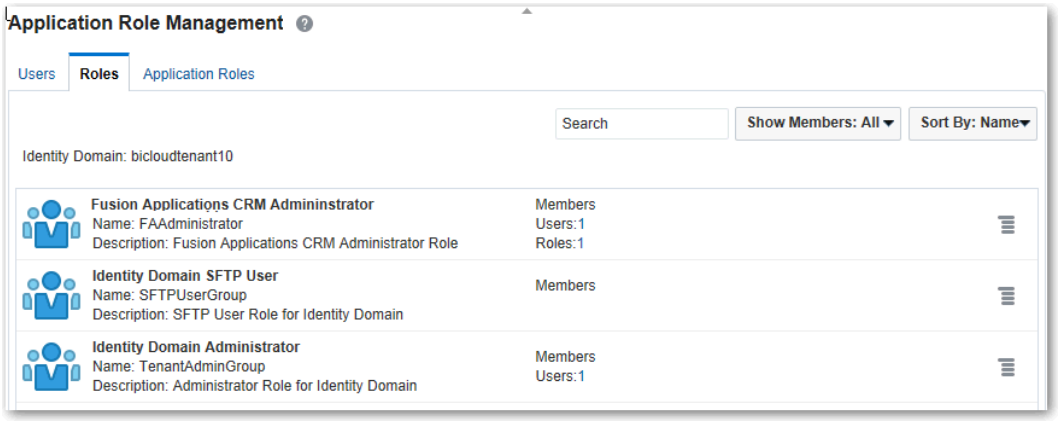
c. Click **OK**.

Assign Application Roles to Multiple Users Through Roles

The Roles page shows you all the roles that people signing in belong to in their identity domain. The list of roles comes directly from the identity domain associated with your instance. It's often quicker to assign privileges to multiple users through their predefined identity domain roles, than it is to assign privileges to users one by one.

You can assign application roles from the Roles page. You can also see who belongs to each role.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click the **Roles** tab.



4. Look in the **Members** area to see who belongs to each role:

   The number of users and roles that are members are displayed on the page. Click a number, such as **1** in this image, to see the members in more detail.

5. To display all available roles, leave the **Search** field blank and **Show Members: All**.

   To filter the list by name, enter all or part of a role name in the **Search** filter and press enter. The search is case-insensitive, and searches both name and display name.

   Alternatively, use the **Show Members** filter to list roles that are members of a particular application role or belong to another role.

6. To see the current application roles assignments:

   a. Select the role.

   b. Click the action menu and select **Manage Application Roles**.

   Current application role assignments display in the **Selected Application Roles** pane.

7. To assign additional application roles or remove them:

   a. Click **Search** to display all available application roles.

      Alternatively, enter all or part of an application role name and click **Search**.

b. Use the shuttle controls to move application roles between the **Available Application Roles** list and the **Selected Application Roles** list.

c. Click **OK**.

Add Your Own Application Roles

Oracle Analytics Server provides a set of predefined application roles. You can also create application roles of your own to suit your own requirements.

You must add new members or privileges after you create the custom application role.

For example, you can create an application role that only allows a select group of people to view specific folders or projects.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click the **Application Roles** tab.
4. Click **Add**.
5. Enter a name and describe the application role. Click **Save**.

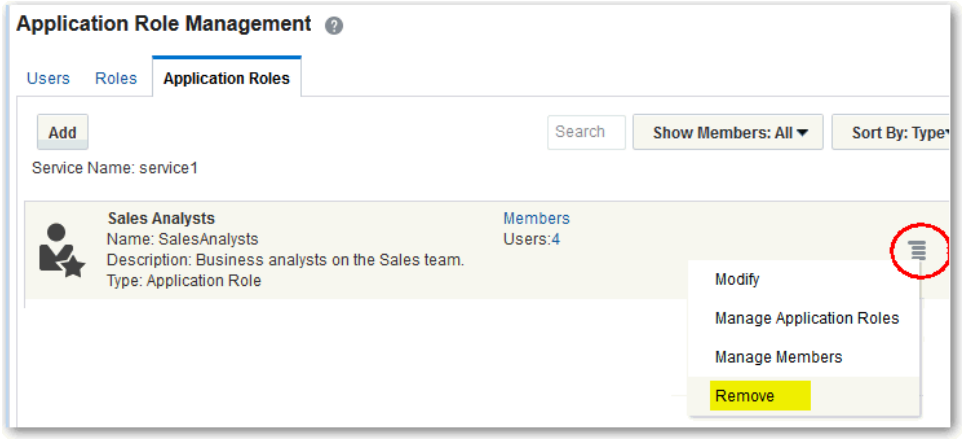
   Initially, new application roles don't have any members or privileges.
6. Add members to the application role:
   a. Click the action menu.
   b. Select **Manage Members**.
   c. Select the members (users, roles or application roles) that you want assigned to this application role and move them to the **Selected** pane on the right.

      For example, you might want an application role that restricts access to everyone in your organization, except sales managers. To do this, move anyone who is a sales manager, to the **Selected** pane.
   d. Click **OK**.
7. Optional: Add privileges to the new application role:
   a. Click the action menu.
   b. Select **Manage Application Roles**.
   c. Click **Search**.
   d. Move all the application roles you want this application role to inherit to the **Selected Application Roles** pane, and click **OK**.

Delete Application Roles

You can delete custom application roles that aren't needed.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click the **Application Roles** tab.
4. Click the action menu for the application role you want to delete and select **Remove**.



5. Click **OK**.

Add One Predefined Application Role to Another (Advanced)

Oracle Analytics Server provides several predefined roles: BI Service Administrator, BI Data Model Author, BI Data Load Author, BI Content Author, DV Content Author, DV Consumer, BI Consumer. There are very few, advanced use cases where you might want to *permanently* include one predefined application role in another.

Any changes that you make to predefined application roles are permanent, so don't perform this task unless you need to.

1. Click **Console**.
2. Click **Snapshots**.
3. Click **New Snapshot** to take a snapshot of your system before the change.

   The only way you can revert predefined application role changes is to restore your service from a snapshot taken before the change.
4. Go back to the Console, click **Users and Roles**.
5. Click the **Application Roles** tab.
6. Click the action menu for the predefined application role you want to change and select **Add Predefined Member (Advanced)**.

7. Click **Yes** to confirm that you've taken a snapshot and want to continue.
8. Select the predefined application role that you want to add.

   You can select only one application role.
9. Click **Yes** to confirm that you've taken a snapshot and want to permanently change the predefined application role.

Grant or Revoke Permission Assignments

Use the `grantPermissionSetsToBIRole` and `revokePermissionSetsFromBIRole` scripts to fine-tune permission assignments.

After you upgrade from Oracle BI EE to Oracle Analytics Server, Oracle Analytics Server automatically assigns any new permissions or permission sets to your application roles to make the new features available to users. Therefore it's important that you review how Oracle Analytics Server assigned these permissions. Use the scripts to make any necessary adjustments.

Certain features work only when permission sets are granted together. If you revoke an individual permission set, you might experience unforeseen side effects.

To grant or revoke permissions for an application role, run the appropriate script:

- *grantPermissionSetsToBIRole.sh*
- *revokePermissionSetsFromBIRole.sh*

Path: *Oracle/Middleware/Oracle_Home/user_projects/domains/bi/bitools/bin*

Usage:

```
./grantPermissionSetsToBIRole.sh [-d domainHome] [-s sikey] -r BIRoleName
-p PermissionSets
```

```
./revokePermissionSetsFromBIRole.sh [-d domainHome] [-s sikey] -r
BIRoleName -p PermissionSets
```

`-d`: Specify the domain home (including the final `domainName` directory). By default, the `DOMAIN_HOME` value is set. If the value isn't set, enter the actual domain home path.

`-s`: Specify the key for the service instance. The default is `ssi`.

`-r`: Specify the application role name.

`-p`: Specify the comma-separated list of permission sets.

For example:

```
./grantPermissionSetsToBIRole.sh -r myAdministrator -p
va.author,customScripts.admin
```

Table 2-1 Permission Sets Available in Oracle Analytics Server

Permission Set Name	Permissions
`actio.admin`	Administrator permissions to view and modify all jobs within the server instance, irrespective of the job owner.
`actio.author`	Permissions to view or modify jobs owned by the user.
`actio.operator`	Permissions to restart jobs. Doesn't include permissions to create jobs.
`actio.viewer`	View job scheduling permissions. (Not for Classic or Publisher)
`bilifecycle.admin`	Corresponding functionality not supported in Oracle Analytics Server.
`bip.administrator`	Publisher administration permissions.
`bip.author`	Publisher author permissions.
`bip.consumer`	Publisher consumer permissions.

Table 2-1 (Cont.) Permission Sets Available in Oracle Analytics Server

Permission Set Name	Permissions
bisecurity.admin	BI security administration permissions. (Internal API)
bisecurity.author	BI security author permissions. (Internal API)
bisecurity.GBUAdmin	Corresponding functionality not supported in Oracle Analytics Server.
bisecurity.impersonate	BI security impersonate permissions.
bisecurity.lifecycle.admin	Corresponding functionality not supported in Oracle Analytics Server.
customScripts.admin	Advanced analytics custom scripts administration permissions.
dataReplication.access	Data replication access permissions.
infer.administrator	Required social and storage providers configuration permissions.
majel.administrator	Mobile administration permissions.
obips.administrator	BI Presentation Server administration permissions.
obis.administrator	BI Server administration permissions.
obisch.administrator	BI Scheduler administration permissions. (For Classic)
obisch.author	BI Scheduler author permissions.
oracle.bi.dss.CustomKnowledge.admin	Data preparation custom knowledge administrator permissions.
oracle.bi.dss.CustomKnowledge.consumer	Data preparation custom knowledge consumer permissions.
oracle.bi.dss.SystemKnowledge.admin	Data preparation custom knowledge administration permissions.
oracle.bi.tech.dv.consumer	Data Visualization basic login permissions.
pod.admin	System settings administration permissions.
rdc.admin	Remote data connections for interoperability with Oracle Analytics Cloud. Corresponding functionality not supported in Oracle Analytics Server.
rdc.consumer	Remote data connections for interoperability with Oracle Analytics Cloud. Corresponding functionality not supported in Oracle Analytics Server.
rdc.monitor	Remote data connections for interoperability with Oracle Analytics Cloud. Corresponding functionality not supported in Oracle Analytics Server.
sac.advanced.approle.administrator	Application role user interface management permissions advanced features.
sac.approle.administrator	Oracle Analytics Console administration permissions to manage Users and Roles, Connections, and Virus Scanner configuration pages.
sac.snapshot.administrator	Snapshot administration permissions.
va.admin	Data Visualization administration permissions.

Table 2-1 (Cont.) Permission Sets Available in Oracle Analytics Server

Permission Set Name	Permissions
`va.author`	Data Visualization author permissions.
`va.interactor`	Data Visualization basic interaction permissions.

Manage Metadata Repository Privileges

Use Identity Manager in the Oracle BI Administration Tool to configure security in the Oracle BI repository.

Topics:

Use the Oracle BI Administration Tool

You use the Oracle BI Administration Tool to configure permissions for users and application roles against objects in the metadata repository.

If you log in to the Oracle BI Administration Tool in online mode, then you can view all users from the WebLogic Server.

If you log in to the Oracle BI Administration Tool in offline mode, then you can only view references to users that have previously been assigned metadata repository permissions directly in the RPD. The best practice is to assign metadata repository permissions to application roles rather than directly to users.

1. Log in to the Oracle BI Administration Tool and open a repository in **Online Mode**.
2. Optional: Select **Manage**, then **Identity**.
3. In the Identity Manager dialog, double-click an application role.
4. In the Application Role <Name> dialog, click **Permissions**.
5. In the **Object Permissions** tab view or configure the **Read** and **Write** permissions for that application role, in relation to objects and folders in the Oracle BI Presentation Catalog.
6. In the Presentation pane, expand a folder, then right-click an object to display the Presentation Table <Table name> dialog.
7. Click **Permissions** to display the Permissions <Table name> dialog.

Set Metadata Repository Privileges for an Application Role

The data model for your instance includes a security policy that defines permissions for accessing different parts of the data model, such as columns and subject areas.

The author of your data model uses the Oracle BI Administration Tool to maintain this security policy including assigning data model permissions to application roles.

When you import an application archive (BAR) file, Oracle Analytics Server uses the security policy for the data model in the archive file.

Best practice is to modify permissions for application roles, not modify permissions for individual users.

To view the permissions for an object in the Presentation pane, right-click the object and choose **Permission Report** to display a list of users and application roles and the permissions for the selected object.

1. Open the repository in the Oracle BI Administration Tool in Online mode.
2. In the Presentation panel, navigate to the subject area or sub-folder for which you want to set permissions.
3. Right-click the subject area or sub-folder, and select **Properties** to display the properties dialog.
4. Click *Permissions*.
5. In Permissions <*subject area name*> properties, click the **Show all users/application roles** if the check box is not checked.
6. In the Permissions <*subject area name*> dialog, update **User/Application Role** permissions to match your security policy.

   For example, to enable users to create dashboards and reports, you might change the repository permissions for an application role from *Read* to *Read/Write*.

Manage Application Roles in the Metadata Repository - Advanced Security Configuration Topic

Application role definitions are maintained in the policy store. The Administrator uses the Oracle Analytics Server Console to make any needed changes.

The repository maintains a copy of the policy store data to facilitate repository development. The Oracle BI Administration Tool displays application role data from the repository's copy; you aren't viewing the policy store data in real time. Policy store changes made while you are working with an offline repository aren't available in the Oracle BI Administration Tool until the policy store next synchronizes with the repository. The policy store synchronizes data with the repository copy whenever the BI Server restarts. If a mismatch in data is found, an error message is displayed.

While working with a repository in offline mode, you might discover that the available application roles do not satisfy the membership or permission grants needed at the time. A placeholder for an application role definition can be created in the Oracle BI Administration Tool to facilitate offline repository development. But this is just a placeholder visible in the Oracle BI Administration Tool and isn't an actual application role. You can't create an actual application role in the Oracle BI Administration Tool.

An application role must be defined in the policy store for each application role placeholder created using the Oracle BI Administration Tool before bringing the repository back online. If a repository with role placeholders created while in offline mode is brought online before valid application roles are created in the policy store, then the application role placeholder disappears from the Oracle BI Administration Tool interface. Always create a corresponding application role in the policy store before bringing the repository back online when using role placeholders in offline repository development.

Manage Session Variables

System session variables are session variables that Oracle BI Server and Oracle BI Presentation Services use for specific purposes.

System session variables have reserved names that can't be used for other kinds of variables such as static or dynamic repository variables and non-system session variables. Every active BI Server session generates session variables and initializes them. Each session variable instance can be initialized to a different value.

See Work with Session Variables.

Manage Server Sessions

The Administration Tool Session Manager is used in online mode to monitor activity.

The Session Manager shows all users logged in to the session, all current query requests for each user, and variables and their values for a selected session. Additionally, an administrative user can disconnect any users and terminate any query requests with the Session Manager.

How often the Session Manager data is refreshed depends on the amount of activity on the system. To refresh the display at any time, click **Refresh**.

You can also use the Oracle Analytics Server Console to check which users are logged in to the session. See Monitor Users Who Are Signed In.

Use the Session Manager

The Session Manager contains an upper pane and a lower pane:

- The top pane, the **Session** pane, shows users currently logged in to the BI Server. To control the update speed, from the **Update Speed** list, select **Normal**, **High**, or **Low**. Select **Pause** to keep the display from being refreshed.
- The bottom pane contains two tabs:
  - The **Request** tab shows active query requests for the user selected in the **Session** pane.
  - The **Variables** tab shows variables and their values for a selected session. You can click the column headers to sort the data.

The tables describe the columns in the Session Manager dialog.

Column Name	Description
Client Type	The type of client connected to the server.

Column Name	Description
Last Active Time	The time stamp of the last activity on the session.
Logon Time	The time stamp that shows when the session initially connected to the BI Server.
Repository	The logical name of the repository to which the session is connected.
Session ID	The unique internal identifier that the BI Server assigns each session when the session is initiated.
User	The name of the user connected.

Column Name	Description
Last Active Time	The time stamp of the last activity on the query.
Request ID	The unique internal identifier that the BI Server assigns each query when the query is initiated.
Session ID	The unique internal identifier that the BI Server assigns each session when the session is initiated.
Start Time	The time of the individual query request.

1. In the Administration Tool, open a repository in online mode and select **Manage** then **Sessions**.
2. Select a session and click the **Variables** tab.
3. To refresh the view, click **Refresh**.
4. To close Session Manager, click **Close**.

Follow these steps to disconnect a user from a session.

1. In the Administration Tool, open a repository in online mode and select **Manage** then **Sessions**.
2. Select the user in the Session Manager top pane.
3. Click **Disconnect**.

   The user session receives a message that indicates that the session was terminated by an administrative user. Any currently running queries are immediately terminated, and any outstanding queries to underlying databases are canceled.
4. To close the Session Manager, click **Close**.

Follow these steps to terminate an active query.

1. In the Administration Tool, open a repository in online mode and select **Manage** then **Sessions**.
2. Select the user session that initiated the query in the top pane of the Session Manager.

   After the user is highlighted, any active query requests from that user are displayed in the bottom pane.
3. Select the request that you want to terminate.
4. Click **Kill Request** to terminate the selected request.

The user receives a message indicating that the query was terminated by an administrative user. The query is immediately terminated, and any outstanding queries to underlying databases are canceled.

Repeat this process to terminate any other requests.

5. To close the Session Manager, click **Close**.

Manage Presentation Services Privileges

The catalog for your instance includes a security policy for Presentation Services privileges. These privileges determine access permission to Presentation Services functionality and catalog objects.

When you import an application archive (BAR) file, Oracle Analytics Server uses the security policy for the Presentation Services functionality and catalog.

You use application roles to manage privileges. When groups are assigned to application roles, the group members are automatically granted associated privileges in Presentation Services. This is in addition to the Oracle Analytics Server permissions.

> **Tip:**
>
> A list of application roles that a user is a member of is available from the **Roles and Groups** tab in the My Account dialog.

About Presentation Services Privileges

Presentation Services privileges are managed in the Administration Manage Privileges page, and they grant or deny access to features, such as the creation of analyses and dashboards.

Being a member of an application role that has been assigned Presentation Services privileges will grant those privileges to the user. The Presentation Services privileges assigned to application roles can be modified by adding or removing privilege grants using the Manage Privileges page in Presentation Services Administration.

Presentation Services privileges can be granted to users both explicitly and by inheritance. However, explicitly denying a Presentation Services privilege takes precedence over user access rights either granted or inherited as a result of group or application role hierarchy.

Topics:

Use Presentation Services Administration Page

You use the Administration page to configure user privileges.

As a best practice, you should assign Presentation Services permissions to application roles rather than directly to users.

1. Log in to Oracle Analytics Server with Administrator privileges.
2. Select the **Administration** link to display the Administration page.
3. Select the **Manage Privileges** link.
4. Select a link for a particular privilege to display the Privilege <*Privilege name*> dialog.
5. Click the **Add users/roles** icon (+) to display the Add Application Roles and Users dialog.

   Use the Add Application Roles and Users dialog to assign application roles to this privilege.

Set Presentation Services Privileges for Application Roles

If you create an application role, you must set appropriate privileges to enable users with the application role to perform various functional tasks.

For example, you might want users with an application role named BISalesAdministrator to be able to create Actions. In this case, you would grant them a privilege named Create Invoke Action.

If you create a new application role to grant Oracle Analytics Server permissions, then you must set Presentation Services privileges for the new role.

Explicitly denying a Presentation Services permission takes precedence over user access rights either granted or inherited as a result of group or application role hierarchy.

Existing Catalog groups are migrated during the upgrade process. Moving an existing Oracle BI Presentation Catalog security configuration to the role-based Oracle Fusion Middleware security model based requires that each Catalog group be replaced with a corresponding application role. To duplicate an existing Presentation Services configuration, replace each Catalog group with a corresponding application role that grants the same Oracle BI Presentation Catalog privileges. You can then delete the original Catalog group from Presentation Services.

1. Log in to Oracle BI Presentation Services as a user with Administrator privileges.
2. From the Home page in Presentation Services, select **Administration**.
3. In the Security area, click Manage Privileges.
4. Click an application role next to the privilege that you want to administer.
   For example, to administer the privilege named Access to Scorecard for the application role named BIConsumer, you would click the **BIConsumer** link next to Access to Scorecard.

   Use the Privilege <*privilege_name*> dialog to add application roles to the list of permissions, and grant and revoke permissions from application roles. For example, to grant the selected privilege to an application role, you must add the application role to the **Permissions** list.

5. Add an application role to the **Permissions** list, as follows:
    a. Click **Add Users/Roles**.
    b. Select **Application Roles** from the list and click **Search**.
    c. Select the application role from the results list.
    d. Use the shuttle controls to move the application role to the **Selected Members** list.
    e. Click **OK**.
6. Set the permission for the application role by selecting **Granted** or **Denied** in the **Permission** list.
7. Save your changes.

Encrypt Credentials (Advanced)

The BI Server and Presentation Services client support industry-standard security for login and password encryption.

When an end user enters a user name and password in a web browser, the BI Server uses the Hypertext Transport Protocol Secure (HTTPS) standard to send the information to a secure Presentation Services port. From Presentation Services, the information is passed through ODBC to the BI Server, using Triple DES (Data Encryption Standard). This provides a high level of security (168 bit) to prevent unauthorized users from accessing data or Oracle Analytics Server metadata.

At the database level, Oracle Analytics Server administrative users can implement database security and authentication. Proprietary key-based encryption provides security to prevent unauthorized users from accessing the metadata repository.

Manage Data Source Access Permissions With Oracle Analytics Server Publisher

You manage the data source access permissions stored in Publisher, using the Publisher Administration pages.

Data source access permissions control application role access to data sources. A user must be assigned to an application role which is granted specific data source access permissions that enable the user to perform the following tasks:

- Create a data model against the data source.
- Edit a data model against a data source.
- View a report created with a data model built from the data source.

Enable High Availability of the Default Embedded Oracle WebLogic Server LDAP Identity Store

Use this procedure to enable high availability in a clustered environment when using the default WebLogic LDAP identity store.

Configure the `virtualize` attribute to enable high availability of the default embedded Oracle WebLogic Server LDAP identity store in a clustered environment. When you set the `virtualize` attribute value to true, Oracle Analytics Server processes look to their local managed server where the processes can authenticate and perform lookups against a local copy of the embedded default Oracle WebLogic Server LDAP identity store.

Use lowercase for the property name `virtualize` . Use uppercase for the property name `OPTIMIZE_SEARCH`.

1. Log in to Fusion Middleware Control.
2. From the navigation pane expand the **WebLogic Domain** folder and select **bi**.
3. Right-click **bi** and select Security, then **Security Provider Configuration** to display the Security Provider Configuration page.
4. Expand **Security Store Provider**, and **Identity Store Provider** area, and click **Configure** to display the Identity Store Configuration page.
5. In the Custom Properties area, use the **Add** option to add the following custom properties:
   - Property `Name=virtualize Value=`*`true`*
   - Property `Name=OPTIMIZE_SEARCH Value=`*`true`*
6. Click **OK** to save the changes.
7. Restart the Administration server, any Managed servers, and Oracle Analytics Server components.

Use runcat to Manage Security Tasks in the Presentation Catalog

You can invoke the command line utility on supported platforms for Oracle Analytics Server such as Linux.

Enter a command such as the following one on Linux for assistance in using the command line utility:

```
./runcat.sh -help
```

Use the following syntax to convert a permission for a catalog group into a permission for an application role.

```
runcat.cmd/runcat.sh -cmd replaceAccountInPermissions -old <catalog_group_name> -
oldType group -new <application_role_name> -newType role -offline <catalog_path>
```

Reporting on Users Privileges for a Set of Presentation Services Catalog Items

Use the following syntax to report on all privileges in the Presentation Services Catalog, and who has those privileges. For example:

```
runcat.cmd/runcat.sh -cmd report -online http://localhost:8080/analytics/saw.dll
-credentials c:/oracle/catmancredentials.properties -outputFile c:/temp/
report.txt -delimiter "\t" -folder "/system/privs" -mustHavePrivilege -type
"Security ACL" -fields "Path:Accounts" "Must Have Privilege"
```

For help use the following command:

```
runcat.sh -cmd report -help
```

3

Use Alternative Authentication Providers

This chapter explains how to configure Oracle Analytics Server to use alternative directory servers for authentication instead of using the default Oracle WebLogic Server LDAP directory.

Topics:

About Alternative Authentication Providers

When you use an alternative authentication provider, you typically use administrative tools provided by your provider vendor to set up your users and groups. You can then assign these users and groups to the application roles defined in Oracle Analytics Server.

You continue to use the other tools such as, the Oracle BI Administration Tool, Oracle Analytics Server Console, and the Presentation Services Administration Page to manage the other areas of the security model.

If you use a directory server other than the default WebLogic LDAP Server, you can view the users and groups from the other directory server in Oracle WebLogic Server Administration Console. However, you must manage the users and groups in the interface for the directory server being used. For example, if you are using Oracle Internet Directory (OID LDAP), you must use OID Console to create and edit users and groups.

For a list of supported identity management systems, see Certification - Indentity Servers and Access.

High-Level Steps for Configuring an Alternative Authentication Provider

Use these steps as a general guide for configuring an alternative authentication provider.

1. Ensure your external Identity Store has all the users and groups setup for use with Oracle Analytics Server.
2. Configure the necessary authentication provider(s).
3. Go to the **myrealm\Users and Groups** tab to verify that the users and groups from the alternative authentication provider are displayed correctly. If the users and groups are displayed correctly, then proceed to the next step. Otherwise, reset your configuration settings and retry.

4. Assign application roles to groups using Oracle Analytics Server Console.

Set Up Groups and Users in the Alternative Authentication Provider

Before you use an alternative authentication provider, you must configure suitable groups and users. You then associate them with the application roles within your Oracle Analytics Server Instance. Follow these steps to set up an alternative authentication provider.

Oracle Analytics Server does not require or mandate any specific users or groups, and in a production environment your corporate Identity Store, for example Oracle Internet Directory (OID), would typically already contain users and groups relevant to you organization.

1. Create groups in the alternative authentication provider similar to the application roles from your Oracle Analytics Server instance. For example, BIServiceAdministrators, BIContentAuthors, BIConsumers.
2. Create users in the alternative authentication provider, corresponding to the created groups. For example, BISERVICEADMIN.
3. Assign the users to respective groups in the alternative authentication provider.

   For example, assign BISERVICEADMIN user to the BIServiceAdministrators group.
4. Make the BIContentAuthors group part of the BIConsumers group in the alternative authentication provider.

   This grouping enables BIContentAuthors to inherit permissions and privileges of BIConsumers.

Configure Oracle Analytics Server to Use Alternative Authentication Providers

Follow these options to configure Oracle Analytics Server to use one or more authentication providers instead of the default Oracle WebLogic Server LDAP directory.

Topics:

Reconfigure Oracle Internet Directory as an Authentication Provider

Use these steps to reconfigure the Oracle Internet Directory (OID) LDAP as the authentication provider.

> **Note:**
>
> If the **User Name Attribute**, or the **Group Name Attribute** is configured to a value other than *cn* in Oracle Internet Directory, you must change corresponding values in Oracle WebLogic Server Administration Console. The LDAP authenticators, including the `OracleInternetDirectoryAuthenticator` and the `ActiveDirectoryAuthenticator`, default to *cn* as the user name and group name attributes. You can use alternative attributes for the user name such as *uid* or *mail*.

1. Log in to Oracle WebLogic Server Administration Console.
2. In the Change Center, click **Lock & Edit**.
3. In Domain Structure, select **Security Realms**, and click **myrealm**.
4. Click the **Providers** tab, then click the **Authentication** tab.
5. Click **New**.
6. In Create a New Authentication Provider, in the **Name** field, type a name for the authentication provider such as *MyOIDDirectory*.
7. From the **Type** list, select *OracleInternetDirectoryAuthenticator*.
8. Click **OK** to save the changes and display the authentication providers list updated with the new authentication provider.
9. In the **Authentication Providers** table, under the **Name** column, click *MyOIDDirectory*.
10. In Settings for *MyOIDDirectory*, click the **Configuration** tab and then click the **Common** tab.
11. From the **Control Flag** list, select *SUFFICIENT*, and then click **Save**.
12. Click the **Provider Specific** tab, in the Connection properties, type your values for **Host**, **Port**, **Principal**, and **Credential**.
13. In the Provider Specific tab, Group area, specify value for the **Group Base DN** (distinguished name).
14. In the Provider Specific tab, Users area, specify the following:
    - **User Base DN**
    - **All Users Filter**
    - **User From Name Filter**
    - **Use Retrieved User Name as Principal**
    - **User Name Attribute**
15. Click **Save**.

You must also complete these tasks:

- Configuring the Default Authenticator Control Flag
- Reordering Authentication Providers

After completing the above tasks, in the Change Center, click **Activate Changes**, and then restart Oracle WebLogic Server.

Oracle Internet Directory Authenticator Provider Specific Reference

Review the table to complete the values required in the Oracle Internet Directory (OID) Authenticator.

Use this table to get the details about the fields in the Provider Settings page of the Settings for MyOIDDirectory.

Section Name	Field Name	Description
Connection	Host	The host name of the Oracle Internet Directory server.
Connection	Port	The port number on which the Oracle Internet Directory server is listening.
Connection	Principal	The distinguished name (DN) of the Oracle Internet Directory user to be used to connect to the Oracle Internet Directory server. For example: *cn=OIDUser,cn=users,dc=us,dc=mycompany,dc=com*.
Connection	Credential	The Password for the Oracle Internet Directory user entered as the *Principal*.
Groups	Group Base DN	The base distinguished name (DN) of the Oracle Internet Directory server tree that contains groups.
Users	User Base DN	The base distinguished name (DN) of the Oracle Internet Directory server tree that contains users.
Users	All Users Filter	The LDAP search filter. Click **More Info...** for details. Leave this blank, because it is the default value for the Active Directory authenticator. Any filter that you add to the **All Users Filter** is appended to all user searches.
Users	User From Name Filter	The LDAP search filter. Click **More Info...** for details.
Users	User Name Attribute	The attribute that you want to use to authenticate such as cn, uid, or mail. For example, to authenticate using a user's email address you set this value to `mail`. The value that you specify must match the User Name Attribute that you are using in the authentication provider.
Users	Use Retrieved User Name as Principal	Specifies whether or not the user name retrieved from the LDAP server should be used as the Principal in the Subject. Oracle recommends that you select this check box as it helps to enforce consistent case usage. For example, if your LDAP user name is JSmith, but you logged in as jsmith (lower case) the Principal is still JSmith (mixed case). This means that any application role memberships granted directly to users, instead of indirectly through groups, are consistently applied at authentication time.

Reconfigure Microsoft Active Directory as the Authentication Provider

Follow this procedure to reconfigure your Oracle Analytics Server installation to use Microsoft Active Directory.

The example data in this section uses a fictional company called XYZ Corporation that wants to set up SSO for Oracle Analytics Server for their internal users.

This example uses the following information:

- Active Directory domain

  The XYZ Corporation has an Active Directory domain, called *xyzcorp.com*, which authenticates all the internal users. When users log in to the corporate network, the log in to the Active Directory domain. The domain controller is *addc.xyzcorp.com*, which controls the Active Directory domain.

- Oracle Analytics Server WebLogic domain

  The XYZ Corporation has a WebLogic domain called *bi*, default name, installed on a network server domain called *bieesvr1.xyz2.com*.

- System Administrator and Test user

  The following system administrator and domain user test the configuration:

  – System Administrator user

    Jo Smith (login=jsmith, hostname=xyz1.xyzcorp.com)

  – Domain user

    Bob Jones (login=bjones hostname=xyz47.xyzcorp.com)

1. Log in to Oracle WebLogic Server Administration Console, and click **Lock & Edit** in the Change Center.
2. Select **Security Realms** from the left pane and click **myrealm**.

   myrealm is the default Security Realm.
3. Display the **Providers** tab, then display the **Authentication** sub-tab.
4. Click **New** to launch the Create a New Authentication Provider page.
5. Enter values in the Create a New Authentication Provider page as follows:
   - **Name**: Enter a name for the authentication provider. For example, ADAuthenticator.

     Type: Select ActiveDirectoryAuthenticator from the list.
   - Click **OK** to save the changes and display the authentication providers list updated with the new authentication provider.
6. Click **DefaultAuthenticator** in the **Name** column to display the Settings page.
7. In the Common Authentication Provider Settings page, change the **Control Flag** from REQUIRED to SUFFICIENT and click **Save**.
8. In the authentication providers table, click **ADDirectory** in the **Name** column to display the Settings page.
9. Display the **Configuration\Common** tab, and use the **Control Flag** list to select 'SUFFICIENT', then click **Save**.

10. Display the **Provider Specific** tab to access the options which apply specifically to connecting to an Active Directory LDAP authentication store.
11. Use the **Provider Specific** tab to specify the provider specific details.
12. Optional: If the User Name attribute, or the Group Name attribute is configured to a value other than *cn* in Microsoft Active Directory, you must change corresponding values in Oracle WebLogic Server Administration Console.

> **Note:**
>
> The LDAP authenticators provided by WebLogic including `OracleInternetDirectoryAuthenticator` and `ActiveDirectoryAuthenticator`, use *cn* as the default user name and group name attributes. You can use alternative attributes for the user name, for example *uid* or *mail*.

13. Click **Save**.
14. In Settings for myrealm page, click the **Providers** tab, then click the **Authentication** tab.
15. Click **Reorder**.
16. In the Reorder Authentication Providers page, select **ADDirectory** and use the arrow buttons to move it into the first position in the list, then click **OK**.
17. In the Change Center, click **Activate Changes**.
18. Restart Oracle WebLogic Server.

Microsoft Active Directory Authentication Provider Specific Reference

Review the table to complete the values required in the Microsoft Authenticator.

Use this table to get the details about the fields in the Provider Settings page of Microsoft Active Directory.

Section Name	Field Name	Description
Connection	Host	The name of the Active Directory server addc.xyzcorp.com.
Connection	Port	The port number on which the Active Directory server is listening (389).
Connection	Principal	The LDAP DN for the user that connects to Active Directory when retrieving information about LDAP users. For example: cn=jsmith,cn=users,dc=us,dc=xyzcorp,dc=com.
Connection	Credential/ Confirm Credential	Password for the specified Principal.
Groups	Group Base DN	The LDAP query used to find groups in AD. Only groups defined under this path will be visible to WebLogic. (CN=Builtin,DC=xyzcorp,DC=com).

Section Name	Field Name	Description
Users	User Base DN	The LDAP query used to find users in AD. CN=Users,DC=xyzcorp,DC=com
Users	User Name Attribute	Attribute used to specify user name in AD. Default value is cn. Do not change this value unless you know your Active Directory is configured to use a different attribute for user name.
Users	All Users Filter	LDAP search filter. Click **More Info...**for details.
Users	User From Name Filter	LDAP search filter. Blank by default in AD. Click **More Info...** for details.
Users	User Object class	The name of the user.
Users	Use Retrieved User Name as Principal	Specifies whether or not the user name retrieved from the LDAP server should be used as the Principal in the Subject. Click **More Info...** for details. Oracle recommends that you select this check box as it helps to enforce consistent case usage. For example, if your LDAP user name is JSmith, but you logged in as jsmith (lower case) the Principal is still JSmith (mixed case). This means that any application role memberships granted directly to users, instead of indirectly through groups, are consistently applied at authentication time.

Configure User and Group Name Attributes in the Identity Store

The LDAP authenticators provided by WebLogic, including OracleInternetDirectoryAuthenticator and ActiveDirectoryAuthenticator, default to using cn as the user name and group name attributes.

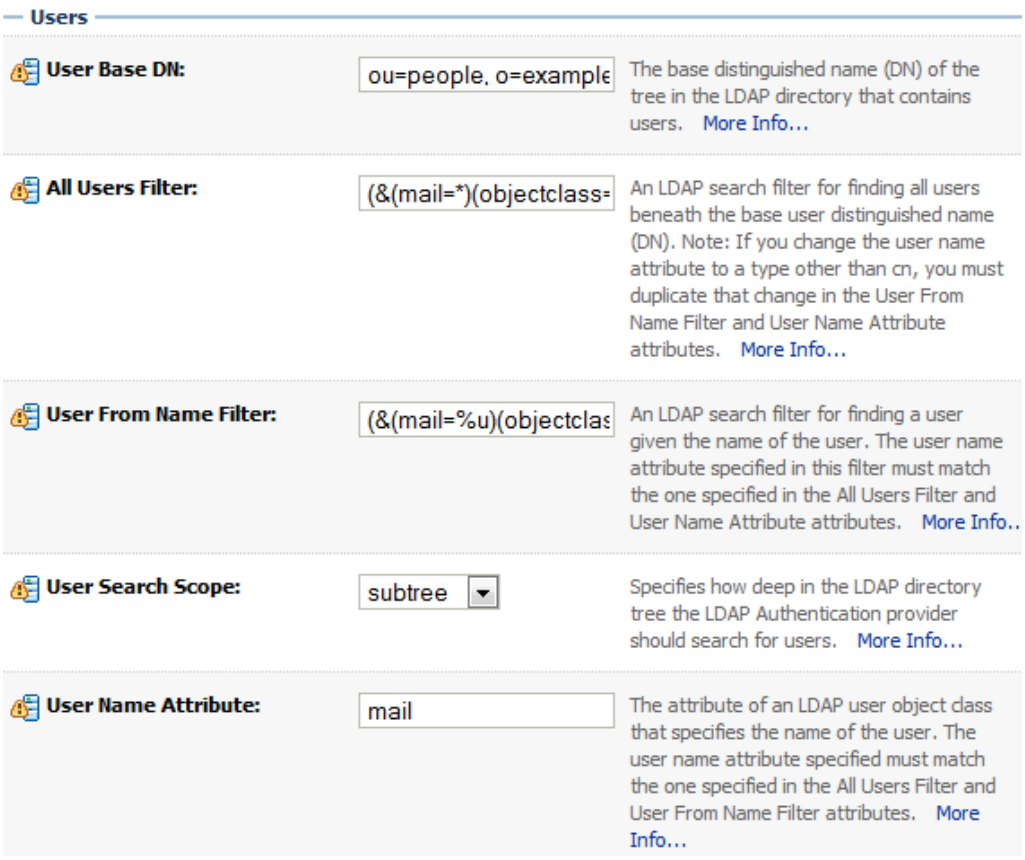
You might need to use alternative attributes for the user name, for example *uid* or *mail*. The need to use different group name attributes is less common. This section explains how to reconfigure user names and group names.

Topics:

Configure User Name Attributes

This section describes how to reconfigure the OracleInternetDirectoryAuthenticator (OID), for example, to use mail as the User Name Attribute.

The **Users** section shows the **User Name Attribute** configured with the value *mail*.



The `UserNameAttribute` in the alternative authentication provider is usually set to the value *cn*. If the `UserNameAttribute` is not set to *cn*, you must make sure the settings for AllUsersFilter and UserFromNameFilter are configured correctly as shown in the table. The table illustrates the default setting using the value *cn*, and a required new setting using a new value in the attribute `AnOtherUserAttribute`.

Attribute Name	Default Setting	Required New Setting
UserNameAttribute	cn	`AnOtherUserAttribute`
AllUsersFilter	(&(cn=*) (objectclass=person))	`(&(AnOtherUserAttribute =*) (objectclass=person))`
UserFromNameFilter	(&(cn=%u) (objectclass=person))	`(&(AnOtherUserAttribute =%u) (objectclass=person))`

Make the changes in the **Provider Specific** tab, substitute the AnOtherGroupAttribute setting with your own value.

Configure Group Name Attributes

You can configure the ActiveDirectoryAuthenticator to use a group name other than *cn*.

If the group name for Active Directory server is set to anything other than the default value *cn*, you must change the group name. If you change the value, you must also

change the values of *AllGroupsFilter* and *GroupFromNameFilter* as in the *AnOtherGroupAttribute* attribute.

Attribute Name	Default Setting	Required New Setting
StaticGroupNameAttribute/ DynamicGroupNameAttribute	cn	*AnOtherGroupAttribute*
AllGroupsFilter	`(&(cn=*) (objectclass=person))`	`(&(`*`AnOtherGroupAttribute`*` =*) (objectclass=person))`
GroupFromNameFilter	`(&(cn=%u) (objectclass=person))`	`(&(`*`AnOtherGroupAttribute`*` =%u) (objectclass=person))`

Make the changes in the **Provider Specific** tab, using the values in the table, substitute the *AnOtherGroupAttribute* setting with your own value. To display the Provider Specific tab, see Reconfigure Microsoft Active Directory as the Authentication Provider.

Configure LDAP as the Authentication Provider and Storing Groups in a Database

The examples provided in this section use Oracle Internet Directory (OID LDAP), and a sample database schema. However, you do not have to use OID LDAP as your LDAP identity store and your database schema does not have to be identical to the sample provided.

Oracle Analytics Server provides an authentication provider for WebLogic Server called BISQLGroupProvider that enables you to use this method. This authentication provider does not authenticate end user credentials but enables external group memberships held in a database table to contribute to an authenticated user's identity.

Topics:

Prerequisites

The following prerequisites must be satisfied before you attempt to configure LDAP authentication as described in this section:

- Oracle Analytics Server must be installed and configured.
- A suitable database schema containing at least one table with the required groups in it, and a mapping table which maps those groups to the names of users authenticated by LDAP must be running and accessible from the Oracle WebLogic Server on which Oracle Analytics Server is running.

- The configuration must include a supported LDAP server to use as the identity store that contains users.
- If you need Oracle Analytics Server to deliver content to members of an application role the following restrictions apply:
  - You can only pair a single LDAP authenticator with a single BISQLGroupProvider.

    When you configure multiple LDAP authenticators and want to retrieve group membership from the BISQLGroupProvider, content cannot be delivered to all members of an application role. In this configuration Oracle Analytics Delivers cannot resolve application role membership based on users and group membership.
  - You cannot define the same group in more than one identity store.

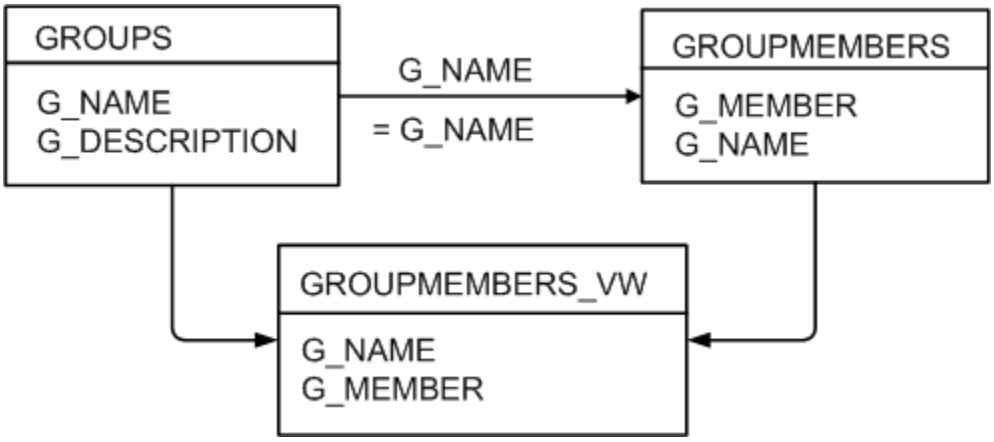
    You cannot have a group with the same name in both LDAP and database groups table. If you do, the security code invoked by Oracle Analytics Delivers cannot resolve application role membership.

Create a Sample Schema for Groups and Group Members

The sample schema described here is deliberately simplistic, and is intended only to illustrate how to configure Oracle Analytics Server to use the schema.

The `ACME_BI_GROUPS` sample schema contains two tables and a view. The `GROUPS` table defines the list of external groups. The `GROUPMEMBERS` table and `GROUPMEMBERS_VW` view describe group membership for users that exist in your primary identity store.

An advantage of defining tables or views identical to those shown in the diagram is that the configuration of the `BISQLGroupProvider` can use the default SQL outlined in the table in Configure the BISQLGroupProvider SQL Authenticator.



You must map the users in your LDAP store to groups in your database table by login name. In the diagram, the value of `G_MEMBER` in the `GROUPMEMBERS` table must match the value of the LDAP attribute used for login, for example, *uid*, *cn*, or *mail*, as specified in the LDAP authenticator. You should not, for example, map the database groups by *uid* if the login attribute is *mail*. Create a `GROUPMEMBERS_VW` view with an outer join between the `GROUPMEMBERS` and `GROUPS` tables.

Configure a Data Source and the BISQLGroupProvider Using Oracle WebLogic Server Administration Console

You configure a data source and the BISQLGroupProvider using Oracle WebLogic Server Administration Console as follows:

Topics:

Configure Oracle Internet Directory as the Primary Identity Store for Authentication Using Oracle WebLogic Server

Use the instructions in the link to configure WebLogic to authenticate your user population against OID LDAP.

See Reconfigure Oracle Internet Directory as an Authentication Provider.

> **Note:**
>
> When following the steps of this task, make a note of the value of the *User Base DN* and *User Name Attribute* in the Provider Specific configuration page for your OID LDAP authenticator for use later.

Install the BISQLGroupProvider

Before you can configure a BISQLGroupProvider authenticator, you must first install the JAR file bi-sql-group-provider.jar, which contains the authenticator. The file is available in the following location:

ORACLE_HOME`/bi/plugins/security/bi-sql-group-provider.jar`

You must copy the file to the following location:

ORACLE_HOME`/wlserver/server/lib/mbeantypes`

After copying the file into the specified location you must restart the Administration Server to enable the new provider to appear in the list of available authenticators.

> **Note:**
>
> If you install to create a clustered environment, then the installation cannot start the scaled-out Managed server because the `bi-sql-group-provider.jar` file is not available. When this situation occurs during installation, copy the Jar file to the correct location and click **Retry** in the installer.

Configure the Data Source Using Oracle WebLogic Server Administration Console

These steps enable you to configure the data source using Oracle WebLogic Server Administration Console.

1. Log in to Oracle WebLogic Server Administration Console, and click **Lock & Edit** in the Change Center.
2. Click **Services**, and click **Data Sources**.
3. In Summary of Data Sources, click **New**, and select **Generic Data Source**.
4. In JDBC Data Sources Properties , enter or select values for the following properties:
   - **Name**, for example, enter `BIDatabaseGroupDS`.

     The name used in the `config.xml` configuration file and throughout the Oracle WebLogic Server Administration Console whenever referring to this data source.

     JNDI Name , for example, enter `jdbc/BIDatabaseGroupDS`.

     The JNDI path to where the JDBC data source is bound.

     Database Type, for example, select *Oracle*.

     The DBMS of the database that you want to connect to.
5. Click **Next**.
6. Select a database driver from the **Database Driver** list.

   > **Note:**
   >
   > If using an Oracle database, select *Oracle's Driver (Thin) for Service Connections; Releases:9.0.1 and later*.
7. Click **Next**.
8. Click **Next**.
9. On the Connection Properties page, enter values for the following properties:
   - **Database Name** - The name of the database that you want to connect to.

     Host Name - for example, enter: `mymachine.example.com`.

     The DNS name or IP address of the server that hosts the database.

> **Note:**
>
> Do not use local host if you intend to use a cluster.

Port - For example, enter: *1521*.

The port on which the database server listens for connections requests.

Database User Name

Typically the schema owner of the tables defined in Create a Sample Schema for Groups and Group Members.

For example, enter `MYUSER`.

- **Password/Confirm Password**

  The password for the **Database User Name**.

  For example, enter *password*.

10. Click **Next**.
11. Check the details on the page are correct, and click **Test Configuration**.
12. Click **Next**.
13. In Select Targets, choose the servers or clusters as deployment targets for your data source.

    You should select the Administration Server and managed servers as your targets, for example:

    - In the Servers pane

      Select the **AdminServer** option.

    - In the Clusters pane

      Select the **bi_server1** check box to deploy to the cluster.

14. Click **Finish**.
15. In the Change Center, click **Activate Changes**.

> **Note:**
>
> In this example, the data source is called *BIDatabaseGroupDS*.

Configure the BISQLGroupProvider SQL Authenticator

Follow these steps to create a BISQLGroupProvider against the BIDatabaseGroupDS data source using an example table structure.

This task explains how to create a BISQLGroupProvider against the BIDatabaseGroupDS data source using the example table structure outlined in Create a Sample Schema for Groups and Group Members. You may need to modify the SQL statements used (table or column names) if your structure differs from the example.

> **Note:**
>
> There is no authentication against the database, as it just stores the groups to be associated with users. Authentication occurs against LDAP and the database is exposed when the BISQLGroupProvider assigns groups to application roles in Oracle WebLogic Server Administration Console.

1. Log in to Oracle WebLogic Server Administration Console as a WebLogic administrator, and click **Lock & Edit** in the Change Center.
2. Select **Security Realms** from the left pane and click **myrealm**.

   The default Security Realm is named **myrealm**.
3. Display the **Providers** tab, then display the **Authentication** sub-tab.
4. Click **New** to launch the Create a New Authentication Provider page.
5. Enter values in the Create a New Authentication Provider page as follows:
   - **Name**: Enter a name for the authentication provider. For example, MySQLGroupProvider.
   - From the **Type** list, select *BISQLGroupProvider*.
   - Click **OK** to save the changes and display the authentication providers list updated with the new authentication provider.
6. In the authentication providers table, click **MySQLGroupProvider** in the **Name** column to display the Settings page.
7. Display the **Provider Specific** tab to specify the SQL statements used to query and authenticate against your database tables.
8. Specify the **DataSource Name**. Don't use the JNDI name. For example: `jdbc/BIDatabaseGroupDS`.
9. Enter all of the SQL statements appropriate to your authenticator.

   The SQL is case sensitive.
10. Click **Save**.
11. Perform the following steps to reorder the authentication providers:
    a. Display the **Providers** tab.
    b. Click **Reorder** to display the Reorder Authentication Providers page
    c. Select **BISQLGroupProvider** and use the arrow buttons to move it into the first position in the list.
    d. Click **OK** to save your changes.
12. Perform the following steps to configure the **Control Flag** setting of **BISQLGroupProvider**:
    a. At the main Settings for myrealm page, display the **Providers** tab, then display the **Authentication** sub-tab, then select BISQLGroupProvider to display its configuration page.
    b. Display the **Configuration\Common** tab and select OPTIONAL from the **Control Flag** list.

c. Click **Save**.

13. In the Change Center, click **Activate Changes**.
14. Restart the Oracle Analytics Server components, use Fusion Middleware Control once the Administration Server has been restarted, Oracle WebLogic Server, and Managed servers.

> **Note:**
>
> Check the **Users and Groups** tab to confirm that the database users and groups appear there.

Configure the Virtualized Identity Store

You configure the virtualized identity store as follows:

Topics:

- Enable Virtualization by Configuring the Identity Store
- Configure SSL Against LDAP
- Configure a Database Adaptor to Retrieve Group Information

Enable Virtualization by Configuring the Identity Store

You configure the identity store to enable virtualization enabling the use of multiple identity stores with the identity store service.

You can split the user profile information across different authentication providers (identity stores), see Configure Identity Store Virtualization Using Fusion Middleware Control.

Configure SSL Against LDAP

If you have configured an LDAP Authenticator to communicate over SSL (one-way SSL only), you must put the corresponding LDAP server's route certificate in an additional keystore used by the virtualization (libOVD) functionality.

See Configure SSL when Using Multiple Authenticators.

Configure a Database Adaptor to Retrieve Group Information

You configure a database adaptor to make it appear like an LDAP server to enable the virtualized identity store provider to retrieve group information from a database using the database adapter.

In this task you create a file containing the elements for an adapter templates that specifies how to use your database tables as an identity store to map groups. The file describes the mapping of the `GROUPMEMBERS_VW` view to a virtual LDAP store. The view uses an outer join to ensure that you can reference fields from more than one table by the database adaptor.

1. Create a file named *bi_sql_groups_adapter_template.xml*.
2. Adapt the following elements to match your table and column attributes against LDAP server attributes.

> **Note:**
>
> For the element:
>
> ```
> <param name="ReplaceAttribute"
> value="uniquemember={cn=%uniquemember%,cn=users,dc=oracle,dc=com}"/>
> ```
>
> This must match the user attribute and root User DN of the main authenticator. For example, for the default authenticator:
>
> ```
> uid=%uniquemember%,ou=people,ou=myrealm,dc=bifoundation_domain
> ```

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<adapters schvers="303" version="1" xmlns="http://www.octetstring.com/
schemas/Adapters" xmlns:adapters="http://www.w3.org/2001/XMLSchema-instance">
   <dataBase id="directoryType" version="0">
      <root>%ROOT%</root>
      <active>true</active>
      <serverType>directoryType</serverType>
      <routing>
         <critical>true</critical>
         <priority>50</priority>
         <inclusionFilter/>
         <exclusionFilter/>
         <plugin/>
         <retrieve/>
         <store/>
         <visible>Yes</visible>
         <levels>-1</levels>
         <bind>true</bind>
         <bind-adapters/>
         <views/>
         <dnpattern/>
      </routing>
      <pluginChains xmlns="http://xmlns.oracle.com/iam/management/ovd/config/
plugins">
         <plugins>
            <plugin>
               <name>VirtualAttribute</name>

<class>oracle.ods.virtualization.engine.chain.plugins.virtualattr.VirtualAttr
ibutePlugin</class>
               <initParams>
                  <param name="ReplaceAttribute"
value="uniquemember={cn=%uniquemember%,cn=users,dc=oracle,dc=com}"/>
               </initParams>
            </plugin>
         </plugins>
         <default>
            <plugin name="VirtualAttribute"/>
         </default>
         <add/>
         <bind/>
         <delete/>
         <get/>
         <modify/>
         <rename/>
      </pluginChains>
```

```
      <driver>oracle.jdbc.driver.OracleDriver</driver>
      <url>%URL%</url>
      <user>%USER%</user>
      <password>%PASSWORD%</password>
      <ignoreObjectClassOnModify>false</ignoreObjectClassOnModify>
      <includeInheritedObjectClasses>true</includeInheritedObjectClasses>
      <maxConnections>10</maxConnections>
      <mapping>
         <joins/>

         <objectClass name="groupofuniquenames" rdn="cn">
            <attribute ldap="cn" table="GROUPMEMBERS_VW" field="G_NAME" type=""/>
            <attribute ldap="groupnameattr" table="GROUPMEMBERS" field="G_NAME"
type=""/>
            <attribute ldap="description" table="GROUPMEMBERS_VW" field="G_NAME"
type=""/>
            <attribute ldap="uniquemember" table="GROUPMEMBERS_VW"
field="G_MEMBER" type=""/>
            <attribute ldap="orclguid" table="GROUPMEMBERS" field="G_NAME"
type=""/>
         </objectClass>
      </mapping>
      <useCaseInsensitiveSearch>true</useCaseInsensitiveSearch>
      <connectionWaitTimeout>10</connectionWaitTimeout>
      <oracleNetConnectTimeout>0</oracleNetConnectTimeout>
      <validateConnection>false</validateConnection>
   </dataBase>
</adapters>
```

3. Customize appropriate sections for the following elements:

   - ReplaceAttribute

     Specifies how to define the unique member for a group. The `%uniquemember%` is a placeholder for a value that is passed at runtime when looking up whether a user is a member of a group.

     The only aspect of this element you may want to change is the specification of the root for your users. While this is notional, by default it must match whatever you specify as the root of your user population when you run the `libovdadapterconfig` script in Step 7.

   - groupofuniquenenames

     Specifies how group attributes are mapped to database fields.

     You must map the following attributes:

     – *cn* maps to a unique name for your group.

     – **uniquemember** maps to the unique name for your user in the user/group mapping table in your database schema.

     Mapping the following attribute is optional:

     – **description** is optional.

     No other attributes are configurable.

4. Copy the adapter file into the following folder:

   ORACLE_HOME`/oracle_common/modules/oracle.ovd/templates/`

5. Open a command prompt/terminal at:

   ORACLE_HOME`/oracle_common/bin`

6. Ensure the following environment variables are set, for example:
   - `ORACLE_HOME=oraclehome`
   - `WL_HOME=`*`ORACLE_HOME`*`/wlserver/`
   - `JAVA_HOME=`*`ORACLE_HOME`*`/jdk/jre`
7. Run the `libovdadapterconfig` script to create a database adapter from the template file. The syntax is:

```
libovdadapterconfig -adapterName <name of adapter> -adapterTemplate <name
(NOT including path) of template file which defines adapater> -host
localhost -port <Admin Server port> -userName <user id of account which has
administrative privileges in the domain> -domainPath <path to the BI domain>
-dataStore DB -root <nominal specification of a pseudo-LDAP query to treat
as the "root" of this adapter - must match that specified in template for
adapter 2 above> -contextName default -dataSourceJNDIName <JNDI name for
DataSource which points at the database being mapped>
```

   For example:

```
./libovdadapterconfig.sh -adapterName biSQLGroupAdapter -adapterTemplate
bi_sql_groups_adapter_template.xml -host localhost -port 9500 -userName
weblogic -domainPath /opt/oracle_bi/user_projects/domains/
bifoundation_domain/ -dataStore DB -root cn=users,dc=oracle,dc=com -
contextName default -dataSourceJNDIName jdbc/BIDatabaseGroupDS
```

> **Note:**
>
> Use the *JNDI* name and not just the *DS* name for the *dataSourceJNDIName*.

> **Note:**
>
> The root parameter value should match the root *dn* specified in the `<param name>="replaceattribute"` element in the adaptor template. For example, if user is specified in the default authenticator, set the root to *ou=people*, *ou=myrealm*, *dc=bifoundation_domain*.

   The script should exit without error.
8. Restart WebLogic Administration Server and Managed servers.

> **Note:**
>
> When you start WebLogic, you can ignore the following `Warning: BISQLGroupsProvider: Connection pool not usable.`

Log in to WebLogic and Oracle Analytics Server using credentials stored in the database.

Test the Configuration by Adding a Database Group to an Application Role

You can test the configuration by adding a database group to an application role.

1. Log in to Fusion Middleware Control, and open WebLogic domain and *bifoundation_domain* in the navigation menu on the left of the page.
2. Right-click **bifoundation_domain** and select **Security**, then **Application Roles** to display the Application Role Configuration page.
3. Add a database group which contains an LDAP user to one of the application roles, for example, BIServiceAdministrator, which that user does not currently have access to.
4. Log in to Oracle Analytics Server as a user that is a member of the group that was newly added to the application role.

   In the top right of the page, you will see the text `Logged in as <user id>`.
5. Click the user id to display a drop down menu.
6. Select **My Account** from the menu.
7. Display the **Roles and Catalog Groups** tab and verify the user now has the new application role.

Correct Errors in the Adaptors

You cannot modify an existing database adapter, so if you make an error in either the libovdadapter command, or the templates you use to create the adapters, you must delete then recreate the adapter.

See Correct Database Adapter Errors by Deleting and Recreating the Adapter.

Configure a Database as the Authentication Provider

This section describes how to configure Oracle Analytics Server to use a database as the authentication provider by using a SQLAuthenticator and a virtualized identity store database adapter, and contains the following topics:

Topics:

Introduction and Prerequisites

User role and profile information can be stored in a database with the help of an adapter that enables the database to appear like an LDAP server. A virtualized identity store provider can retrieve user profile information from a database through a database adapter.

This topic explains how to configure Oracle Analytics Server with a SQLAuthenticator and a virtualized identity store provider including a database adapter, both running against a suitable database schema. The examples given are illustrative only, and your database schema need not be identical to the sample described here.

Use this procedure when you need to authenticate users against a database schema. The preferred identity store for authentication purposes is an LDAP directory service, such as Oracle Internet Directory (OID LDAP).

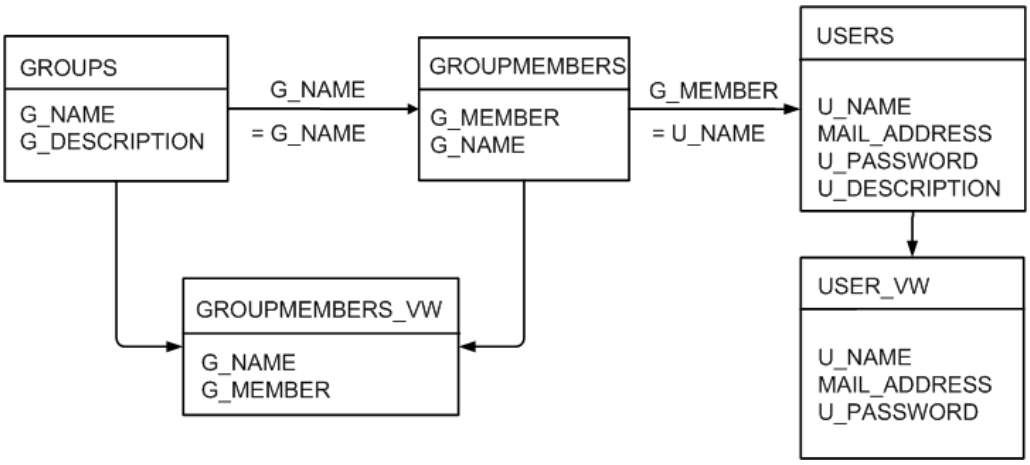
The approach to database authentication described here requires two database columns, one containing users and another containing passwords. This method is not based on database user accounts.

Create a Sample Schema for Users and Groups

You have schemas that you were using in an earlier installation of Oracle Analytics Server. This sample schema is intended to illustrate how to configure the system to use this schema.

> **Note:**
>
> You must use a database schema containing the users, credentials and groups required for authentication that is accessible from the WebLogic Server where Oracle Analytics Server is running.

The diagram shows tables, `USERS`, `USER_VW`, `GROUPMEMBERS`, `GROUPS`, and `GROUPMEMBERS_VW`, where `USER_VW` is a view on the `USERS` table, and `GROUPMEMBERS_VW` is a view joining the `GROUPMEMBERS` and `GROUPS` tables.



If user or group information exists in more than one table, remove `USER_VW` must create a view over the tables of each type of information.

Create a view on the `GROUPMEMBERS` and `GROUPS` tables, for example, `GROUPMEMBERS_VW`, with an outer join on the `GROUPS` table and an inner join on the `GROUPMEMBERS` table, which enables you to see groups in Fusion Middleware Control even when they have no user assigned to them. To present the view shown in the diagram to the database

adapter, you would need to follow the configuration shown in Configure a Database Adaptor.

Configure a Data Source and SQL Authenticator Using the Oracle WebLogic Server Administration Console

You configure a data source and SQL authenticator using the Oracle WebLogic Server Administration Console as follows:

Topics:

- Configure a Data Source Using the Oracle WebLogic Server Administration Console
- Configure a SQL Authenticator Using the Oracle WebLogic Server Administration Console
- SQL Authenticator Select Statement Reference
- Configuring the Default Authenticator Control Flag
- Reordering Authentication Providers

Configure a Data Source Using the Oracle WebLogic Server Administration Console

Use these steps to configure a data source using the Oracle WebLogic Server Administration Console.

The schema owner of the tables is defined in Create a Sample Schema for Users and Groups.

1. Log in to Oracle WebLogic Server Administration Console, navigate to the Change Center, click **Lock & Edit**.
2. Click **Services** and click **Data Sources**.
3. In the Summary of Data Sources page, click **New**, and select **Generic Data Source**.
4. In the JDBC Data Sources Properties page, enter or select values for the following properties:
   - **Name** - For example, enter: UserGroupDS

     The name used in the underlying configuration file (config.xml) and throughout the Administration Console whenever referring to this data source.
   - **JNDI Name** - For example, enter: `jdbc/UserGroupDS`

     The JNDI path to which this JDBC data source is bound.
   - **Database Type** - For example, select: Oracle

     The DBMS of the database that you want to connect to.
5. Click **Next**.
6. Select a database driver from the **Database Driver** list.

   For example, select: Oracle's Driver (Thin) for Service Connections; Releases:9.0.1 and later
7. Click **Next**.
8. Click **Next**.
9. On the Connection Properties page, enter values for the following properties:

- **Database Name** - For example, enter: `ora12c`

  The name of the database that you want to connect to.
- **Host Name** - For example, enter: `mymachine.example.com`

  The DNS name or IP address of the server that hosts the database.
- **Port** - For example, enter: `1521`

  The port on which the database server listens for connections requests.
- **Database User Name**
- **Password/Confirm Password**

  The password for the **Database User Name**.

10. Click **Next**.
11. Check the details on the page are correct, and click **Test Configuration**.
12. Click **Next**.
13. In the Select Targets page select the servers or clusters for deploying the data source.

    You should select the Administration Server and Managed server as your targets, for example:

    - In the Servers pane

      Select the **AdminServer** check box.
    - In the Clusters pane

      Select the **bi_server1** option.
14. Click **Finish**.
15. In the Change Center, click **Activate Changes**.
16. Restart the system.

Configure a SQL Authenticator Using the Oracle WebLogic Server Administration Console

A user with the appropriate privileges can log in to the Oracle WebLogic Server Administration Console using the WebLogic database authenticator.

When creating the SQL authenticator, select the read-only SQL authenticator. The read-only authentication provider type does not write back to the database.

When entering the SQL statements in the Provider Specific tab, if your password column is in plain text as the result of the query supplied for the **SQL Get Users Password** column was not hashed or encrypted, select the **Plaintext Password Enabled** option.

If the **Plaintext Password Enabled** option is cleared, the `SQLAuthenticator` expects passwords hashed using SHA-1, default encryption algorithm. For more information on the supported encryption algorithms, see the documentation for the base `SQLAuthenticator Mbean PasswordAlgorithm` attribute.

See SQL Authenticator Select Statement Reference for help in defining the **Provider Specific** SQL statements.

1. Log in to Oracle WebLogic Server Administration Console.

2. In the Change Center, click **Lock & Edit**.
3. From Domain Structure, select **Security Realms** and click **myrealm**.
4. In Settings for myrealm, click the **Providers** tab, and then click the **Authentication** tab.
5. In **Authentication Provider**s, click **New**.
6. In Create a New Authentication Provider, in **Name** type a name for the authentication providers such as `UserGroupDBAuthenticator`.
7. From the **Type** list, select *ReadOnlySQLAuthenticator*, and click **OK**.
8. From the **Authentication Providers** table, select the provider you just created.
9. In the Settings for <*your new authentication provider name*>, click the **Provider Specific** tab.
10. Optional: In the **Provider Specific** tab, if your password column is in plain text, select **Plaintext Password Enabled**.
11. In the **Data Source Name** field, type the name of an existing data source, for example, *UserGroupsDS*, to use this authentication provider.

    The data source name must match the existing data sources defined in Oracle WebLogic Server Administration Console.
12. In the **Provider Specific** tab, specify the SQL statements used to authenticate user access and to query your database tables.
13. After entering all of the required SQL statements for your authenticator, click **Save**.

You must configure the authentication provider control flag when using multiple authentication providers.

SQL Authenticator Select Statement Reference

Learn options available for creating SQL statements when implementing a SQL authentication provider.

When you create a SQL Authenticator in the **Provider Specific** tab, you specify the SQL statements used to query, and authenticate against, your database tables. See Configuring a SQL Authenticator Using the Oracle WebLogic Server Administration Console.

The table shows SQL statements for the sample schema outlined in Create a Sample Schema for Users and Groups.

If you are using a different table structure, you might need to adapt these SQL statements with the table or column names of your schema. You should use the question mark (?) as a runtime query placeholder rather than hard coding a user or group name.

Query	SQL	Notes
SQL Get Users Password	`SELECT U_PASSWORD FROM USERS WHERE U_NAME = ?`	This SQL statement looks up a user's password. The SQL statement requires a single parameter for the *username* and must return a `resultSet` containing at most a single record containing the password.
SQL User Exists	`SELECT U_NAME FROM USERS WHERE U_NAME = ?`	This SQL statement looks up a user. The SQL statement requires a single parameter for the *username* and must return a `resultSet` containing at most a single record containing the user.

Query	SQL	Notes
SQL List Users	`SELECT U_NAME FROM USERS WHERE U_NAME LIKE ?`	This SQL statement retrieves users that match a specific wildcard search. The SQL statement requires a single parameter for the *usernames* and returns a `resultSet` containing matching *usernames*.
SQL List Groups	`SELECT G_NAME FROM GROUPS WHERE G_NAME LIKE ?`	This SQL statement retrieves group names that match a wildcard. The SQL statement requires a single parameter for the group name and returns a `resultSet` containing matching groups.
SQL Group Exists	`SELECT G_NAME FROM GROUPS WHERE G_NAME = ?`	This SQL statement looks up a group. The SQL statement requires a single parameter for the group name, and must return a `resultSet` containing at most a single record containing the group.
SQL Is Member	`SELECT G_MEMBER FROM GROUPMEMBERS WHERE G_NAME=? AND G_MEMBER LIKE ?`	This SQL statement looks up members of a group. The SQL statement requires two parameters, a group name and a member or group name. This SQL statement must return a `resultSet`.
SQL List Member Groups	`SELECT G_NAME FROM GROUPMEMBERS WHERE G_MEMBER = ?`	This SQL statement looks up the group membership of a user or group. The SQL statement requires a single parameter for the *username* or group name, and returns a `resultSet` containing the names of the groups that matched the criteria.
SQL Get User Description	`SELECT U_DESCRIPTION FROM USERS WHERE U_NAME = ?`	This SQL statement retrieves the description of a specific user. The SQL statement is valid only if `Descriptions Supported` is enabled. The SQL statement requires a single parameter for the *username* and must return a `resultSet` containing at most a single record containing the user description.
SQL Get Group Description	`SELECT G_DESCRIPTION FROM GROUPS WHERE G_NAME = ?`	This SQL statement retrieves the description of a group. The SQL statement is valid only if `Descriptions Supported` is enabled. The SQL statement requires a single parameter for the group name and must return a `resultSet` containing at most a single record containing the group description.

Configure the Default Authenticator Control Flag

Use a JAAS Control Flag for each provider to control how the authentication providers are used in the login sequence.

You must complete this task if you are using multiple authentication providers.

1. From the *myrealm* Settings page, click the **Providers** tab, and then click the **Authentication** tab.
2. From the Authentication Providers table, select **DefaultAuthenticator**.
3. In Settings for DefaultAuthenticator on the **Configuration** page in the **Common** tab, from the **Control Flag** list, select *SUFFICIENT*.

4. Click **Save**.

Reorder Authentication Providers

After adding a new authenticator, you can reorder the Authentication Providers table.

1. From the *myrealm* Settings page, click the **Providers** tab, and then click the **Authentication** tab.
2. In the **Authentication Providers** table, click **Reorder**.
3. In Reorder Authentication Providers, from **Available**, select the provider to use as the default, click the up arrow, and then click **OK**.
4. In the Change Center, click **Activate Changes**.

After restarting the Administration Server, use the Fusion Middleware Control to restart the Oracle Analytics Server components, Oracle WebLogic Server, and managed servers.

Configure the Virtualized Identity Store

Configure the virtualized identity store as follows:

Topics:

- Enabling Virtualization by Configuring the Identity Store
- Configure a Database Adaptor

Configure a Database Adaptor

Follow these steps to configure a database adaptor to make the database appear like an LDAP server. This enables the virtualized identity store provider to retrieve user profile information from a database using the database adapter.

This task shows how to edit and apply adapter templates that specify how to use your database tables as an identity store. The example given here is for the sample schema that is used throughout Configure a Database as the Authentication Provider.

When customizing the `adapter_template_usergroup1.xml` file, map the elements by matching the classes and attributes used in a virtual LDAP schema with the columns in your database. The virtual schema is the same as that of WebLogic Embedded LDAP, you can map database columns to any of the attributes shown in the table.

The following is the schema file example:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<adapters schvers="303" version="1" xmlns="http://www.octetstring.com/schemas/
Adapters" xmlns:adapters="http://www.w3.org/2001/XMLSchema-instance">
   <dataBase id="directoryType" version="0">
      <root>%ROOT%</root>
      <active>true</active>
      <serverType>directoryType</serverType>
      <routing>
         <critical>true</critical>
         <priority>50</priority>
         <inclusionFilter/>
         <exclusionFilter/>
         <plugin/>
         <retrieve/>
         <store/>
```

```
         <visible>Yes</visible>
         <levels>-1</levels>
         <bind>true</bind>
         <bind-adapters/>
         <views/>
         <dnpattern/>
      </routing>
      <pluginChains xmlns="http://xmlns.oracle.com/iam/management/ovd/config/
plugins">
         <plugins>
            <plugin>
               <name>DBGUID</name>

<class>oracle.ods.virtualization.engine.chain.plugins.dbguid.DBGuidPlugin</class>
               <initParams>

                                        <param name="guidAtribute"
value="orclguid"/>
               </initParams>
            </plugin>
         </plugins>
         <default>
            <plugin name="DBGUID"/>
         </default>
         <add/>
         <bind/>
         <delete/>
         <get/>
         <modify/>
         <rename/>
      </pluginChains>
      <driver>oracle.jdbc.driver.OracleDriver</driver>
      <url>%URL%</url>
      <user>%USER%</user>
      <password>%PASSWORD%</password>
      <ignoreObjectClassOnModify>false</ignoreObjectClassOnModify>
      <includeInheritedObjectClasses>true</includeInheritedObjectClasses>
      <maxConnections>10</maxConnections>
      <mapping>
         <joins/>
                        <objectClass name="person" rdn="cn">
                        <attribute ldap="cn" table="USER_VW" field="U_NAME"
type=""/>
                        <attribute ldap="uid" table="USER_VW" field="U_NAME"
type=""/>
                        <attribute ldap="usernameattr" table="USER_VW"
field="U_NAME" type=""/>
                        <attribute ldap="loginid" table="USER_VW" field="U_NAME"
type=""/>
                        <attribute ldap="description" table="USER_VW"
field="U_NAME" type=""/>
                        <attribute ldap="orclguid" table="USER_VW" field="GUID"
type=""/>
                        </objectClass>
      </mapping>
      <useCaseInsensitiveSearch>true</useCaseInsensitiveSearch>
      <connectionWaitTimeout>10</connectionWaitTimeout>
      <oracleNetConnectTimeout>0</oracleNetConnectTimeout>
      <validateConnection>false</validateConnection>
   </dataBase>
</adapters>
```

In the <objectClass> element:

- The name="person" and rdn="cn" values declare the mapping of the LDAP person object class.
- The cn attribute is used as its Relative Distinguished Name (RDN).
- The child elements declare the LDAP attributes mapping to tables and columns in the database, for example:

  The line <attribute ldap="uid" table="USER_VW" field="USER_ID" type=""/> maps the USER_ID field of the USER_VW table to the standard LDAP attribute uid, a unique user id for each user.
- The USER_VW view should have a GUID column to match the orclguid attribute mapped to GUID column in adapter_template_usergroup1.xml, for example:

  You could CREATE or REPLACE VIEW USER_VW as the following:

  ```
  SELECT U_NAME, MAIL_ADDRESS, U_PASSWORD, U_DESCRIPTION, RPAD(U_NAME, 16, '0') AS
  GUID FROM USERS;
  ```

Attribute	Example
description	John Doe
cn	john.doe
uid	john.doe
sn	Doe
userpassword	password
displayName	John Doe
employeeNumber	12345
employeeType	Regular
givenName	John
homePhone	650-555-1212
mail	john.doe@example.com
title	Manager
manager	uid=mary.jones,ou=people,ou=myrealm,dc=wc_domain
preferredLanguage	en
departmentNumber	tools
facsimiletelephonenumber	650-555-1200
mobile	650-500-1200
pager	650-400-1200
telephoneNumber	650-506-1212
postaladdress	200 Oracle Parkway
l	Redwood Shores
homepostaladdress	123 Main St., Anytown 12345

You map groups using the same method as you used for mapping a person. When mapping groups, in the <objectClass name="groupofuniquenames" ...> element, define the unique member for a group. The %uniquemember% value is a placeholder for a value that is passed in

at runtime during the look up to determine if the user is a member of a group. The only aspect of this element you might want to change is the specification of the root for your users. The `%uniquemember%` value matches the root of your user population when you run the `libovdadapterconfig` script.

The `groupofuniquenames` object class specifies how group attributes are mapped to database fields and as with the user, the attributes correspond to the defaults in WebLogic Embedded LDAP. You must map the following attributes:

- `cn` maps to a unique name for your group.
- `uniquemember` maps to the unique name for your user in the user/group mapping table in your database schema.
- `orclguid` maps to a unique id, if available in your database schema.

Mapping the `description` attribute is optional.

1. Create a file named `adapter_template_usergroup1.xml` that maps the user table to a virtual LDAP store.
2. In the `<mapping>` element, add the `<objectclass>` element with attributes similar to the following example:

```
<mapping>
          <joins/>
      <objectClass name="person" rdn="cn">
        <attribute ldap="cn" table="USER_VW" field="U_NAME" type=""/>
        <attribute ldap="uid" table="USER_VW" field="U_NAME" type=""/>
        <attribute ldap="usernameattr" table="USER_VW" field="U_NAME"
type=""/>
        <attribute ldap="loginid" table="USER_VW" field="U_NAME" type=""/>
        <attribute ldap="description" table="USER_VW" field="U_NAME"
type=""/>
        <attribute ldap="orclguid" table="USER_VW" field="GUID" type=""/>
      </objectClass>
      </mapping>
```

3. Create a file, named `adapter_template_usergroup2.xml`, to map the group table to a virtual LDAP store.
4. In the `<objectClass name="groupofuniquenames">` element map the group table to the virtual LDAP store, as shown in the example:

```
  <mapping>
         <joins/>
                        <objectClass name="groupofuniquenames" rdn="cn">
                        <attribute ldap="cn" table="GROUPMEMBERS_VW"
field="G_NAME" type=""/>
                        <attribute ldap="description"
table="GROUPMEMBERS_VW" field="G_NAME" type=""/>
                        <attribute ldap="uniquemember"
table="GROUPMEMBERS_VW" field="G_MEMBER" type=""/>
                               <attribute ldap="orclguid"
table="GROUPMEMBERS_VW" field="G_MEMBER" type=""/>
                        </objectClass>
      </mapping>
```

5. Copy the two adapter files into the following folder:

   ORACLE_HOME`/oracle_common/modules/oracle.ovd/templates/`

6. Open a command prompt/terminal from within:

ORACLE_HOME/oracle_common/bin

7. Verify that the environment variables are set:
   - ORACLE_HOME=*ORACLE_HOME*/oraclehome
   - WL_HOME=*ORACLE_HOME*/wlserver
   - JAVA_HOME=*ORACLE_HOME*/jdk/jre
8. Run the libovdadapterconfig script to create each of the two adapters from the template files using the syntax as follows:

   ```
   libovdadapterconfig -adapterName <name of adapter> -adapterTemplate <name (NOT
   including path) of template file which defines adapter> -host localhost -port
   <Admin Server port> -userName <user id of account which has administrative
   privileges in the domain> -domainPath <path to the BI domain> -dataStore DB -root
   <nominal specification of a pseudo-LDAP query to treat as the "root" of this
   adapter - must match that specified in template for adapter 2 above> -contextName
   default -dataSourceJNDIName <JNDI name for DataSource which points at the database
   being mapped>
   ```

   For example:

   ```
   ./libovdadapterconfig.sh -adapterName userGroupAdapter1 -adapterTemplate
   adapter_template_usergroup1.xml -host localhost -port 9500 -userName weblogic -
   domainPath /opt/oracle_bi/user_projects/domains/bifoundation_domain/ -dataStore DB
   -root cn=users,dc=oracle,dc=com -contextName default -dataSourceJNDIName jdbc/
   UserGroupDS

   ./libovdadapterconfig.sh -adapterName userGroupAdapter2 -adapterTemplate
   adapter_template_usergroup2.xml -host localhost -port 9500 -userName weblogic -
   domainPath /opt/oracle_bi/user_projects/domains/bifoundation_domain/ -dataStore DB
   -root cn=users,dc=oracle,dc=com -contextName default -dataSourceJNDIName jdbc/
   UserGroupDS
   ```
9. Restart WebLogic Administration Server and Managed servers.
10. Sign in to WebLogic and Oracle WebLogic Server using credentials stored in the database.

Troubleshoot the SQL Authenticator

This section provides troubleshooting information on the SQL authenticator in the following topics:

Topics:

Add a User to the Global Admin Role Using the Oracle WebLogic Server Administration Console

You can use this diagnostic test if you are unable to login to Oracle Analytics Server using a database user.

If you cannot log in to Oracle Analytics Server using a database user, a useful diagnostic test is to see whether your user can log in to WebLogic at all. If you do not have other applications on the WebLogic Server which take advantage of WebLogic container authentication, you

can add your user (temporarily) to the WebLogic Global Admin role and see if the user can log in to the Oracle WebLogic Server Administration Console to test whether the SQLAuthenticator is working at all.

If the user can log in to the console, but cannot log in to Oracle Analytics Server, the SQLAuthenticator is working correctly, but there may be issues in the identity store service. Check that you have specified the `virtualize=`*true*, and `OPTIMIZE_SEARCH=`*true* properties in Configure Identity Store Virtualization Using Fusion Middleware Control and that your DBAdapter templates are correct in Configure a Database Adaptor.

1. Log in to Oracle WebLogic Server Administration Console, and click **Lock & Edit** in the Change Center.
2. Select **Security Realms** from the left pane and click **myrealm**.

   The default Security Realm is named *myrealm*.
3. Display the **Roles and Policies** tab, then display the **Realm Roles** tab.
4. In the list of roles, click on the plus sign to expand **Global Roles**, then **Roles**, then click the **View Role Conditions** link for the Admin role.
5. Ensure the conditions specified match your user, directly or by membership in a group.

   For example, a possible condition is User=myadminaccount or Group=Administrators.
6. If you have made any changes, click **Save**.

   Changes are applied immediately.
7. You should now be able to check whether the user in question can log in to the Oracle WebLogic Server Administration Console at `http://<bi server address>:<AdminServer Port>/console`, for example, `http://example.com:9500/console`.

An Incorrect Data Source Name is Specified for the SQLAuthenticator

If you specify the wrong name for the data source field of the SQLAuthenticator, then errors are included in the log files for Administration Server and Managed Servers.

The following is an example of an error written to the log files.

```
Caused by: javax.security.auth.login.FailedLoginException:
[Security:090761]Authentication failed for user jsmith java.sql.SQLException:
[Security:090788]"Problem with DataSource/ConnectionPool configuration, verify
DataSource name wrongdsname is correct and Pool configurations are correct"
      at weblogic.security.providers.authentication.shared.DBMSAtnLoginModuleI
mpl.login(DBMSAtnLoginModuleImpl.java:318)
```

Use the data source name as in the example shown in Configure a Data Source Using the Oracle WebLogic Server Administration Console.

Incorrect SQL Queries

Ensure that the SQL queries that you specify when configuring the SQLAuthenticator are syntactically correct and refer to the correct tables.

For example, the following error occurs in the Administration Server.log file when the wrong table name is specified for the password query:

```
####<Jul 7, 2011 4:03:27 PM BST> <Error> <Security> <gbr20020> <AdminServer> <[ACTIVE]
ExecuteThread: '8' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>>
<> <de7dd0dc53f3d0ed:e0ce69e:131007c1afe:-8000-00000000000007fa> <1310051007798>
<BEA-000000> <[Security:090759]A SQLException occurred while retrieving password
information
java.sql.SQLSyntaxErrorException: ORA-00942: table or view does not exist
     at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:457)
     at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:405)
     at oracle.jdbc.driver.T4C8Oall.processError(T4C8Oall.java:889)
     at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:476)
```

Correct Database Adapter Errors by Deleting and Recreating the Adapter

Use this procedure to create a replacement adapter.

You cannot modify an existing database adapter, if you make an error in the `libovdadapter` command or the templates, you must delete then recreate the adapter.

1. Log in to the Oracle WebLogic Server console by running the `WLST` script.

   ORACLE_HOME`\oracle_common\common\bin\wlst.cmd (Windows)`

2. Connect to your Administration Server using the following syntax:

   `connect ('<WLS admin user name>','<WLS admin password>','t3://<admin server host>:<admin server port>')`

   For example:

   `connect('weblogic','weblogic','t3://myserverexample:9500')`

3. Delete the poorly configured adapter using the following syntax:

   `deleteAdapter(adapterName='<AdapterName>')`

   For example:

   `deleteAdapter(adapterName='userGroupAdapter2')`

4. Exit the WLST console using the `exit()` command.

Recreate the adapter with the correct settings by following the steps outlined in Configure a Database Adaptor.

Configure Identity Store Virtualization Using Fusion Middleware Control

Use these steps to configure identity store virtualization using Fusion Middleware Control.

If you are communicating with LDAP over SSL (one-way SSL only), see Configure SSL when Using Multiple Authenticators.

Configure supported authentication providers as described in Configure Oracle Analytics Server to Use Alternative Authentication Providers.

1. Log in to Fusion Middleware Control.
2. From the navigation pane expand the **WebLogic Domain** folder and select **bi**.
3. Right-click **bi** and select **Security**, then **Security Provider Configuration** to display the Security Provider Configuration page.
4. Expand **Security Store Provider** and **Identity Store Provider**, and click **Configure** to display the Identity Store Configuration page.
5. In the Custom Properties area, use the **Add** option to add the following custom properties:
   - Property Name=*virtualize*
     Value=*true*
   - Property Name=*OPTIMIZE_SEARCH*
     Value=*true*

   > **Note:**
   >
   > Use lowercase for the Property Name `virtualize` , and use uppercase for `OPTIMIZE_SEARCH`.

   > **Note:**
   >
   > If you are using multiple authentication providers, go to Configure Oracle Analytics Server to Use Alternative Authentication Providers and configure the **Control Flag** setting as follows:
   >
   > - If each user appears in only one authentication provider.
   >
   >   Set the value of **Control Flag** for all authentication providers to *SUFFICIENT*.
   > - If users appear in more than one authentication provider.
   >
   >   Set the value of **Control Flag** for all authentication providers to *OPTIONAL*.
   >
   >   For example, if a user's group membership is spread across more than one authentication provider
6. Click **OK** to save the changes.
7. Restart the Administration Server and Managed Servers.

Configure Multiple Authentication Providers

This section explains how to configure an authentication provider so that when it fails, users from other authentication providers can still log in to Oracle Analytics Server.

If you configure Oracle Analytics Server to use multiple authentication providers, and one authentication provider becomes unavailable, users from the other authentication providers cannot log in to Oracle Analytics Server.

When you cannot log in due to an authentication provider becoming unavailable, the following error message is displayed:

```
Unable to Sign In
An error occurred during authentication.
Try again later or contact your system administrator
```

If an authenticator from multiple configured authenticators is unavailable and is not critical, use the following procedure to enable users from other authenticators to log in to Oracle Analytics Server.

1. Open the `adapters.os_xml` file for editing located in

   ORACLE_HOME`\user_projects\domains\bi\config\fmwconfig\ovd\default`

2. Locate the following element in the file:

   `<critical>`*true*`</critical>`

   Change the value of the `<critical>` element to *false* for each authenticator provider that is not critical, as follows:

   `<critical>false</critical>`

3. Save and close the file.
4. Restart WebLogic Administration Server and Managed Servers.

Set the JAAS Control Flag Option

When you configure multiple authentication providers, use the JAAS Control Flag for each provider to control how the authentication providers are used in the login sequence. You can set the JAAS Control Flag in the Oracle WebLogic Server Administration Console.

You can also use the Oracle WebLogic Scripting Tool or Java Management Extensions (JMX) APIs to set the JAAS Control Flag for an authentication provider.

Setting the **Control Flag** attribute for the authenticator provider determines the ordered execution of the authentication providers. The possible values for the **Control Flag** attribute are:

- REQUIRED - This LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers. This setting is the default.
- REQUISITE - This LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is returned to the application.

- SUFFICIENT - This LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.
- OPTIONAL - This LoginModule can succeed or fail. However, if all Authentication providers configured in a security realm have the JAAS Control Flag set to OPTIONAL, the user must pass the authentication test of one of the configured providers.

When additional Authentication providers are added to an existing security realm, by default the **Control Flag** is set to OPTIONAL. If necessary, change the setting of the **Control Flag** and the order of Authentication providers so that each Authentication provider works properly in the authentication sequence.

Configure a Single LDAP Authentication Provider as the Authenticator

This topic explains how to reconfigure Oracle Analytics Server to use a single LDAP authentication provider by disabling the default WebLogic Server LDAP authenticator.

When you install Oracle Analytics Server, the system is automatically configured to use WebLogic Server LDAP as the default authenticator. The install process automatically generates the required users and groups in WebLogic Server LDAP. If you may have your own LDAP directory, for example, Oracle Internet Directory, that you want to use as the default authenticator, you must disable the WebLogic Server default authenticator. A single source authentication provider prevents deriving user names and passwords from multiple authentication sources which could lead to multiple points of attack, or entry from unauthorized users.

Topics:

Configure Oracle Internet Directory LDAP Authentication as the Only Authenticator

Use the examples for configuring Oracle Internet Directory (OID LDAP). You can apply these examples to other LDAP authentication providers with minor changes.

Topics:

Task 1 - Enable Backup and Recovery

Before you begin the process of disabling the WebLogic Server LDAP default method of authentication it is strongly recommended that you back up the system first. Otherwise, if you make an error during configuration you may find that you become locked out of the system or cannot restart it.

To enable backup and recovery, during the re-configuration phase, take a copy of the config.xml file in *ORACLE_HOME*`\user_projects\domains\bi\config` directory.

As you make changes, you keep copies of this file.

Task 2 - Configure the System to use WebLogic Server and an Alternative Authentication Provider

To remove the default WebLogic Server authenticators and use an alternative LDAP source (for example, OID LDAP), you must configure the system to use both WebLogic Server and the alternative method.

See Configure Oracle Analytics Server to Use Alternative Authentication Providers. Your starting point should be that the WebLogic Server LDAP users (default authenticator) and the new alternative LDAP users are both configured to allow access to Oracle Analytics Server.

When you have configured the system to enable you to log on as either a WebLogic Server LDAP user or an OID LDAP user, you can then proceed to follow the steps to remove the WebLogic Server default authenticator, as described in these tasks.

Task 3 - Identify or Create Essential Users Required in OID LDAP

You must ensure that the essential users shown in the table are migrated from WebLogic Server LDAP to OID LDAP.

Standard WebLogic Server Users	New Users Required in OID LDAP
LCMManagerUser	OID_LCMManagerUser; you can use any existing OID LDAP user.
For example, weblogic	OID_Weblogic; you can use any existing OID LDAP user.
OracleSystemUser	OracleSystemUser, this user must exist with this name in OID LDAP which is a fixed requirement of OWSM.

Three users are created during install:

- weblogic or whatever is specified during install or upgrade, so can be different.

  This administrator user is created during the install, sometimes called weblogic, but can have any name. You need to identify or create an equivalent user in OID LDAP but this user can have any name, which needs to be part of a group called Administrators.

- OracleSystemUser

  This user is specifically required by Oracle Web Services Manager - OWSM for the Global Roles mapping, and you must create this user in OID LDAP using this exact name.

Task 4 - Associate OID LDAP Groups with Global Roles in the WebLogic Console

Configure the global roles by mapping to OID LDAP groups.

Global Roles	Current WebLogic Server Groups	New OID LDAP Groups Required
Admin	Administrators	OID_Administrators
AdminChannelUsers	AdminChannelUsers	OID_AdminChannelUsers
AppTester	AppTesters	OID_AppTesters
CrossDomainConnector	CrossDomainConnectors	OID_CrossDomainConnectors
Deployer	Deployers	OID_Deployers
Monitor	Monitors	OID_Monitors
Operator	Operators	OID_Operators
OracleSystemRole	OracleSystemGroup	OracleSystemGroup (fixed requirement)

You must associate the global roles from the table, displayed in the Oracle WebLogic Server Administration Console, with your replacement OID LDAP groups, before you can disable the default WebLogic Server authenticator.

The default Security Realm is named *myrealm*.

Do not do add a new condition for the Anonymous and Oracle System roles, which can both remain unchanged.

1. Log in to Oracle WebLogic Server Administration Console.
2. In the Change Center, click **Lock & Edit**.
3. Select **Security Realms** from the left pane and click **myrealm**.
4. Click **Realm Roles**.
5. Click **Global Roles** and expand **Roles**.
6. Add a new condition for each Role.
7. Click **View Role Conditions**.
8. Select group from the **Predicate steps**.
9. Enter your newly-associated OID LDAP group, for example, assign the Admin role to the *OID_Administrators* role.
10. Save your changes.

After disabling the Default WebLogic Server Authentication, you can remove the old WebLogic Server groups, see Task 8 - Remove WebLogic Server Roles

Task 5 - Set User to Group Membership in OID LDAP

Now that you have created new users and groups in OID LDAP to replicate the users and groups automatically created in WebLogic Server LDAP you must ensure that

these users and groups also have the correct group membership in OID LDAP as shown in the table.

New OID LDAP User	Is A Member Of These New OID LDAP Groups
OID_Weblogic	OID_Administrators OID_BIServiceAdministrators
OracleSystemUser A user with this exact name must exist in OID LDAP.	OracleSystemGroup A group with this exact name must exist in OID LDAP

> **Note:**
>
> In order to achieve the user and group membership shown in the table, you must have suitable access to update your OID LDAP server, or someone else must be able to update group membership on your behalf.

Task 6 - Remove the Default Authenticator

You are now ready to remove the Default Authenticators.

You must create an LDAP authenticator that maps to your LDAP source before performing this task, see Task 2 - Configure the System to use WebLogic Server and an Alternative Authentication Provider.

See Set the JAAS Control Flag Option.

1. Change the **Control Flag** from *SUFFICIENT* to *REQUIRED* in the Oracle WebLogic Server Administration Console.
2. Save the changes.
3. Delete any other authenticators so that your *OID LDAP* authenticator is the single source.

Task 7 - Restart the BI Services

Now you are ready to restart the BI services. You must use the new OID administrator user, for example, OID_Weblogic, because the Oracle WebLogic Server administration user created during installation was removed, and users now exist in the single OID source. The OID administration user must have sufficient privileges, granted by the Global Admin role to start WebLogic.

> **Note:**
>
> When you log in to the Administration Tool online you must now provide the OID LDAP user and password, for example, OID_Weblogic, along with the repository password.

Task 8 - Remove WebLogic Server Roles

Complete this task if everything is working correctly.

The following are examples of WebLogic Server roles to remove using this procedure:

- Admin
- AdminChannelUsers
- AppTester
- CrossDomainConnector
- Deployer
- Monitor
- Operator

See Task 4 - Associate OID LDAP Groups with Global Roles in the WebLogic Console.

Back up your `config.xml` file, before performing this step, see Task 1 - Enable Backup and Recovery.

1. Edit global roles.
2. Remove all WebLogic Server roles that were automatically created, from the `OR` clause.
3. Save your changes.

Task 9 - Stop Alternative Methods of Authentication

You must remove the USER variable and may need to update initialization blocks in the metadata repository.

Oracle Analytics Server allows various forms of authentication methods to be applied at once. While some can see this as a desirable feature it also comes with security risks. To implement a single source of authentication, you must remove the authentication methods that use initialization blocks from the metadata repository.

You stop access through initialization blocks using the Oracle BI Administration Tool. Successful authentication requires a user name, and initialization blocks populate user names using the *USER* system session variable.

1. Remove the *USER* system variable from the metadata repository.
2. Ensure that initialization blocks in the metadata repository have the **Required for authentication** check box cleared.
3. Check that initialization blocks in the metadata repository that set the *PROXY* and *PROXYLEVEL* system session variables do not allow users to bypass security.

   The *PROXY* and *PROXYLEVEL* system variables allow connected users to impersonate other users with their security profile. This method is acceptable when the impersonated user account has less privileges, but if the account has more privileges it can be a security issue.
4. Disable or remove initialization blocks associated with the following system session variables: *USER*, *GROUP*, and *ROLES*.

If you disable an initialization block, then any dependent initialization blocks are also disabled.

You can now be sure that any attempted access using initialization block authentication cannot be successful. However, you must check all of your initialization blocks.

Troubleshoot

You might receive the following error after you have configured Oracle Internet Directory LDAP authentication as the single source:

```
<Critical> <WebLogicServer> <BEA-000386> <Server subsystem failed.

Reason: weblogic.security.SecurityInitializationException: User <oidweblogic> is not
permitted to boot the server. The server policy may have changed in such a way that
the user is no longer able to boot the server. Reboot the server with the
administrative user account or contact the system administrator to update the server
policy definitions.
```

Solution

If when you restart the system as the new WebLogic OID LDAP administrator (oidweblogic), you are locked out, and the message is displayed, it is because the oidweblogic user has insufficient privileges. The oidweblogic user requires the Admin global role to enable it to belong to an OID LDAP Administrator group. You resolve this issue by adding the BIServiceAdministrators group (or an OID LDAP equivalent) to the Admin global role.

> **Note:**
>
> To restore a previously working configuration, you must replace the latest updated version of the config.xml file with a backup version that you have made before changing the configuration, see Task 1 - Enable Backup and Recovery.
> To complete the restoration of the backup config.xml file, restart Oracle Business Intelligence as the original WebLogic administrator user, instead of as the OID LDAP user.

Reset the BI System User Credential

Follow these steps to reset the BI System user credential.

This credential is populated with securely-generated random values at BI domain creation time and is stored in the Credential Store. If at any time you need to reset the user name or password of this credential, follow these steps.

1. From the Fusion Middleware Control target navigation pane, expand the farm, then expand **WebLogic Domain**, and select **bi**.
2. From the WebLogic Domain menu, select **Security**, then **Credentials**
3. Expand the **oracle.bi.system** credential map, select **system.user** and click **Edit**.
4. In the Edit Key dialog, update the user name or password using values that do not match the credentials of a user in your Identity Store.

> **Note:**
>
> `system.user` must not be set to an actual user. It is used for internal authentication between various Business Intelligence components. You must provide a unique, random user name and password that aren't used by an actual system user.

5. Click **OK**.
6. Restart the system.

4

Enable SSO Authentication

These topics provide guidelines for configuring single sign-on (SSO) authentication for Oracle Analytics Server.

Topics:

> **Note:**
>
> Oracle recommends using Oracle Access Manager as an enterprise-level SSO authentication provider with Oracle Fusion Middleware. You can assume that Oracle Access Manager is the SSO authentication provider.

SSO Configuration Tasks for Oracle Analytics Server

The table contains SSO authentication configuration tasks and provides links for obtaining more information.

Task	Description	For More Information
Configure Oracle Access Manager as the SSO authentication provider.	Configure Oracle Access Manager to protect the Oracle Analytics Server URL entry points.	Configure SSO in an Oracle Access Manager Environment
Configure the HTTP proxy.	Configure the web proxy to forward requests from Presentation Services to the SSO provider.	Oracle WebLogic Server Administration Console Online Help
Configure a new authenticator for Oracle WebLogic Server.	Configure the Oracle WebLogic Server domain in which Oracle Analytics Server is installed to use the new identity store.	Configure an OID Authenticator for Oracle WebLogic Server Configure Oracle Analytics Server to Use Alternative Authentication Providers Oracle WebLogic Server Administration Console Online Help

Task	Description	For More Information
Configure a new identity asserter for Oracle WebLogic Server.	Configure the Oracle WebLogic Server domain in which Oracle Analytics Server is installed to use the SSO provider as an asserter.	Configure Oracle Access Manager as a New Identity Asserter for Oracle WebLogic Server Configure Oracle Analytics Server to Use Alternative Authentication Providers Oracle WebLogic Server Administration Console Online Help
Configure custom SSO solutions.	Configure alternative custom SSO solutions to protect the Oracle Analytics Server URL entry points.	Configure Custom SSO Environments
Enable Oracle Analytics Server to accept SSO authentication.	Enable the SSO provider configured to work with Oracle Analytics Server.	Enable Oracle Analytics Server to Use SSO Authentication

Note:

For an example of an Oracle Analytics Server SSO installation scenario, see *Enterprise Deployment Guide for Oracle Business Intelligence*.

Understand SSO Authentication and Oracle Analytics Server

Integrating a single sign-on (SSO) solution enables a user to log on (sign-on) and be authenticated once. Thereafter, the authenticated user is given access to system components or resources according to the permissions and privileges granted to that user.

You can configure Oracle Analytics Server to trust incoming HTTP requests authenticated by a SSO solution that is configured for use with Oracle Fusion Middleware and Oracle WebLogic Server.

When Oracle Analytics Server is configured to use SSO authentication, it accepts authenticated users from whatever SSO solution Oracle Fusion Middleware is configured to use. If SSO is not enabled, then Oracle Analytics Server challenges each user for authentication credentials. When Oracle Analytics Server is configured to use SSO, a user is first redirected to the SSO solution's login page for authentication. After the user is authenticated the SSO solution forwards the user name to Presentation Services where this name is extracted. Next a session with the BI Server is established using the impersonation feature, a connection string between the Oracle BI Presentation Server and the BI Server using credentials that act on behalf of a user being impersonated.

After successfully logging in using SSO, users are still required to have the `oracle.bi.server.manageRepositories` permission to log in to the Administration Tool using a valid user name and password combination.

Configuring Oracle Analytics Server to work with SSO authentication requires minimally that the following be done:

- Oracle Fusion Middleware and Oracle WebLogic Server are configured to accept SSO authentication. Oracle Access Manager is recommended in production environments.
- Oracle BI Presentation Services is configured to trust incoming messages.
- The HTTP header information required for identity propagation with SSO configurations, the user identity and SSO cookie, is specified and configured.

How an Identity Asserter Works

This section describes how Oracle Access Manager authentication provider works with Oracle WebLogic Server using Identity Asserter for single sign-on, providing the following features:

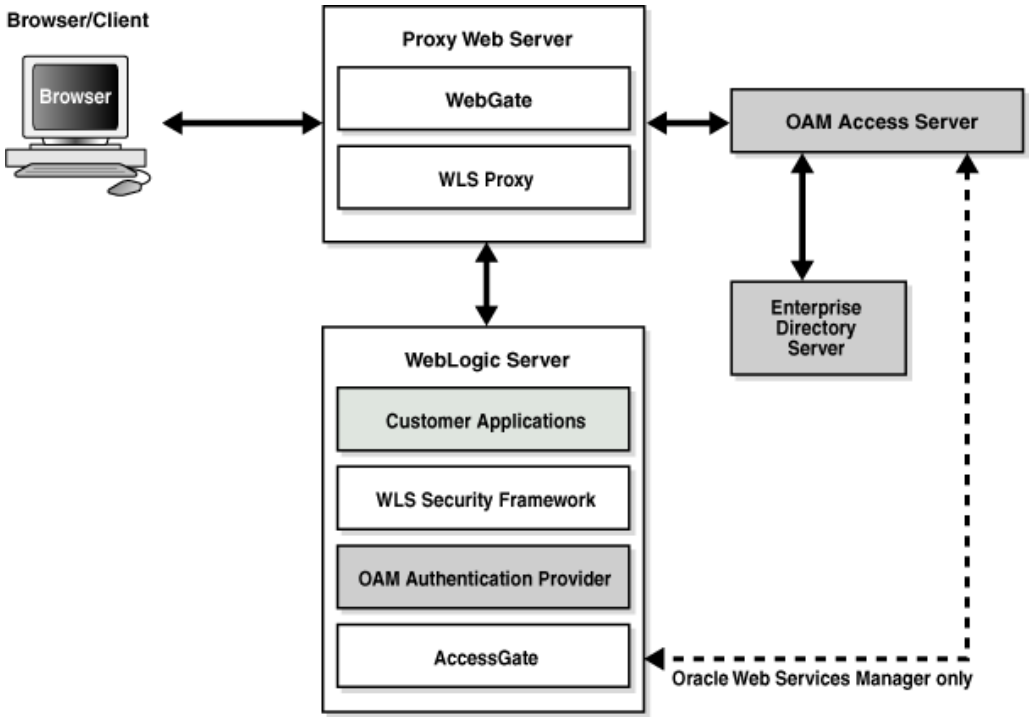
- **Identity Asserter for Single Sign-on**
  This feature uses the Oracle Access Manager authentication services and validates already-authenticated Oracle Access Manager users through a suitable token and creates a WebLogic-authenticated session. It also provides single sign-on between WebGate and portals. WebGate is a plug-in that intercepts web resource (HTTP) requests and forwards them to the Access Server for authentication and authorization.
- **Authenticator**
  This feature uses Oracle Access Manager authentication services to authenticate users who access an application deployed in Oracle WebLogic Server. Users are authenticated based on their credentials, for example a user name and password.

After the authentication provider for Oracle Access Manager is configured as the Identity Asserter for single sign-on, the web resources are protected. Perimeter authentication is performed by WebGate on the web tier and by the *appropriate token* to assert the identity of users who attempt access to the protected WebLogic resources.

All access requests are routed to a reverse proxy web server. These requests are in turn intercepted by WebGate. The user is challenged for credentials based on the authentication scheme configured within Oracle Access Manager (form-based login recommended).

After successful authentication, WebGate generates a token and the web server forwards the request to Oracle WebLogic Server, which in turn invokes Oracle Access Manager Identity Asserter for single sign-on validation. Oracle Access Manager is able to pass various types of heading token, the simplest being an HTTP header called OAM_REMOTE_USER containing the user ID that has been authenticated by Oracle Access Manager. The WebLogic Security Service invokes Oracle Access Manager Identity Asserter for single sign-on, which next gets the token from the incoming request and populates the subject with the `WLSUserImpl` principal. The Identity Asserter for single sign-on adds the `WLSGroupImpl` principal corresponding to the groups the user is a member of. Oracle Access Manager then validates the cookie.

The diagram depicts the distribution of components and the flow of information when the Oracle Access Manager Authentication Provider is configured as an Identity Asserter for SSO with Oracle Fusion Middleware.



How Oracle Analytics Server Operates with SSO Authentication

After SSO authorization has been implemented, Presentation Services operates as if the incoming web request is from a user authenticated by the SSO solution. Presentation Services next creates a connection to the BI Server using the impersonation feature and establishes the connection to the BI Server on behalf of the user. User personalization and access controls such as data-level security are maintained in this environment.

SSO Implementation Considerations

When implementing a SSO solution with Oracle Analytics Server you should consider the following:

When accepting trusted information from the HTTP server or servlet container, you must secure the machines that communicate directly with Presentation Services. In the `instanceconfig.xml` file, specify the list of HTTP Server or servlet container IP addresses in the `Listener\Firewall` node. The `Firewall` node must include the IP addresses of all Oracle BI Scheduler instances, Oracle Presentation Services instances, and Oracle Analytics Server *JavaHost* instances.

If any of these components are co-located with Oracle BI Presentation Services, you must add the 127.0.0.1 address in `Firewall` node. Setting the list of HTTP Server or servlet container IP addresses does not control end-user browser IP addresses. When using mutually-authenticated SSL, you must specify the Distinguished Names (DNs) of all trusted hosts in the `Listener\TrustedPeers` node.

Configure SSO in an Oracle Access Manager Environment

Review the overview about how to configure SSO in an Oracle Access Manager environment, and these additional references.

After the Oracle Fusion Middleware environment is configured, you must do the following to configure Oracle Analytics Server:

- Configure the SSO provider to protect the Oracle Analytics Server URL entry points.
- Configure the web server to forward requests from the Presentation Services to the SSO provider.
- Configure the new identity store as the main authentication source for the Oracle WebLogic Server domain where Oracle Analytics Server has been installed. See Configure an OID Authenticator for Oracle WebLogic Server.
- Configure the Oracle WebLogic Server domain where Oracle Analytics Server is installed to use an Oracle Access Manager identity asserter. See Configure Oracle Access Manager as a New Identity Asserter for Oracle WebLogic Server.
- After the SSO environment configuration is complete, enable SSO authentication for Oracle Analytics Server. See Enable SSO Authentication Using Fusion Middleware Control .

Configure an OID Authenticator for Oracle WebLogic Server

After installing Oracle Analytics Server, the Oracle WebLogic Server embedded LDAP server is the default authentication source (identity store).

To use a new identity store such as Oracle Internet Directory (OID) as the main authentication source, you must configure the Oracle WebLogic Server domain, where Oracle Analytics Server is installed.

For the field details to complete the Provider Specific tab, see Authentication Provider Specific Reference.

1. Click the newly added authenticator in the **authentication providers** table.
2. Navigate to **Settings**, then select the **Configuration\Common** tab:
   - Select **SUFFICIENT** from the **Control Flag** list.
   - Click **Save**.
3. Display the **Provider Specific** tab and specify the following settings using appropriate values for your environment:
4. Click **Save**.
5. Perform the following steps to set up the default authenticator for use with the Identity Asserter:
   a. At the main Settings for myrealm page, display the **Providers** tab, then display the **Authentication** tab, then select **DefaultAuthenticator** to display its configuration page.
   b. Display the **Configuration\Common** tab, from the **Control Flag** list, select *SUFFICIENT*.
   c. Click **Save**.

6. Perform the following steps to reorder providers:
   a. Display the **Providers** tab.
   b. Click **Reorder** to display the Reorder Authentication Providers page
   c. Select a provider name and use the arrow buttons to order the list of providers as follows:
      - OID Authenticator (SUFFICIENT)
      - OAM Identity Asserter (REQUIRED)
      - Default Authenticator (SUFFICIENT)
   d. Click **OK** to save your changes.
7. In the Change Center, click **Activate Changes**.
8. Restart Oracle WebLogic Server.
1. Log in to Oracle WebLogic Server Administration Console.
2. In the Change Center, click **Lock & Edit**.
3. From Domain Structure, select **Security Realms** and click **myrealm**.
4. In Settings for myrealm, click the **Providers** tab, and then click the **Authentication** tab.
5. In **Authentication Providers**, click **New**.
6. In Create a New Authentication Provider, type the **Name** for the authentication providers such as `OID Provider`.
7. From the **Type** list, select *OracleInternetDirectoryAuthenticator*, and click **OK**.
8. From the **Authentication Providers** table, select the provider you just created.
9. Click the **Common** tab, from the **Control Flag** list, select *Sufficient*, and click **Save**.

Use Reordering Authentication Providers to make the OID authenticator the primary authentication used by Oracle WebLogic Server. Reorder the authenticators as follows:

- OID Authenticator (*SUFFICIENT*)
- OAM Identity Asserter (*REQUIRED*)
- Default Authenticator (*SUFFICIENT*)

Authentication Provider Source Reference

This table provides a reference for adding an authentication provider.

Section Name	Field Name	Description
Connection	Host	The LDAP host name. For example, <*localhost*>.
Connection	Port	The LDAP host listening port number. For example, 6050.
Connection	Principal	The distinguished name (DN) of the user that connects to the LDAP server. For example, *cn=orcladmin*.

Section Name	Field Name	Description
Connection	Credential	The password for the LDAP administrative user entered as the Principal.
Users	User Base DN	The base distinguished name (DN) of the LDAP server tree that contains users. For example, use the same value as in Oracle Access Manager.
Users	All Users Filter	The LDAP search filter. For example, (&(uid=*)(objectclass=person)). The asterisk (*) filters for all users. Click **More Info...** for details.
Users	User From Name Filter	The LDAP search filter. Click **More Info...** for details.
Users	User Name Attribute	The attribute that you want to use to authenticate, for example, cn, uid, or mail. Set as the default attribute for user name in the directory server. For example, *uid*. The value that you specify here must match the User Name Attribute that you are using in the authentication provider.
Groups	Group Base DN	The base distinguished name (DN) of the LDAP server tree that contains groups (same as User Base DN).
General	GUID attribute	The attribute used to define object GUIDs in LDAP. orclguid You should not change this default value, in most cases the default value here is sufficient.

Configure Oracle Access Manager as a New Identity Asserter for Oracle WebLogic Server

The Oracle WebLogic Server domain in which Oracle Analytics Server is installed must be configured to use an Oracle Access Manager asserter.

1. Log in to Oracle WebLogic Server Administration Console.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring, for example, **myrealm**.
3. Select **Providers**.
4. Click **New**. Complete the fields as follows:
   - **Name**: *OAM Provider*, or a name of your choosing.
   - **Type**: OAMIdentityAsserter.
5. Click **OK**.
6. Click **Save**.
7. In the **Providers** tab, perform the following steps to reorder **Providers**:
   a. Click **Reorder**

b. In the Reorder Authentication Providers page, select a provider name, and reorder the list of providers as follows:
   - OID Authenticator (SUFFICIENT)
   - OAM Identity Asserter (REQUIRED)
   - Default Authenticator (SUFFICIENT)

c. Click **OK** to save your changes.

8. In the Change Center, click **Activate Changes**.
9. Restart Oracle WebLogic Server.
   You can verify that Oracle Internet Directory is the new identity store (default authenticator) by logging back into Oracle WebLogic Server and verifying the users and groups stored in the LDAP server appear in the console.
10. Enable SSO authentication.
    .

Configure Custom SSO Environments

You can use any Weblogic Identity Asserter combined with a supported Weblogic Authenticator to customize SSO for Oracle Analytics Server.

Custom SSO should be based on the development of a custom Weblogic Asserter. See How to Develop a Custom Identity Assertion Provider. The Weblogic Asserter should be paired with a BI-certified Weblogic Authenticator. See Certification - Identity Servers and Access.

In a typical custom SSO configuration, you include a web tier in front of Oracle Analytics Server to protect Oracle Analytics Server's endpoints. This configuration causes a user to authenticate and interact with an identity provider. After authentication, the web tier sends a token to Oracle Analytics Server that the Weblogic Asserter recognizes and processes.

There are many types of SSO tokens, but a basic implementation of a Weblogic Asserter recognizes a particular HTTP header or cookie (the token) that contains the authenticated user's UserID. The Weblogic Asserter retrieves the UserID from the token and passes it to the chain of Weblogic Authenticators. After this point, the authentication is the same as regular SSO.

Oracle Analytics Server's support for custom SSO starts where a custom asserter is working correctly to pass the authenticated user's UserID to the Weblogic chain of Oracle Analytics-certified authenticators.

Kerberos and SAML2 WebSSO Support

A reference implementation for custom SSO that uses either Kerberos or SAML2 is provided as a Docker-based solution for Oracle Analytics Server. This solution uses a web tier and a Weblogic Asserter.

See SAML 2.0 and Kerberos Single Sign-On Configuration for Oracle Analytics Server.

Enable Oracle Analytics Server to Use SSO Authentication

After you configure Oracle Analytics Server to use the SSO solution, you must enable SSO authentication for Oracle Analytics Server.

After you enable SSO, the default Oracle Analytics Server login page is not available.

Topics:

Enable and Disable SSO Authentication Using WLST Commands

Use WLST commands to enable or disable SSO authentication for Oracle Analytics Server.

SSO is enabled by default. If you leave it enabled, then Oracle Analytics Server uses SSO across the stack regardless of whether you use an external SSO for the initial login. And your configuration will use WLS Asserters for SSO.

If you disable SSO, then your configuration won't use WLS Asserters for Oracle Analytics Server or data visualization, and you'll be prompted a second time for login credentials when navigating from Oracle Analytics Server to data visualization.

If you are using legacy authentication methods such as session variables in initialization blocks, you need to disable lightweight SSO using the `disableBISingleSignOn` command.

- You must have file system and WebLogic Administrator permissions.
- You must perform the enable or disable SSO authentication as an offline activity.
- Validation is limited to URL format. Connectivity and WebLogic configuration is not validated.
- Changing the URL for log off requires that you disable, and then re-enable with new URL.
- A logon URL is not required.

Pre-requisites:

- Configure WebLogic security providers.

Use the table to learn the arguments appropriate for each command.

Command	Arguments	Return	Description
`enableBISingleSignOn`	*DOMAIN_HOME*, *<logoff-url>*	None	Enable SSO and configure logoff URL.
`disableBISingleSignOn`	*DOMAIN_HOME*	None	Disable SSO.

1. Stop the BI system.
2. Enter a SSO management command from the table using the WLST command line.
3. Start WLST using `./wlst.sh` command.
4. Optional: Run the command `help('BILifecycle')` to display help about `enableBISingleSignOn` and `disableBISingleSignOn` commands and their arguments.
5. Run the `enableBISingleSignOn` or `disableBISingleSignOn` command using the arguments appropriate for each command.

   For example: `enableBISingleSignOn('C:/.../user_projects/domains/bi','/bi-security-login/logout?redirect=/va')` or `disableBISingleSignOn('C:/oracle/Middleware/Oracle_Home/user_projects/domains/bi')`

The SSO configuration for Oracle Analytics Server is updated.

6. Restart the Oracle Analytics Server component processes to consume the changes.

Enable SSO Authentication Using Fusion Middleware Control

How you enable SSO authentication for Oracle Analytics Server using the **Security** tab in Fusion Middleware Control.

1. Log in to Fusion Middleware Control.
2. Go to the Security page and display the **Single Sign On** tab.

   Click the **Help for this page** Help menu option to access the page-level help for its elements.
3. Click **Lock and Edit**.
4. Select **Enable SSO**.

   When selected, this checkbox enables SSO to be the method of authentication into Oracle Analytics Server. The appropriate form of SSO is determined by the configuration settings made for the chosen SSO provider.
5. If required, enter the logoff URL for the configured SSO provider.

   The logoff URL (specified by the SSO provider) must be outside the domain and port that the SSO provider protects, because the system does not log users out.
6. Click **Apply**, then **Activate Changes**.
7. Restart the Oracle Analytics Server components using Fusion Middleware Control.

Enable the Online Catalog Manager to Connect

The online Catalog Manager might fail to connect to Oracle BI Presentation Services when the HTTP web server for Oracle Analytics Server is enabled for SSO.

When you enable SSO in Enable SSO Authentication Using Fusion Middleware Control , the Oracle Analytics Server URL `http://hostname:port_number/analytics` becomes protected, and you must point the online Catalog Manager to the URL `http://hostname:port_number/analytics-ws` instead. The URL should remain unprotected. It is configured only to accept SOAP access as used by Publisher, Oracle BI Add-in for Microsoft Office, and the online Catalog Manager.

To log in to the online Catalog Manager when SSO is enabled you must change the URL suffix to point to `analytics-ws/saw.dll`.

5

Configure SSL in Oracle Analytics Server

This chapter describes how to configure Oracle Analytics Server components to communicate over the Secure Socket Layer (SSL).

The SSL Everywhere feature of Oracle Analytics Server enables secure communications between the components. You can configure SSL communication between the Oracle Analytics Server components and between Oracle WebLogic Server for secure HTTP communication across your deployment. This section does not cover configuring secure communications to external services, such as databases and web servers.

Topics:

What is SSL?

SSL is a cryptographic protocol that enables secure communication between applications across a network.

Enabling SSL communication provides several benefits, including message encryption, data integrity, and authentication. An encrypted message ensures confidentiality in that only authorized users have access to it. Data integrity ensures that a message is received intact without any tampering. Authentication guarantees that the person sending the message is who he or she claims to be.

SSL requires that the server possess a public key and a private key for session negotiation. The public key is made available through a server certificate signed by a certificate authority. The certificate also contains information that identifies the server. The private key is protected by the server.

Using SSL in Oracle Analytics Server

Oracle Analytics Server components communicate with each other using TCP/IP by default. Configuring SSL between the Oracle Analytics Server components enables secured network communication.

Oracle Analytics Server components can communicate only through one protocol at a time. It is not possible to use SSL between some components, while using simple TCP/IP communications between others. You must configure the following components to enable secure communication over SSL:

- Oracle BI Server
- Oracle BI Presentation Services
- Oracle BI JavaHost
- Oracle BI Scheduler
- Oracle BI Job Manager
- Oracle BI Cluster Controller
- Oracle BI Server Clients, such as Oracle BI ODBC Client

SSL is configured throughout the Oracle Analytics Server installation from a single centralized point. Certificates are created for you and every Oracle Analytics Server component (except Essbase) is configured to use SSL. The following default security level is configured by SSL:

- SSL encryption is enabled.
- Mutual SSL authentication is not enabled. Since mutual SSL authentication is not enabled, clients do not need their own private SSL keys.
- The default cipher suites are used. See Manually Configure SSL Cipher Suite.
- When scaling out, the centrally managed SSL configuration is automatically propagated to any new components that are added.

If a higher level of security is required, manual configuration might be used to augment or replace the SSL central configuration. This is considerably more complex. For more information about how to configure SSL manually, contact Oracle Support.

Creating Certificates and Keys in Oracle Analytics Server

Secure communication over SSL requires certificates signed by a certificate authority (CA). For internal communication, the SSL Everywhere feature creates both a private certificate authority and the certificates for you. The internal certificates cannot be used for the outward facing web server because user web browsers are not aware of the private certificate authority. The web server must therefore be provided with a web server certificate signed by an externally recognized certificate authority.

Enable End-to-End SSL

To achieve end to end SSL you need to configure both internal SSL and WebLogic SSL.

The internal SSL configuration is highly automated whereas the WebLogic SSL configuration requires multiple manual steps. The two are entirely independent, so can

be performed in either order. Since the WebLogic configuration requires manual steps Oracle advises doing that first.

> **Note:**
>
> This section does not include configuring SSL for Essbase.

Topics:

Configure a Standard Non-SSL Oracle Analytics Server System

This section explains how to configure a standard non-SSL Oracle Analytics Server system.

- Install Oracle Analytics Server.
- Confirm the system is operational.

  Check you can login over HTTP to use:

  – Analytics

  - `http://<Host>:<ManagedServerPort>/analytics`

  – Fusion Middleware Control

  - `http://<Host>:< AdminPort>/em`

  – WebLogic Admin Console

  - `http://<Host>:<AdminPort>/console`

Configure WebLogic SSL

These steps configure WebLogic using the provided demo certificates. These are not secure.

Do not use these tasks in a production environment. Using the demo certificates can help you understand how to configure your environment with real certificates.

To configure with a secure certificate signed by a real Certificate Authority see WebLogic documentation. The certificate authority should return the signed server certificate, and provide a corresponding root CA certificate. Where *demoCA* is mentioned in task steps replace *demoCA* with your real CA certificate.

Topics:

- Restart
- Configure OWSM to Use t3s
- Restart System

Start Only the Administration Server

Starting up just the Administration Server rather than starting everything avoids the need to stop everything while the admin connection properties are in a state of flux, which confuses the stop everything script.

1. Stop everything with:

   `<DomainHome>/bitools/bin/stop.sh`

2. Start up just the Administration server with:

   `<DomainHome>/bitools/bin/start.sh -i Adminserver`

Configure HTTPS Ports

Follow these steps to configure the HTTPS ports.

1. Log in to WebLogic Admin console.
2. Click **Lock and Edit**.
3. Select **environment**, **servers**.
4. For each server on the main **Configuration** tab, select **SSL Listen Port Enabled**.
5. Click **Save**.
6. Click **Activate Changes**.
7. If you're using WebLogic demo certificates, go to URL `https://<host>:<AdminServerSSLPort>` and set up a single browser certificate exception.

   The URL `https://<host>:<AdminServerSSLPort>` is the base URL, without Enterprise Manager or the WebLogic Administration console on the path. By first accessing the base URL, you can set up a single browser certificate exception. If you go directly to the Enterprise Manager or the WebLogic Administration console paths, you must setup multiple certificate exceptions.

8. Enable the certificate exception by going to the base URL.

   You only have to do this once, rather than separately for WebLogic console and Fusion Middleware Control.

   The base URL should give a 404 error once the SSL connection is made. You can ignore the error.

9. Test the secure WebLogic console URL using a URL similar to the following:

   `https://<Host>:<AdminServerSSLPort>/console`

10. Test the secure Fusion Middleware Control URL using a URL similar to the following:

    `https://<Host>:<AdminServerSSLPort>/em`

    Test the HTTPS URL while logged in to Fusion Middleware Control using HTTP.

Don't disable HTTPS.

11. In WebLogic Administration Console, click **Lock and Edit** to begin enabling secure replication.
12. Select **Environment**, select **Clusters**, and then select **bi_cluster**.
13. Select **Configuration**, and select the **Replication** tab.
14. Select **secure replication enabled**.

    If you don't select **secure replication enabled**, the managed servers fail to startup and remain in Administration mode preventing the start scripts from running.
15. Click **Save**.
16. Click **Activate Changes**.

Configure Internal WebLogic Server LDAP to Use LDAPs

If you have configured an external Identity Store, you can skip performing this step. Perform this task if using WebLogic Server LDAP, and the `virtualize` property is not set to *true*.

You can configure an external identity store to use a secure connection. To use an external identity store, you must change the URL in the internal LDAP ID store.

1. Login to Fusion Middleware Control using a URL similar to the following:

   `https://<Host>/<SecureAdminPort>/em`
2. Click **WebLogic Domain**, click **Security**, and click **Security Provider Configuration**.
3. Expand the **Identity Store Provider** segment.
4. Click **Configure**, and click the plus symbol (+) to add a new property.
5. Add a `ldap.url` property using the following format for the *administration server* address rather than the *bi_server1* address:

   `ldaps://<host>:<adminServer HTTPS port>`, for example, `ldaps://myexample_machine.com:9501`.
6. In the Property editor, click **OK**.
7. On the Identity Store Provider page, click **OK**.
8. Open the `jps-config.xml` file located in `<DomainHome>/config/fmwconfig/jps-config.xml`.
9. In the file look for the line, `<property name="ldap.url" value="ldaps://<Host>:<AdminServerSecurePort>"/>` to confirm that the configuration change.

On IBM-AIX an additional configuration step is required to configure the IBM JDK supported cipher suites.

1. Open `<DomainHome>/config/fmwconfig/ovd/default/adapters.os_xml`
2. In the `<ldap>` section of this file, insert the following SSL cipher suites:

```
<ldap id="DefaultAuthenticator" version="0">
<ssl>
    <protocols>TLSv1.2,TLSv1.1</protocols>
    <cipherSuites>
       <cipher>SSL_RSA_WITH_AES_128_CBC_SHA</cipher>
       <cipher>SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</cipher>
```

```
        <cipher>SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256</cipher>
    </cipherSuites>
</ssl>
</ldap>
```

Configure Internal WebLogic Server LDAP Trust Store

You must now provide a trust keystore.

> **Note:**
>
> This section only applies when using WebLogic Server LDAP and when `virtualize=true` is set, as you're explicitly pointing the Administration Server for the embedded WLS LDAP.

1. In a terminal window set the *ORACLE_HOME* and *WL_HOME* environment variables.

   For example, on Linux:

   `setenv ORACLE_HOME <OracleHome>`

   `setenv WL_HOME <OracleHome>/wlserver/`

2. Ensure that both your path and JAVA_HOME point to the JDK 8 installation.

   `setenv JAVA_HOME <path_to_your_jdk8>`

   `setenv PATH $JAVA_HOME/bin`

3. Check the Java version by running:

   `java -version`

4. Run (without the line breaks):

   `<OracleHome>/oracle_common/bin/libovdconfig.sh`

   `-host <Host>`

   `-port <AdminServerNonSSLPort>`

   `-userName <AdminUserName>`

   `-domainPath <DomainHome>`

   `-createKeystore`

   When prompted enter the existing password for*<AdminUserName>*.

   When prompted for the OVD Keystore password, choose a new password.

   For example:

```
oracle_common/bin/libovdconfig.sh -host myhost -port 9500 -userName weblogic
-domainPath /OracleHome/user_projects/domains/bi -createKeystore

Enter AdminServer password:
Enter OVD Keystore password:
OVD config files already exist for context: default
CSF credential creation successful
```

```
Permission grant already available for context: default
OVD MBeans already configured for context: default
Successfully created OVD keystore.
```

The `-port <AdminServerNonSSL>` command doesn't work against the Admin server non-SSL port when it's been disabled. If you enable SSL and then configure LDAPs you would need to temporarily re-enable the non-SSL port on the Administration Server.

5. Check the resultant keystore exists, and see its initial contents, by running:

   keytool -list -keystore <*DomainHome*>/config/fmwconfig/ovd/default/keystores/adapters.jks

6. We now need to export the demo certificate in a suitable format to import into the above keystore.

   In Fusion Middleware Control:

   If using the demo WebLogic certificate you can get the required root CA from the system keystore using Fusion Middleware Control.

   a. Select **WebLogicDomain**, **Security**, **Keystore**.
   b. Expand **System**.
   c. Select **Trust**.
   d. Click **Manage**.
   e. Select **democa**, not olddemoca.
   f. Click **Export**.
   g. Select **export certificate**.
   h. Choose a file name.

      For example, *demotrust.pem*

      If not using the demo WebLogic certificate then you must obtain the root CA of the CA which singed your secure server certificate.

7. Now import into the just created keystore:

   ```
   keytool -importcert -keystore <DomainHome>/config/fmwconfig/ovd/default/keystores/adapters.jks -alias localldap -file <DemoTrustFile>
   ```

8. When prompted enter the keystore password you chose earlier, and confirm that the certificate is to be trusted.

9. If you repeat the keystore `-list` command you should see a new entry under `localldap`, for example:

   ```
   localldap, Jul 8, 2015, trustedCertEntry,
   ```

   Certificate fingerprint (SHA1):

   ```
   CA:61:71:5B:64:6B:02:63:C6:FB:83:B1:71:F0:99:D3:54:6A:F7:C8
   ```

Disable HTTP

After securing the system to use HTTPS, you must also disable HTTP to fully secure the environment.

1. Login to WebLogic Administration console.

2. Click **Lock & Edit**.
3. Select **environment**, **servers**.

   For each server:

   a. Display the **Configuration** tab
   b. Clear **Listen Port Enabled**.
   c. Click **Save**.
4. Click **Activate Changes**.

Verify Server Keystores

You must check that the Administration Server and Managed Servers are configured to use the trust keystore containing your trust certificate.

1. Login to WebLogic Administration console.
2. Click **Lock and Edit**.
3. Select **environment**, **servers**.
4. For each Managed Server.
   a. Display the **Keystores** tab.
   b. Ensure that the value for **Keystores** is `Custom Identity and Custom Trust`.

      > **Note:**
      >
      > If you're using WebLogic demo certificates you must still use `Custom Identity and Custom Trust`, configuring the custom settings to point to the demo keystores as described in these steps. You mustn't use `Demo Identity and Demo Trust` because this overrides the internal channel's SSL configuration.

   c. Verify that the location of the identity keystore points to the correct identity keystore.

      The WebLogic demo identity keystore is `kss://system/demoidentity`.
   d. Verify that the location of the trust keystore points to the correct trust keystore.

      The WebLogic demo trust keystore is `kss://system/trust`.
5. Click **Save**.
6. Click **Activate Changes**.

Restart

Now you must restart Oracle Analytics Server.

You can't login through Oracle Analytics Server since Oracle Web Service Manager (OWSM) uses the disabled HTTP port.

Only the HTTPS one should work.

HTTP should quickly display an error similar to `Unable to connect error`. Don't mix the protocols and ports. The browser can hang when attempting to connect to a running port with the wrong protocol.

1. Stop the Administration Server with `<DomainHome>/bitools/bin/stop.sh`.
2. Start the Administration Server with `<DomainHome>/bitools/bin/start.sh -i AdminServer`.
3. Confirm that HTTP is disabled by logging into both the HTTP and HTTPS WebLogic console URLs.

Configure OWSM to Use t3s

You must now change the Oracle Web Services Manager (OWSM) configuration to use the HTTPS port.

The HTTP(S) OWSM link isn't used when you use a local OWSM.

After you complete this task, you must restart the system and confirm the OWSM configuration. See Restart System.

1. Login to Fusion Middleware Control.

   `https://<Host>/<SecureAdminPort>/em`
2. Select **WebLogic domain**, and **cross component wiring**, **components**.
3. Select **component type**, **OWSM agent**.
4. Select the row **owsm-pm-connection-t3 status 'Out of Sync'**, and click **Bind**.
5. Select **Yes**.

Restart System

You must stop and restart all servers then test Analytics login with HTTPS.

1. Stop all servers using the `<DomainHome>/bitools/bin/stop.sh script`.
2. Use the `<DomainHome>/bitools/bin/start.sh` script to start everything.
3. Confirm your ability to log in to Analytics using a URL similar to the following:

   `https://<Host>:<SecureManagedServerPort>/analytics`

   The WebLogic tier using HTTPS only for its outward facing ports and all WebLogic infrastructure. The internal BI channel and Analytics system components use HTTP.
4. Optional: If you configured OWSM to use t3s, then use the validator to access the policy and confirm the configuration:

   `https://<host>:<ManagedServerSSLPort>/wsm-pm/validator`

Enable Oracle Analytics Server Internal SSL

Follow these steps to enable SSL on internal communication links.

You must run commands from the primary host. Oracle Analytics Server must have been configured by the BI configuration assistant, WebLogic managed servers must have been created, and any scaling out must be complete. Only use this procedure if you have configured security using the configuration assistant.

If you used the Configuration Template for SSL, see Enabling SSL in a Configuration Template Configured System.

You can configure the following advance options:

- Enable server checking of client certificates.
- Specify cipher suite to use.

  See Manually Configure SSL Cipher Suite.

Post conditions:

1. Stop the system using the following command:

   ORACLE_HOME/user_projects/domains/bi/bitools/bin/stop.sh

2. Run the following command to enable SSL on WebLogic internal channels and internal components:

   ORACLE_HOME/user_projects/domains/bi/bitools/bin/ssl.sh internalssl true

3. Optional: Configure advanced options by editing the file:

   ORACLE_HOME/user_projects/domains/bi/config/fmwconfig/biconfig/core/ssl/bi-ssl.xml

4. Restart the domain and Oracle Analytics Server component processes using the following command:

   ORACLE_HOME/user_projects/domains/bi/bitools/bin/start.sh

5. Confirm that WebLogic certificates and the corresponding trust have been correctly configured using the following:

   ORACLE_HOME/user_projects/domains/bi/bitools/bin/ssl.sh report

6. Confirm you can login to Oracle Analytics Server using your environment variables in:

   https://<*host*>:<*SecureManagedServerPort*>/analytics

> **Note:**
>
> You must perform this login to confirm that the HTTPS listener is enabled on each server before you enable end-to-end SSL. Any communication between internal components is encrypted, and is only verifiable using ssl.sh report command, or by checking server traffic.

Post-conditions

- WebLogic servers:
  - Have HTTPS listener enabled on internal channels.
  - The external port configuration is unaltered. See Enable End-to-End SSL for how to enable SSL on the external ports as well.

    There is a separate internal identity (key/certificate pair) for each listener address. The certificate has a common name matching the listening address, which is compatible with standard HTTPS practice. The certificates are signed by the internal certificate authority.

- System components, other than Essbase Studio:
  - Enable an HTTPS listener on internal channels.
  - The external port configuration is unaltered.
  - There is a separate internal identity (key or certificate pair) for each listener address. The certificate has a common name matching the listening address, which is compatible with standard HTTPS practice. The certificates are signed by the internal certificate authority.
- Essbase Studio:
  - No change. Continues with existing connectivity.

Disable Internal SSL

Use this task to disable Oracle Analytics Server SSL on internal communication links.

You must run commands from the primary host. To use this option, you configured Oracle Analytics Server using the configuration assistant, the WebLogic managed servers have been created, and scaling out is complete.

1. Stop the system using:

   `<DomainHome>/bitools/bin/stop.sh`

2. Run the following command to disable SSL on WebLogic internal channels and internal components:

   `<DomainHome>/bitools/bin/ssl.sh internalssl false`

3. Restart the domain using:

   `<DomainHome>/bitools/bin/start.sh`

Post conditions:

- WebLogic servers:
  - Have https listener disabled on internal channels.
  - The external port configuration is unaltered.
- System components, other than Essbase Studio:
  - Only listens on non SSL. SSL connections are not accepted.
- Essbase Studio:
  - No change. Continues with existing connectivity.

Export Trust and Identity for Clients

You can provide the keys and certificates required to allow Oracle Analytics Server clients, for example, the Administration Tool, and Job Manager to connect to SSL-enabled servers.

Assumptions:

- You run commands from the primary host.
- You can complete this operation online and offline.

Prerequisites

- Certificates are created using either the configuration assistant or by running `./ssl.sh` regenerate command.
- SSL on WebLogic is enabled.

  See Configure WebLogic SSL.
- You can perform this task with the system stopped or running.

Use the following command to export client identity and trust to *mydir*:

```
./ssl.sh exportclientcerts mydir
```

Certificates and the zip file are generated.

Post conditions:

- *Mydir* contains *clientcerts.zip* file.
- *Mydir* also contains expanded content of the zip file for immediate use:
  - `clientcert.pem`
  - `clientkey.pem`
  - `identity.jks`
  - `internaltrust.jks`
  - `internaltrust/internalca.pem`
  - `internaltrust/<hashed form of above>`
- Java clients such as Job Manager can successfully connect with secure option **verify server certificate** set using `identity.jks` to define identity, and internaltrust.jks for their trust.
- OpenSSL clients such as the Administration Tool can successfully connect with secure option **verify peer set** using `clientcert.pem` and `clientkey.pem` to define their identity, and `internalca.pem` as the trust file.

Configure SSL for Clients

Use these topics to configure SSL for clients.

You must configure clients accessing the Oracle Analytics Server components to use Oracle Analytics Server certificates. You must export the certificates by running the following command:

```
<DomainHome>/bitools/bin/ssl.sh exportclientcerts <exportDir>
```

Topics:

- Export Client Certificates
- Use SASchInvoke when BI Scheduler is SSL-Enabled
- Configure Oracle BI Job Manager
- Connect the Online Catalog Manager to Oracle Presentation Services
- Configure the Administration Tool to Communicate Over SSL
- Configure an ODBC DSN for Remote Client Access
- Configure Oracle Analytics Publisher to Communicate Over SSL

- Configure SSL when Using Multiple Authenticators

Export Client Certificates

Use these steps to create the passphrase for use when exporting client certificates.

The passphrase is used to protect the export certificates. You must remember this passphrase for use when configuring each client.

The command exports Java keystores for use by Java clients, and individual certificate files for use non Java clients. To make moving the certificates to a remote machine more convenient, the export also packages all the files into a single zip file.

1. Run the following command:

   ```
   <DomainHome>/bitools/bin/ssl.sh exportclientcerts <exportDir>
   ```

2. Type the new passphrase at the prompt.

Use SASchInvoke when BI Scheduler is SSL-Enabled

When the BI Scheduler is enabled for communication over SSL, you can invoke the BI Scheduler using the SASchInvoke command line utility.

The `SASchInvoke` tool is a command line job invocation tool which allows you to run pre-existing Oracle BI Scheduler jobs.

1. Create a new text file containing on a single line the passphrase you used when running the `./ssl.sh exportclientcerts` command.

   Ensure this file has appropriately restrictive file permissions to protect it. Typically it should only be readable by the owner. See Exporting Client Certificates.

2. Locate the `SASchInvoke` tool: `<Domain_Home>/bitools/bin/saschinvoke.cmd`

3. Use the following syntax to run the `SASchInvoke` command:

   ```
   SASchInvoke -u <Admin Name>  (-j <job id> | -i <iBot path>)
       ([-m <machine name>[:<port>]] | -p <primaryCCS>[:<port>] -s
   <secondaryCCS>[:<port>])
       ([(-r <replace parameter filename> | -a <append parameter filename>)]
    | [-x <re-run instance id>])
       [-l [-c <SSL certificate filename> -k <SSL certificate private key
   filename>] [ -w <SSL passphrase>  | -q <passphrase file>  | -y ]
       [-h <SSL cipher list>]
       [-v [-e <SSL verification depth>] -d <CA certificate directory> | -f
   <CA certificate file> [-t <SSL trusted peer DNs>] ] ]

   where:
   -a  File containing additional parameters.
   -c  File containing SSL certificate. SSL certificate filename =
   clientcert.pem
   -d  Certificate authority directory.
   -e  SSL certificate verification depth.
   -f  Certificate authority file.
   -h  SSL cipher list
   -i  Agent path
   -j  Job id
   ```

```
-k  SSL certificate private key filename. SSL certificate private
key filename = clientkey.pem
-l  Use SSL
-m  Machine name:port of scheduler.  Provides direct access to
scheduler.
-p  Primary cluster controller name:port.  Provides access to
clustered scheduler.
-q  Location of the passphrase file created in step 1 containing
the SSL passphrase protecting SSL private key (see -k).
-r  File containing replacement parameters.
-s  Secondary cluster controller name:port.  Provides access to
clustered scheduler.
-t  Distinguished names of trusted peers.
-u  Username
-v  Verify peer
-w  SSL passphrase protecting SSL private key (see -k).
-x  Rerun instance id.
-y  Interactively prompt for SSL passphrase protecting SSL private
key (see -k).
```

4. The command prompts you to enter the administrator password. Once entered, the `SASchInvoke` tool will get the BI Scheduler to run the specified job.

Configure Oracle BI Job Manager

To successfully connect to BI Scheduler that has been enabled for SSL, Oracle BI Job Manager must also be configured to communicate over SSL.

Oracle BI Job Manager is a Java based component and the keys and certificates that it uses must be stored in a Java keystore database.

1. From the **File** menu, select **Oracle BI Job Manager**, then select **Open Scheduler Connection**.
2. In the Secure Socket Layer section, select the **SSL** check box.
3. If the server setting **verify client certificates** is *false* (one way SSL) then you can leave **Key Store** and **Key Store Password** blank. This is the default setting.
4. If the server setting **verify client certificates** is *true* (two way SSL) then you must set **Key Store** and **Key Store Password** as follows:
   - `Key Store=<`*`exportclientcerts_directory`*`>\identity.jks`
   - `Key Store Password =`*`passphrase`*.
5. To provide a secure link you should select the verify server certificate. Without verification the connection works, but a person in the middle attack which impersonates the server is not detectable.
   a. Select the **Verify Server Certificate** check box. When this is checked, the trust store file must be specified. This trust store contains the CA that verifies the Scheduler server certificate.
   b. In the **Trust Store** text box, set the trust store to:

      `<`*`exportclientcerts_directory`*`>\internaltrust.jks`

   c. Set the **Trust Store Password** to the *passphrase*.

Connect the Online Catalog Manager to Oracle Presentation Services

For the online Catalog Manager to connect to Oracle Presentation Services, you might need to import the SSL server certificate or CA certificate.

The online Catalog Manager might fail to connect to Oracle Analytics Server when the HTTP web server for Oracle Analytics Server is enabled for SSL unless Catalog Manager is configured to trust the certificate presented by Oracle Analytics Server. For this configuration, you must export the SSL server certificate or Certificate Authority (CA) certificate from the web server, and then import the certificate into the Java keystore of the Java Virtual Machine (JVM) used by Client Tools.

The method of exporting the certificate from the server depends on the type of web server. One approach is to connect to the web server using a browser and export the certificate from the browser. The method for importing the certificate into the trust store uses the standard keytool utility that comes with the JDK.

Note the following information:

- The location of the Java Keystore when using Catalog Manager from Client Tools is: `<Client tools install folder>\oracle_common\jdk\jre\lib\security\cacerts`
- The location of the Java Keystore when using Catalog Manager on Oracle Analytics Server is: `<JAVA_HOME>/jre/lib/security/cacerts`, where <*JAVA_HOME*> is configured during the Oracle Analytics Server installation.
- The default password for the Java trust store (keystore) is *changeit*.

Use this procedure to connect the online Catalog Manager to Oracle Presentation Services:

1. Navigate to Java's default trust store named `cacerts`.
2. Copy the certificate exported from the web server to the same location as Java's default trust store.

   For example, if using the Oracle WebLogic Server default demonstration certificate, copy the certificate located in `<ORACLE_HOME>/wlserver/server/lib/CertGenCA.der` to the location of the Java keystore.
3. Execute the following command to import the certificate into the default trust store:

   ```
   keytool -importcert -trustcacerts -alias bicert -file <WebServerCertFilename> -
   keystore cacerts -storetype JKS
   ```

   The web server certificate file <*WebserverCertFilename*> is imported into Java's default trust store, under an alias of `bicert`.
4. Restart Catalog Manager using the secure HTTPS URL.

Configure the Administration Tool to Communicate Over SSL

To successfully connect to a BI Server configured to use SSL, you must also configure the Administration Tool to communicate over SSL.

The data source name (DSN) for the BI Server data source is required.

1. Determine the BI Server data source DSN in use by logging into the Presentation Services Administration page as an administrative user.
2. Locate the BI Server **Data Source** field.

The DSN is listed in the following format, `coreapplication_OH<DSNnumber>`.

3. In the Administration Tool, select **File**, then **Open**, then **Online**.
4. Select the DSN from the list.
5. Enter the repository user name and password.

   The Administration Tool is now connected to the BI Server using SSL.

Configure an ODBC DSN for Remote Client Access

You can create an ODBC DSN for the BI Server to enable remote client access.

Configure Oracle Analytics Publisher to Communicate Over SSL

You can configure Oracle Analytics Publisher to communicate securely over the internet using SSL.

Check Certificate Expiry

This task provides a warning if certificates are expired or about to expire.

You must run commands from the primary host with the system running or stopped.

- Run the following command to check certificate expiry:

  `<DomainHome>/bitools/bin/ssl.sh expiry`

Post conditions:

- Detailed expiry information on certificate authority and server certificates is listed.
- The `ssl.sh` command returns the following status:
  - 13 – if certificates expired.
  - 14 – if certificates are due to expire in less than 30 days.
  - 0 – if certificates have more than 30 days life remaining.

Replace the Certificates

Certificate replacement allows replacement of all certificates by new ones.

You may want to do this because:

- The existing certificates have expired, or are about to expire.

  Both server certificates and CA (trust) certificates have defined life spans. Once they expire connections using those certificates do not work.
- Your organization has a policy requiring a different certificate expiry from the default provided by the BI configuration assistant.
- The security of the existing certificates and keys has been compromised.

Assumptions:

- You run commands from the primary host.
- This is an offline operation.

1. Replace internal BIEE or client certificates.

   When you use the regenerate command, it invalidates existing client certificates so you must re-export them.

   ```
   ./ssl.sh regenerate
   ./ssl.sh exportclientcerts mydir
   ```

2. Restart the domain using:

   ```
   ./start.sh
   ```

3. Check WebLogic certificates and corresponding trust are correctly configured using:

   ```
   ./ssl.sh report
   ```

Post conditions

The domain now runs with SSL, and uses the new certificates. Servers will not connect to a WebLogic instance using the old trust.

You can run the `ssl.sh expiry` command to list the new certificates with the new expiry date.

Update Certificates After Changing Listener Addresses

You can update certificates following a change of listener address, for example by setting an explicit listener address in WebLogic console to replace the default (blank).

The ssl.sh scan command shows errors due to incorrect certificate common names. Connections to servers whose certificates do not match their listening addresses will be rejected.

Assumptions:

- You run commands from the primary host.
- This is an offline operation.

1. Update certificates by running:

   ```
   ./ssl.sh rebindchannelcerts
   ```

2. Restart the domain using:

   ```
   ./start.sh
   ```

3. Check WebLogic certificates and corresponding trust are correctly configured using:

   ```
   ./ssl.sh report
   ```

Post conditions

The domain now runs with SSL, and uses the new certificates. The new certificates have the same expiry as existing certificates. The certificates are signed by the existing internal certificate authority so previously exported client trust remains valid.

You can run the `ssl.sh expiry` command to list the new certificates with the new expiry date.

Add New Servers

Follow these steps to achieve the same internal SSL configuration for a new server.

Assumptions:

- You run commands from the primary host.
- This is an offline operation.
- One or more new servers have been created, either by cloning an existing server or creating from scratch.

1. For each new server run the following:

   ```
   ./ssl.sh channel <new_bi_server> <port>
   ```

2. You can run the following more than once:

   ```
   ./ssl.sh internalssl true
   ```

   Run the channel command as indicated in the `internalssl` command's error message.

3. Restart the domain using:

   ```
   ./start.sh
   ```

4. Check WebLogic certificates and corresponding trust are correctly configured using:

   ```
   ./ssl.sh report
   ```

Post conditions

The domain now runs with SSL, with all WebLogic managed servers using the internal SSL. If the servers were cloned, the cloned internal channel port has been replaced by the port given by the channel command. If the servers were created from scratch the internal channel has been created and configured to use SSL.

Enable SSL in a Configuration Template Configured System

This task provides the same SSL internal channel configuration as provided by the BI configuration assistant for systems configured using WLST or by direct application of configuration templates in the WebLogic configuration assistant.

Assumptions:

- You run commands from the primary host.
- This is an offline operation.

1. Run the following commands:

   ```
   <domain_home>/bitools/bin/ssl.sh regenerate <days>
   <domain_home>/bitools/bin/ssl.sh targetapps bi_cluster
   ```

2. For each new server run:

   ```
   ./ssl.sh channel <new_bi_server> <port>
   ```

3. Do one of the following:
   - Run the command:

     ```
     ./ssl.sh internalssl true
     ```

   - Run the `./ssl.sh internalssl true` repeatedly, and run the <<*other commands*>> as indicated in the `internalssl` command's error message

4. Restart the domain using `./start.sh`.

5. Check WebLogic certificates and corresponding trust are correctly configured using:

   ```
   ./ssl.sh report
   ```

Post conditions

The domain runs with SSL and all the WebLogic managed servers using the internal SSL.

Enable SSL Without Internal Oracle Analytics Server SSL

To support SSL on the external ports without using SSL internally you must decouple the internal communications by creating internal channels. Use the steps in this task to create the internal channels configured to use HTTP.

Oracle Analytics Server has system components that need to communicate with Java components running inside WebLogic managed servers, for example at login an Oracle BI Server process calls the BI security service. In a default configuration template configured system, the communication links use the external WebLogic ports. You can configure Oracle WebLogic Server to use HTTPS for its external ports.

If you configure WebLogic to use HTTPS for external ports, the internal components attempt to connect to the HTTPS port without the necessary trust setup. To avoid this problem, you need to configure private channels. These private channels are independent of the external WebLogic ports, with their own ports and their own protocol configuration.

Assumptions:

- Run commands from the primary host.
- Perform this task as an offline operation.
- Do one of the following:
  - Option A, run the following commands:

    `<domain_home>/bitools/bin/ssl.sh regenerate <days>`

    Regenerate the certificates to allow the subsequent channel commands to work. The certificates aren't used unless you subsequently change your mind and enable internal SSL.

    `<domain_home>/bitools/bin/ssl.sh targetapps bi_cluster`

    For each new server run the following using an unused port:

    `./ssl.sh channel <new_bi_server> <port>`

    `./ssl.sh internalssl false`
  - Option B, repeat running the following command using the `internalssl` error checking to prompt you to resolve issues.

    `./ssl.sh internalssl false`

    Run the other commands as indicated in the `internalssl` command's error messages.

Manually Configure SSL Cipher Suite

The default SSL configuration uses default cipher suite negotiation. You can configure the system to use a different cipher suite if your organization's security standards do not allow for the default choice. You can view the default choice in the output from the SSL status report.

This advanced option involves editing a configuration file. Be careful to observe the syntactic conventions of this file type.

A manually configured SSL environment can coexist with a default SSL configuration.

1. Configure SSL.
2. Select the desired Java Cipher Suite.
3. Create an Open SSL Cipher Suite Name that matches the cipher suite.

   For example, the Java Cipher Suite name, `SSL_RSA_WITH_RC4_128_SHA` maps to Open SSL: `RSA+RC4+SHA`.
4. Edit the `bi-ssl.xml` file located at:

   `<DOMAIN_HOME>/config/fmwconfig/biconfig/core/ssl/bi-ssl.xml`

   Add following child element to the `JavaHost/Listener/SSL` element, for example:

   ```
   <EnabledCipherSuites>SSL_RSA_WITH_RC4_128_SHA</EnabledCipherSuites>
   ```
5. Restart the Oracle Analytics Server components using:

   ```
   ./start.sh
   ```

Configure SSL Connections to External Systems

Use these links to see topics about configuring SSL connections to external systems:

Topics:

- Configure SSL for the SMTP Server Using Fusion Middleware Control
- Configure SSL when Using Multiple Authenticators

Configure SSL for the SMTP Server Using Fusion Middleware Control

You must obtain the SMTP server certificate to complete this task.

1. Login to Fusion Middleware Control.
2. Click **Target Navigation**, and then click **biinstance** under **Business Intelligence** to display the Business Intelligence Instance page.
3. Click **Configuration**, and then click **Mail**.

   Click the **Help** button on the page to access the page-level help for its elements.
4. Click **Lock and Edit** in the Change Center.
5. Complete the fields under **Secure Socket Layer (SSL)** as follows:

- **Connection Security**: Select an option, other fields may become active afterward.
- **Specify CA certificate source**: Select **Directory** or **File**.
- **CA certificate directory**: Specify the directory containing CA certificates.
- **CA certificate file**: Specify the file name for the CA certificate.

  Oracle Analytics Server includes a default certificate that you can use for the configuration of SSL for the SMTP server. The certificate's location is:

  `ORACLE_HOME/bi/modules/oracle.bi.publictrust/openssl/cacerts.crt`
- **SSL certificate verification depth**: Specify the verification level applied to the certificate.
- **SSL cipher list**: Specify the list of ciphers matching the cipher suite name that the SMTP server supports, for example, RSA+RC4+SHA.

6. Click **Apply**, then click **Activate Changes** in the Change Center to apply your changes.

Configure SSL when Using Multiple Authenticators

If you are configuring multiple authenticators, and have configured an additional LDAP Authenticator to communicate over SSL (one-way SSL only), you need to put the corresponding LDAP server's root certificate in an additional keystore used by the virtualization (libOVD) functionality.

In the following procedure you set the values for your environment variables: *ORACLE_HOME*, *WL_HOME* and *JAVA_HOME*.

The `createKeystore` command creates an OVD Keystore password. You have to type a value for the OVD Keystore password.

Before completing this task, you must configure the custom property, called `virtualize`, and set the property's value to `true`.

1. Set up the keystore by running libovdconfig.bat on Windows, using the -createKeystore option.
2. Type the command to look similar to the following:

   ```
   libovdconfig.bat -createKeystore -host <hostname> -port <Admin_Server_Port> -
   domainPath <OracleHome>/user_projects/domains/bi -userName <BI Admin User>
   ```
3. At the prompt, type the Oracle Analytics Server administrator user name and password.
4. Type a password for the OVD Keystore password to secure a Keystore file.
5. Export the root certificate from the LDAP directory.
6. Use the following the keytool command to import the root certificate to the `libOVD` keystore:

   ```
   <OracleHome>/jdk/jre/bin/keytool -import -keystore <OracleHome>/user_projects/
   domains/bi/config/fmwconfig/ovd/default/adapters.jks -storepass <KeyStore
   password> -alias <alias of your choice> -file <Certificate filename>
   ```
7. Restart WebLogic Server and Oracle Analytics Server processes.

You should see two new credentials in the Credential Store and a new Keystore file, called `adapters.jks` in the following location, `<OracleHome>/user_projects/domains/bi/config/fmwconfig/ovd/default`.

WebLogic Artifacts Reserved for Oracle Analytics Server Internal SSL Use

The following WebLogic artifacts are reserved for Oracle Analytics Server internal use:

- Virtual hosts:

  bi_internal_virtualhost1
- Channels (on each managed server):

  bi_internal_channel1