Oracle® Analytics

Managing Security for Oracle Analytics Server

F24229-19
July 2024

Oracle Analytics Managing Security for Oracle Analytics Server,

F24229-19

Primary Author: Stefanie Rhone

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

Contents

3 Use Alternative Authentication Providers

4 Enable SSO Authentication

5 Configure SSL in Oracle Analytics Server

A Managing Security for Dashboards and Analyses

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 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 Groups
Fine-tune privileges in the semantic model and Presentation Services	Add and remove the privileges that users and groups have in the Oracle Analytics Server semantic model and in the Classic Home Page. Use Model Administration Tool and the Oracle Analytics Server Classic Administration Page to add and remove these privileges.	Use Model Administration Tool to Manage Metadata Repository Privileges Use Application Roles to Manag Presentation Services Privileges
Decide if you want to deploy single sign-on (SSO) authentication	Configure SSO authentication.	Enable SSO Authentication
Decide if you want to deploy secure socket layer (SSL)	Configure Oracle Analytics Server components to communicate over SSL.	Configure SSL in Oracle Analytics Server

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

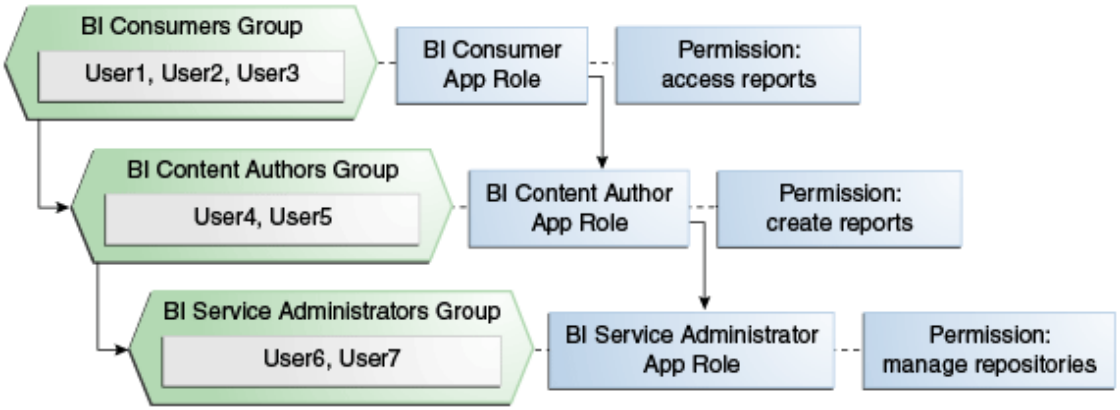
application roles that are assigned to directory server groups and users. For example, application roles BIServiceAdministrator, BI Consumer, and BIContentAuthor.

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 (semantic models).

About the Security Policy

The security policy is split across Presentation Services, the semantic model, 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.

Semantic Model

The semantic model defines the metadata items in the semantic model that user can access with assignment to specific application roles. You can define the security policy using Model 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 semantic model 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 Analytics Server 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 Analytics Server Presentation Catalog. The rights are stored in the catalog and managed by 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 Analytics Server Presentation Catalog:

  Defines which Oracle Analytics Server Presentation Catalog objects and 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.

- Semantic Model:

  Defines access to the specified metadata within the semantic model. Access to this functionality is managed in the Model 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 Model 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

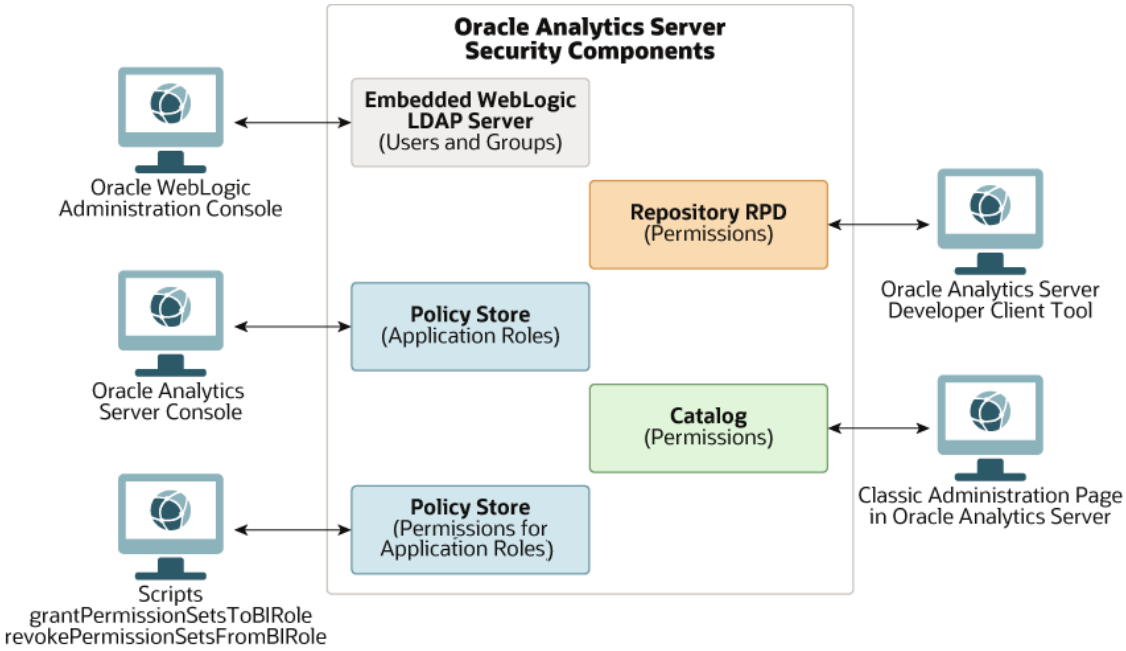
This topic explains how to deploy Oracle Analytics Server security using the embedded WebLogic LDAP Server and the default application.

You can also use the information in this topic to modify Oracle Analytics Server security settings after an application archive (BAR file) was 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.



Overview of Users, Groups, and Application Roles

Users, groups, and application roles determine who can access which Oracle Analytics Server components, folders, reports, data columns, and other objects. Typically, users and groups are stored in an external directory.

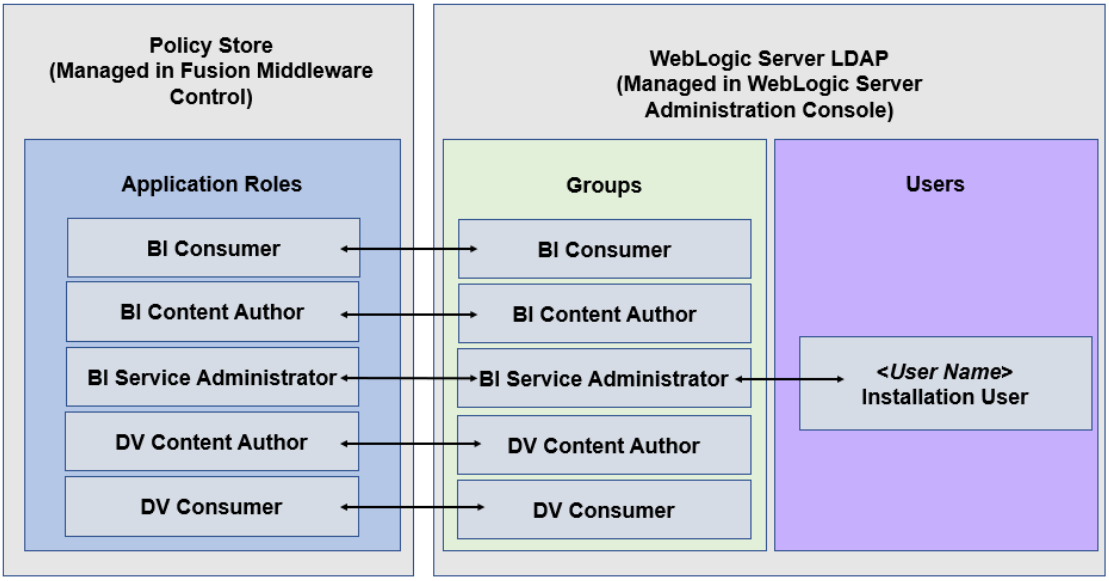
If during installation you selected the **Clean Slate (no predefined application)** option, then Oracle Analytics Server added the Clean Slate application to your service instance.

Clean Slate provides a set of default application roles suitable for general use. You can use these default application roles to configure user and group access to Oracle Analytics Server folders, reports, data columns, and other objects.

For example, following a new installation of Oracle Analytics Server, if you used Clean Slate to populate your initial instance, then the user specified for creating the BI domain during the configuration step is assigned to the BIServiceAdministrator application role. In addition, Clean Slate provides the BIContentAuthor, BIConsumer, BIDataModelAuthor, BIDataLoadAuthor, BIServiceAdministrator, DVConsumer, and DVContentAuthor application roles. These application roles are preconfigured to work together. For example, a user who is a member of the BIServiceAdministrator application role automatically inherits the BIContentAuthor and BIConsumer application roles and is therefore provisioned with all the privileges and permissions associated with all of these application roles.

The roles provided by Clean Slate have appropriate permissions and privileges to enable them to work with the default security policy. For example, the application role BIContentAuthor is preconfigured with permissions and privileges that are required to create dashboards, reports, actions, and so on.

The image below shows application roles that are preconfigured in the Clean Slate installation.



In the Clean Slate installation, the user specified during the Oracle Analytics Server installation is automatically added to the BIServiceAdministrator application role.

See *Installing and Configuring Oracle Analytics Server* and *importServiceInstance* in *Administering Oracle Analytics 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 Analytics Server Presentation Catalog, the semantic model, 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 Presentation Catalog and the semantic model 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 permissions that determine what users can see and do after signing in to Oracle Analytics Server. It's your job as an administrator to assign users and groups to one or more application roles.

There are two types of application role:

Type of Application Role	Description
Predefined	Include a fixed set of permissions.
User-defined	Created by administrators. 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. This application role is assigned all the available permissions.	Administrator who created the service
DV Content Author	Allows users to create workbooks, 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 (workbooks, 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	Allows users to create and manage semantic models using Semantic Modeler.	BI Service Administrator
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.

About Permissions

Permissions allow you to perform specific actions in Oracle Analytics Server. Administrators can grant specific permissions to application roles.

Permissions in Oracle Analytics Server

This table lists Oracle Analytics Server permissions.

Category	Resource Type	Permission	Description	Predefined Application Role
Catalog	**Connections**	**Create and Edit Connections**	Create and edit connections.	DV Content Author
		Create and Edit Connections to OCI Data Science with Resource Principal	Create and edit connections to Oracle Cloud Infrastructure Data Science using a resource principal. Not used in Oracle Analytics Server.	BI Service Administrator
		Create and Edit Connections to OCI Document Understanding with Resource Principal	Create and edit connections to Oracle Cloud Infrastructure Document Understanding using resource principal. Not used in Oracle Analytics Server.	BI Service Administrator
		Create and Edit Connections to OCI Functions with Resource Principal	Create and edit connections to Oracle Cloud Infrastructure Functions using a resource principal. Not used in Oracle Analytics Server.	BI Service Administrator

Category	Resource Type	Permission	Description	Predefined Application Role
		Create and Edit Connections to OCI Language with Resource Principal	Create and edit connections to Oracle Cloud Infrastructure Language using a resource principal. Not used in Oracle Analytics Server.	BI Service Administrator
		Create and Edit Connections to OCI Vision with Resource Principal	Create and edit connections to Oracle Cloud Infrastructure Vision using a resource principal. Not used in Oracle Analytics Server.	BI Service Administrator
	Data Flows	**Create and Edit Data Flows**	Create and edit data flows.	DV Content Author
		Create and Edit Sequences	Create and edit sequences.	DV Content Author
	Datasets	**Create and Edit Datasets**	Create and edit datasets.	DV Content Author
	Workbooks	**Create and Edit Watchlists**	Create and edit watchlists.	DV Content Author
		Create and Edit Workbooks	Create and edit workbooks.	DV Content Author
		Export Workbooks to Documents	Export workbooks to documents, such as PDF.	BI Consumer
		Schedule Workbooks	Set up and edit schedules for workbooks. Not used in Oracle Analytics Server.	BI Service Administrator
		Schedule Workbooks with Bursting	Set up and edit schedules for workbooks with bursting. Not used in Oracle Analytics Server.	BI Service Administrator
		Schedule Workbooks with RunAs User	Set up and edit schedules for workbooks with RunAs user. Not used in Oracle Analytics Server.	BI Service Administrator
		View Navigation Menu	View the curated list of dashboards and workbooks.	BI Consumer

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 four different views: User, Groups, Application Roles, Permissions.

Users and Roles Page	Description
Groups tab	Lists user groups from the identity domain associated with your Oracle Analytics instance. From the Groups tab, you can: • Discover the members (users or groups) directly assigned to each group. • Discover the application roles or any other groups that a group is directly assigned to. • Add or remove application roles assigned to a group. You can't add or remove user groups through the Groups tab. Use your identity management system to manage user groups.
Application Roles tab	Lists the predefined application roles for Oracle Analytics and any user-defined application roles that you add. From the Application Roles tab, you can: • Create your own application roles. • Discover the members (users, groups, application roles) directly assigned to each application role. • Discover the permissions directly granted to each application role. • • Add members or remove members from each application role. • Discover whether an application role is a member of any other application role. • Add or remove memberships for each application role. • Grant permissions to user-defined application roles. • Remove permissions from user-defined application roles. • Generate a report that lists the users assigned to an application role, either directly or indirectly. • Generate a report that lists the groups (or IDCS application roles) assigned to an application role, either directly or indirectly. • Generate a report that lists other application roles assigned to an application role, either directly or indirectly. • Generate a report that lists any other application roles an application role is assigned to, either directly or indirectly.
Permissions tab	Lists the permissions available in Oracle Analytics. From the Permissions tab, you can: • Search for permissions and filter the permissions list. • Discover the application roles a permission is directly assigned to. • Discover the users a permission is directly assigned to.

Add Members to Application Roles

Application roles determine what users 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:

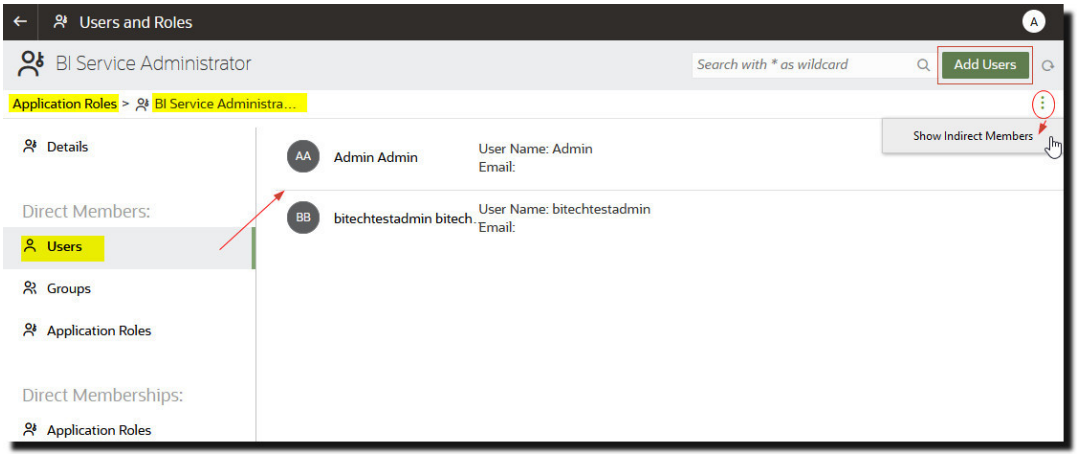
- Members (users, groups, and other application roles) get the permissions granted to an application role.
- Application roles can get permissions granted to other application roles. For example, DV Content Author gets the permissions granted to BI Content Author, DV Consumer, and BI Consumer.

You use the **Users and Roles** page in the Console to assign members to an application role.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles**.

   All the predefined application roles are displayed, together with any user-defined application roles that you've added.
4. Select the name of an application role for more detail, and to see its current members.
5. Under **Direct Members**, click **Users**, **Groups**, or **Application Roles** to view the current, direct members in each category.

   For example, if you click **Users** you see a list of users directly assigned to the application role.



6. To see a list of *all* the members in the selected category that are assigned to the application role (both directly and indirectly), click the menu icon and select **Show Indirect Members**.
7. To add a new member (user, group, application role) to the application role, click **Add Users**, **Add Groups**, or **Add Application Roles**, select one or more members, and then click **Add**.
8. To remove a member from the application role, click the **Delete** icon  next to the member's name.

Why Is the Administrator Application Role Important?

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

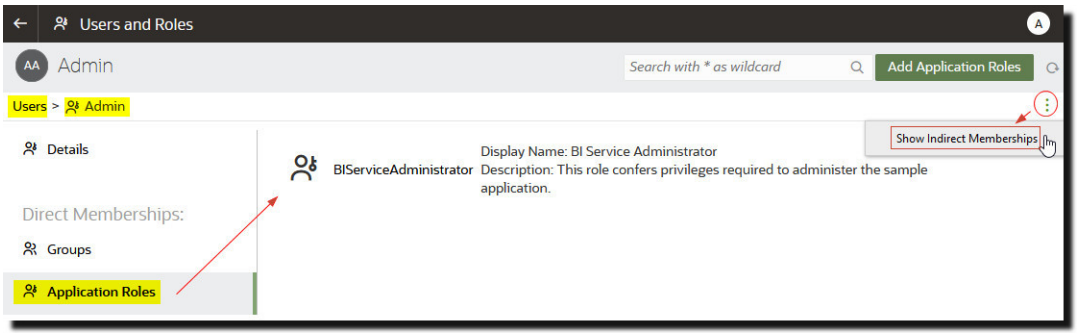
There must always be at least one person in your organization with the **BI Administrator** application role. This ensures there is always someone who can delegate permissions to others. If you remove yourself from the **BI Administrator** role you'll see a warning message.

Assign Application Roles to Users

The Users page lists the users from the identity domain associated with your Oracle Analytics Server instance. As an administrator, you can assign these users to the appropriate application roles.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Users**.
4. On the Users page, click the name of a user.

   To filter the list by name, enter all or part of a user name in the **Search** filter and press enter. If you enter part of the name use * as the wild card. The search is case-insensitive, and searches both name and display name. For example, enter `*admin*` to search for any user that includes the letters `admin`.
5. In the Details page for the user, click **Application Roles** to see a list of application roles directly assigned to this user.



6. Click the menu icon, and select **Show Indirect Memberships** to see a list of *all* the application roles assigned to the user, that is, assigned both directly and indirectly.
7. To assign the user to an additional application role, click **Add Application Roles**.
8. In **Add *user* to Application Roles**, select one or more application roles from the list, and then click **Add**.
9. To remove an application role from the user, click the **Delete** icon  next to the name of the application role you want to delete.

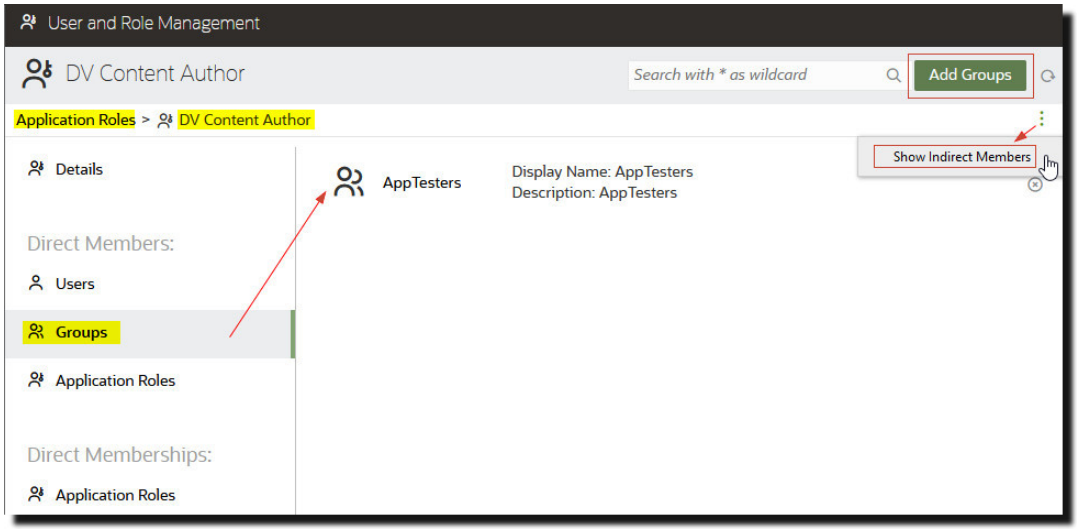
Assign Application Roles to Groups

The Groups page lists user groups from the identity domain associated with the Oracle Analytics Server instance. It's best practice to assign application roles to groups rather than to users.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles**.

   All the predefined application roles are displayed, together with any application roles that you've added.
4. Select the name of the application role you want to assign to a group.
5. Under **Direct Members**, click **Groups** to view the groups currently assigned to this application role.

   For example, there is a group called AppTesters directly assigned to the DV Content Author application role.



6. To see a list of *all* the groups that are assigned to the application role (both directly and indirectly), click the menu icon and select **Show Indirect Members**.
7. To assign a new group of users to the application role, click **Add Groups**, select one or more groups, and then click **Add**.
8. To remove a group from the application role, click the **Delete** icon next to the group's name.

Add Your Own Application Roles

Oracle Analytics Server provides a set of predefined application roles. You can also create user-defined application roles to suit your own requirements. For example, you might create an application role that allows only a select group of people to view specific folders or workbooks. Or you might create an application role with specific permissions assigned to it.

You can create an application role in two ways:

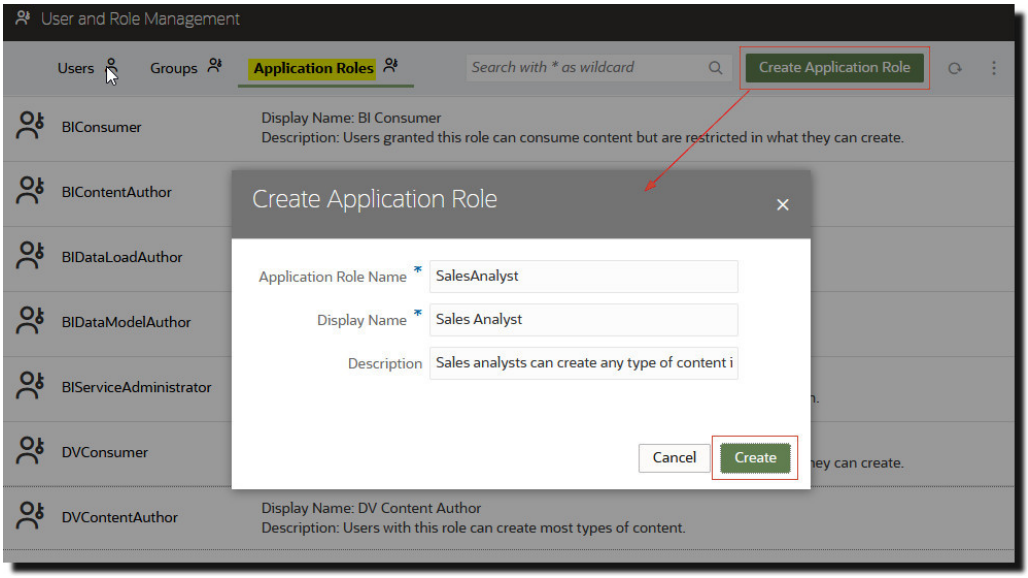
- Create an application role from scratch (no permissions).
- Create an application role with the same permissions as one of the predefined application roles.

After creating the application role, you can grant permissions and add members (users, groups, or other application roles).

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles**.
4. Do one of the following:

   Create an application role from scratch (no permissions):

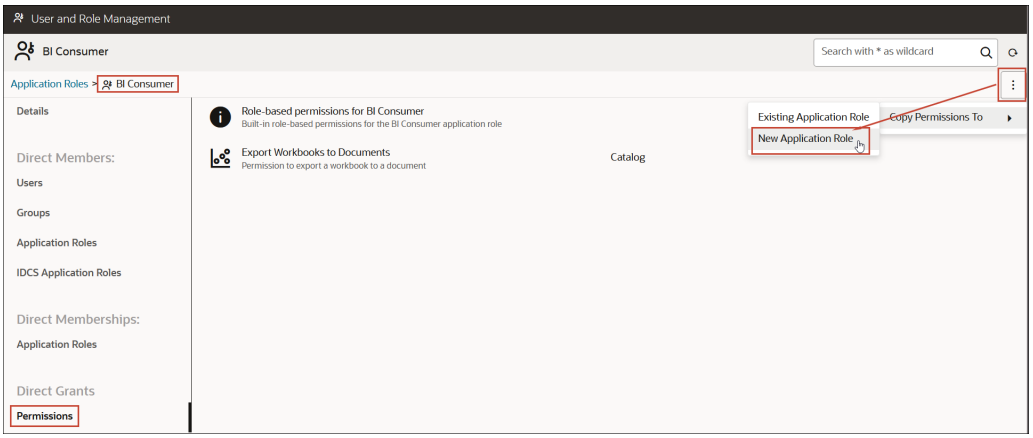
   - Click **Create Application Role**.



Copy the permissions from a predefined application role to a user defined application role:

> **Note:**
>
> In this step, you're copying the permission grants for the predefined application role that you choose. You aren't copying the application role's members or memberships.

- Click the name of the application role you want to copy. For example, BIConsumer.
- Click **Permissions**.
- Click the action menu, and select **Copy Permissions To** and then select **New Application Role**.



5. Enter suitable values for **Application Role Name**, **Display Name**, and **Description**.

   The **Application Role Name** can contain alphanumeric characters (ASCII or Unicode) and other printable characters (such as underscore or square brackets). The **Application Role Name** must not contain any white space.

6. Click **Create**.

   When you create an application role from scratch, it doesn't start with any members or permissions. When you copy the permissions from one of the predefined application roles, the application role starts with the same permissions as the role that you copied.

7. Grant permissions to the application role.

   a. Under **Direct Grants**, select **Permissions**.

   b. Click **Add Permissions**.

      This option is available only to user-defined application roles.

   c. Select one or more permissions, and then click **Add**.

8. Add members (users, groups, or application roles) to the new application role.

   a. Under **Direct Members**, select the type of member you want to add: **Users**, **Groups**, or **Application Roles**.

   b. Click **Add Users**, **Add Groups**, or **Add Application Roles**.

   c. Select one or more members, and then click **Add**.

9. Optional: Create hierarchical relationships between other application roles.

   a. Under **Direct Memberships**, click **Add to Application Roles**.

   b. Select all the application roles you want this application role to inherit privileges from, and then click **Add**.

Copy Permissions to an Existing User-Defined Application Role

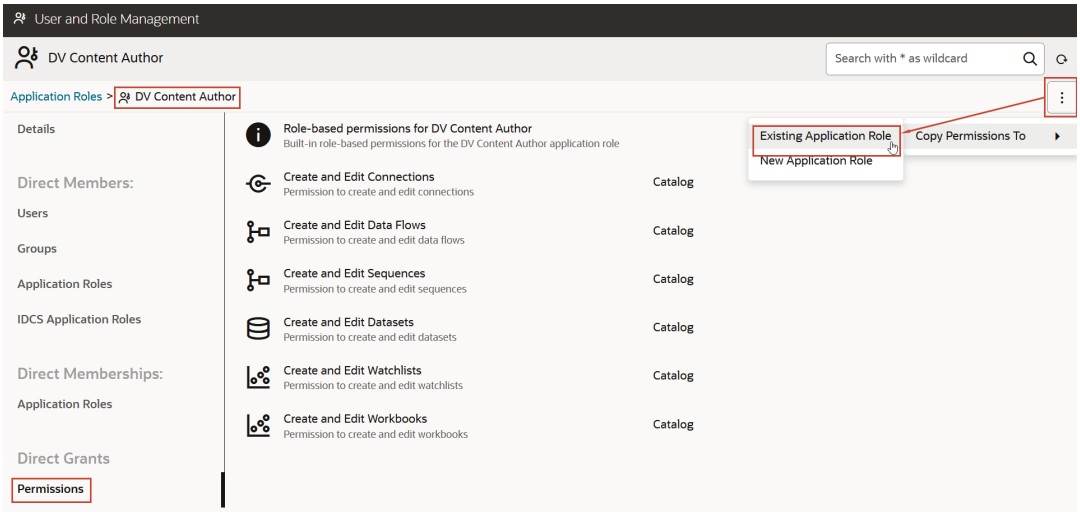
You can copy the permissions directly granted to a predefined application role to a user-defined application role.

After you copy permissions to an existing role, you can grant additional permissions or revoke any of the copied permissions. See Grant and Revoke Permissions for Application Roles.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles**.
4. Click the name of a predefined application role.

   To filter the list by name, enter all or part of a name in the **Search** filter and press enter. If you enter part of the name use * as the wild card. The search is case-insensitive, and searches both name and display name. For example, enter `*admin*` to search for any user that includes the letters `admin`.

5. Click **Permissions** to see the permissions granted to the predefined application role.
6. Click the action menu, select **Copy Permissions To**, and then select **Existing Application Role**.



7. Select an existing application role and click **Copy**.

View Permissions Granted to Application Roles

You can see a list of permissions granted to each *user-defined* application role as well as permissions granted to the predefined application roles from the Application Roles page.

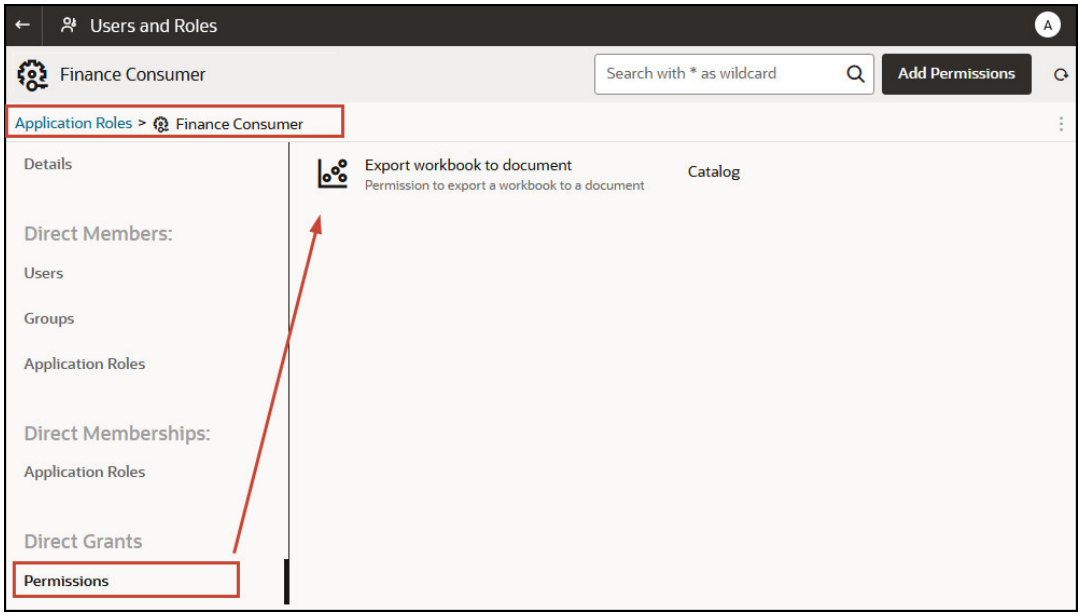
While you can view, add, and remove permissions for user-defined application roles, each predefined application role includes a fixed set of permissions that you can't change. Specifically, each predefined application role has a set of role-based permissions built into it which aren't listed individually, plus zero or more regular permissions which are listed individually but you can't remove them. For example, the predefined application role **BI Consumer** has built-in, role-based permissions plus the permission **Export Workbook to Document**.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles**.
4. Click the name of an application role.

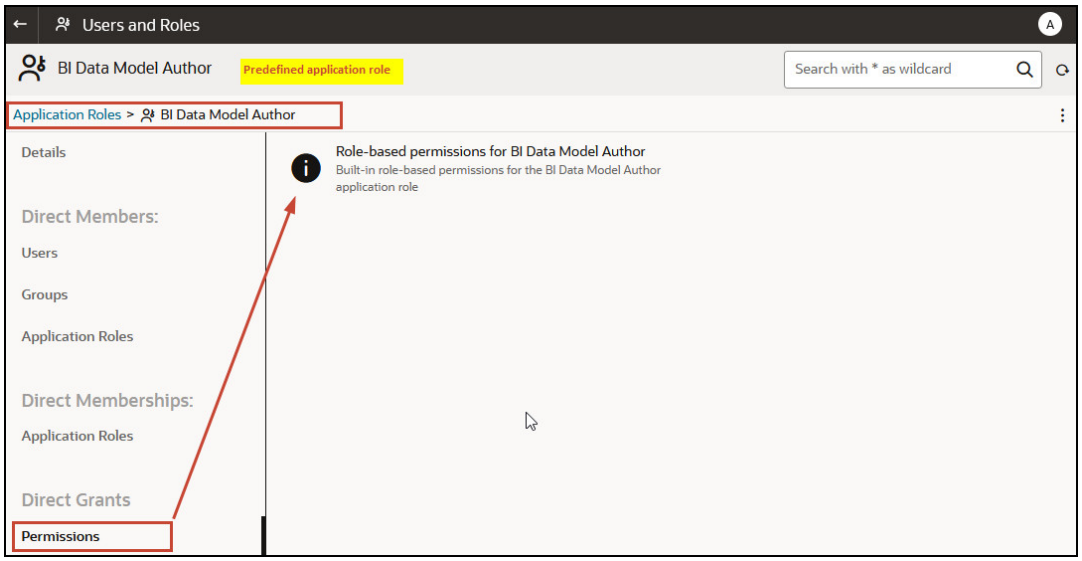
   To filter the list by name, enter all or part of a name in the **Search** filter and press enter. If you enter part of the name use * as the wild card. The search is case-insensitive, and searches both name and display name. For example, enter `*admin*` to search for any application role that includes the letters `admin`.
5. Click **Permissions** to see a list of permissions directly granted to the application role.

   When you select an application role that you created from scratch, you see a list of permissions granted to the role on the right. In this example, only one permission (**Export workbook to document**) is granted to an application role you created (**Finance Consumer**).

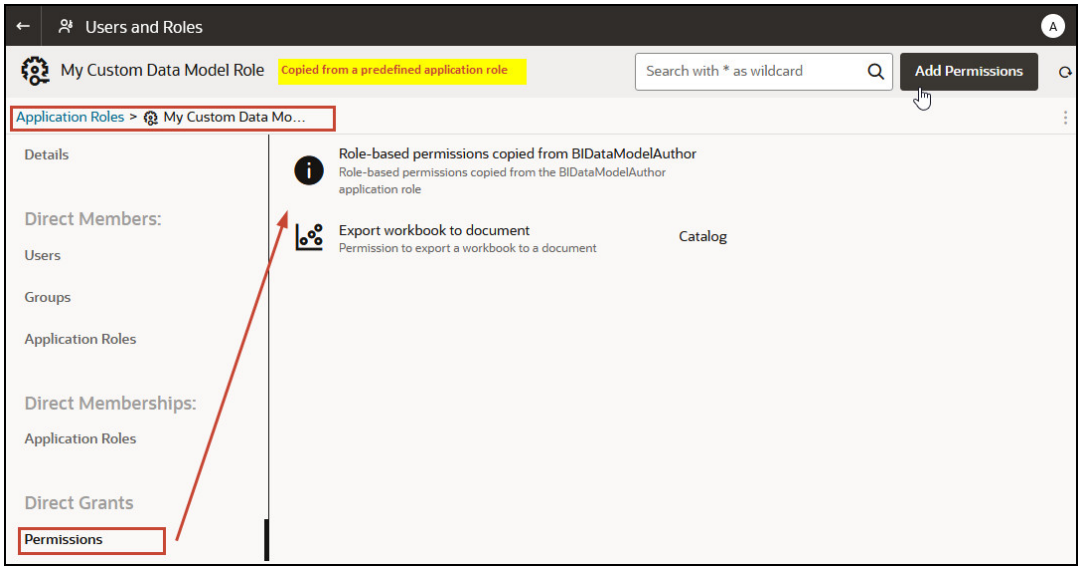
   You can add and delete permissions, as required.



When you select one of the predefined application roles, such as **BI Data Model Author**, you see a message indicating that the role contains a set of built-in, role-based permissions. You can't change the permissions granted to a predefined application role.



When you select a user-defined application role containing permissions copied from one of the predefined application roles, such as **BI Data Model Author**, you see a message indicating that the role contains a set of built-in, role-based permissions, plus any additional permissions assigned to the predefined application role, as well as any permissions that you granted the role.

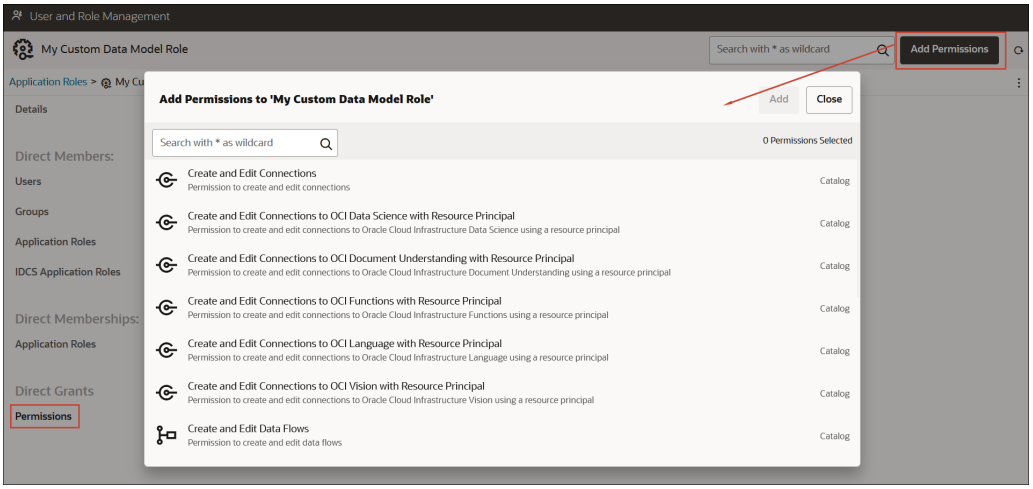


Grant and Revoke Permissions for Application Roles

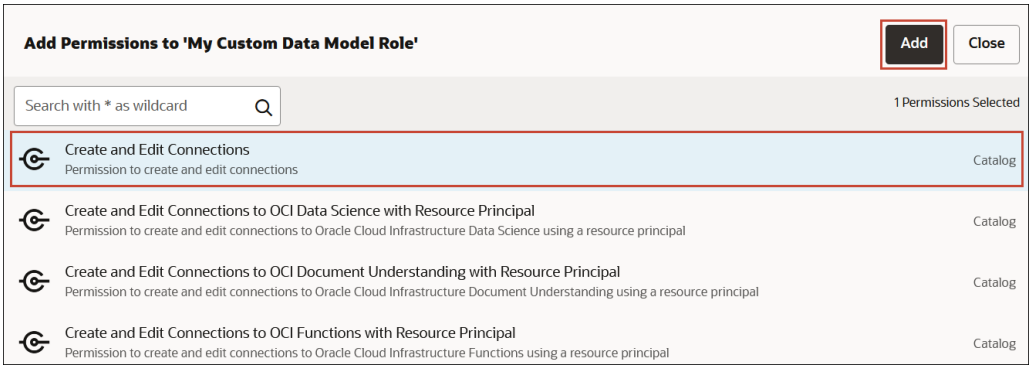
You can grant individual permissions to a *user-defined* application role or revoke permissions that are no longer required. For example, you might want to provide an application role that enables users to export their workbooks to a PDF by granting the permission *Export workbook to document*.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles**.
4. Click the name of a user-defined application role.

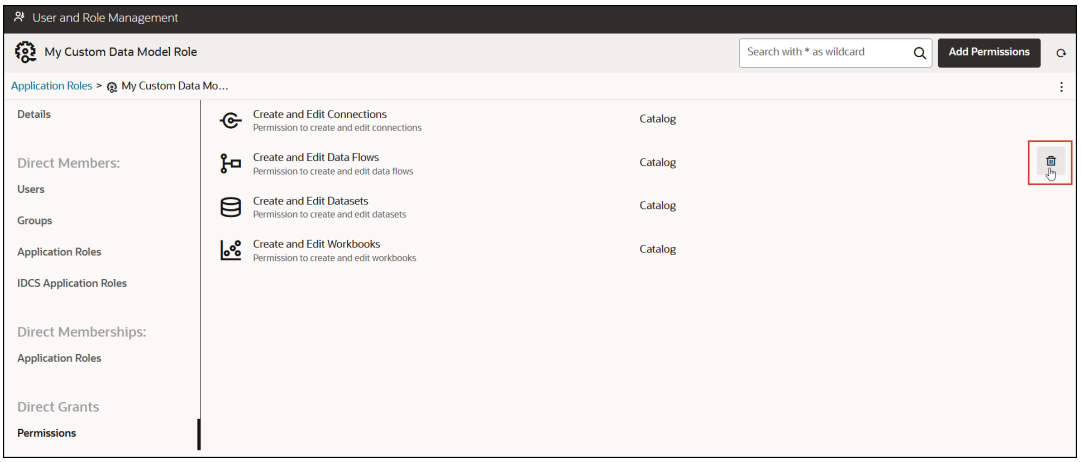
   To filter the list by name, enter all or part of a name in the **Search** filter and press enter. If you enter part of the name use * as the wild card. The search is case-insensitive, and searches both name and display name. For example, enter `*admin*` to search for any user that includes the letters `admin`.
5. Click **Permissions** to see the permissions granted to the user-defined application role.
6. To grant permissions to a user-defined application role.
   a. Click **Add Permissions**.



b. Select the permission you want, and click **Add**.



7. To revoke permissions from the application role.
   a. Navigate to the permission you want to revoke.
   b. Click the **Remove Permission** icon.
   c. To confirm, click **Remove**.



Delete Application Roles

You can delete user-defined application roles that you don't need anymore.

1. Click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles**.
4. Navigate to the user-defined application role you want to delete.

Add One Predefined Application Role to Another (Advanced)

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

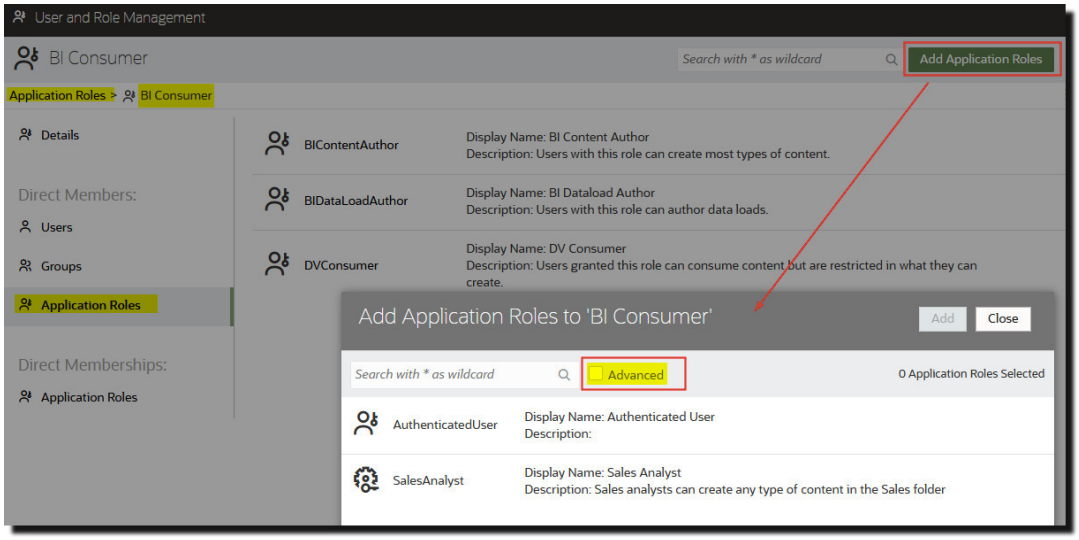
Any changes that you make to predefined application roles are permanent, so don’t perform this task unless you're sure you need to.

1. Take a snapshot of your system before making any predefined application role change.

   Oracle recommends that you always take a snapshot before you start, as the only way you can revert changes to predefined application roles is to restore your service from a snapshot that was taken *before* the change.

   a. Click **Console**.
   b. Click **Snapshots**.
   c. Click **Create Snapshot**.
2. In Console, click **Users and Roles**.
3. Click **Application Roles**.
4. Click the name of the predefined application role you want to change.
5. Under **Direct Members**, click **Application Roles** to see which application roles the selected application role is currently a member of.
6. Click **Add Application Roles**.

   By default, none of the predefined application roles are available.



7. To add a predefined application role, click **Advanced**.

> **WARNING:**
>
> A warning is displayed. Read the information carefully before you proceed. When you add one predefined application role to another, the change is permanent. The only way you can revert predefined application role changes is to restore a snapshot taken before the change.

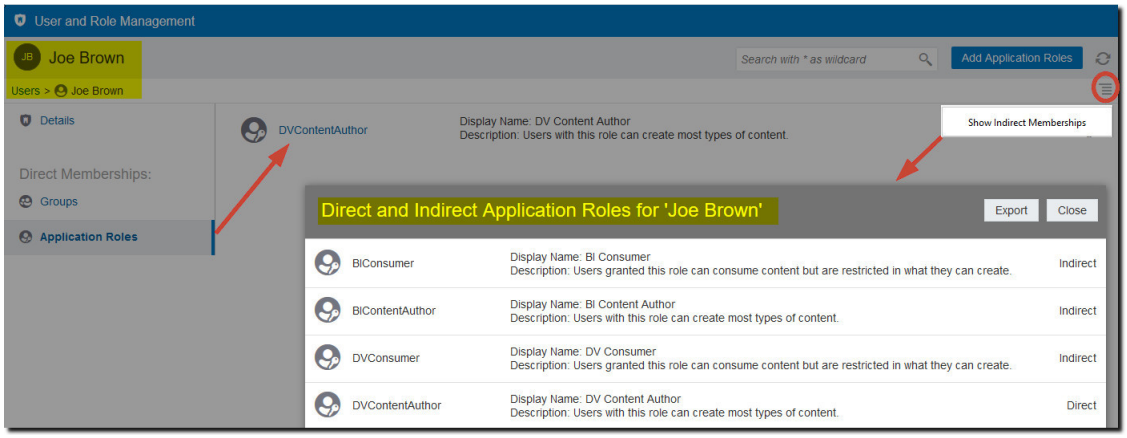
8. Click **OK** to confirm that you've taken a snapshot and you're sure you want to permanently modify the predefined application role you selected.
9. Select one or more predefined application roles from the list, and then click **Add**.
10. To reconfirm that you've taken a snapshot and want to permanently change the predefined application role, click **OK**.

View and Export Detailed Membership Data

Each application role in Oracle Analytics Server can have *direct* members, but they might also have one or more *indirect* members or memberships.

For example, Joe Brown is granted the DV Content Author application role. Joe is a direct member of the DV Content Author role and an indirect member of BI Consumer, BI Content Author, DV Consumer. You can view direct and indirect membership details from the **User and**

Role Management page and you can export this information to a CSV file.



1. Click **Console**.
2. Click **Users and Roles**.
3. To view direct and indirect membership data for a user:
   a. Click the **Users** tab.
   b. Select the name of the user whose membership details you want to see.
   c. Under **Direct Memberships**, click **Application Roles** to see a list of all the or application roles that the user you selected is *directly* assigned to.
   d. Click the menu icon, and select **Show Indirect Memberships** to see a list of *all* the or application roles that this user is both *directly* and *indirectly* assigned to.
4. To view direct and indirect membership data for an application role:
   a. Click the **Application Roles** tab.
   b. Select the name of the application role whose membership details you want to see.
   c. Under **Direct Members** (or **Direct Memberships**), click **Users**, **Groups**, or **Application Roles** to see a list of all the users, groups or application roles that the application role you selected is a *direct* member of (or *directly* assigned to).
   d. Click the menu icon, and select **Show Indirect Members** (or **Show Indirect Memberships**) to see a list of *all* the users, groups, or application roles that this group is both *directly* and *indirectly* a member of (or assigned to).
5. To export both direct and indirect membership data to a CSV file, click **Export**.

Download Membership Data

After displaying a list of the direct and indirect members for a user, group, or application role in Oracle Analytics Cloud, you can download the report to a Comma Separated Values file (.csv).

1. From the **Direct and Indirect *Users*** | ***Groups*** | ***Application Roles*** view, click **Export**.

   The direct and indirect members for the selected user, group, or application role are exported to a file named `RoleReport.csv`.
2. Do one of the following:
   - Click **Open** to open the CSV file in an application of your choice.
   - Click **Save** to save the CSV file to a location of your choice.

Sample Scenarios: User-defined Application Roles

Here are some common scenarios for creating your own application roles .

Topics:

Allow a User to Export Workbooks to PDF

You can give users permission to perform specific actions in Oracle Analytics. For example, you can enable users to export workbooks to PDF through an application role that includes the *Export Workbook to Document* permission.

> **Note:**
>
> The predefined application role **BI Consumer** includes the permission *Export Workbook to Document*. This means that any user who is a member of **BI Consumer** (either directly or indirectly) automatically has this permission.

1. Create a new application role called **Allow Document Export** (or use a similar name).

   See Add Your Own Application Roles.
2. Add the permission **Export Workbook to Document**.

   See Grant and Revoke Permissions for Application Roles.
3. Assign the new application role **Allow Document Export** to a user or a group.

   See Assign Application Roles to Users or Assign Application Roles to Groups.
4. Give users with the **Allow Document Export** application role access to one or more workbooks.

   These users can access workbooks and export the content to PDF.

   See Add or Update Workbook Permissions.

Prevent a User with the BI Consumer Role from Exporting Workbooks to PDF

You can prevent users from performing specific actions in Oracle Analytics. For example, you might want to provide an application role that prevents users with the **BI Consumer** role from exporting workbooks to a PDF by removing the permission *Export Workbook to Document*.

1. Copy the **BI Consumer** application role and name the copy **BI Consumer (prevent export)** (or use a similar name).
   a. Use the option **Copy Permissions to a New Application Role** to create an application role with the same permission set as **BI Consumer**.

b. Provide a suitable name and description for the new role. For example, **BI Consumer (prevent export)**.

See Add Your Own Application Roles.

2. Remove the **Export Workbook to Document** permission.

   See Grant and Revoke Permissions for Application Roles.

3. Assign the new application role **BI Consumer (prevent export)** to a user or a group.

   See Assign Application Roles to Users or Assign Application Roles to Groups.

4. Remove the predefined application role **BI Consumer** from the user or group.

5. Give users with the **BI Consumer (prevent export)** application role access to one or more workbooks and access to the folders where the workbooks are saved.

   When you give the **BI Consumer (prevent export)** application role access to the workbook, you must accept the option to cascade access to any datasets used by the workbook. That is, select the option **Share related artifacts to ensure the workbook is usable** in the **Share Related Artifacts** dialog that displays when you save changes to workbook permissions. See Add or Update Workbook Permissions.

   These users can access workbooks but they can't export the content to PDF.

   See Add or Update Workbook Permissions.

Allow a User to Create Datasets and Workbooks

You can give users permission to perform specific actions in Oracle Analytics. For example, you can enable users to create datasets and workbooks, and access and modify datasets and workbooks through an application role that includes the *Create and Edit Datasets* and *Create and Edit Workbooks* permissions.

> **Note:**
>
> The predefined application role **DV Content Author** includes the permissions *Create and Edit Datasets* and *Create and Edit Workbooks*. This means that any user who is a member of **DV Content Author** (either directly or indirectly) automatically has these permissions.

1. Create a new application role called **Allow Dataset and Workbook Creation** (or use a similar name).

   See Add Your Own Application Roles.

2. Add the permissions **Create and Edit Datasets** and **Create and Edit Workbooks**.

   See Grant and Revoke Permissions for Application Roles.

3. Assign the new application role **Allow Dataset and Workbook Creation** to a user or a group.

   See Assign Application Roles to Users or Assign Application Roles to Groups.

4. Give users with the **Allow Dataset and Workbook Creation** application role access to one or more datasets and one or more workbooks.

   These users can access and edit datasets and workbooks, and create datasets and workbooks.

   See Add or Update Workbook Permissions.

Prevent a User with the DV Content Author Role from Creating or Modifying Specific Object Types

You can prevent users from performing specific actions in Oracle Analytics. For example, you might want to provide an application role that prevents users with the **DV Content Author** role from creating and modifying connections, data flows, sequences, and watchlists.

1. Copy the **DV Content Author** application role and name the copy **DV Content Author (limited create and modify)** (or use a similar name).
   a. Use the option **Copy Permissions to a New Application Role** to create an application role with the same permission set as **DV Content Author**.
   b. Provide a suitable name and description for the new role. For example, **DV Content Author (limited create and modify)**.

   See Add Your Own Application Roles.
2. Remove the **Create and Edit Connections**, **Create and Edit Data Flows**, **Create and Edit Sequences**, and **Create and Edit Watchlists** permissions.

   See Grant and Revoke Permissions for Application Roles.
3. Assign the new application role **DV Content Author (limited create and modify)** to a user or a group.

   See Assign Application Roles to Users or Assign Application Roles to Groups.
4. Remove the predefined application role **DV Content Author** from the user or group.
5. Give users with the **DV Content Author (limited create and modify)** application role access to one or more workbook and datasets and access to the folders where the workbooks and datasets are saved.

   When you give the **DV Content Author (limit create and modify)** application role access to the workbook, you must accept the option to cascade access to any artifacts used by the workbook. That is, select the option **Share related artifacts to ensure the workbook is usable** in the **Share Related Artifacts** dialog that displays when you save changes to workbook permissions. See Add or Update Workbook Permissions.

   These users can access, create, and modify datasets and workbooks, but can't create and modify connections, data flows, sequences, and watchlists.

   See Add or Update Workbook Permissions.

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.

> **Note:**
>
> Oracle Analytics Server includes standard permissions that are assigned to predefined application roles. For example, the Create and Edit Datasets permission is automatically assigned to the DV Content Author role. These standard permissions are included in the permission sets listed below, and in some cases the standard permission are included when you grant a permissions set. If you want to grant or revoke standard permissions to user-defined application roles, use the Console. See Copy Permissions to an Existing User-Defined Application Role.

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. This permission is required to schedule or view the schedules for various objects (for example, data flows).
`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.

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

Permission Set Name	Permissions
bip.administrator	Publisher administration permissions.
bip.author	Publisher author permissions.
bip.consumer	Publisher consumer 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.

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

Permission Set Name	Permissions
`semanticmodeler.author`	Permissions to manage and deploy semantic models. Note that assigning this permission set allows users to bypass the Oracle BI Server security filters. See Grant Semantic Modeler Permissions Assignments.
`va.admin`	Data Visualization administration permissions.
`va.author`	Data Visualization author permissions.
`va.interactor`	Data Visualization basic interaction permissions.

Grant Semantic Modeler Permissions Assignments

Use the `grantPermissionSetsToBIRole` script to grant the permission sets required to access Semantic Modeler.

You'll need to grant these permissions when:

- **Upgrade** - You upgraded from Oracle BI EE to Oracle Analytics Server or from a previous version of Oracle Analytics Server.

  When you upgrade, the BI Data Model Author application role isn't automatically assigned the permissions required for users to manage and deploy semantic models from Semantic Modeler.

- **Snapshot** - You installed Oracle Analytics Server 2024 or later and then you import a snapshot from an earlier version of Oracle Analytics Server. See Restore from a Snapshot.

  Oracle Analytics Server restores the security policy from the snapshot. When you import a snapshot containing application roles and permission set grants different from the Oracle Analytics Server clean slate installation, Oracle Analytics Server restores the security policy from the snapshot. Because Semantic Modeler permissions are new for Oracle Analytics Server 2024, the snapshot from an earlier version won't contain the permissions required to access Semantic Modeler, so you'll need to grant them.

To grant the required permissions, use the `grantPermissionSetsToBIRole` script to assign the BI Data Model Author application role the `semanticmodeler.author` and `obis.administrator` permissions sets.

For example:

```
C:\OA_HOME\bi\modules\oracle.bi.security\scripts\grantPermissionSetsToBIRole.cmd -r BIDataModelAuthor -p semanticmodeler.author
```

> **Note:**
>
> Assigning the `semanticmodeler.author` permission set allows users to bypass the Oracle BI Server security filters.

For information about the script and how to use it, see Grant or Revoke Permission Assignments.

Manage Model Administration Tool Privileges

Use Identity Manager in the Model Administration Tool to configure security in the semantic model.

Topics:

Use Model Administration Tool

You use Model Administration Tool to configure permissions for users and application roles against objects in the semantic model.

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

If you log in to Model Administration Tool in offline mode, you can only view references to users that have previously been assigned permissions directly in the semantic model. The best practice is to assign semantic model permissions to application roles rather than directly to users.

1. Log in to Model Administration Tool and open a semantic model 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 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 Semantic Model Privileges for an Application Role

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

The author of your data model uses the Model 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 semantic model in Model 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 semantic model permissions for an application role from *Read* to *Read/Write*.

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

While working with a semantic model 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 Model Administration Tool to facilitate offline model development. But this is just a placeholder visible in the Model Administration Tool and isn't an actual application role. You can't create an actual application role in the Model Administration Tool.

An application role must be defined in the policy store for each application role placeholder created using the Model Administration Tool before bringing the semantic model back online. If a semantic model 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 Model Administration Tool interface. Always create a corresponding application role in the policy store before bringing the semantic model back online when using role placeholders in offline semantic model development.

Manage Session Variables

System session variables are session variables that Oracle BI Server and Oracle Analytics Server 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 semantic model 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 Model 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.
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 semantic model 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 Model Administration Tool, open a semantic model 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 Model Administration Tool, open a semantic model 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 Model Administration Tool, open a semantic model 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 Analytics Server 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

Presentation Catalog privileges. You can then delete the original Catalog group from Presentation Services.

1. Log in to Oracle Analytics Server 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.

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 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 Model 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).
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.

Section Name	Field Name	Description
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.

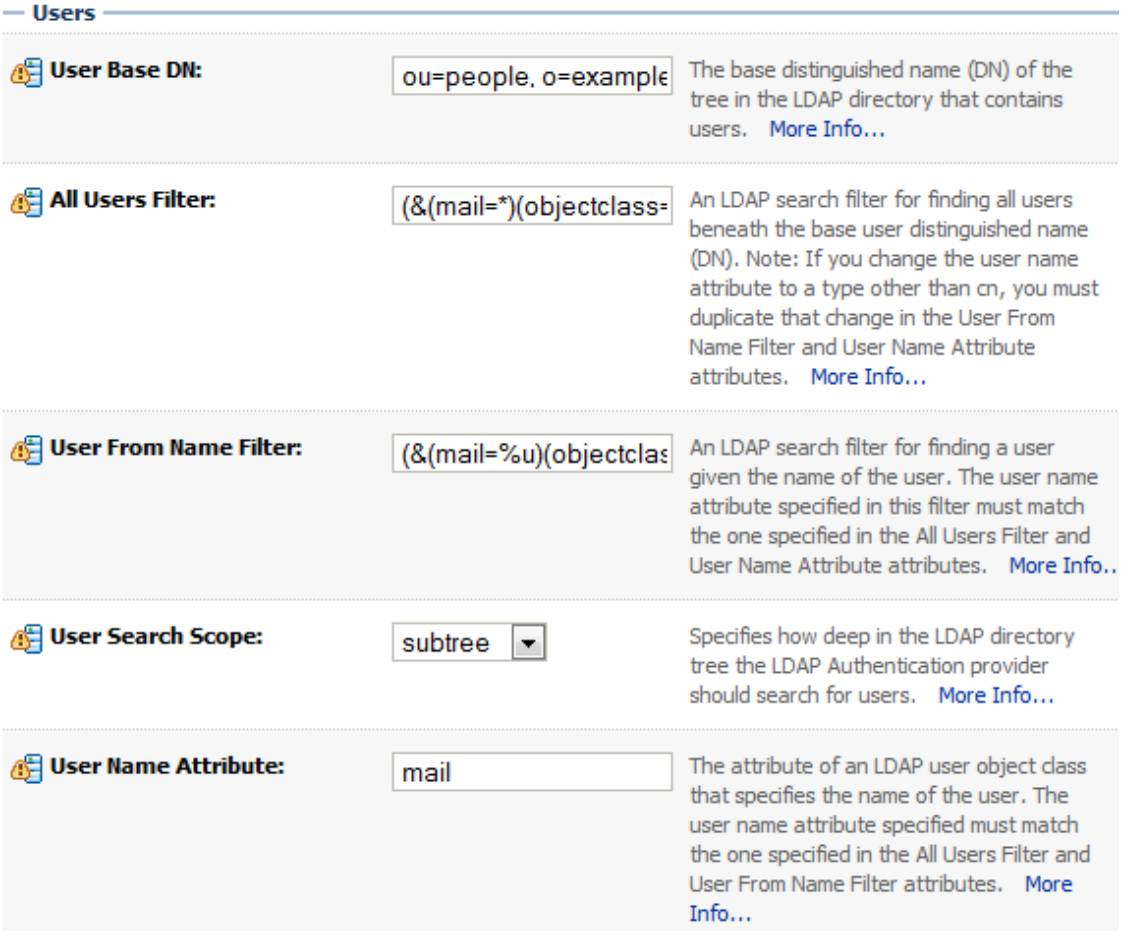
You might need to use alternative attributes for the user name, for example *uid* or *mail*. The need to use different group name attributes is less common. This section explains how to reconfigure user names and group names.

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.

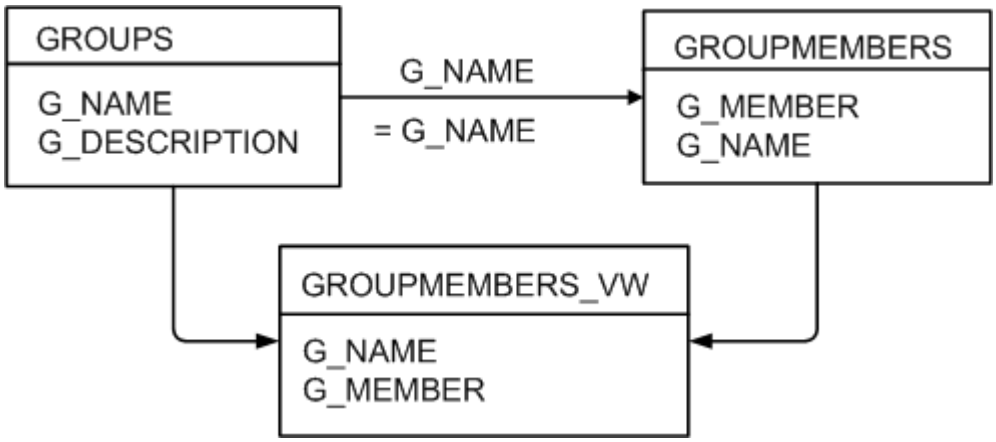
    You cannot have a group with the same name in both LDAP and database groups table. If you do, the security code invoked by Oracle Analytics Delivers cannot resolve application role membership.

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 and if necessary, any intermediate certificates 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.VirtualAttributePlu
gin</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).

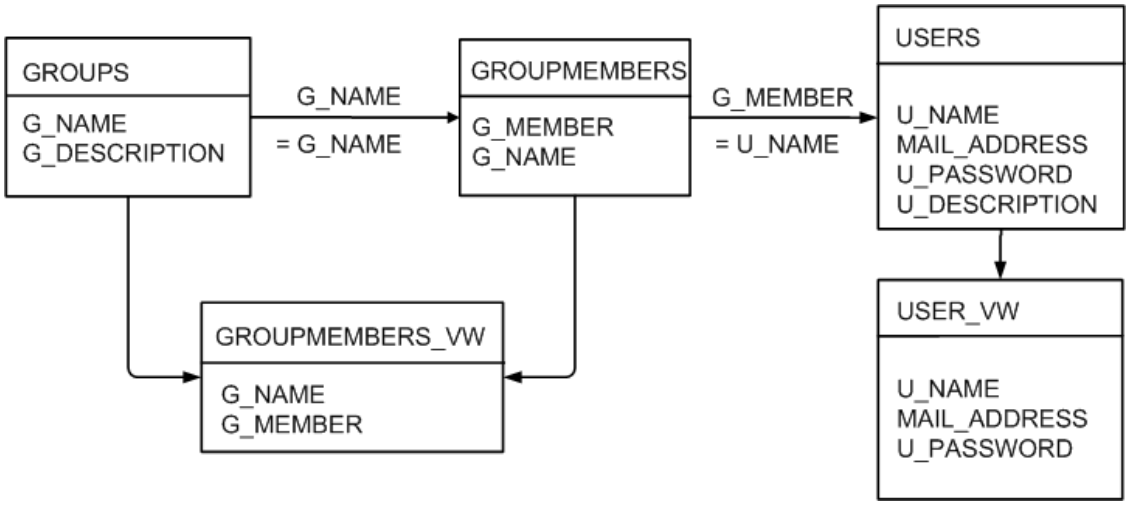
The approach to database authentication described here requires two database columns, one containing users and another containing passwords. This method is not based on database user accounts.

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.
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`.

Query	SQL	Notes
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:

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
- An Incorrect Data Source Name is Specified for the SQLAuthenticator
- Incorrect SQL Queries

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. If the target authenticator is using TLS/SSL, then locate the following element in the file:

   `<secure>`*false*`</secure>`

   Change the value of the `<secure>` element to *true* for each secure authenticator provider, as follows:

   `<secure>true</secure>`

4. Save and close the file.
5. 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
- Troubleshoot

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 Model Administration Tool online you must now provide the OID LDAP user and password, for example, OID_Weblogic, along with the semantic model 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 semantic model.

> **Note:**
>
> Oracle Analytics Server initialization block authentication has been deprecated and is no longer enabled for any use other than integrating with Oracle E-Business Suite Applications. You can use the information in this topic to update your existing initialization blocks.

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 semantic model.

You stop access through initialization blocks using the Model 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 semantic model.
2. Ensure that initialization blocks in the semantic model have the **Required for authentication** check box cleared.
3. Check that initialization blocks in the semantic model 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 Analytics Server as the original WebLogic administrator user, instead of as the OID LDAP user.

Configure Oracle Identity Cloud Integrator as the Authentication Provider

This section describes how to use the Oracle Identity Cloud Integrator provider to integrate Oracle Analytics Server with Oracle Identity Cloud Service for authentication.

In addition to authentication, you can also use Oracle Identity Cloud Service for SSO integration. The authentication steps described in this section are a prerequisite for configuring SSO against Oracle Identity Cloud Service. For more information, see Configure SSO with Oracle Identity Cloud Service and App Gateway.

Topics:

Create a Confidential Application for OAuth Client

In Oracle Identity Cloud Service you must create and set up a confidential application that uses OAuth.

For Oracle WebLogic Server to authenticate users with Oracle Identity Cloud Service, the Oracle Identity Cloud Integrator provider must be associated with an OAuth client that is registered with Oracle Identity Cloud Service. The OAuth client allows the provider access to Oracle Identity Cloud Service.

1. Log into Oracle Identity Cloud Service with tenant administrator credentials.
2. In the Oracle Identity Cloud Service console, expand the Navigation menu, and then click **Applications**.
3. On the Applications page, click **Add** and then in the Add Application dialog click **Confidential Application**.
4. In the Details section, enter a name and description to identify the application, and then click **Next**.
5. In the Client section, click **Configure this application as a client now** to configure the application's authorization settings.
6. In Authorization, click **Client Credentials** in Allowed Grant Types.
7. Scroll to Token Issuance Policy to assign the client to the Identity Domain Administrator application role. Under Grant the client access to Identity Cloud Service Admin APIs, click **Add**.
8. In App Roles, select **Identity Domain Administrator**.
9. Click **Next** until you reach the last step in the wizard, and then click **Finish**.
10. When the Application Added dialog is displayed, record the **Client ID** and **Client Secret** for use later in the configuration.
11. In the application's information page, click **Activate** to activate the application.

Required Configuration Attributes

To configure the Oracle Identity Cloud Integrator provider in Oracle WebLogic Server, you must provide the OAuth client attributes:

The configuration attributes enable communication between the Oracle Identity Cloud Integrator and Oracle Identity Cloud Service.

- **Tenant** - The name of the primary tenant in the Oracle Identity Cloud Service where you provisioned the OAuth client.

  The Oracle Identity Cloud Service tenant name is displayed in the browser URL when you click **My Services** to log in, or if you click **Open Admin Console** from the **Service Instances** section. The tenant name begins with the characters **idcs-** and then is followed by a string of numbers and letters.

- **ClientID** - The OAuth client ID used to access the Oracle Identity Cloud Service identity store.

  To find the OAuth ClientID, go to Oracle Identity Cloud Service, expand the **Navigation** menu, click **Applications**, and in the Applications list locate and open the OAuth application's details.

- **ClientSecret** - The OAuth Client Secret (password) used to generate access tokens.

  To find the OAuth ClientSecret, go to Oracle Identity Cloud Service, expand the **Navigation** menu, click **Applications**, and in the Applications list locate and open the OAuth application's details.

- **Client tenant** - (Optional) The name of the OAuth Client tenant where the Client Id resides. This attribute isn't required if the Client tenant is the same as the primary tenant.

Configure the Oracle Identity Cloud Integrator Provider

Use Oracle Analytics Server Oracle WebLogic Server Administration Console to configure the Oracle Identity Cloud Integrator provider.

The Oracle Identity Cloud provider configuration supplies access to the required users and groups.

To configure the Oracle Identity Cloud provider, you must add the provider to the security realm and specify the configuration attributes required to enable communication between the provider and Oracle Identity Cloud Service.

Note the following list of exceptions when you use the WebLogic Server documentation to configure Oracle Identity Cloud Service as an SSO provider for Oracle Analytics Server:

- Oracle Analytics Server can't use multiple authenticators for users. The Weblogic Server documentation states that you can have multiple authenticators, but this doesn't consider the Oracle Platform Security Services integration, which can only use SCIM or LDAP. Therefore when you use Oracle Identity Cloud Service, you can't use the `virtualize=true` setting.
- SSO uses perimeter authentication. App Gateway enforces the perimater protection and then passes a valid idcs_user_assertion token to Oracle WebLogic Server for an authenticated user.

You need the configuration attributes to complete the Oracle Identity Cloud Integrator configuration. See Required Configuration Attributes.

1. Log into Oracle Analytics Server WebLogic Server Administration Console.

2. Click **Lock and Edit**.
3. Navigate to Security Realms, then myrealm, then Providers, and then New.
4. In the Create a New Authentication Provider dialog, go to the **Name** field and enter a name for the authentication provider.
5. Go to the **Type** field and select **OracleIdentityCloudIntegrator**, and then click **OK**.
6. In the Authentication Providers dialog, move the authentication provider that you created to the top row of the table.
7. Navigate to Security Realms, then myrealm, then Providers, and then the name of the authentication provider that you created.
8. In new authentication provider's Settings page, click the **Common** tab.
9. In the **Control Flag:** field, select **SUFFICIENT**.
10. If you're using Oracle Identity Cloud Service for authentication and not for SSO, then in the **Active Types** field, move *both* idcs_user_assertion active types from the **Chosen** box to the **Available** box.
11. In the Settings page, click the **Provider Specific** tab to configure the Oracle Identity Cloud Integrator.
12. Scroll to Connection. Select the **SSLEnabled** field and provide values in the following fields:
    - **Host** - Enter `identity.oraclecloud.com`.
    - **Port** - Enter the port used to communicate with Oracle Identity Cloud Service. In most cases you can use `443`.
    - **Tenant** - Enter the name of the primary tenant in the Oracle Identity Cloud Service where you provisioned the OAuth client.
    - **Client Id** - Enter the OAuth client ID used to access the Oracle Identity Cloud Service identity store.
    - **Client Secret** - Enter the OAuth Client Secret (password) used to generate access tokens.
    - **Confirm Client Secret** - Reenter the OAuth Client Secret (password).
    - **Client Tenant** - (Optional) Enter the name of the OAuth Client tenant where the Client Id resides. This attribute isn't required if the Client tenant is the same as the primary tenant.
13. Click **Save**.
14. To change the idstore from ldap to scim, open Oracle Analytics Server and go here to open the jps-config.xml file

    `DOMAIN_HOME/bi/config/fmwconfig/jps-config.xml`
15. Locate `<serviceInstanceRef ref="idstore.ldap"/>` and change `.ldap` to `.scim`.
16. Click **Activate changes**.

Configure TLS/SSL for the Oracle Identity Cloud Integrator Provider

The Oracle Identity Cloud Integrator provider supports one-way SSL. To secure the connection using TLS/SSL, you need to establish trust between Oracle WebLogic Server and Oracle Identity Cloud Service.

To do this, you may need to obtain the Oracle Identity Cloud Service SSL certificate and import it into the Oracle WebLogic Server trust store.

In most cases you don't need to import the certificate because Oracle Weblogic Server trusts the Oracle Identity Cloud Service certificate. Oracle Identity Cloud Service contains a certificate signed by a well-known certificate authority (CA) such as Symantec, and your WebLogic domain is using Java Standard Trust.

However, you should use this procedure if you need to configure Oracle Weblogic Server to accept certificates that use wildcards. Or if your domain is configured for custom trust, you may need to import the Intermediate CA and root CA certificates into your trust store, regardless of whether Oracle Identity Cloud Service is using a well-known CA.

1. To configure TLS/SSL, go to the Oracle Identity Cloud Integrator provider and set the **SSLEnabled** attribute to true. Then set the **idcsPort** attribute to the appropriate SSL port for Oracle Identity Cloud Service.
2. To configure host name verification in Oracle WebLogic Server using the wild card host name verifier to allow WebLogic Server to accept certificates containing wildcards, open the `DOMAIN_HOME/bin/setDomainEnv.sh` script.
3. In the `setDomainEnv.sh` script, navigate to the `EXTRA_JAVA_PROPERTIES` section, and add this property:

   ```
   -Dweblogic.security.SSL.hostnameVerifier=weblogic.security.utils.SSLWLSWildcardHostnameVerifier
   ```

4. Restart Oracle Weblogic Server.

Add Users and Groups from Oracle Identity Cloud Service to Oracle Analytics Server

Users and groups from Oracle Identity Cloud Service aren't listed in Oracle WebLogic Server Administration Console. Instead, you add and manage these users and groups from the Console in Oracle Analytics Server.

Adding the Oracle Identity Cloud Service users and groups to Oracle Analytics Server's application roles determines what the users can see and do after signing into Oracle Analytics Server. See Get Started with Application Roles.

1. In the Oracle Analytics Server's Home page, click **Console**.
2. Click **Users and Roles**.
3. Click **Application Roles** and then click the application role to add Oracle Identity Cloud Service users and groups to.
4. To add a new member (user or group) to the application role, click **Add Users** or **Add Groups**. Select one or more members, and then click **Add**.

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 Oracle Analytics Server 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

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 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 Model 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 Analytics Server 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:

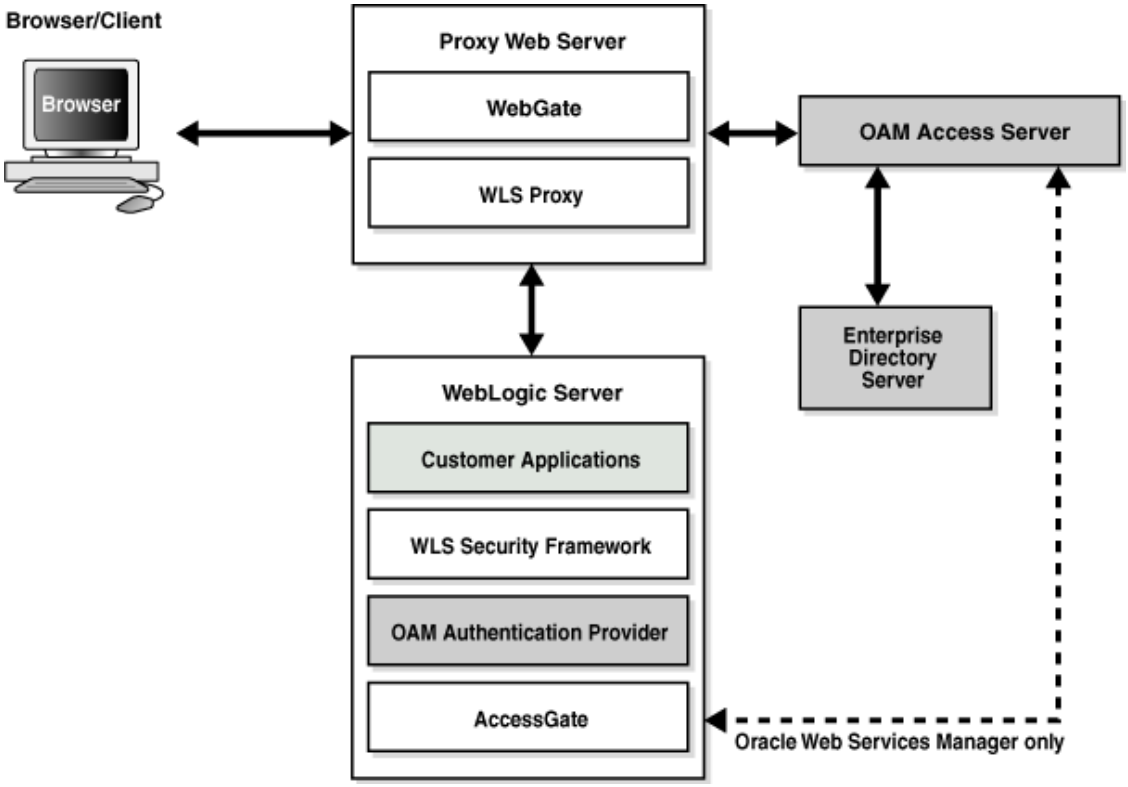
- **Identity Asserter for Single Sign-on**
  This feature uses the Oracle Access Manager authentication services and validates already-authenticated Oracle Access Manager users through a suitable token and creates a WebLogic-authenticated session. It also provides single sign-on between WebGate and portals. 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, Presentation Services instances, and Oracle Analytics Server *JavaHost* instances.

If any of these components are co-located with 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.

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*.
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 SSO with Oracle Identity Cloud Service and App Gateway

This topic describes the process that you need to follow to configure SSO with Oracle Identity Cloud Service and App Gateway.

This topic contains the following sections that explain each security set up step, links to the Oracle guides and topics to follow to configure SSO with Oracle Identity Cloud Service and App Gateway, and any additional configuration information specific to Oracle Analytics Server:

- Configure the Oracle Identity Cloud Integrator Provider In WebLogic Server
- Install and Configure App Gateway
- Create and Configure an Oracle Identity Cloud Service Enterprise Application

- Protect URLs or Make Them Public

Configure the Oracle Identity Cloud Integrator Provider In WebLogic Server

The Oracle Identity Cloud Integrator provider combines authentication and identity assertion into a single provider. The provider establishes identity (the Subject) on WebLogic Server with the authenticated user and the user's groups when the identity store is the Oracle Identity Cloud Service.

You must configure Oracle Identity Cloud Service as Oracle Analytics Server's authentication provider either before or at the same time you configure Oracle Analytics Server to use Oracle Identity Cloud Service as the SSO provider.

If you've already configured Oracle Identity Cloud Service as Oracle Analytics Server's authentication provider, then go to the Oracle Analytics Server WebLogic Server Administration Console and configure the provider to accept the Oracle Identity Cloud Service user assertion tokens. These are the tokens that provide SSO for a user. To update the configuration, go to the **Active Types** field and move idcs_user_assertion and ldcs_user_assertion from the **Chosen** box to the **Available** box.

Install and Configure App Gateway

App Gateway acts as a reverse proxy protecting web applications by restricting unauthorized network access to them. App Gateway intercepts any HTTP request to these applications and ensures that the users are authenticated with Oracle Identity Cloud Service before forwarding the request to these applications. App Gateway propagates the authenticated user's identity to the applications using a token.

Use App Gateway to:

- Integrate enterprise applications hosted either on-premises or in a cloud infrastructure with Oracle Identity Cloud Service for authentication purposes.
- Expose intranet web applications to internet access.
- Integrate with applications that lack a native authentication mechanism and don't support SAML federation, OAuth, or OpenID Connect integration methods.
- Integrate with applications that support the HTTP header-based authentication.

For information about how to install App Gateway, see Set Up an App Gateway.

Create and Configure an Oracle Identity Cloud Service Enterprise Application

When you add App Gateway to the SSO configuration, then you need to go to Oracle Identity Cloud Service and add an enterprise application that interacts with App Gateway.

For information about how to create and configure the enterprise application, see Add an Enterprise Application.

Protect Oracle Analytics Server URLs or Make Them Public

In the Oracle Identity Cloud Service enterprise application, you must add the following Oracle Analytics Server URLs (resources), specify if they are public (public resources) or protected (resources protected by form or token), and select the **Allow CORS** and **Require Secure Cookies** authentication policies to apply them to the URLs. App Gateway enforces these policies in the enterprise application.

Resource	Public	Protected
/analytics/?.*	-	Yes

Resource	Public	Protected
/analytics/saw.dll/wsdl?.*	Yes	-
/analytics-bi-adf/?.*	Yes	-
/analytics-ws?.*	Yes	-
/api/?.*	Yes	-
/aps/?.*	Yes	-
/aps/JAPI/?.*	Yes	-
/aps/SmartView/?.*	-	Yes
/bicontent/?.*	-	Yes
/bi-lcm/?.*	Yes	-
/biinfer/?.*	-	Yes
/bi-sac-config-mgr/?.*	-	Yes
/bisearch/?.*	-	Yes
/bi-security-login/?.*	Yes	-
/biserviceadministration/?.	-	Yes
/biservices/?.*	Yes	-
/cds/?.*	-	Yes
/dv/?.*	-	Yes
/mapviewer/?.*	-	Yes
/mapviewer/dataserver/?.*	Yes	-
/mapviewer/foi/?.*	Yes	-
/mapviewer/mcserver/?.*	Yes	-
/mapviewer/wms/?.*	Yes	-
/mapviewer/wmts/?.*	Yes	-
/mobile/?.*	-	Yes
/security/?.*	-	Yes
/xmlpserver/?.*	-	Yes
/xmlpserver/Guest?.*	Yes	-
/xmlpserver/report_service/?.*	Yes	-
/xmlpserver/ ReportTemplateService.xls?.*	Yes	-
/xmlpserver/services/?.*	Yes	-
/bimajel/?.*	-	Yes
/analytics/res/?.*	Yes	-
/dv/public/?.*	Yes	-
/dv/ui/api/?.*	Yes	-
/dv/static/?.*	Yes	-
/logout	-	Yes

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

To configure a fully supported integration of Oracle Analytics Server using Kerberos and SAML2 WebSSO, you can use Oracle Access Manager in front of Oracle Analytics Server. The appropriate Oracle Access Manager license is required for this configuration.

Alternatively, you can use open source components for Kerberos and SAML2 WebSSO. A reference implementation for custom Kerberos and SAML2 WebSSO using open source components is provided.

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.

- 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 isn't 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.

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 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:

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 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:

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

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

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

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

   DomainHome/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:

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

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

   DomainHome/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, Model Administration Tool, 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 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 Model 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 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:

   Windows: *<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 the Model Administration Tool to Communicate Over SSL

To successfully connect to a BI Server configured to use SSL, you must also configure the Model 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 Model Administration Tool, select **File**, then **Open**, then **Online**.
4. Select the DSN from the list.
5. Enter the semantic model user name and password.

   The Model 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.

For Oracle Analytics Publisher to use Oracle Analytics Server as a data source when SSL is enabled, you must update the default connection string. See Set Up a JDBC Connection to the Oracle Analytics Server and Update the JMS Configuration.

If Oracle Analytics Publisher doesn't work after configuring SSL, you might need to reconfigure the HTTPs protocol and SSL Port. See Integrate with Presentation Services.

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 Oracle Analytics Server 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.

> **Note:**
>
> If the LDAP server is using TLS/SSL and is using a certificate signed by an intermediate certificate authority, you need to import the intermediate and root CA certificates into the libOVD trust store.

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 and any intermediate certificates from the LDAP directory.
6. Use the following keytool command to import the root and any intermediate certificates into 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

A

Managing Security for Dashboards and Analyses

This appendix explains how to manage security for dashboards and analyses such that users have only:

- Access to objects in the Oracle BI Presentation Catalog that are appropriate to them.
- Access to features and tasks that are appropriate to them.
- Access to saved customizations that are appropriate to them.

This appendix contains the following sections:

Managing Security for Users of Presentation Services

As a system administrator, you must configure a business intelligence system to ensure that all functionality including administrative functionality is secured by providing access only to authorized users that are allowed to perform appropriate operations. You must configure the system to secure all middle-tier communications.

This overview section contains the following topics:

Security Settings in Presentation Services

Security settings that affect users of Presentation Services are made in the following Oracle Analytics Server components:

- Use the Model Administration Tool to perform the following tasks:
  - Set permissions for business models, tables, columns, and subject areas.
  - Specify database access for each user.
  - Specify filters to limit the data accessible by users.
  - Set authentication options.
- Presentation Services Administration enables setting privileges for users to access features and functions such as editing views and creating agents and prompts.

- Presentation Services enables assigning permissions for objects in the Presentation Catalog.

> **Note:**
>
> Security Administrators should advise report users to not edit Subject Area security privileges within Presentation Services. The Security Administrator should enforce data security.

What Are the Security Goals in Oracle BI Presentation Services?

This topic provides guidelines for security with Oracle BI Presentation Services.

When maintaining security in Presentation Services, you must ensure the following:

- Only the appropriate users can sign in and access Presentation Services. You must assign sign-in rights and authenticate users through the BI Server.

  Authentication is the process of using a user name and password to identify someone who is logging on. Authenticated users are then given appropriate authorization to access a system, in this case Presentation Services. Presentation Services doesn't have its own authentication system; it relies on the authentication system that it inherits from the BI Server.

  All users who sign in to Presentation Services are granted the `AuthenticatedUser` role and any other roles that they were assigned in Fusion Middleware Control.

  For information about authentication, see About Authentication.
- Users can access only the objects that are appropriate to them. You apply access control in the form of permissions, as described in *Visualizing Data in Oracle Analytics Server*.
- Users have the ability to access features and functions that are appropriate to them. You apply user rights in the form of privileges. Example privileges are *Edit system wide column formats* and *Create agents*.

  Users are either granted or denied a specific privilege. These associations are created in a privilege assignment table, as described in Managing Presentation Services Privileges.

You can configure Oracle Analytics Server to use the single sign-on feature from the web server. Presentation Services can use this feature when obtaining information for end users. See Enable SSO Authentication.

How Are Permissions and Privileges Assigned to Users?

When you assign permissions and privileges in Presentation Services, you can assign them in one of the following ways:

- To application roles — This is the recommended way of assigning permissions and privileges. Application roles provide much easier maintenance of users and their assignments. An application role defines a set of permissions granted to a user or group that has that role in the system's identity store. An application role is assigned in accordance with specific conditions. As such, application roles are granted dynamically based on the conditions present at the time authentication occurs.

  See About Application Roles.

- To individual users — You can assign permissions and privileges to specific users, but such assignments can be more difficult to maintain and so this approach isn't recommended.

Using Oracle BI Presentation Services Administration Pages

You can use the Administration pages in Oracle BI Presentation Services to perform the tasks that are described in the following sections:

- Understanding the Administration Pages
- Managing Presentation Services Privileges
- Managing Sessions in Presentation Services

Understanding the Administration Pages

The main Oracle BI Presentation Services Administration page contains links that allow you to display other administration pages for performing various functions, including those related to users in Presentation Services.

You can obtain information about all these pages by clicking the Help button in the upper-right corner.

> **Note:**
>
> Use care if multiple users have access to the Administration pages, because they can overwrite each other's changes. Suppose User A and User B are both accessing and modifying the Manage Privileges page in Presentation Services Administration. If User A saves updates to privileges while User B is also editing them, then User B's changes are overwritten by those that User A saved.

Managing Presentation Services Privileges

This section contains the following topics about Presentation Services privileges:

- What Are Presentation Services Privileges?
- Presentation Services Privileges

What Are Presentation Services Privileges?

Presentation Services privileges control the rights that users have to access the features and functionality of Presentation Services. Privileges are granted or denied to specific application roles and individual users using a privilege assignment table.

Like permissions, privileges are either explicitly set or are inherited through role or group membership. Explicitly denying a privilege takes precedence over any granted, inherited privilege. For example, if a user is explicitly denied access to the privilege to edit column formulas, but is a member of an application role that has inherited the privilege, then the user can't edit column formulas.

Privileges are most commonly granted to the BIContentAuthor or BIConsumer roles. This allows users access to common features and functions of Presentation Services.

See Setting Presentation Services Privileges for Application Roles.

Presentation Services Privileges

You can manage privilege assignments:

1. Click **My Profile** and **Administration**.
2. Under Security click **Manage Privileges**.

Access to Oracle Analytics Server Actions

You must set the Action privileges that determine whether the Actions functionality is available to users, and specify which user types can create Actions.

The following list describes these privileges:

- Create Navigate Actions

  The Create Navigate Actions privilege indicates whether the user can create a Navigate action type. Users who are denied this privilege don't have the user interface components that allow the creation of Navigate Actions. Users without the Create Navigate Actions privilege can add saved actions to analyses and dashboards, and execute an action from an analysis or dashboard that contains an action.

- Create Invoke Actions

  The Create Invoke Actions privilege indicates whether the user can create an Invoke action type. The Invoke Actions options include Invoke a Web Service, and Invoke an HTTP Request. However, users who are denied this privilege can add saved actions to analyses and dashboards. And, users who are denied this privilege can execute an action from an analysis or dashboard that contains an action.

- Save Actions Containing Embedded HTML

  The Save Actions Containing Embedded HTML privilege indicates whether users can embed HTML code in customized web service action results. You should use extreme care in assigning the Save Actions Containing Embedded HTML privilege, because users with this privilege can pose a security risk allowin users to run HTML code.

Access to Oracle BI for Microsoft Office Privilege

If your users have the Access to Oracle BI for Microsoft Office privilege, they can interact with Microsoft Office from Oracle Analytics Server.

When a user has the Access to Oracle BI for Microsoft Office privilege, then the Smart View for MS Office link is available from the Download Desktop Tools menu on the Oracle Analytics Server Home page.

The Access to Oracle BI for Microsoft Office privilege doesn't affect the display of the **Copy** link for analyses. The link is always available there.

Save Content with HTML Markup Privilege

By default, Presentation Services is secured against cross-site scripting (XSS).

Securing against XSS escapes input in fields in Presentation Services and renders it as plain text. For example, an unscrupulous user can use an HTML field to enter a script that steals data from a page.

By default, end users can't save content that's flagged as HTML. Only administrators who have the Save Content with HTML Markup privilege can save content that contains HTML code. Users that have the Save Content with HTML Markup privilege can save an image with the *fmap* prefix. If users try to save an image with the *fmap* prefix when they don't have this privilege assigned, then they see an error message. See EnableSavingContentWithHTML.

Users with this privilege can also save mission and vision statements in Oracle Scorecard and Strategy Management.

EnableSavingContentWithHTML

The EnableSavingContentWithHTML element along with the Save Content With HTML Markup and Save Actions Containing Embedded HTML privileges determine whether the **Contains HTML Markup** option is available in properties dialogs when editing analyses.

As the BI Service Administrator, you can use the EnableSavingContentWithHTML element to enable all HTML editing and you can grant the related privileges to users. You set the EnableSavingContentWithHTML element to *true* in the instanceconfig.xml file, and you grant users the Save Content With HTML Markup and Save Actions Containing Embedded HTML privileges in the Manage Privileges page to enable the **Contains HTML Markup** option. See Default Presentation Services Privileges Assignments and Making Advanced Configuration Changes for Presentation Services.

For the location of the instanceconfig.xml file, see Configuration Files.

Managing Sessions in Presentation Services

Using the Session Management page in Presentation Services Administration, you can view information about active users and running analyses, cancel requests, and clear the cache.

1. From the Home page in Presentation Services, select **Administration**.
2. Click the **Manage Sessions** link.

   The Session Management screen is displayed with the following tables:

   - The Sessions table, which gives information about sessions that have been created for users who have logged in:
   - The Cursor Cache table, which shows the status of analyses:

To cancel all running requests:

1. Click **Cancel Running Requests**.
2. Click **Finished**.

Cancel one running analysis as shown below.

- In the Cursor Cache table, identify the analysis and click the **Cancel** link in the **Action** column.

  The user receives a message indicating that the analysis was canceled by an administrator.

Use these steps to clear the web cache.

1. In the Cursor Cache table, identify the analysis and click **Close All Cursors**.
2. Click **Finished**.

Clear the cache entry associated with an analysis as described below.

- In the Cursor Cache table, identify the analysis and click the **Close** link in the **Action** column.

View the query file for information about an analysis as described below.

- In the Cursor Cache table, identify the analysis and click the **View Log** link.

> **Note:**
>
> Query logging must be turned on for data to be saved in this log file.

Determining a User's Privileges and Permissions in Presentation Services

Presentation Services privileges and Presentation Services Catalog item permissions, use an Access Control List (ACL) to control who has privilege to access Presentation Services functionality and what permissions any given user can have on Presentation Services Catalog items. Privileges are set using the Administration pages in Oracle BI Presentation Services. Permissions are set for Presentation Services Catalog objects through the Analytics user interface.

When you try to access functionality in Presentation Services, the appropriate privilege is checked; for example, to view the Oracle Analytics Server page you must have the Access to Answers privilege. Also, when you try to perform any action on a Presentation Services Catalog item, that item's permissions are checked; for example, to view an item in Oracle Analytics Server, the item's permissions are checked to see if you have read access.

The types of records that you may add to an ACL:

- Individual user records

  It is difficult to administer individual user records especially when there might be thousands of users, and hundreds of thousands of Catalog items.

- Application roles records

  This is the recommended way of managing ACLs.

Oracle Analytics Server determines user access by sequentially checking the types of records. A user's effective privileges or permissions are deduced using the ACL records, looking for an explicit record for the user (if there's one); and then looking for any records with application roles granted to the user either explicitly or implicitly.

This section contains the following topics:

Rules for Determining a User's Privileges or Permissions

The following tasks describe the sequential checks completed to determine a user's effective privileges and permissions.

> **Note:**
>
> Each earlier step takes precedence over any later step.

> **Note:**
>
> Within an individual step, a privilege access control (ACL) record that's *Denied* always takes precedence over any other grants. Within an individual step, a permission ACL record that has *No Access* always takes precedence over any access grant.
>
> The privilege *Denied* is the same as the permission *No Access*. The term *deny* is used interchangeably for both privileges and permissions.

Task 1 - Check for an explicit record for this user

The following sequence represents the checks completed for a user record.

1. If there's an explicit record for this user, then return that access, *Done*.
2. If there's no explicit record for this user. Go to Step 2.

Task 2 - Check records for this user's application roles

The following sequence represents the checks completed for a user's application roles.

1. Get all the application roles for this user, including both direct, explicit application roles and indirect, implicit application roles.

   For example, if a user is explicitly granted the BI Author application role, then the user also implicitly has the BI Consumer application role too.
2. Check for any ACL record that matches any of the set of application roles.
   - If any records deny access, then return access denied. Done.
   - Else, if any records grant access, return the union of all those access grants. So if one application role had read access, and another application role had write access, then the user has read and write access. Done.
   - Else there are no records for this user's application roles.
3. Else there were no records for this user's application roles. Go to Task 3.

Task 3 - Fall back default behavior

The following sequence represents the checks completed for a specific application role called Authenticated User.

> **Note:**
>
> The Authenticated User application role is deliberately not included in the list of application roles for a user in Task 2, even though that user does technically have this application role.

1. If there's a record for the authenticated user application role, return that record's access. Done.
2. Else there's no record for the special application role. Go to Task 4.

Task 4 - No matching records at all

Return access denied. Done.

Example of Determining a User's Privileges with Application Roles

The diagram shows an example of how privileges are determined with application roles.

At the top of the diagram is a rectangle labelled User1, which specifies that User1 has been explicitly given the application roles Executive and BI Author. Attached beneath the User1 rectangle are two more rectangles - one on the left that represents the Executive role and one on the right that represents the BI Author role.

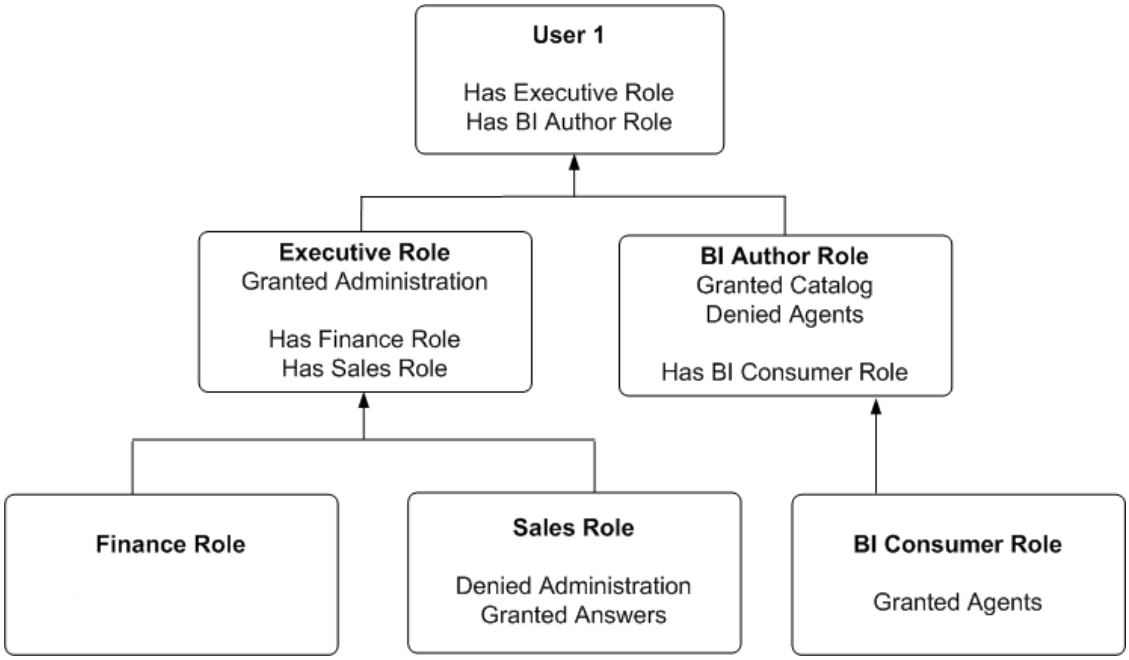
- The Executive role rectangle specifies that Executive is granted the Access to Administration privilege, and that the application roles Finance and Sales have in turn been given to Executive.
- The BI Author role rectangle specifies that BI Author is granted the Catalog privilege, is Denied the Agents privilege, and that the application role BI Consumer has in turn been given to BI Author.

Attached beneath the Executive Role rectangle are two more rectangles - one on the left that represents the Finance role and one on the right that represents the Sales role:

- The Sales Role rectangle specifies that Sales is Denied the Access to Administration privilege and granted the Access to Answers privilege.

And finally, attached beneath the BI Author Role rectangle is a rectangle that represents the BI Consumer role:

- The BI Consumer Role rectangle specifies that BI Consumer is granted the Catalog privilege and is granted the Agents privilege.



In this example:

- User1 explicitly has the Executive role, and thus implicitly has Finance role and also Sales role.
- User1 also explicitly has the BI Author role, and thus also implicitly has BI Consumer role.
- So User1's flattened list of application roles is Executive, BI Author, Finance, Sales and BI Consumer.
- The effective privileges from Executive Role are Denied Administration privilege, granted Scorecard privilege, and granted Answers privilege. The Sales' Denied Administration privilege takes precedence over Executive's granted privilege, as Deny always takes precedence.
- The effective privileges from the BI Author role are granted Catalog privilege, and Denied Agents privilege. The BI Author's Denied Agents privilege takes precedence over BI Consumer's granted, as deny always takes precedence.

The total privileges granted to User1 are as follows:

- Denied Administration privilege, because the privilege is specifically denied for Sales.
- Granted Scorecard privilege.
- Granted Answers privilege.
- Granted Catalog privilege.
- Denied Agents privilege, because the privilege is specifically denied for BI Author.

Example of Determining a User's Permissions with Application Roles

The diagram below shows an example of how permissions are determined with application roles.

At the top of the diagram is a rectangle labelled User1, which specifies that User1 has been explicitly given the application roles Executive and BI Author. Attached beneath the User1

rectangle are two more rectangles - one on the left that represents Executive Role and one on the right that represents BI Author Role.

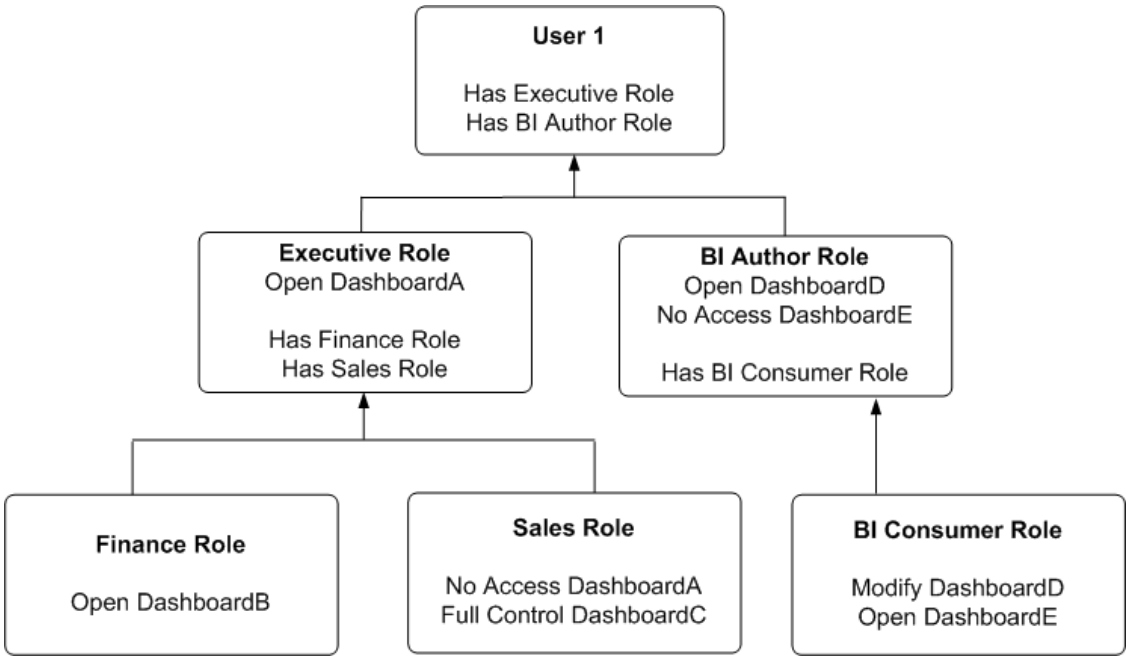
- The Executive Role rectangle specifies that Executive has no access to DashboardA, and that the application roles Finance and Sales have in turn been given to Executive.
- The BI Author Role rectangle specifies that BI Author role has open access to DashboardD, has no access to DashboardE, and that the BI Consumer role has in turn been given to BI Author.

Attached beneath the Executive Role rectangle are two more rectangles, one on the left that represents Finance role and one on the right that represents Sales role:

- The Finance Role rectangle specifies that Finance role has open access to DashboardB.
- The Sales Role rectangle specifies that Sales role has no access to DashboardA and full control of DashboardC.

And finally, attached beneath the BI Author Role rectangle is a rectangle that represents BI Consumer role:

- The BI Consumer Role rectangle specifies that BI Consumer role has modify access to DashboardD and open access to DashboardE.



In this example:

- User1 explicitly has Executive role, and thus implicitly has Finance role and also Sales role.
- User1 also explicitly has BI Author role, and thus also implicitly has BI Consumer role.
- So User1's flattened list of application roles is Executive, BI Author, Finance, Sales and BI Consumer.
- The effective permissions from Executive role are no access to DashboardA, open access to DashboardB, and full control for DashboardC. The Sales role's No Access to DashboardA takes precedence over Executive role's Open, as Deny always takes precedence.

- The effective privileges from BI Author role are Open&Modify access to DashboardD, and No Access to DashboardE. The BI Author role's No Access to DashboardE takes precedence over BI Consumer role's Open, as Deny always takes precedence.

The total permissions and privileges granted to User1 are as follows:

- No Access to DashboardA, because access is specifically denied for Sales role.
- Open Access to DashboardB.
- Full Control for DashboardC.
- Open&Modify access to DashboardD, the union of Role2's and Role5's access.
- No Access to DashboardE, because access is specifically denied for BI Author role.

Providing Shared Dashboards for Users

This section contains the following topics on providing shared dashboards for users:

Understanding the Catalog Structure for Shared Dashboards

Learn about the catalog structure of My Folders and Shared Folders for shared dashboards.

The Oracle BI Presentation Catalog has two main folders:

- My Folders contain the personal storage for individual users. Includes a Subject Area Contents folder where you save objects such as calculated items and groups.
- Shared Folders contain objects and folders that are shared across users. Dashboards that are shared across users are saved in a Dashboards subfolder under a common subfolder under the `/Shared Folders` folder

> **Note:**
>
> If a user is given permission to an analysis in the Oracle BI Presentation Catalog that references a subject area to which the user doesn't have permission, then the BI Server still prevents the user from executing the analysis.

Creating Shared Dashboards

After setting up the Oracle BI Presentation Catalog structure and setting permissions, you can create shared dashboards and content for use by others.

One advantage to creating shared dashboards is that pages that you create in the shared dashboard are available for reuse. Users can create their own dashboards using the pages from your shared dashboards and any new pages that they create. You can add pages and content as described in *Visualizing Data in Oracle Analytics Server*.

If you plan to allow multiple users to modify a shared default dashboard, then consider putting these users into an application role. For example, suppose that you create an application role called Sales and create a default dashboard called SalesHome. Of the 40 users that have been assigned the Sales application role, suppose that there are three who must have the ability to create and modify content for the SalesHome dashboard. Create a SalesAdmin application role, with the same permissions as the primary Sales application role. Add the three users who are allowed to make changes to the SalesHome dashboard and content to this new SalesAdmin application role, and give this role the appropriate permissions in the Oracle BI Presentation Catalog. This allows those three users to create and modify content for the SalesHome dashboard. If a user no longer requires the ability to modify dashboard content, then you can change the user's role assignment to Sales. If an existing Sales role user must have the ability to create dashboard content, then the user's role assignment can be changed to SalesAdmin.

Testing the Dashboards

Before releasing dashboards and content to the user community, perform some tests.

1. Verify that users with appropriate permissions can correctly access it and view the intended content.
2. Verify that users without appropriate permissions can't access the dashboard.
3. Verify that styles, skins, and themes are displayed as expected, and that other visual elements are as expected.
4. Correct any problems you find and test again, repeating this process until you're satisfied with the results.

Releasing Dashboards to the User Community

What to do after testing is complete.

Notify the user community that the dashboard is available, ensuring that you provide the relevant network address.

Controlling Access to Saved Customization Options in Dashboards

This section provides an overview of saved customizations and information about administering saved customizations. It contains the following topics:

Overview of Saved Customizations in Dashboards

Saved customizations allow users to save and view dashboard pages in their current state with their most frequently used or favorite choices for items such as filters, prompts, column sorts, drills in analyses, and section expansion and collapse.

By saving customizations, users need not make these choices manually each time that they access the dashboard page.

Users and groups with the appropriate permissions and dashboard access rights can perform the following activities:

- Save various combinations of choices as saved customizations, for their personal use or use by others.
- Specify a saved customization as the default customization for a dashboard page, for their personal use or use by others.
- Switch between their saved customizations.

You can restrict this behavior in the following ways:

- Users can view only the saved customizations that are assigned to them.
- Users can save customizations for personal use only.
- Users can save customizations for personal use and for use by others.

Administering Saved Customizations

This topic describes the privileges and permissions that are required to administer saved customizations.

In Oracle BI Presentation Services Administration, the following privileges in the Dashboards area, along with permission settings for key dashboard elements, control whether users or groups can save or assign customizations:

- Save Customizations
- Assign Default Customizations

You can set either privilege, one privilege, or both privileges for a user or group, depending on the level of access desired. For example, a user who has neither privilege can view only the saved customization that's assigned as his or her default customization.

Permissions are required so users can administer Oracle BI Presentation Catalog on shared and personal saved customizations.

Permission and Privilege Settings for Creating Saved Customizations

The topic describes user roles and specific permission settings that you can grant to users for creating saved customizations.

User Role	Permission and Privilege Settings
Power users such as IT users perform the following tasks: • Create and edit underlying dashboards. • Save dashboard view preferences as customizations. • Assign customizations to other users as default customizations.	In the Shared section of the catalog, requires Full Control permission to the following folders: • `dashboard_name` • `_selection` • `_defaults` You don't need to assign additional privileges.

User Role	Permission and Privilege Settings
Technical users such as managers perform the following tasks: • Save customizations as customizations for personal use. • Save customizations for use by others. Users can't create or edit underlying dashboards, or assign view customizations to others as default customizations.	In the Shared section of the catalog, requires `View` permission to the following folders: • `dashboard_name` In the Shared section of the catalog, requires `Modify` permission to the following folders: • `_selections` • `_defaults` You don't need to assign additional privileges.
Everyday users that save customizations for personal use only.	In Oracle BI Presentation Services Administration, requires the following privilege to be set: • Save Customizations In the dashboard page, requires that the following option is set: • **Allow Saving Personal Customizations** In the catalog, you don't need to assign additional privileges.
Casual users who must view only their assigned default customization.	In the Shared section of the catalog, the user needs `View` permission to the following folders: • `dashboard_name` • `_selections` • `_defaults` In the catalog, you don't need to assign additional privileges.

Example Usage Scenario for Saved Customization Administration

Depending on the privileges set and the permissions granted, you can achieve various combinations of user and group rights for creating, assigning, and using saved customizations.

For example, suppose a group of power users can't change dashboards in a production environment, but they're allowed to create saved customizations and assign them to other users as default customizations. The following permission settings for the group are required:

- Open access to the dashboard, using the Catalog page.
- Modify access to the `_selections` and `_defaults` subfolders within the dashboard folder in the Oracle BI Presentation Catalog, which you assign using the Dashboard Properties dialog in the Dashboard Builder. After selecting a page in the list in the dialog, click **Specify Who Can Save Shared Customizations** and **Specify Who Can Assign Default Customizations**.