Oracle® Fusion Middleware

Administrator's Guide for Oracle Business Intelligence Publisher

11*g* Release 1 (11.1.1)

E22255-05

December 2014

Explains how to administer Oracle Business Intelligence Publisher, including how to configure security, set up data source connections, define delivery servers, manage the scheduler and delivery server load, and configure run-time properties.

ORACLE

Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Publisher, 11*g* Release 1 (11.1.1)

E22255-05

Primary Author: Leslie Grumbach Studdard

Contributor: The Oracle Business Intelligence Publisher product management, development, and quality assurance teams.

Contents

3 Alternative Security Options

12 Diagnostics and Performance Monitoring

13 Adding Translations for the BI Publisher Catalog and Reports

14 Moving Catalog Objects Between Environments

15 Customizing the BI Publisher User Interface

A Scheduler Configuration Reference

E Converting Oracle Reports

F Enabling Memory Guard Features

Index

Preface

Welcome to Release 11g (11.1.1) of the *Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Publisher*.

Intended Audience

This document is intended for system administrators who are responsible for managing Oracle Business Intelligence Publisher processes, logging, caching, monitoring, data source connections, delivery servers, security, and configuration.

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 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.

Related Documentation and Other Resources

See the Oracle Business Intelligence documentation library for a list of related Oracle Business Intelligence documents.

In addition:

- Go to the Oracle Learning Library for Oracle Business Intelligence-related online training resources.
- Go to the Product Information Center support note (Article ID 1338762.1) on My Oracle Support at https://support.oracle.com.

System Requirements and Certification

Refer to the system requirements and certification documentation for information about hardware and software requirements, platforms, databases, and other information. Both of these documents are available on Oracle Technology Network (OTN).

The system requirements document covers information such as hardware and software requirements, minimum disk space and memory requirements, and required system libraries, packages, or patches:

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html

The certification document covers supported installation types, platforms, operating systems, databases, JDKs, and third-party products:

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

New Features for Administrators

This preface describes changes to Oracle BI Publisher administration features for Oracle Business Intelligence Publisher 11*g* Release 1 (11.1.1).

This preface contains the following topics:

New Features and Changes for Oracle BI Publisher 11*g* Release 1 (11.1.1.9)

New features for Oracle BI Publisher 11g Release 1 (11.1.1.9) include:

Oracle Endeca No Longer Supported as a Data Source

Oracle BI Publisher no longer supports integration with Oracle Endeca as a data source. The Oracle Endeca option has been removed from the Integration section of the Administration page.

Support for Oracle WebCenter Content Server as a Delivery Destination

Oracle WebCenter Content Server is now supported as a delivery destination. For more information, see Section 10.8, "Adding a Content Server."

Configure Memory Guard on the Properties Page

BI Publisher provides a set of features to provide options to protect Oracle BI Publisher server instances from out-of-memory errors by blocking report requests that generate excessive amounts of data. These "memory guard" features consist of a set of properties that you set in a configuration file. The properties enable you to configure conditions and processing points at which data size is inspected to determine whether the system continues to process a report request or terminates processing. When processing terminates due to data size, an error message is returned to the user.

For information about implementing memory guard features, see Appendix F, "Enabling Memory Guard Features."

Support for Maximum Threads Constraint Work Manager on Oracle WebLogic Server

During the processing of large BI Publisher reports Oracle WebLogic Server can use multiple concurrent threads to generate the report. If the threads are not constrained, out of memory errors can occur when Oracle WebLogic Server allots too many threads to report generation. To address this situation, you can now create a Work Manager in Oracle WebLogic Server to limit the number of threads that are allotted to BI Publisher report processing.

For more information, see Section F.4, "Configuring a Maximum Threads Constraint to Avoid Out of Memory Errors."

New Features for Oracle BI Publisher 11*g* Release 1 (11.1.1.7)

New features for Oracle BI Publisher 11g Release 1 (11.1.1.7) include:

- New Features for 11.1.1.7.10
- New Features for 11.1.1.7.0

New Features for 11.1.1.7.10

Note: Information about Oracle BI Publisher 11*g* Release 1 (11.1.1.7.10) is only applicable to customers using Oracle Fusion Applications 11*g* Release 8 (11.1.8).

New features for administrators in Oracle BI Publisher 11*g* Release 1 (11.1.1.7.10) include:

- Support for Named, Shared Web Service as a Data Source
- Support for Named, Shared HTTP as a Data Source
- Guard Against Out of Memory Errors
- Customize the BI Publisher User Interface

Support for Named, Shared Web Service as a Data Source

Previously, for web service-based data sets the connection to the web service had to be configured for each data set. This release introduces centrally administered web service connections. Web service connections set up by the administrator are available to data model developers through a list of values. Therefore data models can share the central connection and no longer require set up for each data set. For more information, see Section 9.7, "Setting Up a Connection to a Web Service."

Support for Named, Shared HTTP as a Data Source

Previously, for HTTP-based data sets the connection to the HTTP server had to be configured for each data set. This release introduces centrally administered HTTP connections. HTTP server connections set up by the administrator are available to data model developers through a list of values. Therefore data models can share the central connection and no longer require set up for each data set. For more information, see Section 9.8, "Setting Up a Connection to a HTTP XML Feed."

Guard Against Out of Memory Errors

BI Publisher provides a set of features to provide options to protect Oracle BI Publisher server instances from out-of-memory errors by blocking report requests that generate excessive amounts of data. These "memory guard" features consist of a set of properties that you set in a configuration file. The properties enable you to configure conditions and processing points at which data size is inspected to determine whether the system continues to process a report request or terminates processing. When processing terminates due to data size, an error message is returned to the user.

With the release of 11.1.1.9.0 this feature is now configurable on the Properties page. See "Configure Memory Guard on the Properties Page."

Customize the BI Publisher User Interface

You can now customize the skins and styles used by the BI Publisher user interface. The default style has also been enhanced. BI Publisher now uses the "Skyros" style, which replaces the blafplus (browser look-and-feel plus) style. You can switch back to the blafplus style or you can customize the style to match your company's branding. For more information, see Chapter 15, "Customizing the BI Publisher User Interface."

New Features for 11.1.1.7.0

New features for administrators in Oracle BI Publisher 11*g* Release 1 (11.1.1.7) include:

- Integration with Oracle Endeca
- PDF to PCL Converter
- Support for Private Data Sources

Integration with Oracle Endeca

BI Publisher now supports integration with Oracle Endeca as a data source for reports. As of 11.1.1.9.0 this feature has been removed.

PDF to PCL Converter

BI Publisher now provides a PDF to PCL converter to enable the printing of PDF output to a PCL printer. You can embed PCL commands into RTF templates to invoke the PCL commands at a specific position on the PCL page, for example to use a font installed on the printer for routing and account numbers on a check For more information about the set up, see Section 10.2, "Adding a Printer."

Support for Private Data Sources

Data model developers can now create and manage private JDBC and OLAP data source connections for use in SQL or MDX data sets without having to depend on Administrators. However, Administrator users can still view, modify, and delete private data source connections, and extend access to other users. For more information, see Section 9.1.1, "About Private Data Source Connections."

New Features for Oracle BI Publisher 11*g* Release 1 (11.1.1.6)

New features for administrators in Oracle BI Publisher 11*g* Release 1 (11.1.1.6) include:

- Reporting Organization Support Enabled with E-Business Suite Security Integration
- Simplified Configuration Steps for Enabling Performance Monitoring and User Auditing

Reporting Organization Support Enabled with E-Business Suite Security Integration

In 11.1.1.5 support for E-Business Suite security was enhanced to support data-level security based on responsibility. In 11.1.1.6, BI Publisher also recognizes reporting organization. After setting up integration with E-Business Suite security, when a user logs in he can choose both responsibility and reporting organization from the **My Account** dialog. For more information, see Section 5.4, "Integrating with Oracle E-Business Suite."

Simplified Configuration Steps for Enabling Performance Monitoring and User Auditing

In previous releases enabling performance monitoring and user auditing required the manual editing of configuration files. In 11.1.1.6, this process has been simplified and now you simply select a check box on the Administration **Server Configuration** page. For more information, see Section 8.6, "Enabling Monitor and Audit" and Section 12.6, "Enabling Monitoring and Auditing."

New Features for Oracle BI Publisher 11*g* Release 1 (11.1.1.5)

New features in Oracle BI Publisher 11*g* Release 1 (11.1.1.5) include:

- Enhanced Integration with Oracle E-Business Suite
- Auditing and Monitoring
- Reports Development Life Cycle Management

Enhanced Integration with Oracle E-Business Suite

BI Publisher now supports Oracle E-Business Suite's data level security based on user responsibility and enables you to switch responsibilities in-session. You can use BI Publisher 11.1.1.5 to view, manage, and deliver reports against E-Business Suite data and join and aggregate with data from other data sources. For more information, see Section 5.4, "Integrating with Oracle E-Business Suite."

Auditing and Monitoring

Auditing is not just about compliance, it's a way to improve customer service by understanding what your users like to do, when and how they access and view reports. BI Publisher 11.1.1.3 included a framework to help Administrators collect data required for auditing. BI Publisher 11.1.1.5 enhances the audit and performance information captured and introduces a method to store the data into a database. Now you can use BI Publisher to visualize, analyze and report on your auditing and performance information. For more information, see Section 12.5, "About Performance Monitoring and User Auditing."

Reports Development Life Cycle Management

Developing reports often involves multiple phases and processes before going live. Reports must be moved to a testing environment after the development and then to the production environment. Now you can use the BI Publisher Catalog Utility to move your reports either one at a time or in batch from one environment to another all without shutting down servers. For more information, see Chapter 14, "Moving Catalog Objects Between Environments."

New Features for Oracle BI Publisher 11*g* Release 1 (11.1.1.3)

New features in Oracle BI Publisher 11*g* Release 1 (11.1.1.3) include:

- User Interface Enhancements
- Scheduler Enhancements
- Shared Catalog with Oracle Business Intelligence Enterprise Edition
- Integration with Oracle Fusion Middleware Security
- Enhanced Catalog Security
- User Auditing and Performance Monitoring

User Interface Enhancements

The user interface has undergone major improvements in several areas, including a new Home page and redesigned editors and panes. These improvements are intended to make working with Oracle BI Publisher easier and more consistent. For information about working in the new interface, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Publisher*.

Scheduler Enhancements

The updated architecture of BI Publisher Scheduler in 11*g* uses Java Messaging Service (JMS) queues to provide a highly scalable, performing and robust architecture for report scheduling and delivery. This architecture enables you to add multiple BI Publisher servers to a cluster and then dedicate each server to a particular function: report generation, document generation, or specific delivery channels. The new interface also enables you to configure the number of threads per processor. To facilitate diagnosing issues with the scheduler, a Scheduler Diagnostics page is now available to provide run-time information regarding JMS configuration, JMS queues, Cluster instance status, Scheduler Database status, Toplink status, and Scheduler (Quartz) status. For more information, see Chapter 7, "Configuring the Scheduler."

Shared Catalog with Oracle Business Intelligence Enterprise Edition

For installations of BI Publisher with the Oracle BI Enterprise Edition, BI Publisher now shares the same catalog with Oracle BI Presentation services. The catalog integration is configured during installation. For information about the improved catalog, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Publisher*.

Integration with Oracle Fusion Middleware Security

The Oracle Fusion Middleware security model is built upon the Oracle Fusion Middleware platform, which incorporates the Java security model. The Java model is a role-based, declarative model that employs container-managed security where resources are protected by roles that are assigned to users. When implemented with the Oracle BI Enterprise Edition, this provides a much tighter and more rational security integration across products. For more information about using the Oracle Fusion Middleware security model for standalone implementation of BI Publisher, see Chapter 2, "Configuring Oracle Fusion Middleware Security Model." For more information about security when BI Publisher is integrated with Oracle BI Enterprise Edition, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*

Enhanced Catalog Security

A user's role must now be granted explicit permissions on catalog folders to enable functional privileges. This provides much more control over which users can perform actions on particular objects in the catalog. Permissions are granted by an administrator within the catalog interface. For more information, see Section 3.5, "About Catalog Permissions."

User Auditing and Performance Monitoring

Performance monitoring enables you to monitor the performance of queries, reports and document generation and to analyze the provided details. BI Publisher 11*g* collects performance statistics through JMX Management Beans or Mbeans. Each MBean reveals attributes, operations, and relevant statistics gathered by the Oracle Dynamic Monitoring Service (DMS). For more information, see Section 12.5, "About Performance Monitoring and User Auditing."

1

Introduction to Oracle BI Publisher Administration

This chapter describes tasks required to administer BI Publisher.

It includes the following topics:

1.1 Introduction

Oracle BI Publisher is an enterprise reporting solution for authoring, managing, and delivering all your highly formatted documents, such as operational reports, electronic funds transfer documents, government PDF forms, shipping labels, checks, sales and marketing letters, and much more.

Administering BI Publisher requires setting up and maintaining the following system components:

- BI Publisher security
- Data source connections
- Report delivery destinations
- BI Publisher Scheduler configurations
- Runtime configuration settings
- Server configuration settings

See the guides that are outlined in Table 1–1 for more information about using the product for other business roles.

Table 1–1 Other Guides to Consult

Role	Sample Tasks	Guide
Data Model developer	Fetching and structuring the data to use in reports	*Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*
Application developer or integrator	Integrating BI Publisher into existing applications using the application programming interfaces	*Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Publisher*
Report consumer	Viewing reports Scheduling report jobs Managing report jobs	*Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*
Report designer	Creating report definitions Designing layouts	*Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*

1.2 Configurations Performed by the BI Platform Installer

After installation is complete, the following pieces are configured:

- The security model is configured to use Oracle Fusion Middleware Security
- The scheduler is configured to use Oracle WebLogic JMS. The schema tables are installed and configured in the database.
- The BI Publisher catalog and repository are configured to ${xdo.server.config.dir}/repository

1.3 Flow of Tasks for First Time Setup of BI Publisher

If you are setting up BI Publisher for the first time, then consult the following table for the recommended flow of tasks to get the system up and running.

Table 1–2 Recommended Flow of Tasks

Task	Where to Get Information
Define a Local Superuser Set up this Superuser to ensure access to all administrative functions in case of problems with the current security setup.	Section 4.1, "Enabling a Local Superuser"
Set up the chosen security model and test	Chapter 2, "Configuring Oracle Fusion Middleware Security Model" Chapter 3, "Alternative Security Options" Chapter 5, "Integrating with Other Oracle Security Models"
Set up the data sources and test	Chapter 9, "Setting Up Data Sources"
Set up the delivery servers and test	Chapter 10, "Setting Up Delivery Destinations"
Configure server properties	Chapter 8, "Configuring Server Properties"
Configure system runtime properties	Chapter 11, "Defining Run-Time Configurations"

1.4 Starting and Stopping BI Publisher

Use the Oracle WebLogic Server Administration Console to centrally manage Oracle Business Intelligence Publisher.

For detailed information about Oracle WebLogic Server, see:

- *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*

Display Oracle WebLogic Server Administration Console, using one of the following methods:

- Using the Start menu in Windows
- Clicking a link on the Overview page in Fusion Middleware Control
- Entering a URL into a Web browser window

The Oracle WebLogic Server Administration Console is available only if the Administration Server for WebLogic Server is running.

To display Oracle WebLogic Server Administration Console:

1. If the Administration Server for WebLogic Server is not running, start it.
2. Display the Oracle WebLogic Server Administration Console using one of the following methods:

   Using the Windows Start menu:

   a. From the **Start** menu, select **All Programs**, Oracle WebLogic**, User Projects**, **bifoundation_domain**, and **Admin Server Console**.

      The Oracle WebLogic Server Administration Console login page is displayed.

   Clicking a link on the Overview page in Fusion Middleware Control:

   a. Display Oracle Fusion Middleware Control.

   b. Expand the WebLogic Domain node and select the bifoundation_domain.

   c. Click the Oracle WebLogic Server Administration Console link in the Summary region.

      The Oracle WebLogic Server Administration Console login page is displayed.

   Using a URL in a Web browser window:

   a. Enter the following URL into the browser:

      http://<host>:<port>/console/

      For example, http://mycomputer:7001/console/

      where host is the DNS name or IP address of the Administration Server and port is the listen port on which the Administration Server is listening for requests (port 7001 by default).

      If you have configured a domain-wide Administration port, then use that port number. If you configured the Administration Server to use Secure Socket Layer (SSL), then you must add the letter 's' after http as follows:

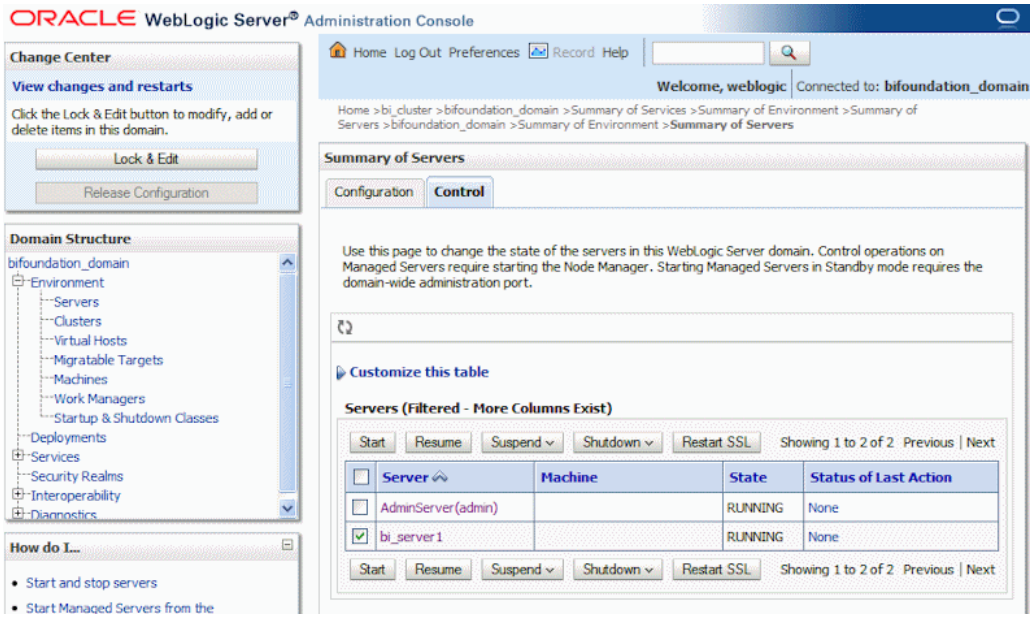
      https://<host>:7001/console/

1.4.1 Using Oracle WebLogic Server Administration Console

To use the Oracle WebLogic Server Administration Console to start and stop BI Publisher:

1. Start the Oracle WebLogic Server Administration Console.
2. Under the Domain Structure, expand **Environment**.
3. Click **Servers** to display the **Summary of Servers** table.
4. Click **Control**. Select the server and then click the appropriate action, as shown in Figure 1–1.

Figure 1–1 Administration Console

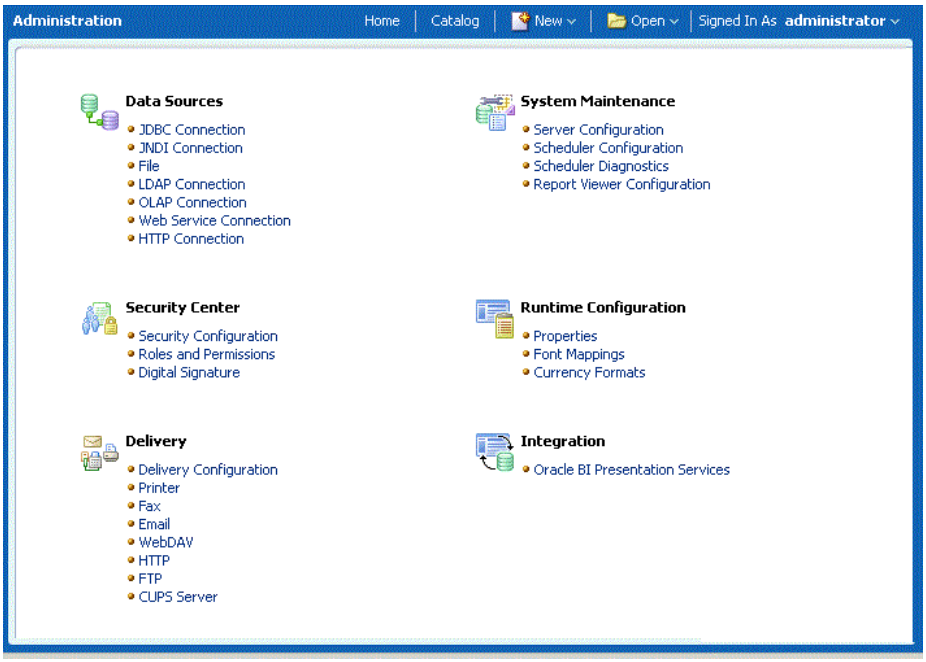


1.5 About the Administration Page

Many of the tasks described in the Administration section of this guide are performed from the BI Publisher Administration page. You must be granted Administrator privileges to access the Administration page.

The Administration page is accessed from the Administration link in the global header. Figure 1–2 shows the Administration page:

Figure 1–2 Administration Page



1.6 About Integration with Oracle Business Intelligence Enterprise Edition

If you installed Oracle BI Publisher with the Oracle Business Intelligence Enterprise Edition, then you must perform the Administration tasks in the BI Publisher Administration page, as described in the following table. Navigate to the BI Publisher Administration page as follows:

In the global header, click **Administration**, on the Administration page, click **Manage BI Publisher**.

Table 1–3 BI Publisher Administration Tasks

Task	Where to Get Information
Set up data source connections for reporting	Chapter 9, "Setting Up Data Sources"
Grant access to data sources for user roles defined in Oracle Business Intelligence	Section 3.8.4, "Granting Data Access"
Configure the connections to delivery servers (for example, printers, e-mail servers, FTP servers, and so on)	Chapter 10, "Setting Up Delivery Destinations"
Configure the scheduler processors	Chapter 7, "Configuring the Scheduler"
Configure system runtime properties such as PDF security properties, properties specific to each output format, template type properties, font mappings, and currency formats.	Section 11.1, "Setting Run-Time Properties"
Configure server properties such as caching specifications, database failover properties, and database fetch size.	Chapter 8, "Configuring Server Properties"

1.7 About the Security Model Options

BI Publisher offers the following security options:

- Oracle Fusion Middleware Security

  After installation, BI Publisher is configured to use Oracle Fusion Middleware Security. For more information, see Chapter 2, "Configuring Oracle Fusion Middleware Security Model." If you prefer to use another security model, then choose from the alternative options.

- BI Publisher Security

  Use BI Publisher's Users and Roles paradigm to control access to reports and data sources. For more information see Chapter 3, "Alternative Security Options."

- Integration with an LDAP server

  Set up the BI Publisher roles in your LDAP server then configure BI Publisher to integrate with it. For more information see Chapter 3, "Alternative Security Options."

- Oracle E-Business Suite

  Upload a DBC file to recognize your Oracle E-Business Suite users. For more information see Chapter 5, "Integrating with Other Oracle Security Models."

- Oracle BI Server

  You can still leverage the 10*g* legacy BI Server authentication method if you choose not to upgrade to Oracle Fusion Middleware Security. For more information see Chapter 5, "Integrating with Other Oracle Security Models."

- Oracle Database

  Set up the BI Publisher roles in your Oracle Database and then configure BI Publisher to integrate with it. For more information see Chapter 5, "Integrating with Other Oracle Security Models."

- Oracle Siebel CRM Security Model

  For more information see Chapter 5, "Integrating with Other Oracle Security Models."

1.8 About the Data Source Connections

BI Publisher reports rely on XML data. BI Publisher supports retrieving data from a variety of data sources.

The following data sources must be first set up in BI Publisher through the Administration page:

- Database connections

  BI Publisher supports direct JDBC connections and connections through a JNDI pool (recommended)

- LDAP connections
- OLAP connections
- File directory connections - you can use existing XML files, Microsoft Excel files, or CSV files stored in a directory that BI Publisher can access
- Web Service connections

- HTTP XML connections

For more information on setting up these data source connections, see Chapter 9, "Setting Up Data Sources."

If you have integrated your system with Oracle Business Intelligence you can also take advantage of the following data sources:

- Oracle BI Analysis
- Oracle BI Server subject area

You can also upload some file types stored locally.

1.9 About Report Delivery Destinations

The BI Publisher delivery manager supports the following delivery channels:

- Printer
- Fax
- E-mail
- HTTP notification
- FTP
- Web Folder (or WebDAV)

For more information on setting up the delivery options, see Chapter 10, "Setting Up Delivery Destinations."

1.10 About Setting Runtime Configuration Properties

Use the Runtime Configuration page to enable configuration settings for your system. The properties include settings that

- Control the processing for different output types
- Enable digital signature
- Tune for scalability and performance
- Define font mappings

For more information on setting configuration properties and font mappings, see Section 11.1, "Setting Run-Time Properties."

1.11 About the Server Configuration Settings

BI Publisher administration also includes a set of system maintenance settings and tasks. These are:

- Configuring the catalog
- Setting caching properties
- Setting retry properties for failover
- Enabling Auditing and Monitoring

For more information on these tasks and settings, see Chapter 8, "Configuring Server Properties."

2

Configuring Oracle Fusion Middleware Security Model

This chapter describes how to configure Oracle Fusion Middleware security model for BI Publisher.

It includes the following topics:

2.1 Understanding the Security Model

The Oracle Fusion Middleware security model is built upon the Oracle Fusion Middleware platform, which incorporates the Java security model. The Java model is a role-based, declarative model that employs container-managed security where resources are protected by roles that are assigned to users. However, extensive knowledge of the Java-based architecture is unnecessary when using the Oracle Fusion Middleware Security model. When using this security model, BI Publisher can furnish uniform security and identity management across the enterprise.

After installation BI Publisher is automatically installed into an Oracle WebLogic Server domain, which is a logically related group of WebLogic Server resources that are managed as a unit. After a Simple installation type the WebLogic Server domain that is created is named **bifoundation_domain**. This name might vary depending upon the installation type performed. One instance of WebLogic Server in each domain is configured as an Administration Server. The Administration Server provides a central point for managing a WebLogic Server domain. The Administration Server hosts the Administration Console, which is a Web application accessible from any supported Web browser with network access to the Administration Server. BI Publisher is part of the active security realm configured for the Oracle WebLogic Server domain into which it is installed.

For more information about the Oracle Fusion Middleware platform and the common security framework, see *Oracle Fusion Middleware Application Security Guide*. For more

information about managing the Oracle WebLogic Server domain and security realm, see *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server* and *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

2.2 Key Security Elements

The Oracle Fusion Middleware security model depends upon the following key elements to provide uniform security and identity management across the enterprise:

- Application policy

  BI Publisher permissions are granted to members of its application roles. In the default security configuration, each application role conveys a predefined set of permissions. Permission grants are defined and managed in an **application policy**. After an application role is associated with an application policy, that role becomes a **Grantee** of the policy. An application policy is specific to a particular application.

- Application role

  After permission grants are defined in an application policy, an application role can be mapped to that policy, and the application role then becomes the mechanism to convey the permissions. In this manner an **application role** becomes the container that grants permissions to its members. The permissions become associated with the application role through the relationship between *policy* and *role*. After groups are mapped to an application role, the corresponding permissions are granted to all members equally. Membership is defined in the application role definition. Application roles are assigned in accordance with specific conditions and are granted dynamically based on the conditions present at the time authentication occurs. More than one user or group can be members of the same application role.

- Authentication provider

  An **authentication provider** is used to access user and group information and is responsible for authenticating users. The default authentication provider that BI Publisher uses during a Simple or Enterprise installation is named DefaultAuthenticator. This is the same default authenticator used by a basic Oracle WebLogic Server installation. An Oracle WebLogic Server authentication provider enables you to manage users and groups in one place.

  An **identity store** contains user name, password, and group membership information. An authentication provider accesses the data in the identity store and authenticates against it. 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. The BI Publisher default authentication provider authenticates against Oracle WebLogic Server embedded directory server.

- 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.

  Groups are organized collections of users that have something in common. Users should be organized into groups with similar access needs to facilitate efficient security management.

- Security realm

During installation an Oracle WebLogic Server domain is created and BI Publisher is installed into that domain. BI Publisher security is managed within the **security realm** for this Oracle WebLogic Server domain. 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. BI Publisher authentication is performed by the authentication provider configured for the default security realm for the WebLogic Server domain in which it is installed. Oracle WebLogic Server Administration Console is the administration tool used for managing an Oracle WebLogic Server domain.

2.3 Permission Grants and Inheritance

BI Publisher provides application-specific permissions for accessing different features. BI Publisher permissions are typically granted by becoming a member in an application role. Permissions can be granted two ways: through membership in an application role (direct) and through group and role hierarchies (inheritance). Application role membership can be inherited by nature of the application role hierarchy. In the default security configuration, each application role is preconfigured to grant a predefined set of permissions. Groups are mapped to an application role. The mapping of a group to a role conveys the role's permissions to all members of the group. In short, permissions are granted in BI Publisher by establishing the following relationships:

- A group defines a set of users having similar system access requirements. Users are added as members to one or more groups according to the level of access required.
- Application roles are defined to represent the role a user typically performs when using BI Publisher. The default security configuration provides the following preconfigured application roles: BIAdministrator (an administrator), BIAuthor (an author of content), and BIConsumer (a consumer of content).
- The groups of users are mapped to one or more application roles that match the type of access required by the population.
- Application policies are created and BI Publisher permissions are mapped that grant a set of access rights corresponding to role type.
- An application role is mapped to the application policy that grants the set of permissions required by the role type (an administrator, an author, a consumer).
- Group membership can be inherited by nature of the group hierarchy. Application roles mapped to inherited groups are also inherited, and those permissions are likewise conveyed to the members.

How a user's permissions are determined by the system is as follows:

1. A user enters credentials into a Web browser at login. The user credentials are authenticated by the authentication provider against data contained in the identity store.
2. After successful authentication, a Java subject and principal combination is issued, which is populated with the user name and the user's groups.
3. A list of the user's groups is generated and checked against the application roles. A list is created of the application roles that are mapped to each of the user's groups.

4. A user's permission grants are determined from knowing which application roles the user is a member of. The list of groups is generated only to determine what roles a user has, and is not used for any other purpose.

A user can also be granted permissions if they inherit other application roles. Members of application roles can include other groups and application roles. The result is a hierarchical role structure where permissions can be *inherited* in addition to being *explicitly granted*. This hierarchy provides that a group is granted the permissions of the application role for which it is a member, and the permissions granted by all roles *descended* from that role.

For example, the default security configuration includes several predefined groups and application roles. The default BIAdministrator application role includes the BIAdministrators group, the BIAuthor application role includes the BIAuthors group, and the BIConsumer application role includes the BIConsumers group. The default BIAdministrator application role is a member the BIAuthor application role, and the BIAuthor application role is a member of the BIConsumer application role. The members of these application roles inherit permissions as follows. Members of the BIAdministrators group are granted all the permissions of the BIAdministrator role, the BIAuthor role, and the BIConsumer role. By nature of this role hierarchy, the user who is a member of a particular group is granted permissions both explicitly and through inheritance. For more information about the default application roles and groups, see Section 2.4.2, "Default Application Roles and Permissions."

Note: By themselves, groups and group hierarchies do not enable any privilege to access resources controlled by an application. Privileges are conveyed by the permission grants defined in an application policy. A user, group, or application role becomes a Grantee of the application policy. The application policy Grantee conveys the permissions and this is done by direct association (user) or by becoming a member of the Grantee (group or application role).

Figure 2–1 shows these relationships between the default groups and application roles.

Figure 2–1 Relationships Between Default Groups and Application Roles

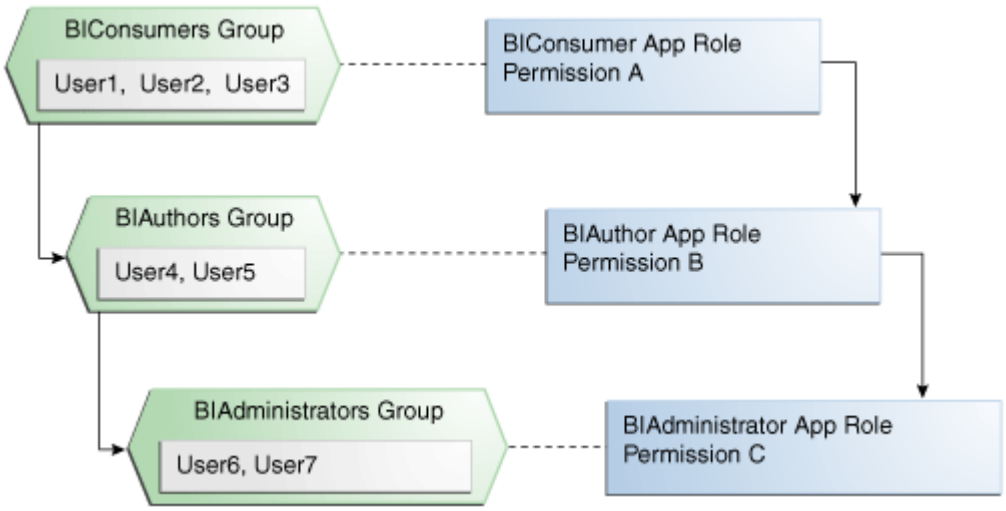


Table 2–1 summarizes how permissions are granted explicitly or are inherited in the previous example and figure.

Table 2–1 Permissions Granted by the Role Hierarchy Example

User Name	Group Membership: Explicit/Inherited	Application Role Membership: Explicit/Inherited	Permission Grants: Explicit/Inherited
User1, User2, User3	BIConsumers: Explicit	BIConsumer: Explicit	Permission A: Explicit
User4, User5	BIAuthors: Explicit BIConsumers: Inherited	BIAuthor: Explicit BIConsumer: Inherited	Permission B: Explicit Permission A: Inherited
User6, User7	BIAdministrators: Explicit BIAuthors: Inherited BIConsumers: Inherited	BIAdministrator: Explicit BIAuthor: Inherited BIConsumer: Inherited	Permission C: Explicit Permission B: Inherited Permission A: Inherited

2.4 Default Security Configuration

Access control of system resources is achieved by requiring users to authenticate at login and by restricting users to only those resources for which they are authorized. A default security configuration is available for immediate use after BI Publisher is installed and is configured to use the Oracle Fusion Middleware security model. BI Publisher is installed into the Oracle WebLogic Server domain and uses its security realm. The default configuration includes three predefined security stores available for managing user identities, credentials, and BI Publisher-specific permission grants. Users can be added to predefined groups that are mapped to preconfigured application roles. Each application role is preconfigured to grant specific BI Publisher permissions.

The BI Publisher default security stores are configured as described in Table 2–2 during installation.

Table 2–2 BI Publisher Default Security Stores

Store Name	Purpose	Default Provider	Options
Identity store	▪ Used to control authentication. ▪ Stores the users and groups, and the users group for Oracle WebLogic Server embedded directory server.	▪ Oracle WebLogic Server embedded directory server. ▪ Managed with Oracle WebLogic Server Administration Console.	BI Publisher can be configured to use alternative authentication providers. For a complete list, see System Requirements and Certification.

Table 2–2 (Cont.) BI Publisher Default Security Stores

Store Name	Purpose	Default Provider	Options
Policy store	▪ Used to control authorization. ▪ Stores the application role definitions and the mapping definitions between groups and application roles.	▪ system.jazn-data.xml file. Default installation location is MW_HOME/user_projects/domain/*your_domain*/config/fmwconfig ▪ Managed with Oracle Enterprise Manager Fusion Middleware Control.	BI Publisher can be configured to use Oracle Internet Directory as the policy store provider.
Credential store	Stores the passwords and other security-related credentials either supplied or system-generated.	▪ cwallet.sso file. ▪ Managed using Fusion Middleware Control.	BI Publisher can be configured to use Oracle Internet Directory as the credential store provider.

2.4.1 Default Users and Groups

Table 2–3 lists the default user names and passwords added to the BI Publisher identity store provider after installation. These defaults can be changed to different values and additional users can be added to the identity store by an administrative user using Oracle WebLogic Server Administration Console.

Table 2–3 Default Names and Passwords

Default User Name and Password	Purpose	Description
Name: *administrator user* **Password:** *user supplied*	Is the administrative user.	This user name is entered by the person performing the installation, it can be any desired name, and does not need to be named Administrator. The password entered during installation can be changed later using the administration interface for the identity store provider. This single administrative user is shared by BI Publisher and Oracle WebLogic Server. This user is automatically made a member of the Oracle WebLogic Server default Administrators group after installation. This enables this user to perform all Oracle WebLogic Server administration tasks, including the ability to manage Oracle WebLogic Server's embedded directory server.

Table 2–3 (Cont.) Default Names and Passwords

Default User Name and Password	Purpose	Description
Name: *BISystemUser* **Password**: *system generated*	A fixed user created during installation for trusted communication between components when using Oracle BI Analysis as a data source for your BI Publisher Data Model. If you are integrating BI Publisher with Oracle Business Intelligence Enterprise Edition, the recommendation is to use this default user name for trusted communication with Oracle BI Presentation Services. This is the default configuration automatically configured during installation.	**Important:** This is a highly privileged user whose credentials should be protected from non-administrative users. Using a separate trusted system account for secure inter-component communication enables you to change the password for the system administrator account without affecting communication between components. The name of this user can be changed or a different user can be created for the purpose of inter-component communication.

Table 2–4 lists the default group names and group members added to the identity store provider during installation. These defaults can be changed to different values and additional group names can be added by an administrative user using Oracle WebLogic Server Administration Console.

Table 2–4 Default Group Names and Members

Default Group Name and Members	Purpose	Description
Name: BIAdministrators **Members**: Any *administrator user*	Contains the BI Publisher administrative users.	Members of the BIAdministrators group are granted administrative permissions because this group is mapped to the BIAdministrator application role at installation. All users requiring administrative permissions should be added to the BIAdministrators group when using the default security configuration.
Name: BIAuthors **Members**: BIAdministrators group	Contains the BI Publisher authors.	Members of the BIAuthors group have the permissions necessary to create content for other users to use, or to consume.
Name: BIConsumers **Members**: BIAuthors group and Oracle WebLogic Server LDAP server users group.	Contains the BI Publisher consumers.	Members of the BIConsumers group have the permissions necessary to use, or consume, content created by other users. The BIConsumers group represents all users that have been authenticated by BI Publisher. By default, every authenticated user is automatically added to this group. Oracle WebLogic Server LDAP server users group members have the permissions necessary to log in to and use Oracle WebLogic Server Administration Console.

2.4.2 Default Application Roles and Permissions

Table 2–5 lists the BI Publisher permissions and the application role that grants these permissions. This mapping exists in the default policy store.

Table 2–5 lists the permissions explicitly granted by membership in the corresponding default application role. Permissions can also be inherited from group and application role hierarchies. For more information about permission inheritance, see Section 2.3, "Permission Grants and Inheritance."

Table 2–5 BI Publisher Permissions and Application Roles

BI Publisher Permission	Description	Default Application Role Granting Permission Explicitly
oracle.bi.publisher.administerServer	Enables the Administration link to access the Administration page and grants permission to set any of the system settings. **Important:** See Section 2.4.2.1, "Granting the BIAdministrator Role Catalog Permissions" for additional steps required to grant the BIAdministrator permissions on Shared Folders.	BIAdministrator
oracle.bi.publisher.developDataModel	Grants permission to create or edit data models.	BIAuthor
oracle.bi.publisher.developReport	Grants permission to create or edit reports, style templates, and sub templates. This permission also enables connection to the BI Publisher server from the Template Builder.	BIAuthor
oracle.bi.publisher.runReportOnline	Grants permission to open (execute) reports and view the generated document in the report viewer.	BIConsumer
oracle.bi.publisher.scheduleReport	Grants permission to create or edit jobs and also to manage and browse jobs.	BIConsumer
oracle.bi.publisher.accessReportOutput	Grants permission to browse and manage job history and output.	BIConsumer
BIConsumer permissions granted implicitly	The authenticated role is a member of the BIConsumer role by default and, as such, all authenticated role members are granted the permissions of the BIConsumer role implicitly.	Authenticated Role

The **authenticated role** is a special application role provided by the Oracle Fusion Middleware security model and is made available to any application deploying this security model. BI Publisher uses the authenticated application role to grant permissions implicitly derived by the role and group hierarchy of which the authenticated role is a member. The authenticated role is a member of the BIConsumer role by default and, as such, all authenticated role members are granted the permissions of the BIConsumer role implicitly. By default, every authenticated user is automatically added to the BIConsumers group. The authenticated role is not stored in the **obi** application stripe and is not searchable in the BI Publisher policy store.

However, the authenticated role is displayed in the administrative interface for the policy store, is available in application role lists, and can be added as a member of another application role. You can map the authenticated role to another user, group, or application role, but you cannot remove the authenticated role itself. Removal of the authenticated role would result in the inability to log in to the system and this right would need to be granted explicitly.

For more information about the Oracle Fusion Middleware security model and the authenticated role, see *Oracle Fusion Middleware Application Security Guide*.

2.4.2.1 Granting the BIAdministrator Role Catalog Permissions

The BIAdministrator role is granted only Read permissions on the catalog by default. This means that before a BIAdministrator can manage Shared Folders the BIAdministrator role must be granted Write and Delete permissions on the Shared Folders node. See Section 3.8.3, "Granting Catalog Permissions" for a detailed description of granting permissions in the catalog.

2.5 Managing Authentication

Authentication is the process of verifying identity by confirming the user is who he claims to be. Oracle WebLogic Server embedded directory server is the authentication provider for the default security configuration. Users, groups, and passwords are managed using Oracle WebLogic Server Administration Console. It is fine to use the default authentication provider for a development or test environment. In a production environment, best practice is to use a full featured authentication provider.

> **Note:** Refer to the system requirements and certification documentation for information about hardware and software requirements, platforms, databases, and other information. These documents are available on Oracle Technology Network (OTN).

During installation an Oracle WebLogic Server domain is created. BI Publisher is installed into that domain and uses the Oracle WebLogic Server security realm. The security realm can have multiple authentication providers configured but only one provider can be active at a time. The order of providers in the list determines priority. The effect of having multiple authentication providers defined in a security realm is not cumulative; rather, the first provider in list is the source for all user and password data needed during authentication. This enables you to switch between authentication providers as needed. For example, if you have separate LDAP servers for your development and production environments, you can change which directory server is used for authentication by re-ordering them in the Administration Console. For information about how to configure a different authentication provider, see Section 2.8.1, "Configuring a New Authentication Provider."

Detailed information about managing an authentication provider in Oracle WebLogic Server is available in its online help. For more information, log in to Oracle WebLogic Server Administration Console and launch *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

2.5.1 Accessing Oracle WebLogic Server Administration Console

Oracle WebLogic Server is automatically installed and serves as the default administration server. The Administration Console is browser-based and is used to manage the embedded directory server that is configured as the default authenticator.

It is launched 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 number of the administration server. By default, the port number is 7001.

To launch the Oracle WebLogic Server Administration Console:

1. Log in to Oracle WebLogic Server by entering its URL into a Web browser.

   For example, http://*hostname:7001*/console. The Administration Console login page displays, as shown in Figure 2–2.

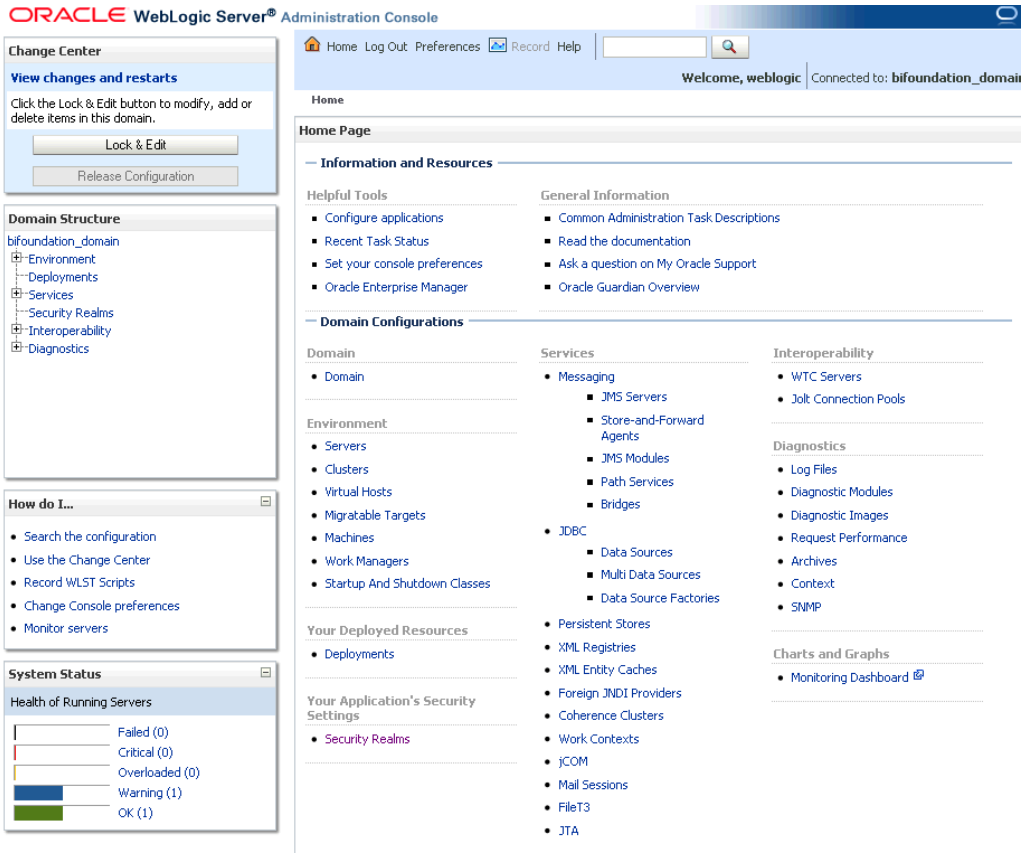
Figure 2–2 Administration Console Login Page

2. Log in using the BI Publisher administrative user and password and click **Login**.

   The password is the one you supplied during the installation of BI Publisher. If these values have been changed, then use the current administrative user name and password combination.

   The Administration Console displays, as shown in Figure 2–3.

Figure 2–3 Administration Console



2.5.2 Managing Users and Groups Using the Default Authentication Provider

Managing a group is more efficient than managing a large number of users individually. Best practice is to first organize all BI Publisher users into groups that have similar system access requirements. These groups can then be mapped to application roles that provide the correct level of access. If system access requirements change, then you need only modify the permissions granted by the application roles, or create a new application roles with appropriate permissions. Once your groups are established, continue to add or remove users directly in the identity store using its administration interface as you normally would.

To create a user in the default directory server:

1. If needed, launch Oracle WebLogic Server Administration Console.

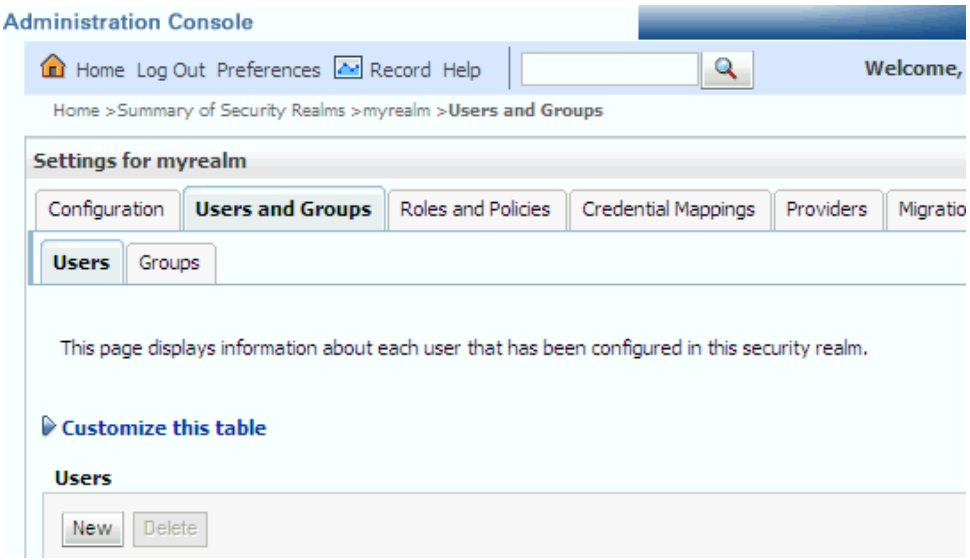
   For more information, see Section 2.5.1, "Accessing Oracle WebLogic Server Administration Console."

2. Log in as an administrative user.

3. In the Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.

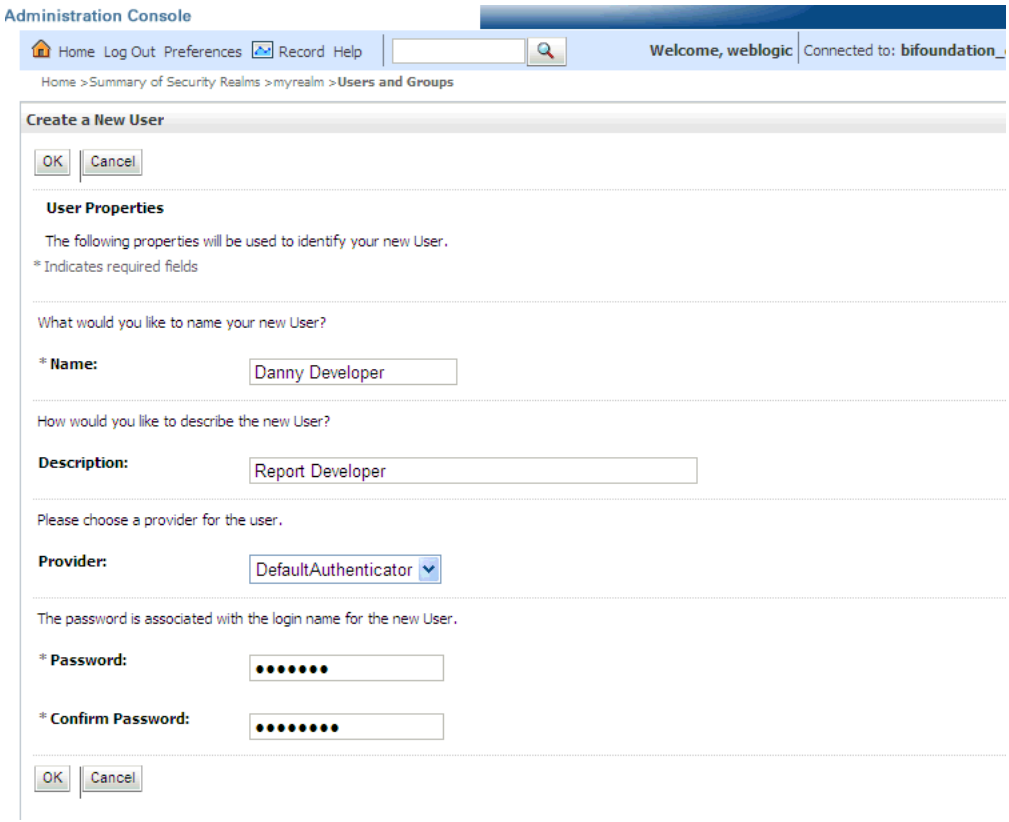
4. Select **Users and Groups** tab (shown in Figure 2–4), then **Users**. Click **New**.

Figure 2–4 Users and Groups tab



5. In the **Create a New User** page (shown in Figure 2–5) provide the following information:
   - **Name**: Enter the name of the user. See online help for a list of invalid characters.
   - (Optional) **Description**: Enter a description.
   - **Provider**: Select the authentication provider from the list that corresponds to where the user information is contained. DefaultAuthenticator is the name for the default authentication provider.
   - **Password**: Enter a password for the user that is at least 8 characters long.
   - **Confirm Password**: Re-enter the user password.

Figure 2–5 Create a New User Page



6. Click **OK**.

   The user name is added to the User table.

To create a group in the default directory server:

1. If needed, launch Oracle WebLogic Server Administration Console.

   For more information, see Section 2.5.1, "Accessing Oracle WebLogic Server Administration Console."

2. Log in as an administrative user.
3. In the Administration Console, select **Security Realm** from the left pane and click the realm you are configuring. For example, **myrealm**.
4. Select **Users and Groups** tab, then **Groups**. Click **New**.
5. In the **Create a New Group** page provide the following information:
   - **Name**: Enter the name of the Group. Group names are case insensitive but must be unique. See the online help for a list of invalid characters.
   - (Optional) **Description**: Enter a description.
   - **Provider**: Select the authentication provider from the list that corresponds to where the group information is contained. DefaultAuthenticator is the name for the default authentication provider.
6. Click **OK**.

   The group name is added to the Group table.

To add a user to a group in the default directory server:

1. If needed, launch Oracle WebLogic Server Administration Console.

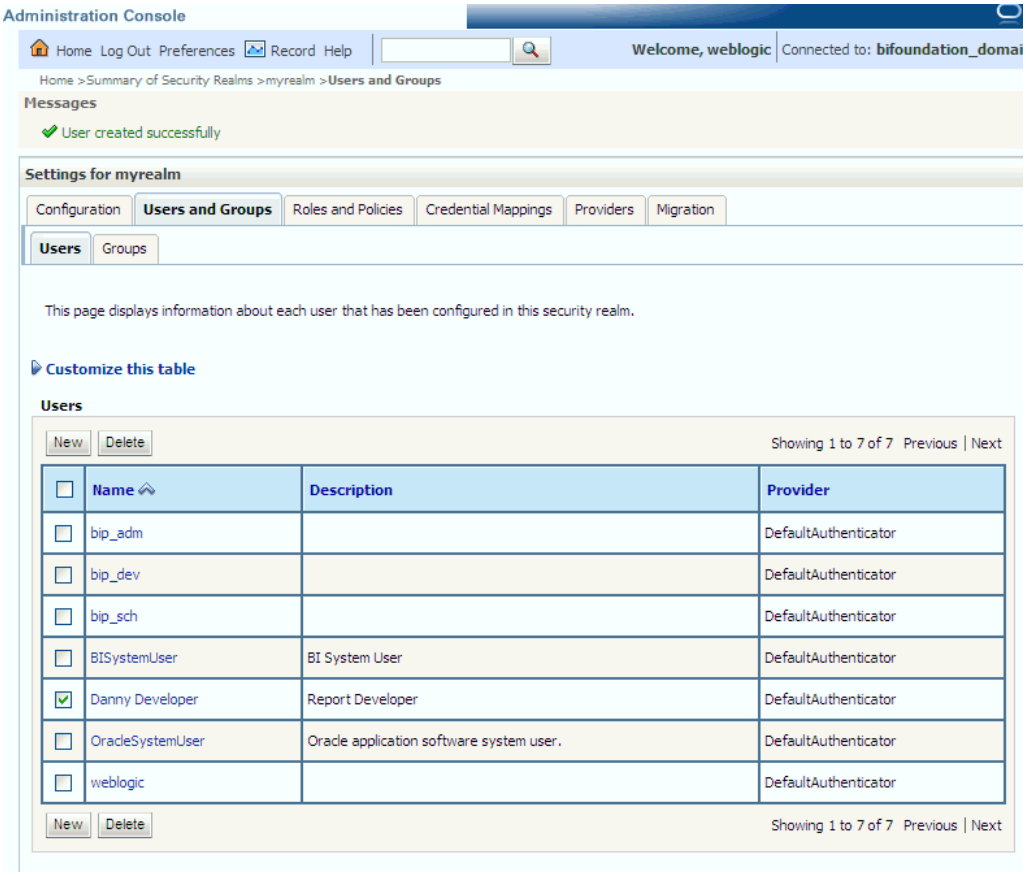
   For more information, see Section 2.5.1, "Accessing Oracle WebLogic Server Administration Console."

2. Log in as an administrative user.

3. In the Administration Console, select **Security Realm** from the left pane and click the realm you are configuring. For example, **myrealm**.

4. Select **Users and Groups** tab, then **Users**, as shown in Figure 2–6. Select the user from **Name**.

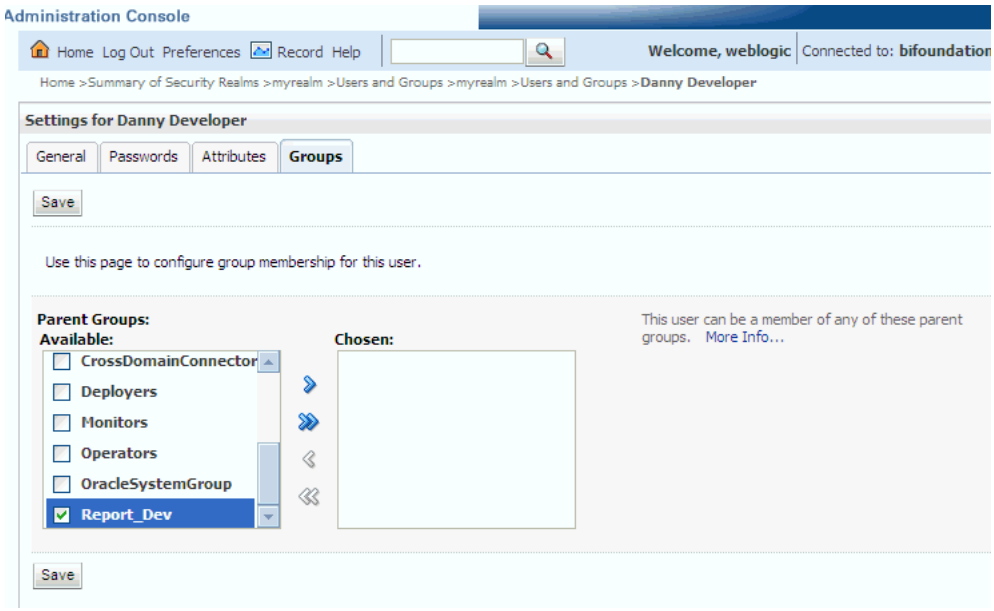
Figure 2–6 Users Tab



5. From the **Settings** page, select the **Groups** tab to display the list of available groups.

6. Select one or more groups from the **Available** list and use the shuttle controls to move them to the **Chosen** list, as shown in Figure 2–7.

Figure 2–7 Available List and Chosen List



7. Click **Save**.

   The user is added to the group.

To change a user password in the default directory server:

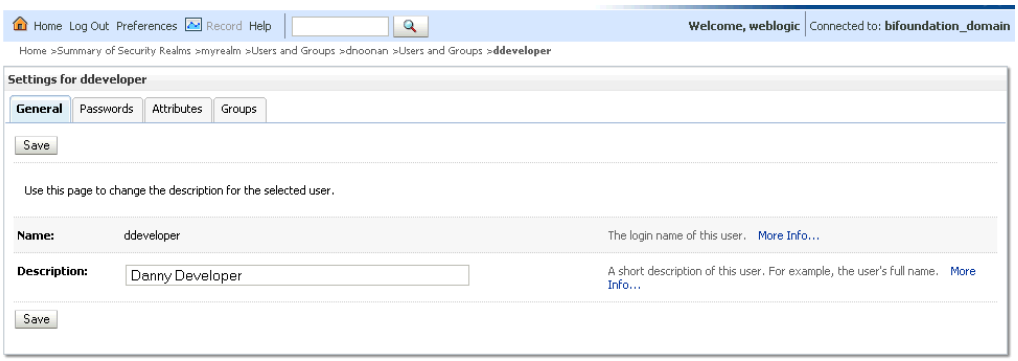
1. If needed, launch Oracle WebLogic Server Administration Console.

   For more information, see Section 2.5.1, "Accessing Oracle WebLogic Server Administration Console."

2. Log in as an administrative user.
3. In the Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.
4. Select **Users and Groups** tab, then **Users**.
5. In the Users table select the user you want to change the password for.

   The settings page for the user displays, as shown in Figure 2–8.

Figure 2–8 Settings Page



6. Select the Passwords tab and enter the password in the **New Password** and **Confirm Password** fields.

7. Click **Save**.

2.6 Managing Authorization

After a user is authenticated, further access to BI Publisher resources is controlled by the granting of permissions, also known as authorization. The policy store contains the system and application-specific policies and roles required for BI Publisher. A policy store can be file-based or LDAP-based and holds the mapping definitions between the default BI Publisher application roles, permissions, users and groups. BI Publisher permissions are granted by mapping users and groups from the identity store to application roles and permission grants located in the policy store. These mapping definitions between users and groups (identity store) and the application roles (policy store) are also kept in the policy store.

Note: Best practice is to map groups instead of individual users to application roles. Controlling membership in a group reduces the complexity of tracking access rights for multiple individual users. Group membership is controlled in the identity store.

The system-jazn-data.xml file is installed and configured as the default policy store. You can continue to use the default store and modify it as needed for your environment, or you can migrate its data to an LDAP-based provider. Oracle Internet Directory is the supported LDAP server in this release.

The policy store and credential store must be of the same type in your environment. That is, both must be either file-based or LDAP-based.

Permissions must be defined in a manner that BI Publisher understands. All valid BI Publisher permissions are premapped to application policies, which are in turn premapped to the default application roles. You cannot create new permissions in the policy store. However, you can customize the default application policy permission grants and application role mappings and you can create your own.

For more information about the default BI Publisher permissions grants, see Section 2.4.2, "Default Application Roles and Permissions." For more information about customizing application roles and permission grants, see Section 2.8.3, "Customizing the Policy Store."

2.6.1 Accessing Oracle Enterprise Manager Fusion Middleware Control

Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer a farm. A farm is a collection of components managed by Fusion Middleware Control. It can contain Oracle WebLogic Server domains, one Administration Server, one or more Managed Servers, clusters, and the Oracle Fusion Middleware components that are installed, configured, and running in the domain. During installation an Oracle WebLogic domain is created and BI Publisher is installed into that domain. If you performed a Simple or Enterprise installation type, this domain is named **bifoundation_domain** and is located within the WebLogic Domain in the Fusion Middleware Control target navigation pane.

Launch Fusion Middleware Control by entering its URL into a Web browser. The URL includes the name of the host and the administration port number assigned during the installation. This URL takes the following form: http://*hostname:port_number*/em. The default port is 7001. For more information about using Fusion Middleware Control, see *Oracle Fusion Middleware Administrator's Guide*.

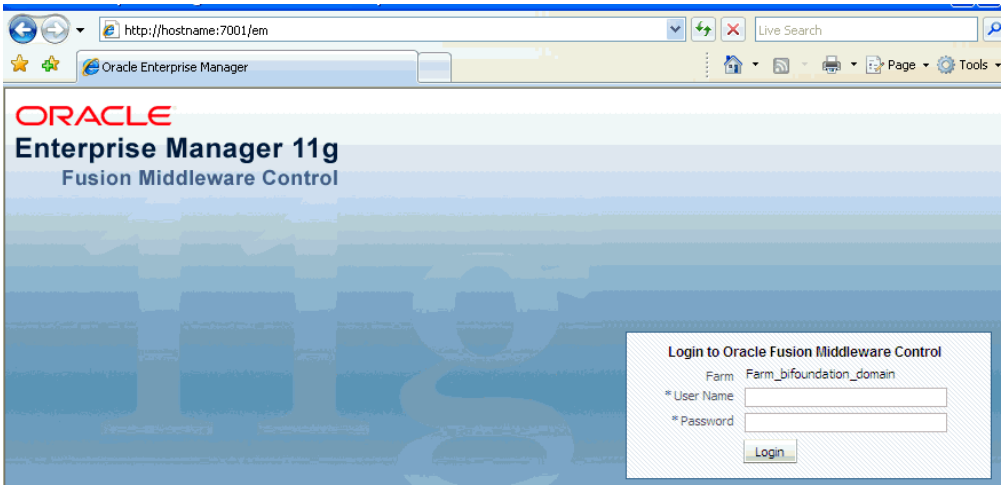
To display the Security menu in Fusion Middleware Control:

1. Log into Oracle Enterprise Manager Fusion Middleware Control by entering the URL in a Web browser.

   For example, http://*hostname:7001*/em.

   The Fusion Middleware Control login page displays, as shown in Figure 2–9.

Figure 2–9 Fusion MIddleware Control Login Page



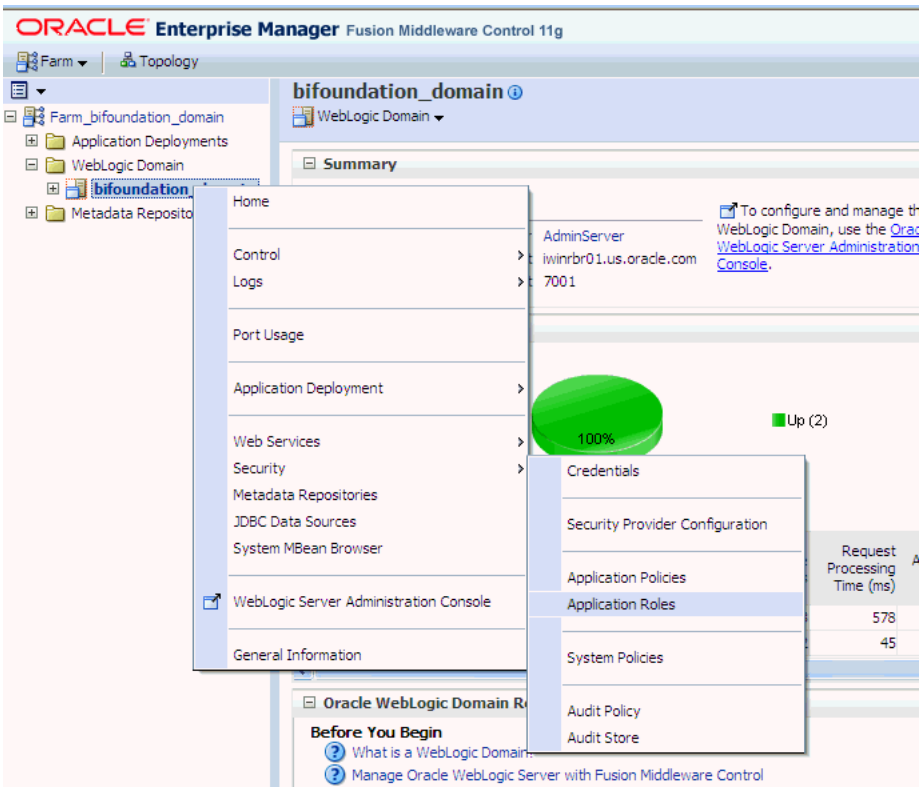
2. Enter the BI Publisher administrative user name and password and click **Login**.

   The password is the one you supplied during the installation of BI Publisher. If these values have been changed, then use the current administrative user name and password combination.

3. From the target navigation pane, open **WebLogic Domain** to display **bifoundation_domain**. Display the **Security** menu by selecting one of the following methods:

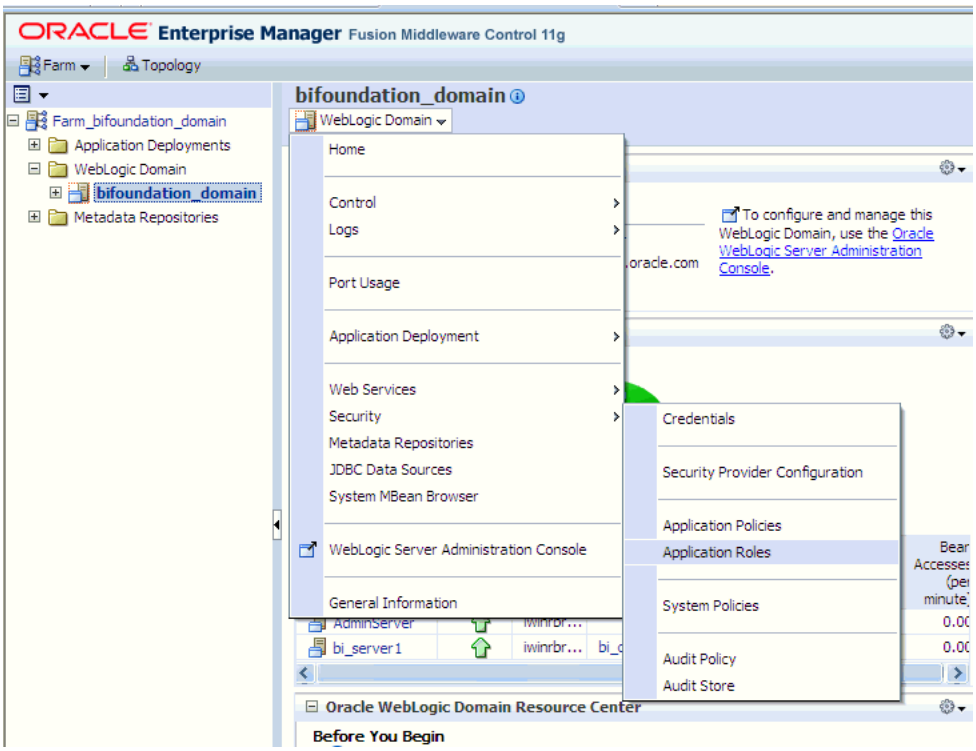
   - Right-click **bifoundation_domain** to display the **Security** menu, as shown in Figure 2–10. Select **Security** to display a submenu.

Figure 2–10 Security Submenu



- From the content pane, display the **WebLogic Domain** menu and select **Security**. Select **Security** to display a submenu, as shown in Figure 2–11.

Figure 2–11 Security Submenu



2.6.2 Managing the Policy Store Using Fusion Middleware Control

Use Fusion Middleware Control to manage the BI Publisher application policies and application roles maintained in the policy store whether it is file-based or LDAP-based. For more information about configuring an LDAP-based policy store, see Section 2.8.2, "Configuring a New Policy Store and Credential Store Provider."

> **Caution:** Oracle recommends you make a copy of the original system-jazn-data.xml policy file and place it in a safe location. Use the copy of the original file to restore the default policy store configuration, if needed. Changes to the default security configuration might lead to an unwanted state. The default installation location is MW_HOME/user_projects/domain/*your_domain*/config/fmwconfig.

The following are common policy store management tasks:

- Modifying the membership of an application role. For more information, see Section 2.6.4, "Modifying Membership in an Application Role."
- Modifying the permission grants for an application role. For more information, see Section 2.8.3.3, "Changing Permission Grants for an Application Policy."
- Creating a new application role from the beginning. For more information, see Section 2.8.3.1, "Creating Application Roles Using Fusion Middleware Control."
- Creating a new application role based on an existing application role. For more information, see Section 2.8.3.1, "Creating Application Roles Using Fusion Middleware Control."

2.6.3 Modifying Application Roles Using Fusion Middleware Control

Members can be added or deleted from an application role using Fusion Middleware Control. You must perform these tasks while in the WebLogic Domain that BI Publisher is installed in. For example, bifoundation_domain.

> **Caution:** Be very careful when changing the permission grants and membership for the default application roles. Changes could result in an unusable system.

2.6.4 Modifying Membership in an Application Role

Valid members of an application role are users, groups, or other application roles. The process of becoming a member of an application role is called *mapping*. That is, being mapped to an application role is to become a member of an application role. Best practice is to map groups instead of individual users to application roles for easier maintenance.

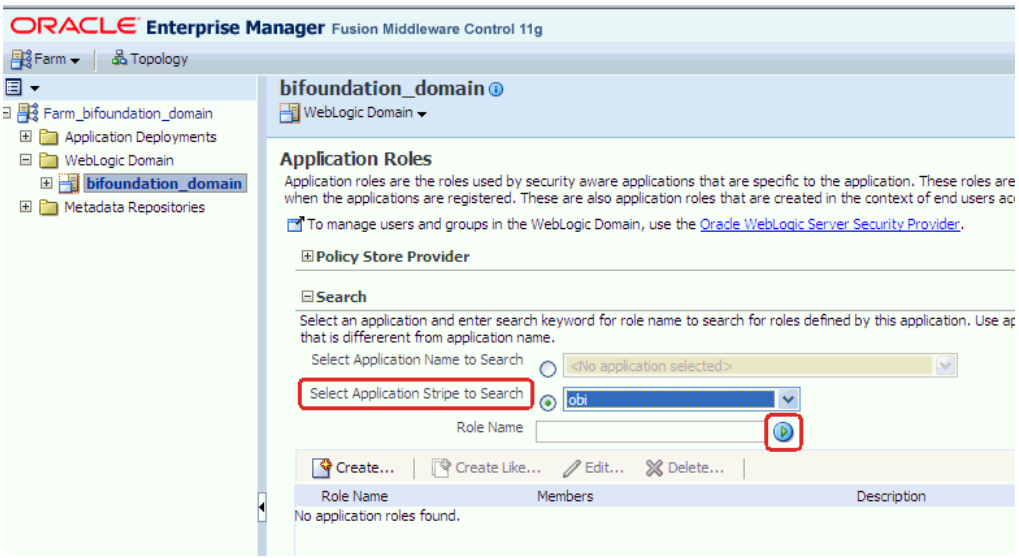
To add or remove members from an application role:

1. Log into Fusion Middleware Control, navigate to **Security**, then select **Application Roles** to display the **Application Roles** page.

   For information about navigating to the **Security** menu, see Section 2.6.1, "Accessing Oracle Enterprise Manager Fusion Middleware Control."

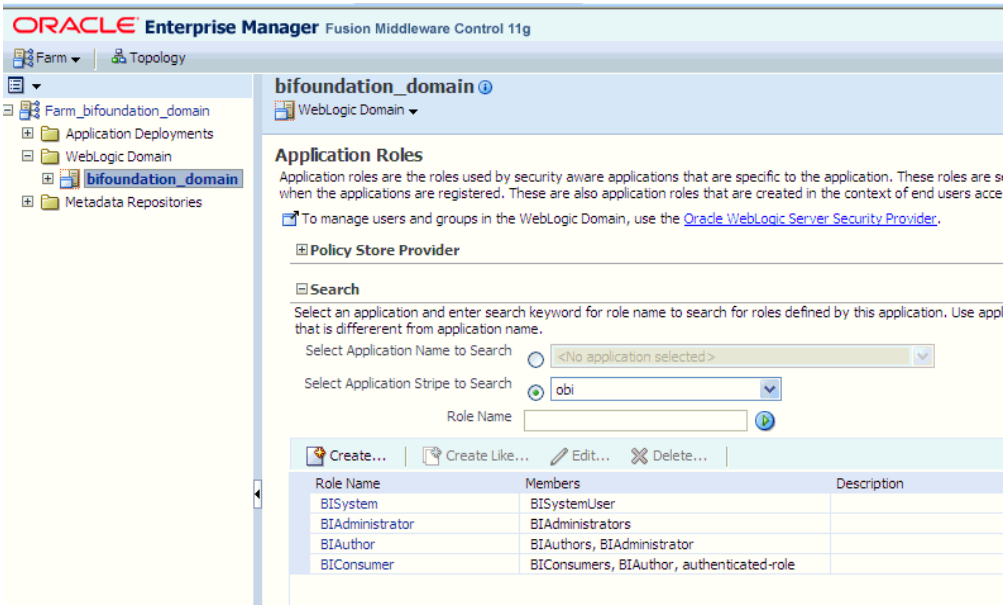
2. Choose **Select Application Stripe to Search**, then select the **obi** from the list. Click the search icon next to **Role Name**, as shown in Figure 2–12.

Figure 2–12 Search Icon



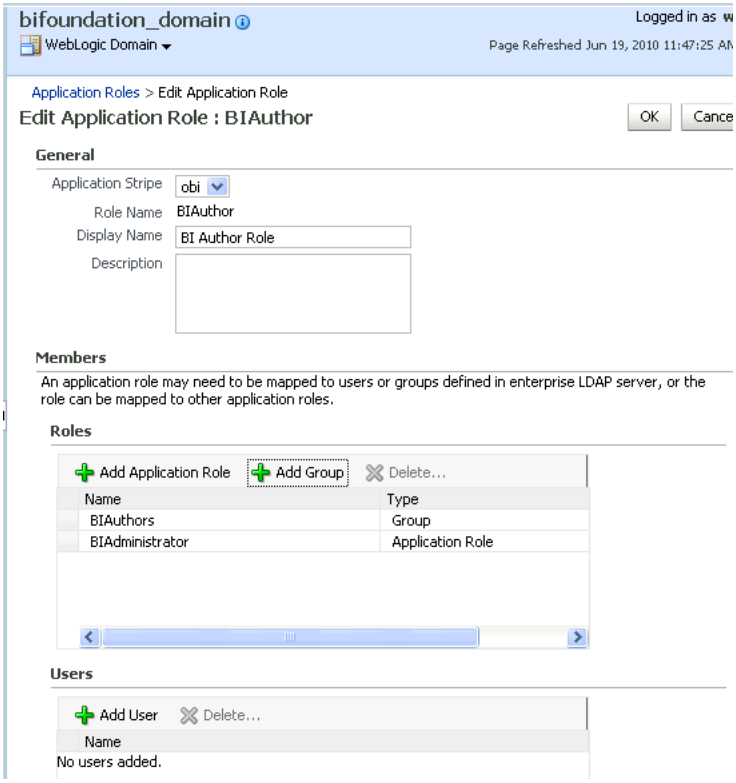
The BI Publisher application roles are displayed. Figure 2–13 shows the default application roles.

Figure 2–13 Default Application Roles



3. Select the cell next to the application role name and click **Edit** to display the **Edit Application Role** page. In Figure 2–14, the BIAuthor application role has been selected.

Figure 2–14 BIAuthor Application Role



You can add or delete members from the **Edit Application Role** page. Valid members are application roles, groups, and users.

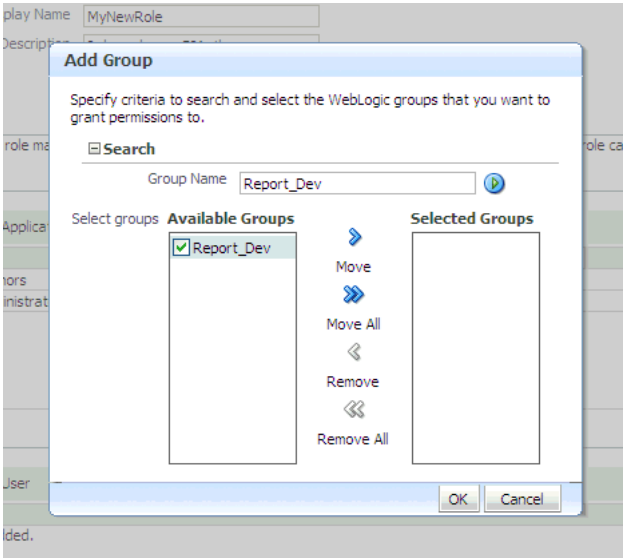
4. Select from the following options:

   - **To delete a member**: From **Members**, select from **Name** the member to activate the **Delete** button. Click **Delete**.
   - **To add a member**: Click the **Add** button that corresponds to the member type being added. Select from **Add Application Role**, **Add Group**, and **Add User**.

5. If adding a member, complete **Search** and select from the available list. Use the shuttle controls to move the member to the selected field. Click **OK**.

   For example, Figure 2–15 shows the **Add Group** dialog after the **Report_Dev** group has been selected.

Figure 2–15 Add Group Dialog



The added member displays in the **Members** column corresponding to the application role modified in the **Application Roles** page.

2.7 Managing Credentials

Credentials used by the system are stored in a single secure credential store. Oracle Wallet is the default credential store file (cwallet.sso). The credential store alternatively can be LDAP-based and Oracle Internet Directory is the supported LDAP server in this release. LDAP-based credential stores are configured and administered using Oracle Enterprise Manager Fusion Middleware Control or WLST commands.

Each credential is uniquely identified by a *map name* and a *key name*. Each map contains a series of keys and each key is a credential. The combination of map name and key name must be unique for all credential store entries. The following credential maps are used by BI Publisher:

- oracle.bi.system: Contains the credentials that span the entire BI Publisher platform.
- oracle.bi.publisher: Contains the credentials used by only BI Publisher.

The following two credential types are supported:

- Password: Encapsulates a user name and a password.
- Generic: Encapsulates any customized data or arbitrary token, such as public key certificates.

To facilitate getting started with your development environment, default credentials are inserted into the file-based credential store during installation. Be aware that BI Publisher credentials such as user passwords are stored in the identity store and managed with its corresponding administrative interface.

2.7.1 Managing the Credential Store

Credentials can be managed either in Fusion Middleware Control or using WLST command. For more information about both methods, see "Managing the Domain Credential Store" in *Oracle Fusion Middleware Application Security Guide*.

2.7.2 Managing BISystemUser Credentials

If using Oracle Business Intelligence as a data store, BI Publisher establishes system communication with it as BISystemUser. If you change the BISystemUser password in the identity store administrative interface, you also must change the password in the credential store (oracle.bi.system credential map). This applies if you have created a custom application role to take the place of the default BISystemUser. Components cannot communicate with each other if the credentials are out-of-sync. For more information about how Oracle Business Intelligence uses BISystemUser for trusted system communication, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

2.8 Customizing the Default Security Configuration

You can customize the default security configuration in the following ways:

- Configure a new authentication provider. For more information, see Section 2.8.1, "Configuring a New Authentication Provider."
- Configure new policy store and credential store providers. For more information, see Section 2.8.2, "Configuring a New Policy Store and Credential Store Provider."
- Migrate policies and credentials from one store to another. For more information, see Section 2.8.2.1, "Reassociating the Policy Store and Credential Store."
- Create new application roles. For more information, see Section 2.8.3.1, "Creating Application Roles Using Fusion Middleware Control."
- Create new application policies. For more information, see Section 2.8.3.2, "Creating Application Policies Using Fusion Middleware Control."
- Modify the permission grants for an application policy. For more information, see Section 2.8.3.3, "Changing Permission Grants for an Application Policy."

2.8.1 Configuring a New Authentication Provider

You can configure another supported LDAP server to be the authentication provider. Configuring BI Publisher to use an alternative external identity store is performed using the Oracle WebLogic Server Administration Console. BI Publisher delegates authentication and user population management to the authentication provider and identity store configured for the domain it is a part of. For example, if configured to use Oracle WebLogic Server's default authentication provider, then management is performed in the Oracle WebLogic Server Administration Console. If configured to use Oracle Internet Directory (OID), then the OID management user interface is used, and so on.

If using an authentication provider other than the one installed as part of the default security configuration, the default users and groups that are discussed in Section 2.4.1, "Default Users and Groups" are not automatically present. You can create users and groups with names of your own choosing or re-create the default user and group names if the authentication provider supports this. After this work is completed, you must map the default BI Publisher application roles to different groups again. For example, if the corporate LDAP server is being used as the identity store and you are unable to re-create the BI Publisher default users and groups in it, you must map the default application roles to different groups specific to the corporate LDAP server. Use Fusion Middleware Control to map the groups to application roles.

For information about how to configure a different authentication provider, see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help* and Oracle Fusion Middleware Securing Oracle WebLogic Server.

2.8.2 Configuring a New Policy Store and Credential Store Provider

The policy store and credential store can be file-based or LDAP-based. The supported LDAP server for both stores in this release is Oracle Internet Directory. The pre-requisites for using an LDAP-based store are the same as for both the policy store and credential store. For more information, see "Configuring LDAP-Based Policy and Credential Stores" in *Oracle Fusion Middleware Application Security Guide*.

2.8.2.1 Reassociating the Policy Store and Credential Store

Migrating policies and credentials from one security store to another is called reassociation. Both policy store and credential store data can be reassociated (migrated) from a file-based store to an LDAP-based store, or from an LDAP-based store to another LDAP-based store

Because the credential store and the policy store must both be of the same type, when reassociating one store you must reassociate the other.

For more information about reassociation and the steps required to migrate credential store and policy store data to Oracle Internet Directory, see "Reassociating with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

2.8.3 Customizing the Policy Store

The Fusion Middleware Security model can be customized for your environment by creating your own application policies and application roles. Existing application roles can be modified by adding or removing members as needed. Existing application policies can be modified by adding or removing permission grants. For more information about managing application policies and application roles, see *Oracle Fusion Middleware Application Security Guide*.

Note: Before creating a new application policy or application role and adding it to the default BI Publisher security configuration, familiarize yourself with how permission and group inheritance works. It is important when constructing a role hierarchy that circular dependencies are not introduced. Best practice is to leave the default security configuration in place and first incorporate your customized application policies and application roles in a test environment. For more information, see Section 2.3, "Permission Grants and Inheritance.".

2.8.3.1 Creating Application Roles Using Fusion Middleware Control

There are two methods for creating a new application role:

- **Create New** — A new application role is created. Members can be added at the same time or you can save the new role after naming it and add members later.
- **Copy Existing** — A new application role is created by copying an existing application role. The copy contains the same members as the original, and is made a Grantee of the same application policy. You can modify the copy as needed to finish creating the new role.

To create a new application role:

1. Log into Fusion Middleware Control, navigate to **Security**, then select **Application Roles** to display the **Application Roles** page.

   For information, see Section 2.6.1, "Accessing Oracle Enterprise Manager Fusion Middleware Control."

2. Choose **Select Application Stripe to Search**, then select **obi** from the list. Click the search icon next to **Role Name**.

   The BI Publisher application roles display.

3. Click **Create** to display the **Create Application Role** page. You can enter all information at once or you can enter a **Role Name**, save it, and complete the remaining fields later. Complete the fields as follows:

   In the **General** section:

   - **Role Name** — Enter the name of the application role.
   - (Optional) **Display Name** — Enter the display name for the application role.
   - (Optional) **Description** — Enter a description for the application role.

   In the **Members** section, select the users, groups, or application roles to be mapped to the application role, Select **Add Application Role** or **Add Group** or **Add Users** accordingly. To search in the dialog box that displays:

   - Enter a name in **Name** field and click the blue button to search.
   - Select from the results returned in the **Available** box.
   - Use the shuttle controls to move the desired name to the **Selected** box.
   - Click **OK** to return to the **Create Application Role** page.
   - Repeat the steps until all members are added to the application role.

4. Click **OK** to return to the **Application Roles** page.

   The application role just created displays in the table at the bottom of the page.

To create an application role based on an existing one:

1. Log into Fusion Middleware Control, navigate to **Security**, then select **Application Roles** to display the **Application Roles** page.

   For information, see Section 2.6.1, "Accessing Oracle Enterprise Manager Fusion Middleware Control."

2. Choose **Select Application Stripe to Search**, then select the **obi** from the list. Click the search icon next to **Role Name**.

   The BI Publisher application roles display.

3. Select an application role from the list to enable the action buttons.

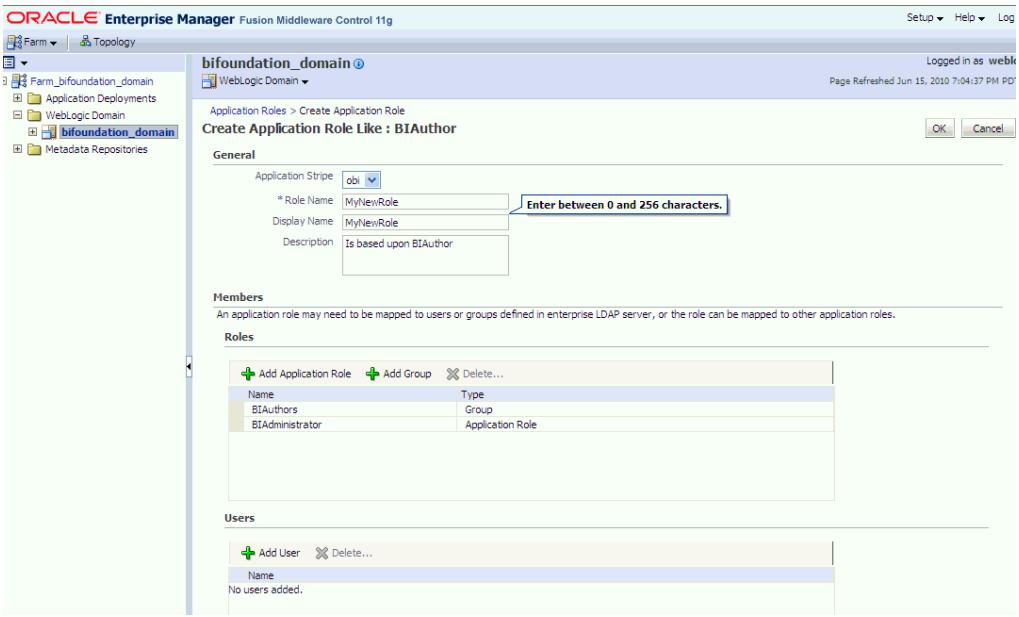
4. Click **Create Like** to display the **Create Application Role Like** page.

   The Members section is completed with the same application roles, groups, or users that are mapped to the original role.

5. Complete the **Role Name**, **Display Name**, and **Description** fields.

   Figure 2–16 shows an application role based upon BIAuthor after being named **MyNewRole**, as an example.

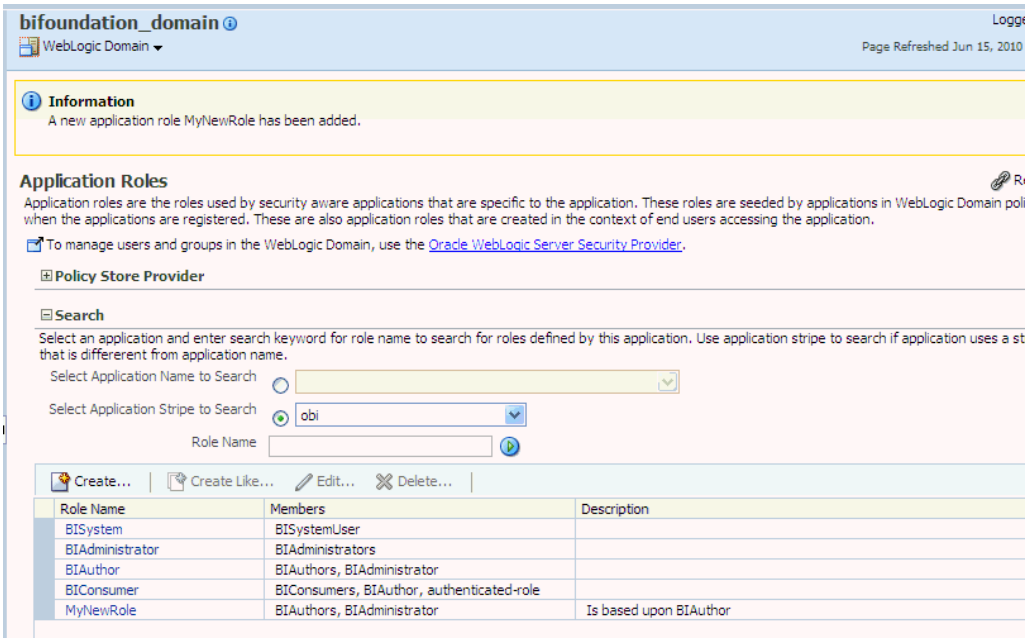
Figure 2–16 Role Based on BIAuthor



6. Use **Add** and **Delete** to modify the members as appropriate and click **OK**.

   The just created application role displays in the table at the bottom of the page. Figure 2–17 shows the example **MyNewRole** that is based upon the default BIAuthor application role.

Figure 2–17 MyNewRole Based on Default BIAuthor Role



2.8.3.2 Creating Application Policies Using Fusion Middleware Control

All BI Publisher permissions are provided and you cannot create new permissions. Permission grants are controlled in the Fusion Middleware Control **Application Policies** page. The permission grants are defined in an application policy. An

application role, user, or group, is then mapped to an application policy. This process makes the application role, user, or group a **Grantee** of the application policy.

There are two methods for creating a new application policy:

- **Create New** — A new application policy is created and permissions are added to it.
- **Copy Existing** — A new application policy is created by copying an existing application policy. The copy is named and existing permissions are removed or permissions are added as needed.

To create a new application policy:

1. Log into Fusion Middleware Control, navigate to **Security**, then select **Application Policies** to display the **Application Policies** page.

   For information, see Section 2.6.1, "Accessing Oracle Enterprise Manager Fusion Middleware Control."

2. Choose **Select Application Stripe to Search**, then select the **obi** from the list. Click the search icon next to **Permission**.

   The BI Publisher application policies are displayed. The **Principal** column displays the name of the policy **Grantee**.

3. Click **Create** to display the **Create Application Grant** page.

4. To add permissions to the policy being created, click **Add** in the **Permissions** area to display the **Add Permission** dialog.

   - Complete the **Search** area and click the blue search button next to the **Resource Name** field.

     All permissions located in the **obi** application stripe are displayed. For information about the BI Publisher permissions, see Section 2.4.2, "Default Application Roles and Permissions."

   - Select the desired BI Publisher permission and click **OK**. Repeat until all desired permissions are selected. Selecting non-BI Publisher permissions has no effect in the policy.

   - To remove any items, select it and click **Delete**.

   You are returned to the **Create Application Grant** page. The selected permissions display in the **Permissions** area.

5. To add an application role to the policy being created, click **Add Application Role** in the **Grantee** area to display the **Add Application Role** dialog.

   - Complete the **Search** area and click the blue search button next to the **Resource Name** field.

   - Select from the **Available Roles** list and use the shuttle controls to move it to **Selected Roles**.

   - Click **OK**.

   You are returned to the **Application Policies** page. The **Principal** (Grantee) and **Permissions** of the policy just created are displayed in the table.

To create an application policy based on an existing one:

1. Log into Fusion Middleware Control navigate to **Security**, then select **Application Policies** to display the **Application Policies** page.

For information, see Section 2.6.1, "Accessing Oracle Enterprise Manager Fusion Middleware Control."

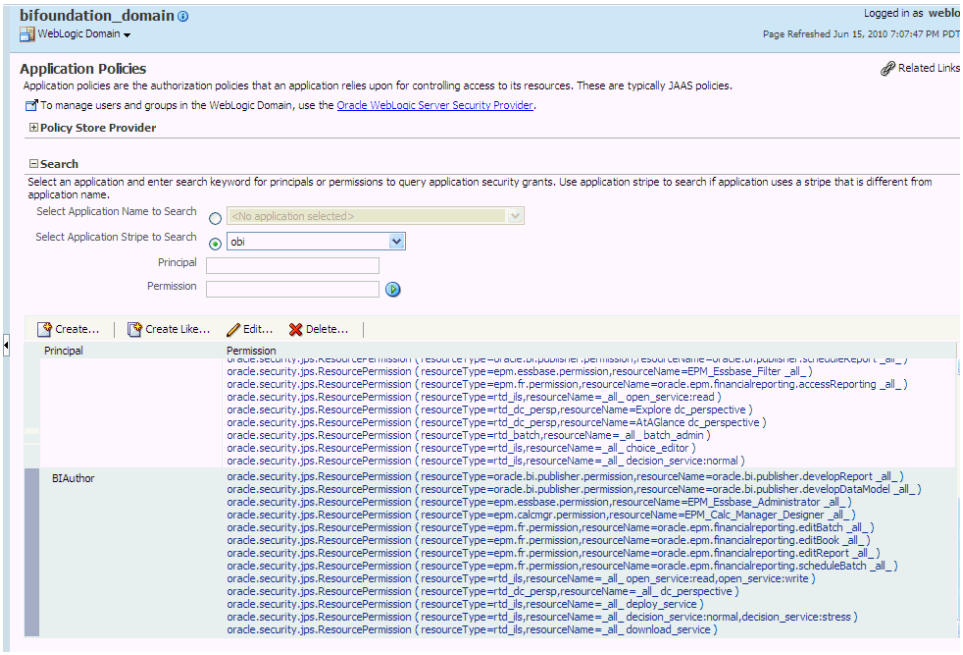
2. Choose **Select Application Stripe to Search**, then select **obi** from the list. Click the search icon next to **Permission**.

   The BI Publisher application policies are displayed. The **Principal** column displays the name of the policy **Grantee**.

3. Select an existing policy from the table.

   For example, Figure 2–18 shows the **BIAuthor** Principal (Grantee) selected and the **Create Like** button activated.

Figure 2–18 BIAuthor Principal (Grantee)



4. Click **Create Like** to display the **Create Application Grant Like** page. The Permissions table displays the names of the permissions granted by the policy selected.

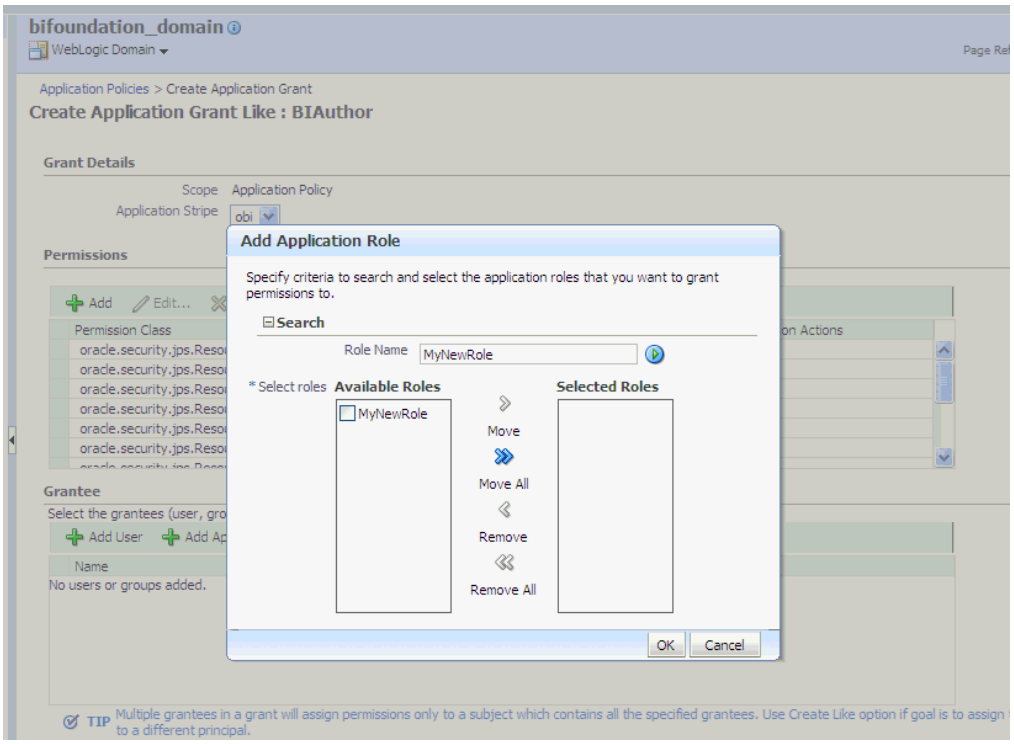
5. To remove any items, select it and click **Delete**.

6. To add application roles to the policy, click **Add Application Role** in the **Grantee** area to display the **Add Application Role** dialog.

   The following figures use the **MyNewRole** application role as an example.

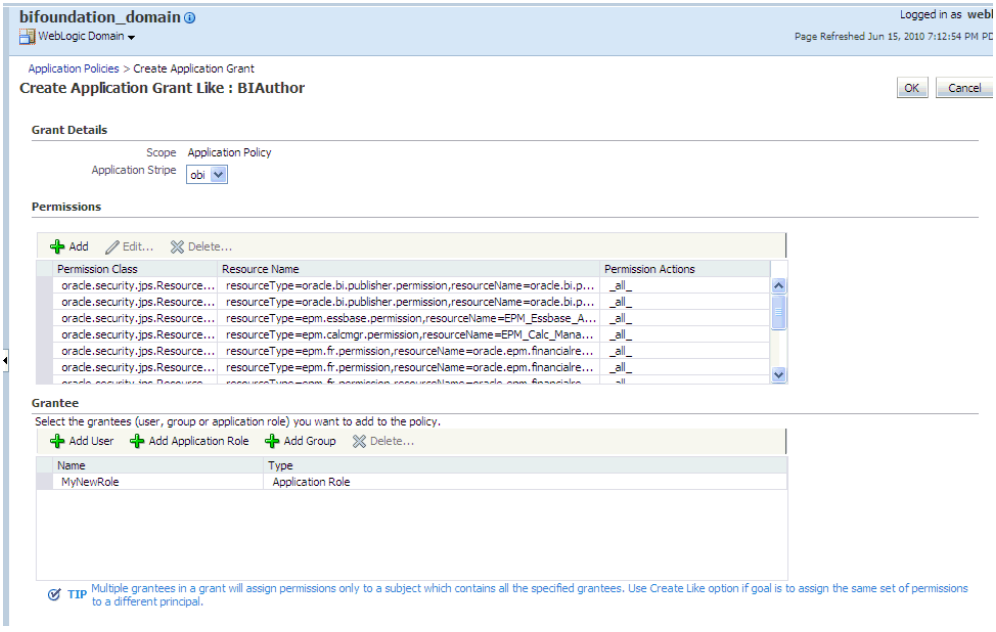
   - Complete the **Search** area and click the blue search button next to the **Resource Name** field, as shown in Figure 2–19.

Figure 2–19 Add Application Role Dialog



- Select from the **Available Roles** list and use the shuttle controls to move it to **Selected Roles**. The **Create Application Grant Like** page displays (as shown in Figure 2–20) with the selected application role added as **Grantee**.

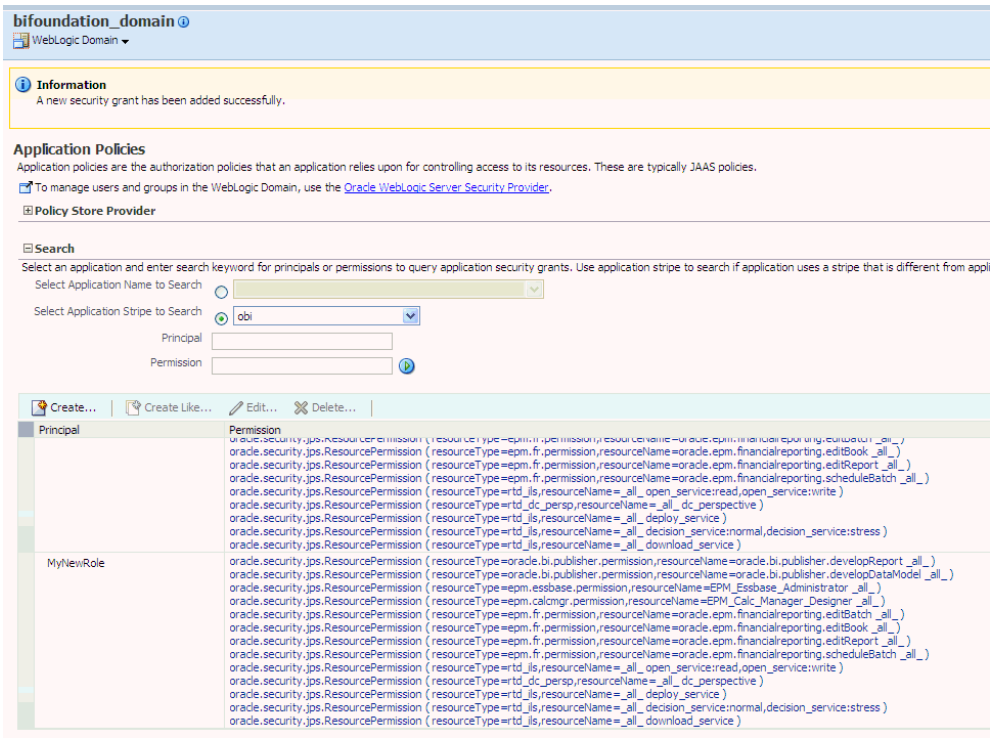
Figure 2–20 Create Application Grant Like Page



- Click **OK**.

You are returned to the **Application Policies** page. The Principal and Permissions of the policy created are displayed in the table, as shown in Figure 2–21.

Figure 2–21 Principal and Permissions of Policy



2.8.3.3 Changing Permission Grants for an Application Policy

You can change one or more permissions granted by an application policy.

To add or remove permission grants from an application policy:

1. Log into Fusion Middleware Control, navigate to **Security**, then select **Application Policies** to display the **Application Policies** page.

   For information, see Section 2.6.1, "Accessing Oracle Enterprise Manager Fusion Middleware Control."

2. Choose **Select Application Stripe to Search**, then select **obi** from the list. Click the search icon next to **Role Name**.

   The BI Publisher application policies are displayed. The **Principal** column displays the name of the policy **Grantee**.

3. Select the name of the application role from the **Principal** column and click **Edit**.

4. Add or delete permissions from the **Edit Application Grant** view and click **OK** to save the changes.

3

Alternative Security Options

This chapter describes alternative security options for BI Publisher, including Single Sign-on (SSO), LDAP options, Oracle Access Manager (OAM), and Microsoft Active Directory.

It covers the following topics:

3.1 About Alternative Security Options

This chapter describes security concepts and options for a standalone implementation of Oracle BI Publisher, that is, not installed as part of the Oracle Business Intelligence Enterprise Edition. Note the following:

- If you have installed the Oracle BI Enterprise Edition, then see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for information about security.
- If you have installed BI Publisher on its own and you plan to use Oracle Fusion Middleware Security, then see Section 2.1, "Understanding the Security Model." The following topics will be of interest in this chapter:
  - Section 3.5, "About Catalog Permissions"
  - Section 3.7, "About Access to Data Sources"

- To configure BI Publisher with these other Oracle security models:
  - Oracle BI Server security
  - Oracle E-Business Suite security
  - Oracle Database security
  - Siebel CRM security

  See Chapter 5, "Integrating with Other Oracle Security Models."

Use the information in this chapter to configure the following:

- BI Publisher Security
- Integration with an LDAP provider

> **Note:** Any identity store provider that is supported by Oracle WebLogic Server can be configured to be used with BI Publisher. Configuring BI Publisher to use an alternative external identity store is performed using the Oracle WebLogic Server Administration Console. For this configuration, see Section 2.8, "Customizing the Default Security Configuration."

- Integration with a Single Sign-On provider

3.2 Authentication and Authorization Options

BI Publisher supports several options for authentication and authorization. You can choose a single security model to handle both authentication and authorization; or, you can configure BI Publisher to use a Single Sign-On provider or LDAP provider for authentication with another security model to handle authorization.

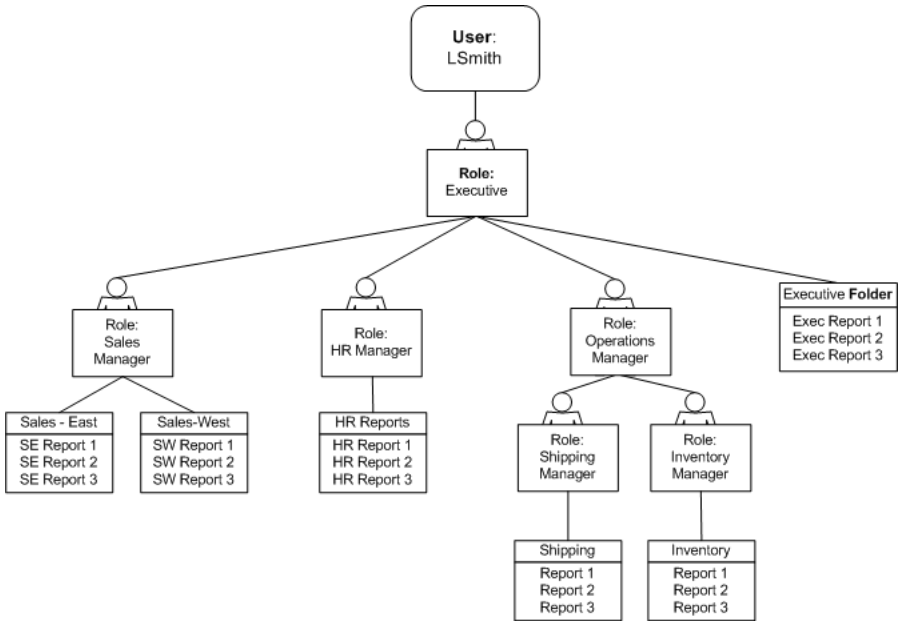
3.3 Understanding BI Publisher's Users, Roles, and Permissions

A user is assigned one or multiple **Roles**. A **Role** can grant any or all of the following:

- Privileges to use functionality
- Permissions to perform actions on catalog objects
- Access to data sources

You can create a hierarchy of roles by assigning roles to other roles. In this way the privileges and permissions of multiple roles can roll up to higher level roles. Figure 3–1 shows an example of the hierarchy structure of User, Role, and Folder.

Figure 3–1 Hierarchy Structure of User, Role, and Folder



3.3.1 Options for Configuring Users and Roles

There are three options for setting up users and roles:

- Set up users and roles in the BI Publisher Security Center

  For this option, follow the instructions in this section.

- Configure BI Publisher with your LDAP server

  For this option, see Section 3.10.2, "Configuring BI Publisher to Use an LDAP Provider for Authentication and Authorization."

- Set up users and roles in a supported Oracle security model. For this option, see Chapter 5, "Integrating with Other Oracle Security Models."

3.4 About Privileges to Use Functionality

BI Publisher provides a set of functional roles to grant access to specific functionality within the application. Assign these roles to users based on their need to perform the associated tasks. These roles cannot be updated or deleted.

Table 3–1 shows the privileges granted to each functional role.

Table 3–1 Privileges Granted to Functional Roles

Role	Privilege
BI Publisher Scheduler	View Export History Schedule
BI Publisher Template Designer	View Export History (public reports only) Enables access to Layout Editor Enables log on from Template Builder
BI Publisher Developer	View Export Schedule History Edit Report Enables access to Layout Editor Enables log on from the Template Builder Enables access to the Data Model Editor
BI Publisher Administrator	Enables the privileges of all other roles Grants access to the Administration page and all administration tasks

Roles assigned these privileges cannot perform any actions on objects in the catalog until they are also granted permissions on the catalog objects.

3.5 About Catalog Permissions

To perform the actions allowed by the functional roles above, a role must also be granted permissions to access the objects in the catalog. Table 3–2 describes permissions for roles.

Each of these permissions can be granted at the folder level to enable the operations on all items within a folder.

Table 3–2 Permissions for Roles

Permission	Description
Read	Enables a role to display an object in the catalog. If the object resides within a folder, a role must be granted the Read permission on the object and its parent Folder.
Write	▪ Report - requires the BI Publisher Developer role ▪ Data Model - requires the BI Publisher Developer role ▪ Sub Template and Style Template - requires the BI Publisher Developer Role or the BI Publisher Template Designer Role
Delete	Enables a role to delete an object.
Run Report Online	Enables a role to run a report and view it in the report viewer.
Schedule Report	Enables a role to schedule a report.
View Report Output	Enables a role to access the Report Job History for a report.

It is important to note that for a report consumer to successfully run a report, his role must have read access to every object that is referenced by the report.

For example, a report consumer must run a report in a folder named Reports. The data model for this report, resides in a folder named Data Models. This report references a Sub Template stored in a folder named Sub Templates, and also references a Style Template stored in a folder named Style Templates. The report consumer's role must be granted Read access to all of these folders and the appropriate objects within.

3.6 How Functional Privileges and Permissions Work Together

It is important to understand the following rules regarding the behavior of privileges and permissions:

- A role assigned a functional privilege cannot perform any actions in the catalog until catalog permissions are also assigned
- A role can be assigned a set of permissions on catalog objects without being assigned any functional privileges
- If a role is assigned a functional privilege, when catalog permissions are assigned, some permissions are inherited

3.6.1 A Role Must Be Assigned Catalog Permissions

A role assigned a functional role cannot perform any actions in the catalog until catalog permissions are granted. Note that the functional roles themselves (BI Publisher Developer, BI Publisher Scheduler, and so on) cannot be directly assigned

permissions in the catalog. The functional roles must first be assigned to a custom role and then the custom role is available in the catalog permissions table.

3.6.2 A Role Can Be Granted Catalog Permissions Only

The permissions available directly in the catalog enable running reports, scheduling reports, and viewing report output. Therefore if your enterprise includes report consumers who have no other reason to access BI Publisher except to run and view reports, then the roles for these users consist of catalog permissions only.

3.6.3 Inherited Permissions

When a role is assigned one of the functional roles, and that role is granted permissions on a particular folder in the catalog, then some permissions are granted automatically based on the functional role.

For example, assume that you create a role called Financial Report Developer. You assign this role the BI Publisher Developer role. For this role to create reports in the Financial Reports folder in the catalog, you grant this role Read, Write, and Delete permissions on the folder. Because the BI Publisher Developer role includes the run report, schedule report, and view report history privileges, these permissions are automatically granted on any folder to which a role assigned the BI Publisher Developer role is granted Read access.

3.7 About Access to Data Sources

A role must be granted access to a data source to view reports that run against the data source or to build and edit data models that use the data source. Add access to data sources in the **Roles and Permissions** page. See Section 3.8.4, "Granting Data Access."

3.8 Configuring Users, Roles, and Data Access

The following procedures describe:

- Creating Roles
- Creating Users and Assigning Roles to a User
- Granting Catalog Permissions
- Granting Data Access

3.8.1 Creating Roles

To create a new role in BI Publisher:

1. Navigate to the BI Publisher **Administration** page.
2. Under **Security Center**, click **Roles and Permissions**.
3. Click **Create Role**.
4. Enter a **Name** for the role and optionally, enter a **Description**.
5. Click **Apply**.
6. Click **Assign Roles** to assign roles to the user.
7. Use the shuttle buttons to move **Available Roles** to **Assigned Roles**. Click **Apply**.
8. To add a role to a role, click **Add Roles**.

9. Use the shuttle buttons to move **Available Roles** to **Included Roles**. Click **Apply**.

To add data sources to a role, see Section 3.8.4, "Granting Data Access."

3.8.2 Creating Users and Assigning Roles to a User

To create a new user in BI Publisher:

1. Navigate to the BI Publisher **Administration** page.
2. Under **Security Center**, click **Users**.
3. Click **Create User**.
4. Add the **User Name** and **Password** for the user.
5. Click **Apply**.
6. Click **Assign Roles** to assign roles to the user.
7. Use the shuttle buttons to move **Available Roles** to **Assigned Roles**. Click **Apply**.

3.8.3 Granting Catalog Permissions

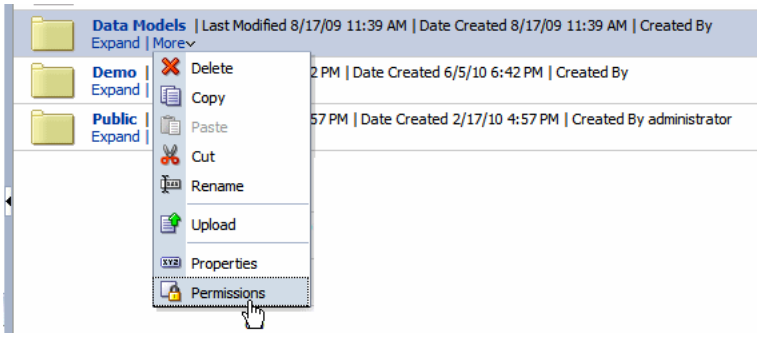
For a role to access an object in the catalog, the role must be granted Read permissions on both the object and the folder in which the object resides. Permissions can be granted at the folder level and applied to all the objects and subfolders it contains, or applied to individual objects.

To grant catalog permissions to a role:

1. Navigate to the Catalog.
2. Locate the folder or object on which to grant permissions and click **More**. From the menu (shown in Figure 3–2), select **Permissions**. Alternatively, you can select the folder and click **Permissions** in the **Tasks** region.

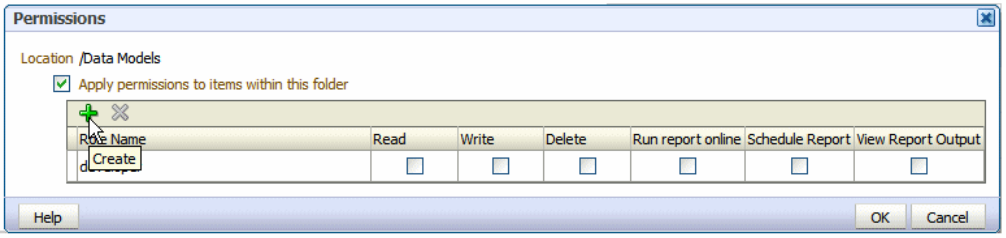
> **Note:** Permissions cannot be granted on the root Shared folder.

Figure 3–2 More Menu



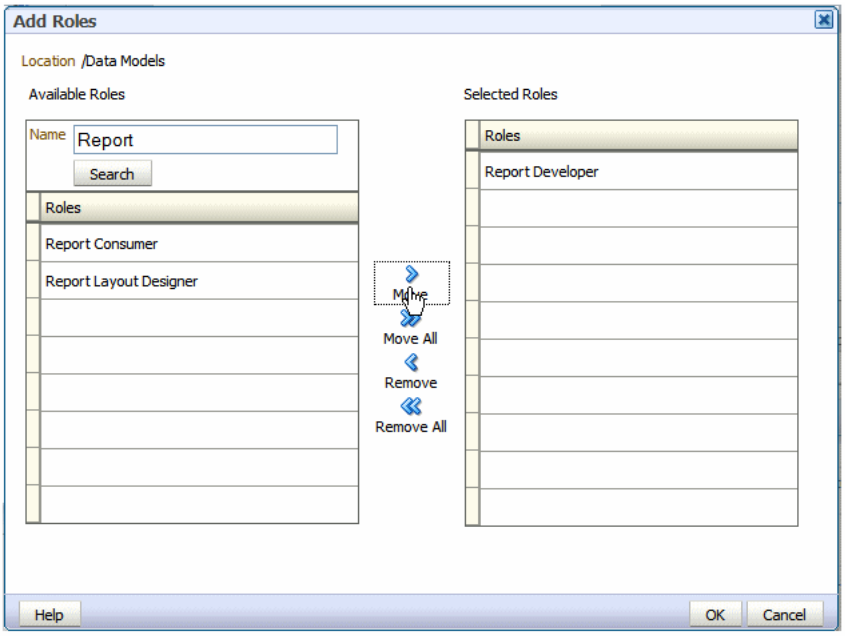
3. On the **Permissions** dialog (shown in Figure 3–3), click **Create**.

Figure 3–3 Permissions Dialog



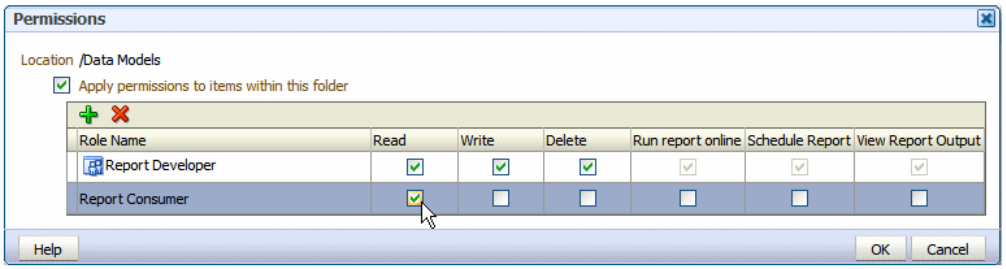
4. On the **Add Roles** dialog (shown in Figure 3–4), enter a search string to find a role, or simply click **Search** to display all roles. Use the shuttle buttons to move roles from the **Available Roles** list to the **Selected Roles** list.

Figure 3–4 Add Roles Dialog



5. When finished, click **OK** to return to the **Permissions** dialog.
6. On the **Permissions** dialog (shown in Figure 3–5), configure the permissions required by the role.

Figure 3–5 Permissions Dialog



Note the following:

- The icon next to the Report Developer role indicates that this role is assigned one of the BI Publisher functional roles (in this case, the BI Publisher Developer role).
- Once the Report Developer role is assigned access to this folder, the following permissions are automatically granted based on the privileges that comprise the BI Publisher Developer Role: Run report online, Scheduler Report, View Report Output.

7. If you are granting permissions on a Folder, select **Apply permissions to items within this folder**, if the permissions should apply to all objects.

3.8.4 Granting Data Access

A role must be granted access to a data source if the role must:

- Run or schedule a report built on a data model that retrieves data from the data source
- Create or edit a data model that retrieves data from the data source

To grant a role access to a data source:

1. Navigate to the BI Publisher **Administration** page.
2. Under **Security Center**, click **Roles and Permissions**.
3. On the **Roles and Permissions** page, locate the role, then click **Add Data Sources**.
4. On the **Add Data Sources** page you see a region for each of the following types of data sources:
   - Database Connections
   - File Directories
   - LDAP Connections
   - OLAP Connections
5. Use the shuttle buttons to move the required data sources from the **Available Data Sources** list to the **Allowed Data Sources** list.
6. When finished, click **Apply**.

3.9 Security and Catalog Organization

Because permissions are granted in the catalog, it is very important to be aware of this design when creating roles for your organization and when structuring the catalog.

For example, assume that your organization requires the roles that are described in Table 3–3.

Table 3–3 Example Role Requirements

Role	Required Permissions
Sales Report Consumer	Needs to view and schedule Sales department reports.
Financial Report Consumer	Needs to view and schedule Financial department reports.
Executive Report Consumer	Needs to consume both Sales and Financial reports and executive level reports.
Sales Report Developer	Needs to create data models and reports for Sales department only.

Table 3–3 (Cont.) Example Role Requirements

Role	Required Permissions
Financials Report Developer	Needs to create data models and reports for Financials department only.
Layout Designer	Needs to design report layouts for all reports.

You might consider setting up the catalog structure as described in Table 3–4.

Table 3–4 Example Catalog Setup

Folder	Contents
Sales Reports	All reports for Sales Report Consumer. Also contains any Sub Templates and Style Templates associated with Sales reports.
Sales Data Models	All data models for Sales reports.
Financials Reports	All reports for Financials Report Consumer. Also contains any Sub Templates and Style Templates associated with Financials reports.
Financials Data Models	All data models for Financials reports
Executive Reports	All executive-level reports and data models.

Set up the roles as follows:

Example Role Configuration

Sales Report Consumer:

Grant catalog permissions:

- To the Sales Reports folder add the Sales Report Consumer and grant:

  Read

  Schedule Report

  Run Report Online

  View Report Online

  Select **Apply permissions to items within this folder**
- To the Sales Data Models folder add the Sales Report Consumer and grant:

  Read

Grant Data Access:

On the **Roles** page, locate the role, then click **Add Data Sources**. Add all data sources used by Sales reports.

Financials Report Consumer

Grant catalog permissions:

- To the Financials Reports folder add the Financials Report Consumer and grant:

  Read

  Schedule Report

  Run Report Online

  View Report Online

Select **Apply permissions to items within this folder**

- To the Financials Data Models folder add the Financials Report Consumer and grant:

  Read

Grant Data Access:

On the **Roles** page, locate the role, then click **Add Data Sources**. Add all data sources used by Financials reports.

Executive Report Consumer

Assign Roles:

On the **Roles** tab, assign the Executive Report Consumer the Sales Report Consumer and the Financials Report Consumer roles.

Grant catalog permissions:

- To the Executive Reports folder add the Executive Report Consumer and grant:

  Read

  Schedule Report

  Run Report Online

  View Report Online

  Select **Apply permissions to items within this folder**

Grant Data Access:

On the **Roles** tab, locate the role, then click **Add Data Sources**. Add all data sources used by Executive reports.

Sales Report Developer

Assign Roles:

On the **Roles** tab, assign the Sales Report Developer the BI Publisher Developer Role and the BI Publisher Template Designer Role.

Grant Data Access:

On the **Roles** tab, locate the Sales Report Developer and click **Add Data Sources**. Add all data sources from which Sales data models are built.

Grant Catalog Permissions:

- In the catalog, to the Sales Data Models folder add the Sales Report Developer and grant:

  Read, Write, Delete

- To the Sales Reports folder, add the Sales Report Developer and grant:

  Read, Write, Delete

Financials Report Developer

Assign Roles:

On the **Roles** tab, assign the Financials Report Developer the BI Publisher Developer Role, and the BI Publisher Template Designer Role.

Grant Data Access:

On the **Roles** tab, locate the Financials Report Developer and click **Add Data Sources**. Add all data sources from which Financials data models are built.

Grant Catalog Permissions:

- In the catalog, to the Financials Data Models folder add the Financials Report Developer and grant:

  Read, Write, Delete

- To the Financials Reports folder, add the Financials Report Developer and grant:

  Read, Write, Delete

Layout Designer

Assign Roles:

On the **Roles** tab, assign the Layout Designer the BI Publisher Template Designer Role and the BI Publisher Developer Role.

Grant Catalog Permissions:

- In the catalog, to the Financials Data Models and the Sales Data Models folders add the Layout Designer Role and grant:

  Read

- To the Financials Reports and Sales Reports folders, add the Layout Designer and grant:

  Read, Write, Delete

3.10 Using LDAP with BI Publisher

You can use BI Publisher with an LDAP provider for authentication only or for both authentication and authorization.

> **Important:** By default, BI Publisher allows every LDAP user to log in to the system even when no BI Publisher-specific roles are assigned to the user. Users cannot perform any functions that require roles, such as creating reports or data models; however if a user is assigned a role that is assigned permissions on catalog objects (such as traverse and open) the user will be able to perform those tasks.
>
> To prevent users from logging in to BI Publisher unless they have a BI Publisher role assigned, see Section 3.10.3, "Disable Users Without BI Publisher-Specific Roles from Logging In."

- Section 3.10.1, "Configuring BI Publisher to Use an LDAP Provider for Authentication Only"
- Section 3.10.2, "Configuring BI Publisher to Use an LDAP Provider for Authentication and Authorization"

3.10.1 Configuring BI Publisher to Use an LDAP Provider for Authentication Only

To use an LDAP provider for authentication in conjunction with another security model for authorization, perform the following in BI Publisher:

To configure BI Publisher to use LDAP for authentication only:

1. On the **Administration** page, under **Security Center** click **Security Configuration**.
2. Create a Local Superuser.

   Enter a **Superuser Name** and **Password** and select Enable Local Superuser check box. Enabling a local superuser ensures that you can access the Administration page of BI Publisher in case of security model configuration errors.
3. Scroll down to the **Authentication** region. Select the **Use LDAP** check box.
4. Enter the following:
   - **URL**

     For example: ldap://example.com:389/

     If you are using LDAP over SSL, then note the following:
     - the protocol is "ldaps"
     - the default port is 636

     An example URL would be: ldaps://example.com:636/
   - **Administrator Username** and **Password** for the LDAP server

     The Administrator user entered here must also be a member of the XMLP_ ADMIN group.
   - **Distinguished Name for Users**

     For example: cn=Users,dc=example,dc=com

     The distinguished name values are case-sensitive and must match the settings in the LDAP server.
   - **JNDI Context Factory Class**

     The default value is com.sun.jndi.ldap.LdapCtxFactory
   - **Attribute used for Login Username**

     Enter the attribute that supplies the value for the Login user name. This is also known as the Relative Distinguished Name (RDN). This value defaults to cn.
   - **Attribute used for user matching with authorization system** - enter the attribute that supplies the value to match users to the authorization system. For example, orcleguid.
5. Click **Apply**. Restart the BI Publisher server.

3.10.2 Configuring BI Publisher to Use an LDAP Provider for Authentication and Authorization

BI Publisher can be integrated with the LDAP provider to manage users and report access. Create the users and roles within the LDAP server, then configure the BI Publisher server to access the LDAP server.

In the BI Publisher security center module, assign folders to those roles. When users log in to the server, they have access to those folders and reports assigned to the LDAP roles.

Integrating the BI Publisher server with Oracle LDAP consists of three main tasks:

1. Set up users and roles in the LDAP provider
2. Configure BI Publisher to recognize the LDAP server

3. Assign catalog permissions and data access to roles

For information on supported LDAP servers, see "System Requirements and Certification" for the most up-to-date information on supported hardware and software.

3.10.2.1 Set Up Users and Roles in the LDAP Provider

The following steps must be performed in the LDAP provider. See the documentation for the provider for details on how to perform these tasks.

To set up users and roles:

1. In the Domain root node of the LDAP provider, create the roles that are described in Table 3–5 to integrate with BI Publisher. See Section 3.3, "Understanding BI Publisher's Users, Roles, and Permissions" for full descriptions of the required functional roles.

Table 3–5 Roles to Integrate with BI Publisher

BI Publisher System Group	Description
XMLP_ADMIN	The administrator role for the BI Publisher server. You must assign the Administrator account used to access your LDAP server the XMLP_ADMIN group.
XMLP_DEVELOPER	Allows users to create and edit reports and data models.
XMLP_SCHEDULER	Allows users to schedule reports.
XMLP_TEMPLATE_DESIGNER	Allows users to connect to the BI Publisher server from the Template Builder for Word and to upload and download templates. Allows users to design layouts using the BI Publisher Layout Editor.

2. Create other functional roles as required by your implementation (for example: HR Manager, Warehouse Clerk, or Sales Manager), and assign the appropriate BI Publisher functional roles.
3. Assign roles to users.

> **Note:** Ensure that you assign the Administrator account the XMLP_ADMIN role.

3.10.2.2 Configure the BI Publisher Server to Recognize the LDAP Server

To configure the BI Publisher server to recognize the LDAP server, update the Security properties in the BI Publisher Administration page.

> **Important:** Ensure that you understand your site's LDAP server configuration before entering values for the BI Publisher settings.

To configure the BI Publisher Server for the LDAP Server:

1. On the **Administration** page, under **Security Center** click **Security Configuration**.
2. Create a Local Superuser.

   Enter a **Superuser Name** and **Password** and select Enable Local Superuser check box. Enabling a local superuser ensures that you can access the Administration page of BI Publisher in case of security model configuration errors.

3. Scroll down to the **Authorization** region. Select LDAP for the **Security Model**.
4. Enter the following:
   - **URL**

     For example: ldap://example.com:389/

     If you are using LDAP over SSL, then note the following:

     – the protocol is "ldaps"

     – the default port is 636

     For example: ldaps://example.com:636/
   - **Administrator Username** and **Password** for the LDAP server

     The Administrator user entered here must also be a member of the XMLP_ ADMIN group.
   - **Distinguished Name for Users**

     For example: cn=Users,dc=example,dc=com

     The distinguished name values are case-sensitive and must match the settings in the LDAP server.
   - **Distinguished Name for Groups**

     For example: cn=Groups,dc=us,dc=oracle,dc=com

     The default value is cn=OracleDefaultDomain,cn=OracleDBSecurity,cn=Products,cn=OracleContext,dc=example,dc=com
   - **Group Search Filter**

     The default value is (&(objectclass=groupofuniquenames)(cn=*))
   - **Group Attribute Name**

     The default value is cn
   - **Group Member Attribute Name**

     The default value is uniquemember
   - **Member of Group Attribute Name**

     (Optional) Set this attribute only if memberOf attribute is available for User and Group. Group Member Attribute is not required when this attribute is available. Example: memberOf or wlsMemberOf
   - **Group Description Attribute Name**

     The default value is description
   - **JNDI Context Factory Class**

     The default value is com.sun.jndi.ldap.LdapCtxFactory
   - **Group Retrieval Page Size**

     Setting this value enables support of the LDAPv3 control extension for simple paging of search results. By default, the BI Publisher server does not use pagination. This value determines the number of results to return on a page (for example, 200). Your LDAP server must support control type 1.2.840.113556.1.4.319 to support this feature, such as Oracle Internet Directory

10.1.4. Ensure that you check your LDAP server documentation for support of this control type before entering a value.

For more information about LDAP pagination and the required control type, see the article: RFC 2696 - LDAP Control Extension for Simple Paged Results Manipulation (http://www.faqs.org/rfcs/rfc2696.html).

- **Attribute used for Login Username**

  Enter the attribute that supplies the value for the Login user name. This is also known as the Relative Distinguished Name (RDN). This value defaults to cn.

- **Automatically clear LDAP cache** - to schedule the automatic refresh of the LDAP cache the LDAP cache per a designated interval, select this box. After you select this box the following additional fields become enabled:
  - Enter an integer for **Ldap Cache Interval**. For example, to clear the LDAP cache once a day, enter 1.
  - Select the appropriate **Ldap Cache Interval Unit**: Day, Hour, or Minute.

- **Default User Group Name**

  (Optional) Use this option if your site has the requirement to allow all authenticated users access to a set of folders, reports, or other catalog objects. The user group name that you enter here is added to all authenticated users. Any catalog or data source permissions that you assign to this default user group are granted to all users.

- **Attribute Names for Data Query Bind Variables**

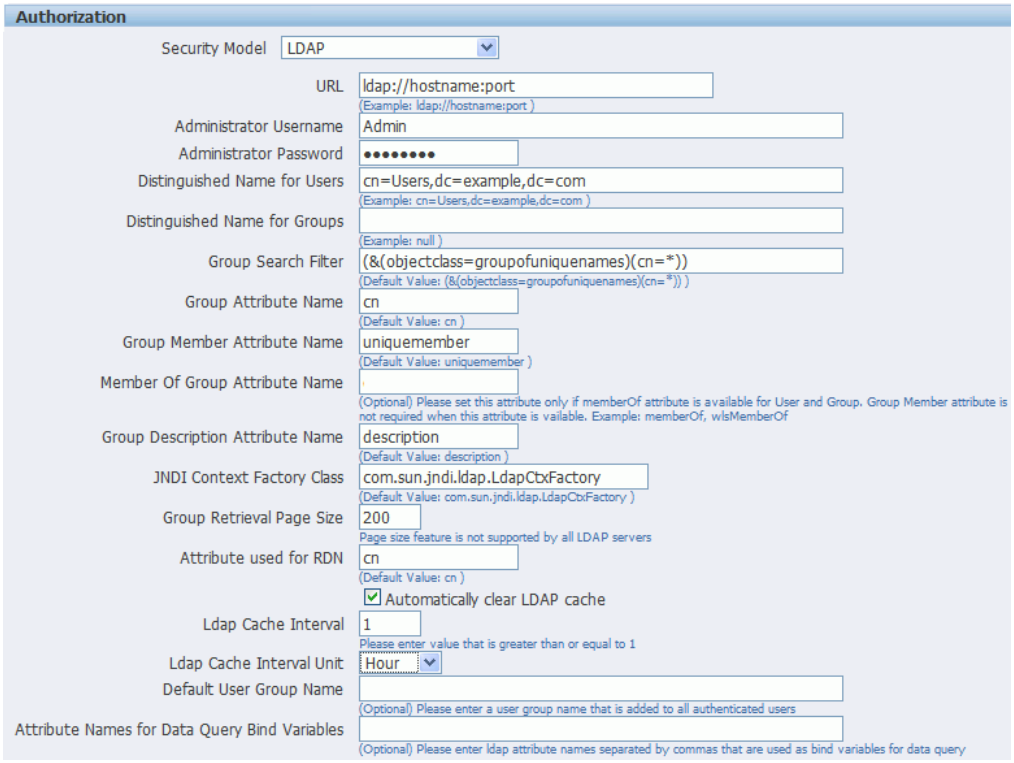
  (Optional) Use this property to set attribute values to be used as bind variables in a data query. Enter LDAP attribute names separated by a commas for example: memberOf, primaryGroupID,mail

  See the section "Creating Bind Variables from LDAP User Attributes" in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

5. Click **Apply**. Restart the BI Publisher server.

Figure 3–6 shows a sample of the LDAP security model entry fields from the Security Configuration page.

Figure 3–6 Sample of LDAP Security Model Entry Fields



If you are configuring BI Publisher to use LDAP over SSL, then you must also configure Java keystore to add the server certificate to JVM. For more information, see Section 4.3, "Configuring BI Publisher for Secure Socket Layer (SSL) Communication."

3.10.2.3 Assign Data Access and Catalog Permissions to Roles

To assign data access and catalog permissions to roles:

1. Log in to BI Publisher as a user assigned the XMLP_ADMIN role in the LDAP provider.
2. On the **Administration** page click **Roles and Permissions**.

   You see the roles that you created in the LDAP provider to which you assigned the XMLP_ roles. Note the following:

   - The XMLP_X roles are not shown because these are controlled through the LDAP interface.
   - The Users tab is no longer available under the Security Center because users are now managed through your LDAP interface.
   - Roles are not updatable in the BI Publisher interface, except for adding data sources.
3. Click **Add Data Sources** to add BI Publisher data sources to the role. A role must be assigned access to a data source to run reports from that data source or to build data models from the data source. For more information see Section 3.8.4, "Granting Data Access."
4. Grant catalog permissions to roles. See Section 3.5, "About Catalog Permissions" and Section 3.8.3, "Granting Catalog Permissions" for details on granting catalog permissions to roles.

Users can now log in using their LDAP username/password.

3.10.3 Disable Users Without BI Publisher-Specific Roles from Logging In

To disable users without BI Publisher-specific roles from logging in to the BI Publisher server, set a configuration property in the xmlp-server-config.xml file located at:

```
<repository>/Admin/Configuration/xmlp-server-config.xml
```

In the xmlp-server-config.xml file, add the following property and setting:

```
<property name="REQUIRE_XMLP_ROLE_FOR_LOGIN" value="true"/>
```

3.11 Integrating with Microsoft Active Directory

Microsoft Active Directory supports the LDAP interface and therefore can be configured with BI Publisher using LDAP Security.

3.11.1 Configuring the Active Directory

To configure the active directory:

1. Add users who must access BI Publisher.

   Add the users under "Users" or any other organization unit in the Domain Root.

2. Add the BI Publisher system groups. The Scope of the groups must be Domain Local.

   Table 3–6 describes the BI Publisher system groups that must be added.

Table 3–6 BI Publisher System Groups

BI Publisher System Group	Description
XMLP_ADMIN	The administrator role for the BI Publisher server. You must assign the Administrator account used to access your LDAP server the XMLP_ADMIN group.
XMLP_DEVELOPER	Allows users to create and edit reports and data models.
XMLP_SCHEDULER	Allows users to schedule reports.
XMLP_TEMPLATE_DESIGNER	Allows users to connect to the BI Publisher server from the Template Builder for Word and to upload and download templates. Allows users to design layouts using the BI Publisher Layout Editor.

3. Grant BI Publisher system groups to global groups or users.

   You can grant BI Publisher system groups directly to users or through global groups.

Example 1: Grant Users the BI Publisher Administrator Role

1. Under the **Active Directory User and Computers**, open the XMLP_ADMIN group and click the **Members** tab.
2. Click **Add** to add users who need to BI Publisher Administrator privileges.

Example 2: Grant Users Access to Scheduling Reports

The "HR Manager" global group is defined under "Users".

All users in this group need to schedule reports.

To achieve this, add "HR Manager" as a Member of the XMLP_SCHEDULER group.

3.11.2 Configuring BI Publisher

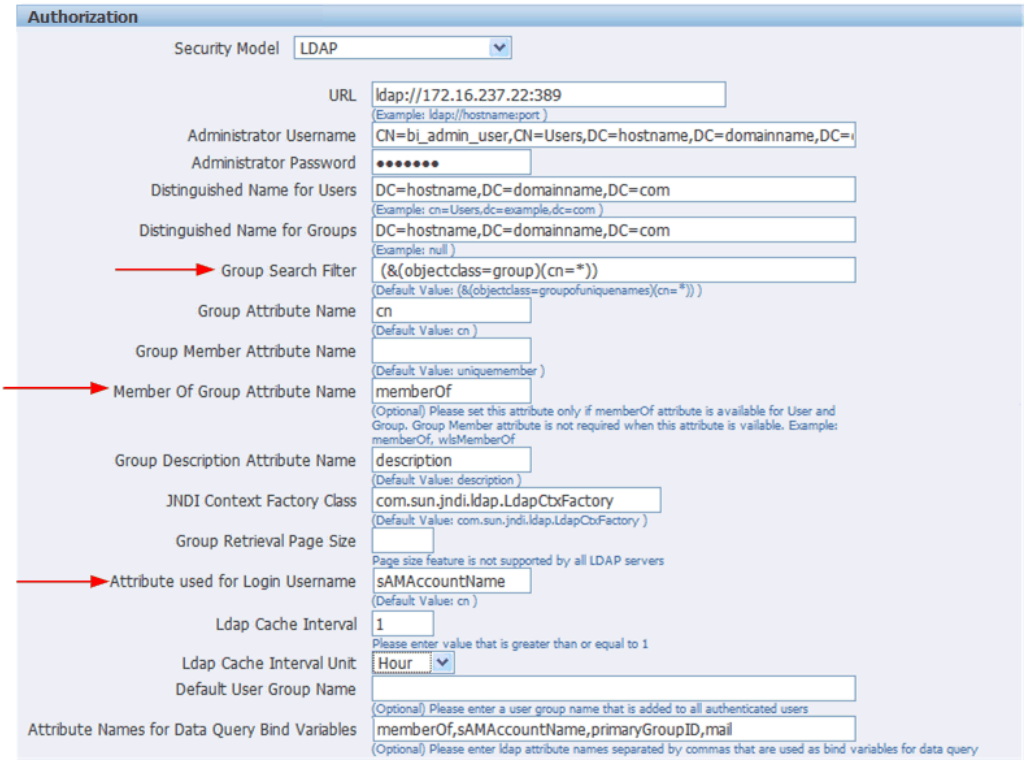
To configure BI Publisher:

1. On the **Administration** page, click **Security Configuration**.
2. Set up a Local Superuser if one has not been configured. This is very important in case the security configuration fails, you must still be able to log in to BI Publisher using the Superuser credentials.
3. In the **Authorization** region of the page, select LDAP from the **Security Model** list.
4. Enter the details for the Active Directory server, as described in Section 3.10.2.2, "Configure the BI Publisher Server to Recognize the LDAP Server," noting the following specific information for Active Directory:
   - Set **Group Search Filter** objectclass to "group"
   - Set **Member of Group Member Attribute Name** to "memberOf" (**Group Member Attribute Name** can be left blank).
   - Set **Attribute used for Login Username** to "sAMAccountName".
   - If your are using LDAP over SSL note the following:
     - the protocol is "ldaps"
     - the default port is 636

     An example URL would be: ldaps://example.com:636/

   Figure 3–7 shows an example configuration highlighting the recommendations stated above.

Figure 3–7 Example Configuration



5. Click **Apply**. Restart the BI Publisher application.

If you are configuring BI Publisher to use LDAP over SSL, then you must also configure Java keystore to add the server certificate to JVM. For more information, see Section 4.3, "Configuring BI Publisher for Secure Socket Layer (SSL) Communication."

3.11.3 Logging In to BI Publisher Using the Active Directory Credentials

The User login name defined in **Active Directory Users and Computers** >**User Properties** >**Account** is used for the BI Publisher login name. Add the Domain to the user name to log in to BI Publisher. For example: "scott_tiger@domainname.com".

Note the following:

- The **Attribute used for Login Username** can be sAMAccountName instead of userPrincipalName.
- You must use sAMAccountName for the **Attribute used for Login Username** when the "User logon name (pre-Windows 2000)" is required to use for the BI Publisher login username.
- User names must be unique across all organization units.

3.11.4 Assign Data Access and Catalog Permissions to Roles

To assign data access and catalog permissions to roles:

1. Log in to BI Publisher as a user assigned the XMLP_ADMIN role in Active Directory.
2. On the **Administration** page click **Roles and Permissions**.

You see the roles that you created in Active Directory to which you assigned the XMLP_ roles. Note the following:

- The XMLP_X roles are not shown because these are controlled through the Active Directory interface.
- The Users tab is no longer available under the Security Center because users are now managed through Active Directory.
- Roles are not updatable in the BI Publisher interface, except for adding data sources.

3. Click **Add Data Sources** to add BI Publisher data sources to the role. A role must be assigned access to a data source to run reports from that data source or to build data models from the data source. For more information see Section 3.8.4, "Granting Data Access."
4. Grant catalog permissions to roles. See Section 3.5, "About Catalog Permissions" and Section 3.8.3, "Granting Catalog Permissions" for details on granting catalog permissions to roles.

3.12 Configuring BI Publisher with Single Sign-on (SSO)

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. Oracle Business Intelligence can be configured to trust incoming HTTP requests authenticated by a SSO solution that is configured for use with Oracle Fusion Middleware and Oracle WebLogic Server. For more information about configuring SSO for Oracle Fusion Middleware, see "Configuring Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*.

When BI Publisher 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 BI Publisher challenges each user for authentication credentials. When BI Publisher is configured to use SSO, a user is first redirected to the SSO solution's login page for authentication.

Configuring BI Publisher 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.
- BI Publisher is configured to trust incoming messages.
- The HTTP header information required for identity propagation with SSO configurations (namely, user identity and SSO cookie) is specified and configured.

3.12.1 How BI Publisher Operates with SSO Authentication

After SSO authorization has been implemented, BI Publisher operates as if the incoming web request is from a user authenticated by the SSO solution. User personalization and access controls such as data-level security are maintained in this environment.

3.12.2 Tasks for Setting Up SSO Authentication with BI Publisher

Table 3–7 contains SSO authentication configuration tasks and provides links for obtaining more information.

Table 3–7 Task Map: Configuring SSO Authentication for BI Publisher

Task	Description	For More Information
Configure Oracle Access Manager as the SSO authentication provider.	Configure Oracle Access Manager to protect the BI Publisher URL entry points.	Section 3.13, "Configuring SSO in an Oracle Access Manager Environment" Also see: "Configuring Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*
Configure the HTTP proxy.	Configure the web proxy to forward requests from BI Publisher to the SSO provider.	"Configuring Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*
Configure a new authenticator for Oracle WebLogic Server.	Configure the Oracle WebLogic Server domain in which BI Publisher is installed to use the new identity store.	Section 3.13.1, "Configuring a New Authenticator for Oracle WebLogic Server" Also see: *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*
Configure a new identity asserter for Oracle WebLogic Server.	Configure the Oracle WebLogic Server domain in which BI Publisher is installed to use the SSO provider as an asserter.	Section 3.13.2, "Configuring OAM as a New Identity Asserter for Oracle WebLogic Server" Also see: *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*
Configure the new trusted system user to replace the default BISystemUser.	Add the new trusted system user name from Oracle Internet Directory to become a member of the BISystem application role.	See "Configuring a New Trusted User (BISystem User)" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
Enable BI Publisher to accept SSO authentication.	Enable the SSO provider configured to work with BI Publisher.	Section 3.13.3, "Configuring BI Publisher for Oracle Fusion Middleware Security"

Note: For an example of an Oracle Business Intelligence SSO installation scenario, see *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence*.

3.13 Configuring SSO in an Oracle Access Manager Environment

For information about how to configure Oracle Access Manager as the SSO authentication provider for Oracle Fusion Middleware with WebLogic Server, see "Configuring Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*. For more information about managing Oracle Access Manager, see *Oracle Fusion Middleware Administrator's Guide for Oracle Access Manager*.

After the Oracle Fusion Middleware environment is configured, in general the following must be done to configure BI Publisher:

- Configure the SSO provider to protect the BI Publisher URL entry points.
- Configure the web server to forward requests from BI Publisher to the SSO provider.
- Configure the new identity store as the main authentication source for the Oracle WebLogic Server domain in which BI Publisher has been installed. For more information, see Section 3.13.1, "Configuring a New Authenticator for Oracle WebLogic Server.".
- Configure the Oracle WebLogic Server domain in which BI Publisher is installed to use an Oracle Access Manager asserter. For more information, see Section 3.13.2, "Configuring OAM as a New Identity Asserter for Oracle WebLogic Server.".
- After configuration of the SSO environment is complete, enable SSO authentication for BI Publisher. For more information, see Section 3.13.3, "Configuring BI Publisher for Oracle Fusion Middleware Security.".

3.13.1 Configuring a New Authenticator for Oracle WebLogic Server

After installing BI Publisher, the Oracle WebLogic Server embedded LDAP server is the default authentication source (identity store). To use a new identity store (for example, OID), as the main authentication source, you must configure the Oracle WebLogic Server domain (where BI Publisher is installed).

For more information about configuring authentication providers in Oracle WebLogic Server, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

To configure a new authenticator in Oracle WebLogic Server:

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 **Providers** tab, then display the **Authentication** sub-tab.
4. Click **New** to launch the **Create a New Authentication Provider** page.

   Complete the fields as follows:

   - **Name**: *OID Provider*, or a name of your choosing.
   - **Type**: OracleInternetDirectoryAuthenticator
   - Click **OK** to save the changes and display the authentication providers list updated with the new authentication provider.
5. Click the newly added authenticator in the **authentication providers** table.
6. Navigate to **Settings**, then select the **Configuration\Common** tab:

   - Select **SUFFICIENT** from the **Control Flag** list.
   - Click **Save**.
7. Display the **Provider Specific** tab and specify the following settings using appropriate values for your environment:

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*. **Note**: 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

For more information about configuring authentication providers in Oracle WebLogic Server, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

8. Click **Save**.
9. 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** sub-tab, then select **DefaultAuthenticator** to display its configuration page.
   b. Display the **Configuration\Common** tab and select 'SUFFICIENT' from the **Control Flag** list.
   c. Click **Save**.
10. 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.

11. In the Change Center, click **Activate Changes**.
12. Restart Oracle WebLogic Server.

3.13.2 Configuring OAM as a New Identity Asserter for Oracle WebLogic Server

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

For more information about creating a new asserter in Oracle WebLogic Server, see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

To configure Oracle Access Manager as the new asserter for Oracle WebLogic Server:

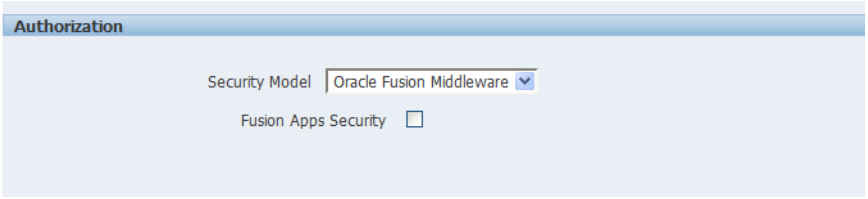
1. Log in to Oracle WebLogic Server Administration Console.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**. Select **Providers**.
3. Click **New**. Complete the fields as follows:
   - **Name**: *OAM Provider*, or a name of your choosing.
   - **Type**: OAMIdentityAsserter.
4. Click **OK**.
5. Click **Save**.
6. 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 use the arrows beside the list to order the providers as follows:
      - OID Authenticator (SUFFICIENT)
      - OAM Identity Asserter (REQUIRED)
      - Default Authenticator (SUFFICIENT)
   c. Click **OK** to save your changes.
7. In the Change Center, click **Activate Changes**.
8. 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.
9. Use Fusion Middleware Control to enable SSO authentication.

3.13.3 Configuring BI Publisher for Oracle Fusion Middleware Security

After Oracle WebLogic Server has been configured, navigate to the BI Publisher Administration **Security Configuration** page. In the Authorization region select Oracle Fusion Middleware as the Security Model as shown in the following figure:

Figure 3–8 Fusion Middleware Security Selection



3.14 Setting Up Oracle Single Sign-On

To set up Oracle Single Sign-On, first configure WebLogic Server using the instructions in the chapter, "Configuring Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*. BI Publisher must be configured to use Oracle Internet Directory as the default LDAP server.

Note: When using Oracle SSO, BI Publisher assumes that a login user name can be derived from Osso-User-Dn, which is HTTP Header value. For example, if the Osso-User-Dn on HTTP Header looks like this:

cn=admin,cn=users, dc=us,dc=oracle,dc=com

Then BI Publisher assumes the value of first cn= is the login user name (that is, "admin" in this case).

Therefore if your Osso-User-Dn does not contain a login user name as the first cn value, then select "Other SSO Type" to configure the settings (even if you use Oracle SSO).

3.14.1 Setup Procedure

To set up SSO:

1. Modify the application server configuration file to protect the xmlpserver. See the section "Configuring mod_osso to Protect Web Resources" in *Oracle Fusion Middleware Application Security Guide*.
2. In the mod_osso.conf add a new "Location" directive as follows:

```
<!-- Protect xmlpserver -->
<Location /xmlpserver>
      require valid-user
      AuthType Basic
</Location>
```

3. To allow Web service communication between BI Publisher and its client component (the Template Builder) you must make additional modifications to the mod_osso.conf file. To open up the xmlpserver to allow these Web services, enter the following directives:

```
<Location /xmlpserver/services/>
  require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>

<Location /xmlpserver/report_service/>
  require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>

Location /xmlpserver/ReportTemplateService.xls/>
  require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>
```

4. For integration with Oracle BI Presentation Services, you must disable SSO for Web services between the BI Presentation Services server and the BI Publisher server. If you made this entry when performing the previous step, then you do not need to repeat this setup.

   To open up the xmlpserver to allow the Web service, enter the following directive in the mod_osso.conf file:

```
<Location /xmlpserver/services/>
 require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>
```

   A sample mod_osso.conf file with the entries discussed in this section is shown below:

```
LoadModule osso_module libexec/mod_osso.so

 <IfModule mod_osso.c>
     OssoIpCheck off
     OssoIdleTimeout off
     OssoConfigFile /home/as1013/ohome/Apache/Apache/conf/osso/osso.conf

   <Location /xmlpserver>
     require valid-user
     AuthType Basic
   </Location>


<Location /xmlpserver/services/>
  require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>
```

```
<Location /xmlpserver/report_service/>
  require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>

Location /xmlpserver/ReportTemplateService.xls/>
  require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>

<Location /xmlpserver/Guest/>
 require valid-user
     AuthType Basic
     Allow from All
     Satisfy any
</Location>
 #
 # Insert Protected Resources: (see Notes below for how to protect resources)
 #


 #______-
 #
 # Notes
 #
 #______-
 #
 # 1. Here's what you need to add to protect a resource,
 #    e.g. <ApacheServerRoot>/htdocs/private:
 #
 #      <Location /private>
 #      require valid-user
 #      AuthType Basic
 #      </Location>
 #
 </IfModule>

 #
 # If you would like to have short hostnames redirected to
 # fully qualified hostnames to allow clients that need
 # authentication through mod_osso to be able to enter short
 # hostnames into their browsers uncomment out the following
 # lines
 #
 #PerlModule Apache::ShortHostnameRedirect
 #PerlHeaderParserHandler Apache::ShortHostnameRedirect
```

5. Restart the HTTP server.
6. In BI Publisher: Set up the Single Sign-Off URL on the BI Publisher Security Configuration page.

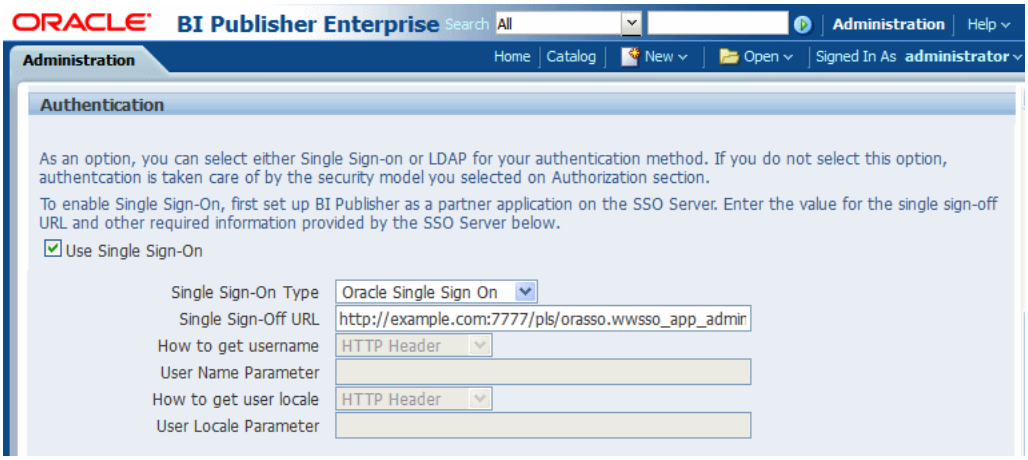
   On the **Administration** page, click **Security Configuration**. In the **Authentication** region:

   - Select **Use Single Sign-On**

- From the **Single Sign-On Type** list, select **Oracle Single Sign On**
- Enter the **Single Sign-Off URL** with the value you wrote down in the preceding step. The remaining fields are not applicable to Oracle SSO.

  A sample BI Publisher Security Configuration page is shown in Figure 3–9.

Figure 3–9 Sample BI Publisher Security Configuration Page



7. Create a BI Publisher Local Superuser to ensure access to BI Publisher regardless of your selected security configuration. See Section 4.1, "Enabling a Local Superuser" for more information.
8. Click **Apply**. Restart the application through the Oracle Fusion Middleware Control page.
9. Enter the URL to access the BI Publisher Enterprise application, and you are redirected to the SSO login page.

4

Other Security Topics

This chapter describes additional BI Publisher security topics including SSL configuration, proxy settings, enabling a local superuser, and enabling a guest user.

It covers the following topics:

4.1 Enabling a Local Superuser

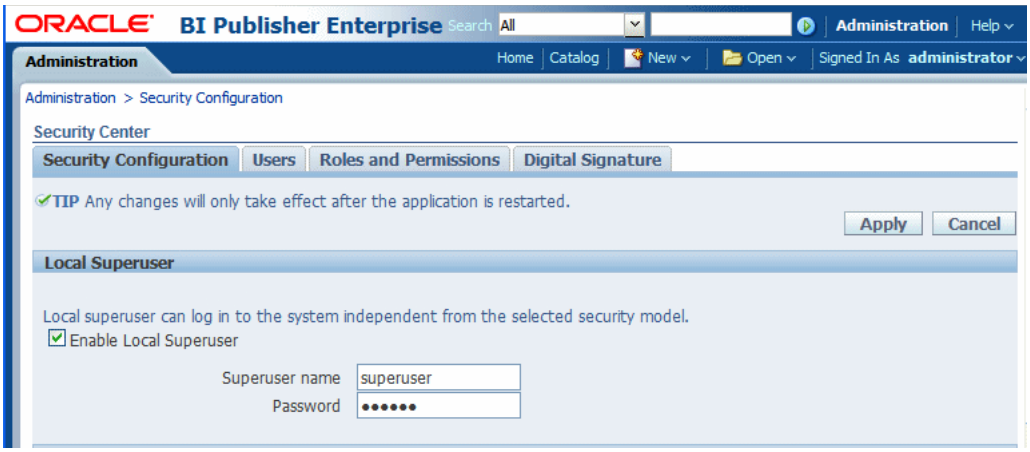
BI Publisher enables you to define an administration Superuser. Using the Superuser credentials you can directly access the BI Publisher administrative functions without logging in through the defined security model.

Set up this Superuser to ensure access to all administrative functions in case of failures with the configured security model. It is highly recommended that you set up a Superuser.

To enable a local superuser:

1. Click **Administration**.
2. Under **Security Center** click **Security Configuration**.
3. Under **Local Superuser**, select the box and enter the credentials for the Superuser, as shown in Figure 4–1.

Figure 4–1 Superuser Credentials



4. Restart the BI Publisher application.

4.2 Enabling a Guest User

BI Publisher allows you configure public access to specific reports by defining a "Guest" folder. Any user can access the reports in this folder without entering credentials.

Note: Guest access is not supported when BI Publisher uses a shared catalog or is installed with Oracle Business Intelligence Enterprise Edition.

Guest access is not supported with Single Sign-On.

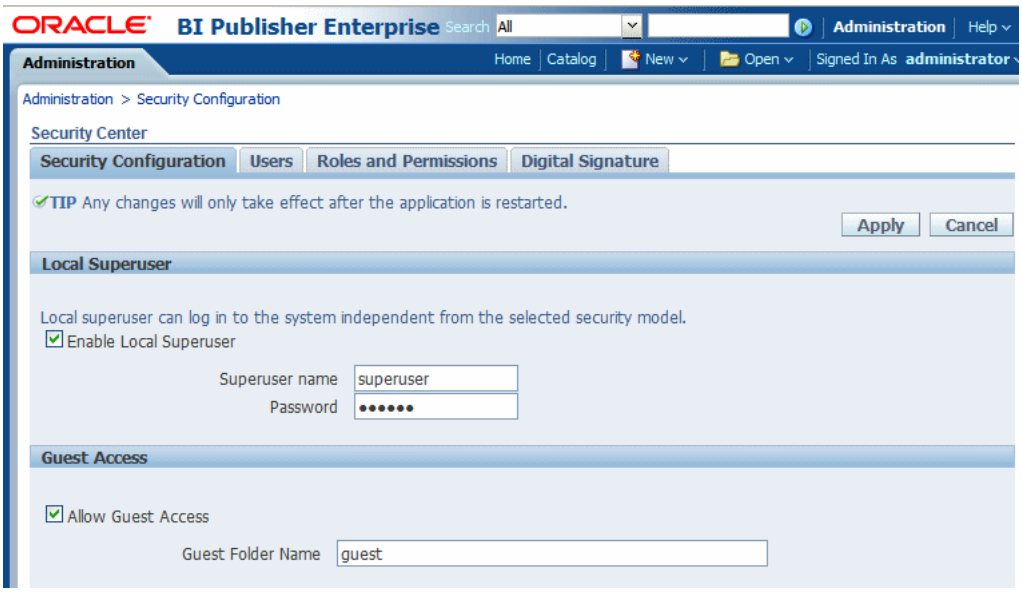
All objects that are required to view a report must be present in the Guest folder because the Guest folder is the only folder the guest user has any access rights to. Therefore the report and the data model must be present in the Guest folder and Sub Templates and Style Templates, if applicable. The guest user must have read access only.

The Guest user must also be granted access to the report data source.

To enable guest access:

1. Under Shared Folders, create the folder to which you want to grant public access.
2. Click **Administration**.
3. Under **Security Center** select **Security Configuration**.
4. Under **Guest Access**, select **Allow Guest Access**.
5. Enter the name of the folder that you created for public access. as shown in Figure 4–2.

Figure 4–2 Public Access Folder



6. Restart the BI Publisher application.
7. Add the objects to the Guest folder that the guest users can access: folders, reports, data models, Sub Templates and Style Templates.

   Note: The report must reference the data model that is stored in the guest folder. Therefore, if you copy a report with its data model from another location, then ensure that you open the report and reselect the data model so that the report references the data model inside the guest folder.

   Similarly, any references to Sub Templates or Style Templates must also be updated.
8. Grant access to the data sources used by data models in your Guest folder. See Chapter 9, "Setting Up Data Sources" for information on granting Guest access to a data source.

Users who access BI Publisher see the Guest button on the log on page. Users can select this button and view the reports in your chosen guest folder without presenting credentials.

4.3 Configuring BI Publisher for Secure Socket Layer (SSL) Communication

If BI Publisher is communicating with other applications over SSL you might need to perform additional configuration to ensure operability.

Note: It is strongly recommended that you enable Secure Socket Layer (HTTPS) on the middle tier hosting the Web services because the trusted username/password that is passed can be intercepted. This also pertains to Web services that are used for communication between BI Publisher and Oracle BI Presentation Services.

- Point BI Publisher to the System-Wide Keystore
- Import certificates to the Java keystore
- Configure the Delivery Manager

4.3.1 Pointing BI Publisher to the System-Wide Keystore

By default, BI Publisher uses the Java keystore: {java.home}/lib/security/cacerts

If you are using a different location as your keystore, then set the JAVA_OPTS environment variable for your Web server to tell the BI Publisher server where to find the keystore, as follows:

```
set JAVA_OPTS=-Djavax.net.ssl.trustStore=<keystore file>
```

4.3.2 Importing Certificates for Web Services Protected by SSL

If you make calls to Web services that are protected through Secure Sockets Layer (SSL), then you must export the certificate from the Web server hosting the Web service and import it into the Java keystore on the computer that is running BI Publisher.

To import certificates for Web services:

1. Navigate to the HTTPS site where the WSDL resides.
2. Download the certificate by following the prompts; the prompts that you see vary depending on the browser that you are using.
3. Install the Certificate into your keystore using the Java keytool, as follows:

   ```
   keytool -import -file <certfile> -alias <certalias> -keystore <keystore file>
   ```

4. Restart the application server.

These steps should not be required if the server certificate is linked to some certificate authority (such as Verisign). But if the Web service server is using a self-generated certificate (for example, in a testing environment), then these steps are required.

4.3.3 Configuring the Delivery Manager

If you want to use the default certificates built-in with BI Publisher, then no further configuration is required. SSL works with the default certificate if the server uses the certificate signed by a trusted certificate authority such as Verisign.

If the user uses the SSL with a self-signed certificate, then the certificate information must be entered in the Delivery Configuration page. A self-signed certificate means that the certificate is signed by a non-trusted certificate authority (usually the user).

4.4 Configuring Proxy Settings

To use external Web Services or HTTP data sources when the BI Publisher server is configured behind a firewall or requires a proxy to access the internet, you must configure Oracle WebLogic Server to allow the Web service requests and to be aware of the proxy. When configuring the proxy setting, you must also configure WebLogic Server to be aware of any hosts that BI Publisher must connect to directly (not through the proxy) for example, the Oracle BI Enterprise Edition host.

Define the proxy host and the non-proxy hosts to WebLogic Server by setting the following parameters:

- `-Dhttp.proxyHost` - specifies the proxy host. For example:

  -Dhttp.proxyHost=www-proxy.example.com

- `-Dhttp.proxyPort` - specifies the proxy host port. For example:

  -Dhttp.proxyPort=80

- `-Dhttp.nonProxyHosts` - specifies the hosts to connect to directly, not through the proxy. Specify the list of hosts, each separated by a "|" character; a wildcard character (*) can be used for matching. For example:

  -Dhttp.nonProxyHosts="localhost|*.example1.com|*.example2.com

To set these proxy parameters and the Web service configuration for your WebLogic Server add the following to the WebLogic setDomainEnv script as follows:.

1. Open the setDomainEnv script (.sh or .bat) in the MW_HOME/user_projects/domains/DOMAIN_NAME/bin/ directory.
2. Enter the following parameters:

```
EXTRA_JAVA_PROPERTIES="-Dhttp.proxyHost=www-proxy.example.com
-Dhttp.proxyPort=80
-Dhttp.nonProxyHosts=localhost|*.mycompany.com|*.mycorporation.com|*.otherhost.
com ${EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES


EXTRA_JAVA_
PROPERTIES="-Djavax.xml.soap.MessageFactory=oracle.j2ee.ws.saaj.soap.MessageFac
toryImpl
-Djavax.xml.soap.SOAPFactory=oracle.j2ee.ws.saaj.SOAPFactoryImpl
-Djavax.xml.soap.SOAPConnectionFactory=oracle.j2ee.ws.saaj.client.p2p.HttpSOAPC
onnectionFactory  ${EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES
```

   where

   `www-proxy.example.com` is an example proxy host

   `80` is the example proxy port

   `localhost|*.mycompany.com|*.mycorporation.com|*.otherhost.com` are example non-proxy hosts

5

Integrating with Other Oracle Security Models

This chapter describes BI Publisher support for security models of other Oracle products including Oracle E-Business Suite security, Oracle Database security, and Oracle Siebel CRM security.

It covers the following topics:

5.1 About Integrating with Other Oracle Security Models

This chapter describes how to integrate BI Publisher with other Oracle product security models. In most cases you must first define the BI Publisher functional roles in the other Oracle product and then configure BI Publisher to use the other Oracle product security for authorization. You can use one of the Oracle product authorization methods described here in conjunction with a supported authentication method (SSO or LDAP) described in Chapter 3, "Alternative Security Options."

For conceptual information regarding BI Publisher roles and permissions, see Section 3.3, "Understanding BI Publisher's Users, Roles, and Permissions."

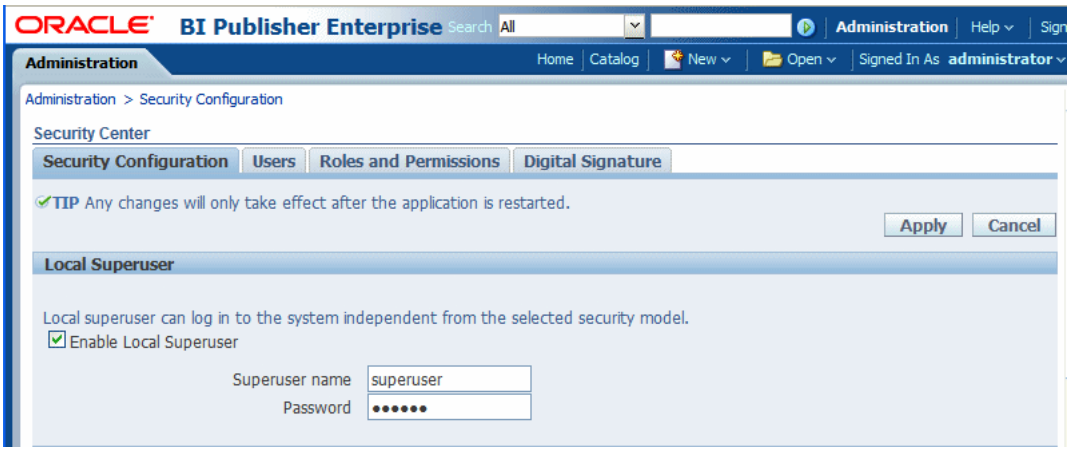
5.2 Before You Begin: Create a Local Superuser

Before you implement any of these security models, first create a local superuser. The local superuser credentials ensure that you can access the Administration pages of Oracle BI Publisher in case of any unexpected failures in the configured security settings.

To create a local superuser:

1. On the **Administration** page, click **Security Configuration**.
2. On the **Security Configuration** tab, under the **Local Superuser** region, select **Enable Local Superuser**, as shown in Figure 5–1.

Figure 5–1 Enabling Local Superuser



3. Enter a name and password for your superuser.
4. Restart BI Publisher for the Superuser to become activated in the system.

5.3 Integrating with Oracle BI Server Security

If you have installed BI Publisher as part of the Oracle Business Intelligence Enterprise Edition and you have configured Oracle BI Enterprise Edition to use legacy Oracle BI Server authentication, then follow these procedures to configure BI Publisher to use BI Server security:

- Section 5.3.1, "Configuring BI Publisher for Oracle BI Server Security"
- Section 5.3.2, "Adding Data Sources to BI Server Roles"

> **Note:** The Oracle BI Server security option is for customers who want to use legacy 10*g* authentication. This section does not apply to you if you have configured Oracle Fusion Middleware Security.

These procedures assume that you have performed the configuration required in the BI Server. For information on configuring legacy Oracle BI security, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*

5.3.1 Configuring BI Publisher for Oracle BI Server Security

To configure BI Publisher for BI Server Security:

1. Log in to BI Publisher with administrator credentials. Navigate to the BI Publisher Administration page. On the Administration page click **Security Configuration**.

> **Note:** To log in directly to the BI Publisher server, use the login URL with the /xmlpserver suffix, for example: http://example.com:9704/xmlpserver

2. In the **Authorization** region of the page, select Oracle BI Server from **Security Model** list. Provide the following connection information for the Oracle BI Server:
   - JDBC **Connection String** — Example: jdbc:oraclebi://host:port/

If you do not know the connection string to the BI Server, then you can copy it from data source connection page. From the **Administration** page, under **Data Sources**, click **JDBC Connection**. Locate the Oracle BI EE server and copy the connection string. If this has not been configured, then see Section B.4, "Setting Up a JDBC Connection to the Oracle BI Server."

- **Administrator Username** and **Administrator Password**

  Enter the BISystemUser and password.

- **Database Driver Class** — Example: oracle.bi.jdbc.AnaJdbcDriver

3. Click **Apply**. Restart the BI Publisher application for the security changes to take effect.

5.3.2 Adding Data Sources to BI Server Roles

To add data sources to BI Server roles:

1. Log in to Oracle Business Intelligence as an administrator.
2. On the global header click **Administration**. On the Oracle BI Administration page, click **Manage BI Publisher**.
3. On the BI Publisher **Administration** page click **Roles and Permissions**. The groups to which you assigned the BI Publisher groups are displayed as available roles.
4. Find the group (role) to add data sources to and click **Add Data Sources**.

   Alternatively, you can navigate to the data source and add the roles that require access to the data source.
5. Locate the appropriate data sources in the **Available Data Sources** list and use the shuttle buttons to move the sources to the **Allowed Data Sources** list for the role.
6. Click **Apply**.
7. Repeat for all roles that need access to report data sources.

5.4 Integrating with Oracle E-Business Suite

BI Publisher can leverage your E-Business Suite security to enable your users to log in to BI Publisher using their E-Business Suite credentials. The BI Publisher security integration recognizes the user's E-Business Suite responsibility and org_id combinations.

When users log in they are prompted to select a responsibility. Reports that users run against the E-Business Suite data tables then filter the data based on their responsibility and org_id combination. Users can switch responsibilities and reporting organization while still logged in using the **My Account** dialog.

When you integrate with the E-Business Suite security, your E-Business Suite responsibilities appear as roles in the BI Publisher security center. You can then add BI Publisher catalog permissions and data access privileges to the imported roles/responsibilities. See Section 3.3, "Understanding BI Publisher's Users, Roles, and Permissions."

Follow these procedures to integrate BI Publisher with Oracle E-Business Suite:

- Section 5.4.2, "Configuring BI Publisher to Use E-Business Suite Security"
- Section 5.4.3, "Adding Data Sources to the E-Business Suite Roles"

- Section 5.4.4, "Granting Catalog Permissions to the E-Business Suite Roles"

> **Note:** In this release, users cannot access or execute reports that are stored on the E-Business Suite instance. Reports must reside in the BI Publisher catalog. The E-Business Suite data security is enforced when BI Publisher connects to the E-Business Suite data tables to retrieve the report data.

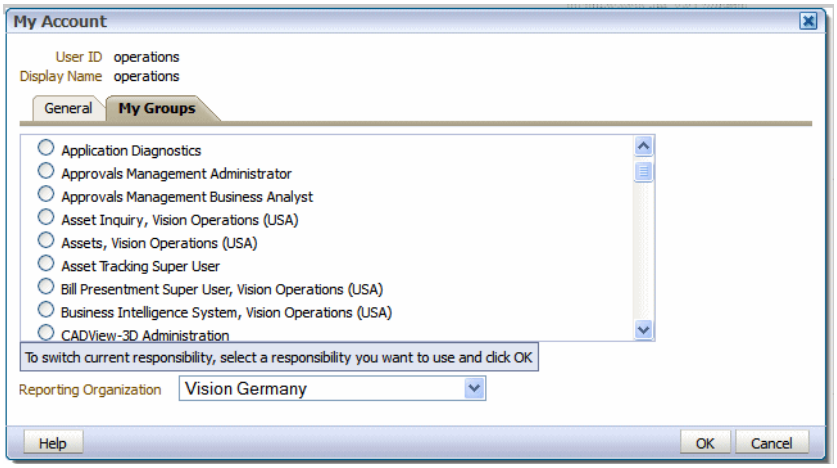
Oracle BI Publisher relies on information stored in the DBC file to connect to the E-Business Suite instance. Ensure that you can locate and have access to this file. The DBC file is typically located under the $FND_SECURE directory.

5.4.1 Features of the Integration with E-Business Suite Security

When BI Publisher is integrated with E-Business Suite security, the following features are enabled:

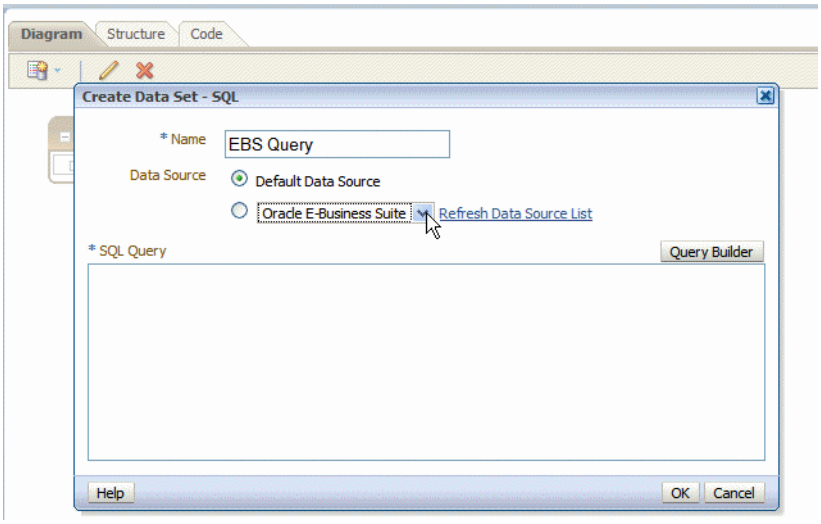
- When users log in to BI Publisher using their E-Business Suite credentials, they are prompted to choose a responsibility, as shown in Figure 5–2.

Figure 5–2 Selecting a Responsibility at Log In



- Users can switch responsibilities or reporting organizations using the **My Account** dialog.
- The data source connection to the E-Business Suite instance is automatically configured and available in the data model editor, as shown in Figure 5–3.

Figure 5–3 E-Business Suite Data Source Option in Data Model Editor



5.4.2 Configuring BI Publisher to Use E-Business Suite Security

To configure BI Publisher for E-Business Suite Security:

1. In the Oracle E-Business Suite, log in as a System Administrator and create the following responsibilities to correspond to the BI Publisher functional roles:
   - XMLP_ADMIN — Serves as the administrator role for the BI Publisher server.
   - XMLP_DEVELOPER — Allows users to build reports in the system.
   - XMLP_SCHEDULER — Allows users to schedule reports.
   - XMLP_TEMPLATE_DESIGNER — Allows users to connect to the BI Publisher server from the Template Builder and to upload and download templates. Allows users to design layouts using the BI Publisher Layout Editor.
2. Add these new BI Publisher responsibilities to the appropriate users.

   Note: Ensure that you assign at least one user to the XMLP_ADMIN group.

3. Log in to Oracle BI Publisher. On the Administration page, select **Security Configuration**.
4. In the **Authorization** region of the page, select Oracle E-Business Suite from the **Security Model** list.
5. Load the DBC file from the E-Business Suite instance. This is typically located under the $FND_SECURE directory. If you do not have access to this file, then contact your E-Business Suite system administrator. This file specifies how BI Publisher should access the E-Business Suite instance.
6. Click **Apply**. Restart BI Publisher for the security changes to take effect.

When you restart the system, the E-Business Suite responsibilities to which BI Publisher roles have been assigned are visible as roles in the BI Publisher security center.

5.4.3 Adding Data Sources to the E-Business Suite Roles

To view a report generated from a particular data source, a report consumer's role must be granted access to the data source. Similarly, to create a data model based on a particular data source, the report author's role must be granted access to the data source.

To grant a role access to a data source:

1. On the **Administration** tab, under **Security Configuration**, click **Roles and Permissions**. The responsibilities that are assigned BI Publisher roles in the E-Business Suite instance are displayed as available roles.
2. Find the role to which you want to add data sources and click **Add Data Sources**. The **Add Data Sources** page is displayed.
3. Locate the appropriate data sources in the **Available Data Sources** list and use the shuttle buttons to move the sources to the **Allowed Data Sources** list for the role.
4. Click **Apply**.
5. Repeat for all roles that need access to report data sources.

5.4.4 Granting Catalog Permissions to the E-Business Suite Roles

For a role to access objects in a folder, you must grant the role permissions to the catalog object. You can grant permissions at the folder level, so that a role has the same access to every object in a folder, or you can assign access individually to each object in a folder.

See the following sections for more information:

- Section 3.3, "Understanding BI Publisher's Users, Roles, and Permissions"
- Section 3.4, "About Privileges to Use Functionality"
- Section 3.5, "About Catalog Permissions"
- Section 3.6, "How Functional Privileges and Permissions Work Together"

To grant permissions to E-Business Suite roles:

1. In the catalog, navigate to a catalog object required for a role.
2. Click the **More** link for the object and then click **Permissions** to open the **Permissions** dialog.
3. Click the **Create** icon to open the **Add Roles** dialog.
4. Click **Search** to populate the list of **Available Roles**.
5. Use the **Move** button to move the appropriate roles from the **Available Roles** list to the **Selected Roles** list.
6. Click **OK**.
7. Enable the appropriate permissions for the role by selecting the check boxes.
8. If you have selected a folder: To apply the selections to all items within a folder, select **Apply permissions to items within this folder**.

5.5 Integrating with Oracle Database Security

BI Publisher offers integration with Oracle Database security to enable you to administer the BI Publisher users with your Oracle Database users. Follow these procedures to integrate BI Publisher with Oracle E-Business Suite:

- Section 5.5.1, "Defining the BI Publisher Functional Roles in the Oracle Database"
- Section 5.5.2, "Adding Data Sources to Roles"
- Section 5.5.3, "Granting Catalog Permissions to Roles"

> **Note:** For information on setting up Oracle Database security, see the *Oracle Database Security Guide*.

When you restart the server, the roles to which BI Publisher roles have been assigned are visible as roles in the BI Publisher security center.

5.5.1 Defining the BI Publisher Functional Roles in the Oracle Database

To define the BI Publisher functional roles in the Oracle Database:

1. In the Oracle Database, create the following roles to correspond to the BI Publisher functional roles:
   - XMLP_ADMIN — Serve as the administrator role for the BI Publisher server.
   - XMLP_DEVELOPER — Allows users to build reports in the system.
   - XMLP_SCHEDULER — Allows users to schedule reports.
   - XMLP_TEMPLATE_DESIGNER — Allows users to connect to the BI Publisher server from the Template Builder and to upload and download templates.
2. Assign these roles to the appropriate Database roles and users. You might also want to create additional reporting roles that you can use when setting up your report privileges on the BI Publisher side. For example, you might create a role called "HUMAN_RESOURCES_MANAGER" that you can assign a Human Resources Folder of reports to. You can then assign that role to any user requiring access to the Human Resources reports.
3. Ensure to assign the XMLP_ADMIN role to a user with administration privileges, such as SYSTEM.
4. Log in to BI Publisher application with Administrator privileges. From the Administration page, select **Security Configuration**.
5. In the **Authorization** region of the page, select Oracle Database from the **Security Model** list. Provide the following connection information:
   - JDBC **Connection String** — Example: jdbc:oracle:thin:@mycompany.com:1521:orcl
   - **Administrator Username** and **Administrator Password** — Note the following requirements for this user:
     - The user must be granted the XMLP_ADMIN role
     - The user must have privileges to access data from the dba_users/_roles/_role_privs tables.
   - **Database Driver Class** — Example: oracle.jdbc.driver.OracleDriver

6. Click **Apply**. Restart BI Publisher for the security changes to take effect.

5.5.2 Adding Data Sources to Roles

To view a report generated from a particular data source, a report consumer's role must be granted access to the data source. Similarly, to create a data model based on a particular data source, the report author's role must be granted access to the data source.

To grant a role access to a data source:

1. On the **Administration** tab, under **Security Configuration**, click **Roles and Permissions**.
2. Find the role to which you want to add data sources and click **Add Data Sources**. The **Add Data Sources** page is displayed.
3. Locate the appropriate data sources in the **Available Data Sources** list and use the shuttle buttons to move the sources to the **Allowed Data Sources** list for the role.
4. Click **Apply**.
5. Repeat for all roles that need access to report data sources.

5.5.3 Granting Catalog Permissions to Roles

For a role to access objects in a folder, you must grant the role permissions to the catalog object. You can grant permissions at the folder level, so that a role has the same access to every object in a folder, or you can assign access individually to each object in a folder.

See the following sections for more information:

- Section 3.3, "Understanding BI Publisher's Users, Roles, and Permissions"
- Section 3.4, "About Privileges to Use Functionality"
- Section 3.5, "About Catalog Permissions"
- Section 3.6, "How Functional Privileges and Permissions Work Together"

To grant catalog permissions to a role:

1. In the catalog, navigate to a catalog object required for a role.
2. Click the **More** link for the object and then click **Permissions** to open the **Permissions** dialog.
3. Click the **Create** icon to open the **Add Roles** dialog.
4. Click **Search** to populate the list of **Available Roles**.
5. Use the **Move** button to move the appropriate roles from the **Available Roles** list to the **Selected Roles** list.
6. Click **OK**.
7. Enable the appropriate permissions for the role by selecting the check boxes.
8. If you have selected a folder: To apply the selections to all items within a folder, select **Apply permissions to items within this folder**.

5.6 Integrating with Oracle Siebel CRM Security

To configure BI Publisher to integrate with Siebel security, perform the tasks in the following sections:

- Section 5.6.1, "Setting Up BI Publisher Roles as Siebel CRM Responsibilities"
- Section 5.6.2, "Configuring BI Publisher to Use Siebel Security"
- Section 5.6.3, "Adding Data Sources to Roles"
- Section 5.6.4, "Granting Catalog Permissions to Roles"

5.6.1 Setting Up BI Publisher Roles as Siebel CRM Responsibilities

To set up BI Publisher roles as Siebel CRM responsibilities:

1. Using Siebel Administrator credentials, navigate to Administration - Application, and then Responsibilities.
2. In the Responsibilities list, add a new record for each of the BI Publisher functional roles:
   - XMLP_ADMIN — Serves as the administrator role for the BI Publisher server.
   - XMLP_DEVELOPER — Allows users to build reports in the system.
   - XMLP_SCHEDULER — Allows users to schedule reports.
   - XMLP_TEMPLATE_DESIGNER — Allows users to connect to the BI Publisher server from the Template Builder and to upload and download templates and grants access to the layout editor.
3. Assign these roles to the appropriate users. You might also want to create additional reporting roles that you can use when setting up your report privileges in the BI Publisher. For example, you might create a role called "EXECUTIVE_SALES" that you can assign a executive-level report folder. You can then assign that role to any user requiring access to the Executive reports.
4. Ensure to assign the XMLP_ADMIN role to a user with administration privileges.

5.6.2 Configuring BI Publisher to Use Siebel Security

To configure BI Publisher to use Siebel Security:

1. Log in to BI Publisher with Administrator privileges. From the **Administration** page select **Security Configuration**.
2. In the **Authorization** region of the page, select Siebel Security from the **Security Model** list. Provide the following connection information:
   - **Siebel Web Service Endpoint String**
   - **Administrator Username**.
   - **Administrator Password**
3. Click **Apply**. Restart BI Publisher for the security changes to take effect.

When you log back in to BI Publisher, the responsibilities to which you added the BI Publisher functional roles are displayed in the **Roles and Permissions** page.

5.6.3 Adding Data Sources to Roles

To view a report generated from a particular data source, a report consumer's role must be granted access to the data source. Similarly, to create a data model based on a particular data source, the report author's role must be granted access to the data source.

To grant a role access to a data source:

1. On the **Administration** tab, under **Security Configuration**, click **Roles and Permissions**.
2. Find the role to which you want to add data sources and click **Add Data Sources**. The **Add Data Sources** page is displayed.
3. Locate the appropriate data sources in the **Available Data Sources** list and use the shuttle buttons to move the sources to the **Allowed Data Sources** list for the role.
4. Click **Apply**.
5. Repeat for all roles that need access to report data sources.

5.6.4 Granting Catalog Permissions to Roles

For a role to access objects in a folder, you must grant the role permissions to the catalog object. You can grant permissions at the folder level, so that a role has the same access to every object in a folder, or you can assign access individually to each object in a folder.

See the following sections for more information:

- Section 3.3, "Understanding BI Publisher's Users, Roles, and Permissions"
- Section 3.4, "About Privileges to Use Functionality"
- Section 3.5, "About Catalog Permissions"
- Section 3.6, "How Functional Privileges and Permissions Work Together"

To grant catalog permissions to a role:

1. In the catalog, navigate to a catalog object that is required for a role.
2. Click the **More** link for the object and then click **Permissions** to open the **Permissions** dialog.
3. Click the **Create** icon to open the **Add Roles** dialog.
4. Click **Search** to populate the list of **Available Roles**.
5. Use the **Move** button to move the appropriate roles from the **Available Roles** list to the **Selected Roles** list.
6. Click **OK**.
7. Enable the appropriate permissions for the role by selecting the check boxes.
8. If you have selected a folder: To apply the selections to all items within a folder, select **Apply permissions to items within this folder**.

6

Implementing a Digital Signature

This chapter describes how to implement a digital signature in PDF documents generated by BI Publisher.

It covers the following topics:

6.1 Introduction

BI Publisher supports digital signatures on PDF output documents. Digital signatures enable you to verify the authenticity of the documents you send and receive. Oracle BI Publisher can access your digital ID file from a central, secure location and at runtime sign the PDF output with the digital ID. The digital signature verifies the signer's identity and ensures that the document has not been altered after it was signed.

For additional information on digital signatures, see the following sources:

- Digital ID Introduction by Verisign
  - http://www.verisign.com/support/tlc/per/whitepaper.htm
- Digital Signature by Adobe
  - http://www.adobe.com/security/digsig.html
- Digital Signatures in PDF and Acrobat
  - http://acrobatusers.com/tutorials/digital-signatures-pdf-acrobat

6.2 Prerequisites and Limitations

Before you can implement digital signatures with Oracle BI Publisher output documents, you need the following:

A digital ID obtained from a public certificate authority or from a private/internal certificate authority (if for internal use only). You must copy the digital ID file to a

secure location of the file system on the server that is accessible by the BI Publisher server.

Use of digital signatures with Oracle BI Publisher output documents has the following limitations:

- Only a single digital ID can be registered with BI Publisher.
- Only reports submitted through BI Publisher's Schedule Report Job interface can include the digital signature.
- The digital signature is enabled at the report level; therefore, multiple templates assigned to the same report share the digital signature properties.

6.3 Obtaining Digital Certificates

To obtain a digital certificate, do one of the following:

- Purchase one from a certificate authority, such as Verisign, and save it to your computer. This method is recommended because it is easier to verify (and therefore trust) the authenticity of the certificate that you purchase. Next, use Microsoft Internet Explorer 7 or later to create a PFX file based on the certificate you purchased. See Section 6.4, "Creating PFX Files."
- Create a self-signed certificate using a software program, such as Adobe Acrobat, Adobe Reader, OpenSSL, or OSDT. This method is less preferred because anyone can create a self-signed certificate. Therefore, it is more difficult to verify and trust the authenticity of the certificate.

  Typically, when you create a self-signed certificate using a software program, the program saves the certificate as part of a PFX file. If this is the case, you do not need to create another PFX file (as described in Section 6.4, "Creating PFX Files").

  To create a self-signed certificate using Adobe Reader:

  1. Open Adobe Reader.
  2. On the Document menu click Security Settings.
  3. Select Digital IDs on the left.
  4. On the toolbar, click Add ID.
  5. Follow the steps in the Add Digital ID wizard. For assistance, refer to the documentation provided with Adobe Reader.
  6. When prompted, save your self-signed certificate as part of a PFX file to an accessible location on your computer.

  After you create your self-signed certificate as part of a PFX file, you can use the PFX file to sign PDF documents by registering it with BI Publisher. See Section 6.5, "Implementing a Digital Signature."

6.4 Creating PFX Files

If you obtained a digital certificate from a certificate authority, you can create a PFX file using that certificate and Microsoft Internet Explorer 7 or later.

Note: If you created a self-signed certificate using a software program such as Adobe Reader, it is likely that the program created the certificate in a PFX file. If this is the case, you don't have to create another PFX file. You can use the one you have.

To create a PFX file with Microsoft Windows Explorer 7 or later:

1. Ensure that your digital certificate is saved on your computer.
2. Open Microsoft Internet Explorer.
3. On the **Tools** menu, click I**nternet Options** and then click the **Content tab**.
4. Click Certificates.
5. In the Certificates dialog, click the tab that contains your digital certificate and then click the certificate.
6. Click Export.
7. Follow the steps in the Certificate Export Wizard. For assistance, refer to the documentation provided with Microsoft Internet Explorer.
8. When prompted, select Use DER encoded binary X.509 as your export file format.
9. When prompted, save your certificate as part of a PFX file to an accessible location on your computer.

After you create your PFX file, you can use it to sign PDF documents.

6.5 Implementing a Digital Signature

The following steps provide an overview of the tasks required to set up and sign your output PDF documents with a digital signature.

1. Register the digital ID in the BI Publisher Administration page and specify the roles that are authorized to sign documents, as described in Section 6.5.1, "Registering Your Digital Signature ID and Assigning Authorized Roles."
2. Specify the display field location, as described in Section 6.5.2, "Specifying the Signature Display Field or Location."
3. Enable Digital Signature for the report using the report properties.
4. Log in to BI Publisher as a user with an authorized role and submit the report through the BI Publisher scheduler, choosing PDF output. When the report completes, it is signed with your digital ID in the specified location of the document.

6.5.1 Registering Your Digital Signature ID and Assigning Authorized Roles

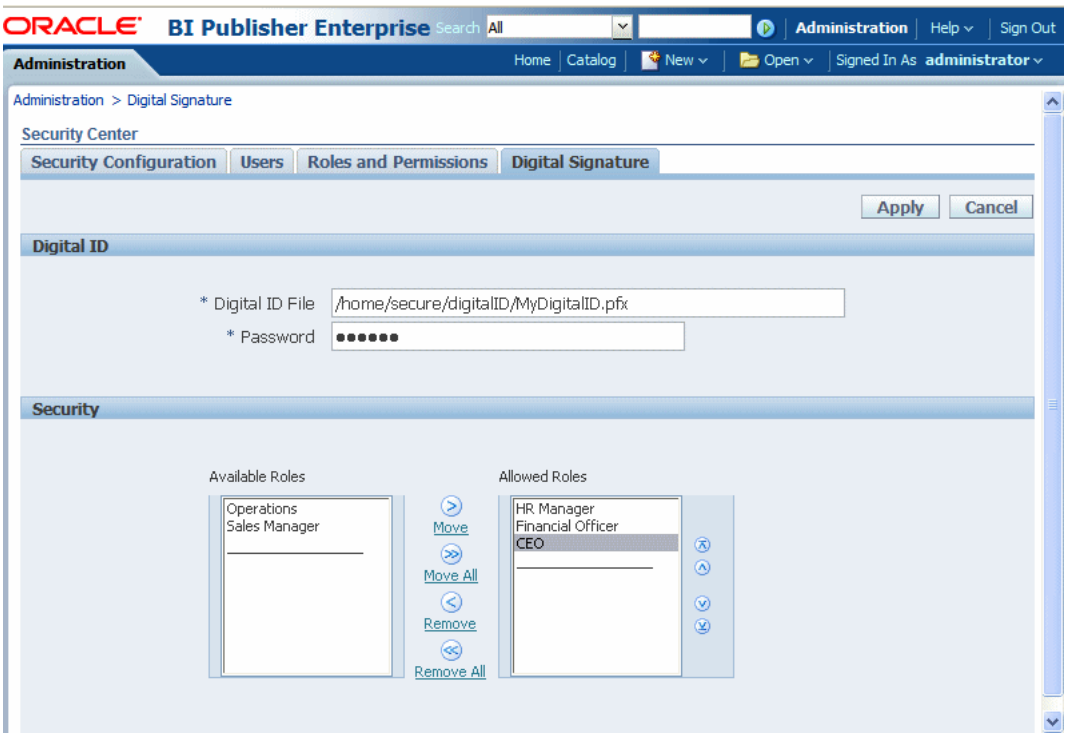
BI Publisher supports the identification of a single digital ID file.

To register a digital ID in the BI Publisher Administration page:

1. On the **Administration** tab, under **Security Center**, click **Digital Signature**.
2. On the **Digital Signature** subtab, enter the file path to the digital ID file and enter the password for the digital ID.
3. Enable the Roles that must have the authority to sign documents with this digital ID. Use the shuttle buttons to move **Available Roles** to the **Allowed Roles** list.

4. Click **Apply**. Figure 6–1 shows the **Digital Signature** subtab.

Figure 6–1 Digital Signature Subtab



6.5.2 Specifying the Signature Display Field or Location

You must specify the location for the digital signature to appear in the completed document. The methods available depend on whether the template type is PDF or RTF.

If the template is PDF, use one of the following options:

- Section 6.5.3, "Specifying a Template Field in a PDF Template for the Digital Signature"
- Section 6.5.4, "Specifying the Location for the Digital Signature in the Report Properties"

If the template is RTF, use the following option:

- Section 6.5.4, "Specifying the Location for the Digital Signature in the Report Properties"

6.5.3 Specifying a Template Field in a PDF Template for the Digital Signature

See the chapter: Creating a PDF Template, topic: "Adding or Designating a Field for a Digital Signature" in *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* for instructions on including a field in the PDF template for the digital signature.

6.5.4 Specifying the Location for the Digital Signature in the Report Properties

When you specify a location in the document to place the digital signature, you can either specify a general location (Top Left, Top Center, or Top Right) or you can specify x and y coordinates in the document. You can also specify the field height and width.

This is done through properties on the Runtime Configuration page. Therefore you do not need to alter the template to include a digital signature.

To specify the location for the digital signature:

1. In the catalog, navigate to the report.
2. Click the **Edit** link for the report to open the report for editing.
3. Click **Properties** and then click the **Formatting** tab.
4. Scroll to the **PDF Digital Signature** group of properties.
5. Set **Enable Digital Signature** to **True**.
6. Specify the location in the document where you want the digital signature to appear by setting the appropriate properties as follows (note that the signature is inserted on the first page of the document only):
   - **Existing signature field name** — Does not apply to this method.
   - **Signature field location** — Provides a list containing the following values:

     Top Left, Top Center, Top Right

     Select one of these general locations and BI Publisher places the digital signature in the output document sized and positioned appropriately.

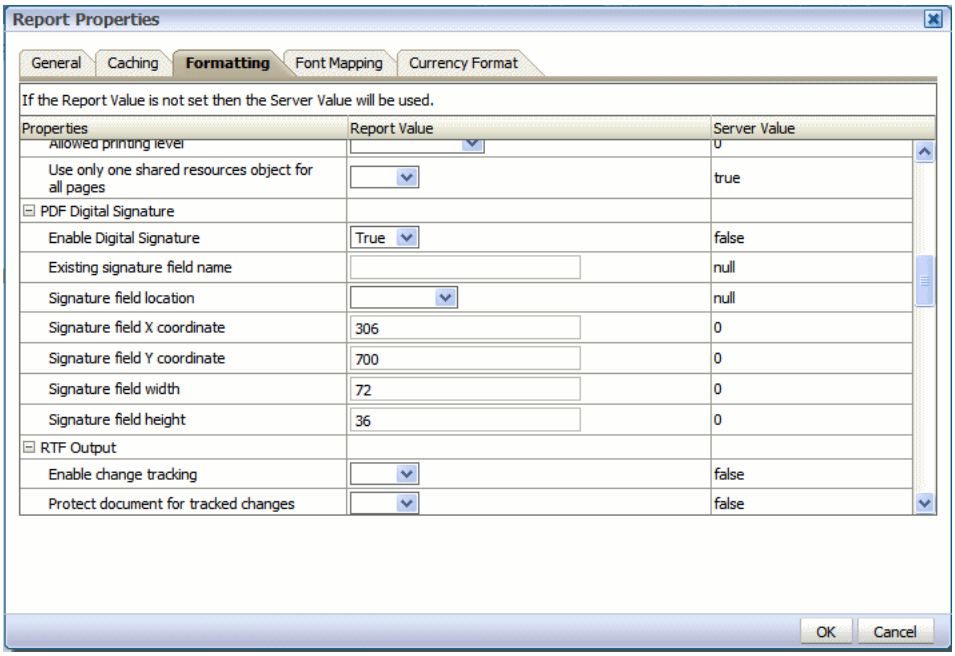
     If you set this property, then do not enter X and Y coordinates or width and height properties.
   - **Signature field X coordinate** — Using the left edge of the document as the zero point of the X axis, enter the position in points to place the digital signature from the left.

     For example, to place the digital signature horizontally in the middle of an 8.5 inch by 11 inch document (that is, 612 points in width and 792 points in height), enter 306.
   - **Signature field Y coordinate** — Using the bottom edge of the document as the zero point of the Y axis, enter the position in points to place digital signature from the bottom.

     For example, to place the digital signature vertically in the middle of an 8.5 inch by 11 inch document (that is, 612 points in width and 792 points in height), enter 396.
   - **Signature field width** — Enter in points the desired width of the inserted digital signature field. This applies only if you are setting the X and Y coordinates.
   - **Signature field height** — Enter in points the desired height of the inserted digital signature field. This applies only if you are setting the X and Y coordinates.

Figure 6–2 shows a report that is configured to place the digital signature at specific x and y coordinates in the document.

Figure 6–2 Report Properties for Digital Signature



6.6 Running and Signing Reports with a Digital Signature

Users assigned a role with the digital signature privilege can attach the digital signature to their generated reports configured to include the digital signature. The digital signature can be inserted only on scheduled reports.

To sign reports with a digital signature:

1. Log in to BI Publisher as a user with a role granted digital signature privileges.
2. In the catalog, navigate to the report that has been enabled for digital signature and click **Schedule**.
3. Complete the fields in the **Schedule Report Job** page, selecting PDF output, and then submit the job.

   The completed PDF displays the digital signature.

7

Configuring the Scheduler

This chapter describes the features, architecture, diagnostics, and configuration of the BI Publisher scheduler.

It covers the following topics:

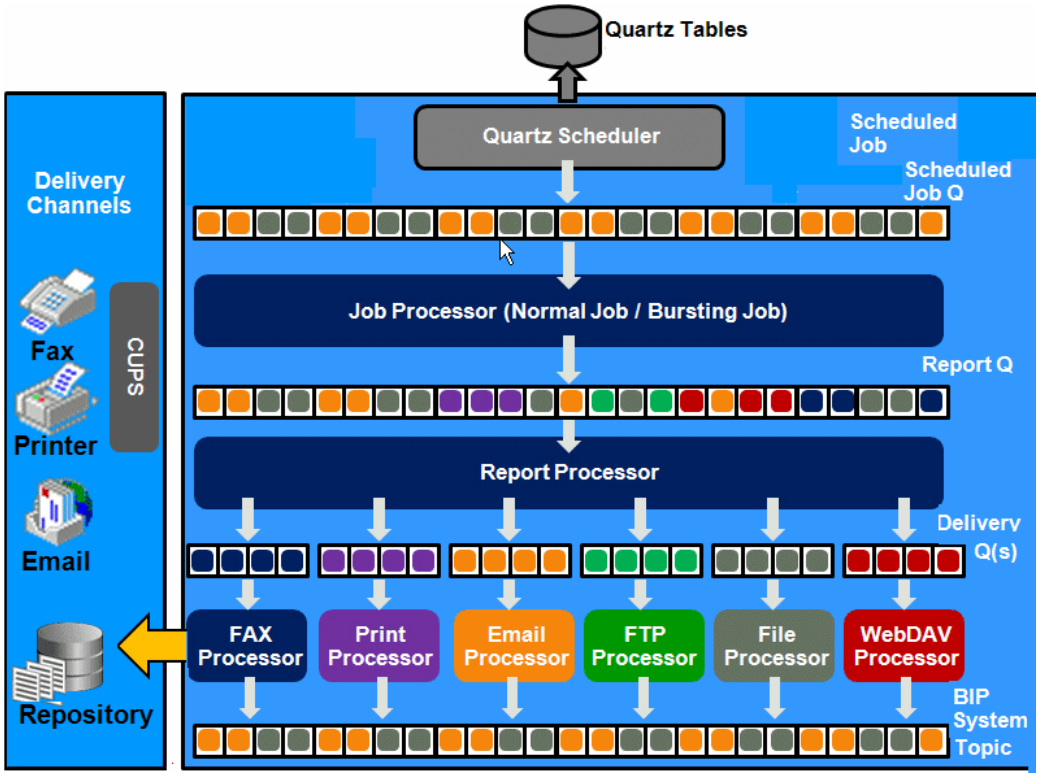
7.1 Understanding the BI Publisher Scheduler

The updated architecture of the 11*g* BI Publisher Scheduler uses the Java Messaging Service (JMS) queue technology. This architecture enables you to add multiple BI Publisher servers to a cluster and then dedicate each server to a particular function: report generation, document generation, or specific delivery channels.

7.1.1 Architecture

The architecture of the BI Publisher Scheduler uses JMS queues and topics to provide a highly scalable, highly performing and robust report scheduling and delivery system. Figure 7–1 displays the scheduler architecture.

Figure 7–1 Scheduler Architecture



The following list describes the tasks performed by the scheduler when a job is submitted:

1. Submit Job
   - Stores job information and triggers in Quartz tables
2. Job Processor
   - When quartz trigger is fired, puts job information in Scheduler job queue
3. Bursting Engine / Batch Job Process
   - Bursting Engine Listener
     - Takes the scheduled job information from the queue
     - Extracts data from data source
     - Splits data according to bursting split by definition
     - Stores data temporarily in temp folder
     - Puts report metadata into Report Queue
   - Batch Job Process
     - Takes the scheduled job information from the queue
     - Extracts data from data source
     - Stores data temporarily in temp folder
     - Puts report metadata into Report Queue
4. FO Report Processor

- Listens to Report Q
- Generates report based on metadata
- Stores report in shared TEMP directory
- Puts report delivery information in Delivery Queue

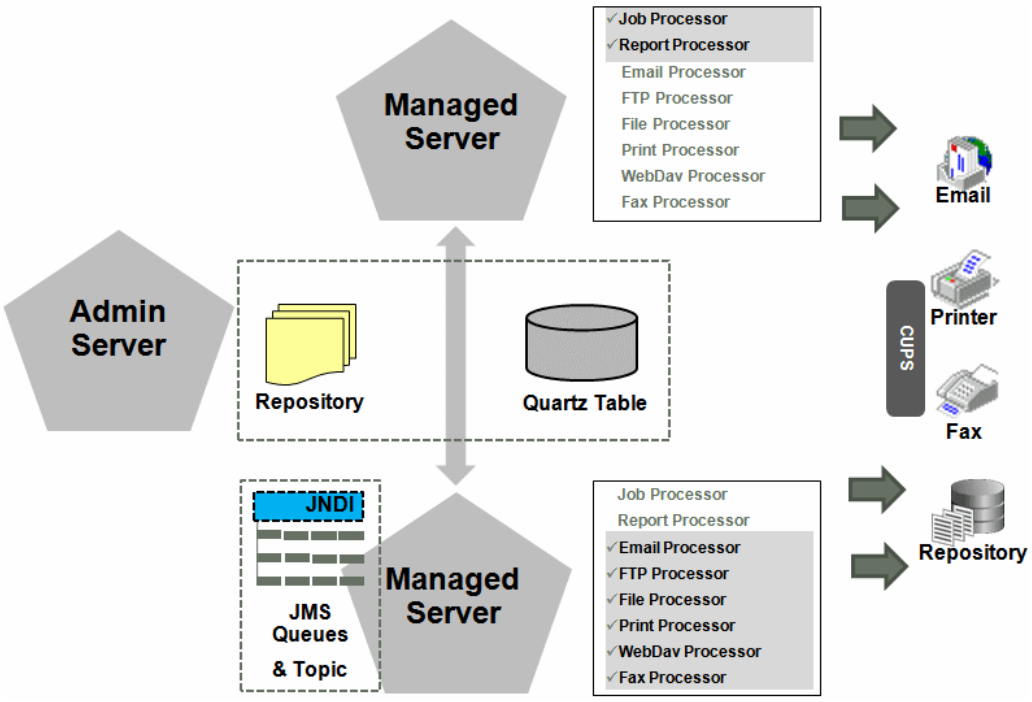
5. Delivery (E-mail, File, FTP) Processors
   - Listen to Delivery queue
   - Call delivery API to deliver to different channels
6. BI Publisher (BIP) System Topic

   The BIP System Topic publishes the runtime status and health of the scheduling engine. The topic publishes the status of all instances, the thread status of messages in the JMS queues, the status of all scheduler configurations such as database configuration, JNDI configuration of JMS queues and so on.

7.1.2 About Clustering

BI Publisher clustering support enables you to add server instances on demand to handle processing and delivery load. Figure 7–2 illustrates clustering in an Oracle WebLogic Server. Note that the report repository and the scheduler database are shared across the multiple instances; also, the JMS queues for scheduling and JMS topic for publishing diagnostic information are shared across the server by registering JMS queues and topics through JNDI services.

Figure 7–2 BI Publisher Clustering



Each managed server instance points to the same report repository. In each managed server instance all the processes (Job Processor, Report Processor, E-mail Processor, FTP Processor, Fax Processor, File Processor, Print Processor, and Web Dav Processor) are configured. Therefore the moment a server instance pointing to the same

repository is deployed, it is added to the cluster and all the processors in this instance are ready to run.

You can select the process to enable on any server instance, thereby using the resources optimally. Moreover, if there is a demand to process heavier jobs you can add more instances for report processing. Similarly, if e-mail delivery is the most preferred delivery channel, then more instances can be added to scale up e-mail delivery.

For more information about clustering and high availability, see *Oracle Fusion Middleware High Availability Guide*.

7.1.3 How Failover Works

BI Publisher provides a robust failover mechanism so that no report fails to deliver due to server unavailability. Achieve this by balancing each process of the Scheduler using two or more nodes in a cluster thereby ensuring that a failure of any node must be backed up by the second node without any loss of data. For example, by enabling the Job Processor in two nodes, if one node fails, then the second node can process the jobs.

Important: If a node goes down, the other nodes continue to service the queue. However, if a report job is in one of the following stages of execution: data retrieval, data formatting, or report delivery, the job is marked as failed, and must be manually resubmitted.

7.2 Set Up Considerations

Following are topics to consider before setting up the scheduler.

7.2.1 Choosing JNDI or JDBC Connection

By default, the BI Platform installer configures the WebLogic JNDI connection URL. JDBC is not recommended for production use. JDBC should only be used for low volume local testing.

7.2.2 Supported JMS Providers

When you install BI Publisher, the scheduler is automatically configured to use WebLogic JMS. To configure BI Publisher to use ActiveMQ instead, see Section A.2, "Configuring BI Publisher for ActiveMQ."

7.3 About the Scheduler Configuration

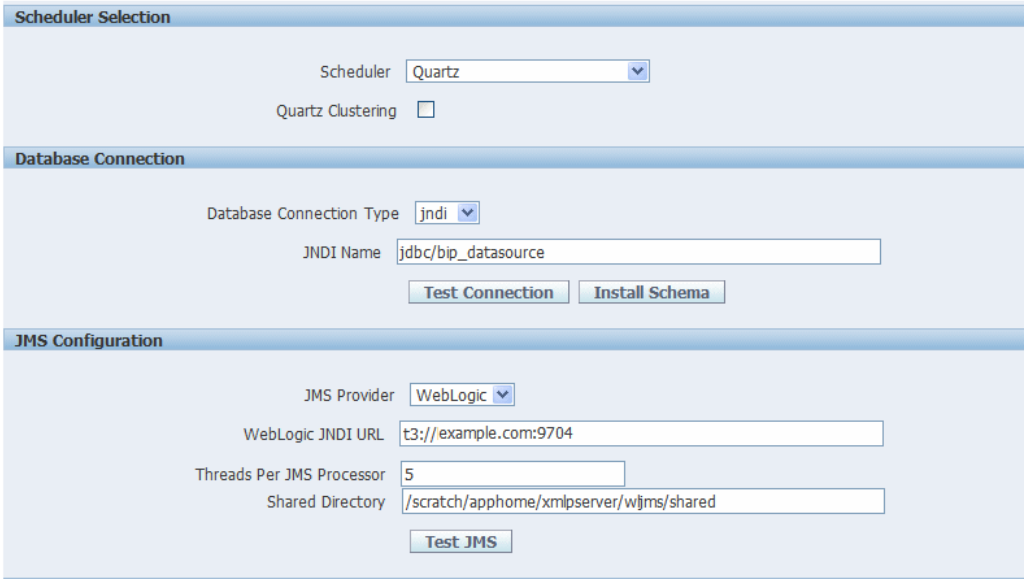
After you install BI Publisher using the BI Platform Installer and start the servers, the BI Publisher scheduler is running and the following are configured:

- The scheduler schema is installed to the database by the Repository Creation Utility as a preinstall step.
- JMS is configured in your server for BI Publisher.
- The WebLogic JNDI URL is configured.
- Default threads per processor is set to 5.

See *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence* for more information on configurations performed by the Oracle BI Platform Installer.

You can see this configuration in the **Scheduler Configuration** page: From the **Administration** page, under **System Maintenance**, click **Scheduler Configuration**. Figure 7–3 shows the Database Connection and JMS Configuration regions of the Scheduler Configuration page.

Figure 7–3 Database Connection and JMS Configuration Regions



7.3.1 Configuring the Shared Directory

The Shared Directory is used to temporarily store data and files used by the scheduler while jobs are executing. After a job completes, the temporary data for the job is deleted. If the BI Publisher scheduler is configured to run on different nodes or machines, you must define this directory. The directory is used to exchange data and document information among all the BI Publisher nodes and therefore must be accessible by all BI Publisher nodes. The size of the directory depends on the total size of the job data, output documents, and the number of concurrent jobs. The directory should be big enough to hold all the XML data and documents for all the parallel running jobs. If BI Publisher runs on different machines while this directory is not configured, the scheduler may fail.

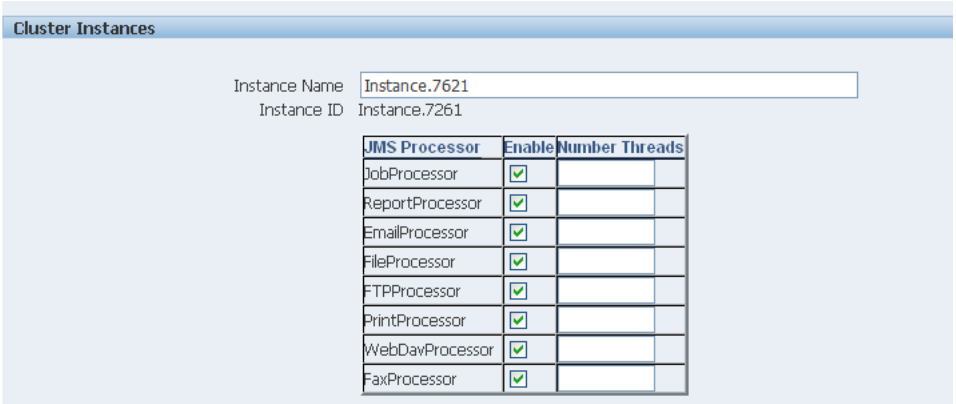
If BI Publisher runs on a single machine, defining a shared directory is optional. BI Publisher uses the application server's temporary directory to store this data.

7.4 Configuring Processors and Processor Threads

For each cluster instance that you have configured, a processor configuration table is displayed. Use the tables to enable and disable processors and specify threads for each processor.

The default number of threads for each processor is set by the **Threads per JMS Processor** property under **JMS Configuration**, as shown in Figure 7–3. Edit the threads for a specific processor in the **Cluster Instances** region by updating the **Number Threads** setting, as shown in Figure 7–4. Note that processors that use the default setting show no entry in the table. Enter a **Number Threads** value only to set a thread count for a particular processor to differ from the default.

Figure 7–4 Cluster Instances Region



The optimum number of threads per processor depends on the requirements of the system. You can use the **Scheduler Diagnostics** page to help in assessing load in the system. See Section 7.6, "Scheduler Diagnostics."

To add managed servers to the system, see Section 7.5, "Adding Managed Servers."

7.5 Adding Managed Servers

Add managed servers in the Oracle WebLogic Administration Console and then configure the cluster instances in the BI Publisher Administration page.

7.5.1 Adding a Managed Server

For detailed information on using the Oracle WebLogic Administration Console see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*. For additional information about Fusion Middleware Control and how to use it, see *Oracle Fusion Middleware Administrator's Guide*.

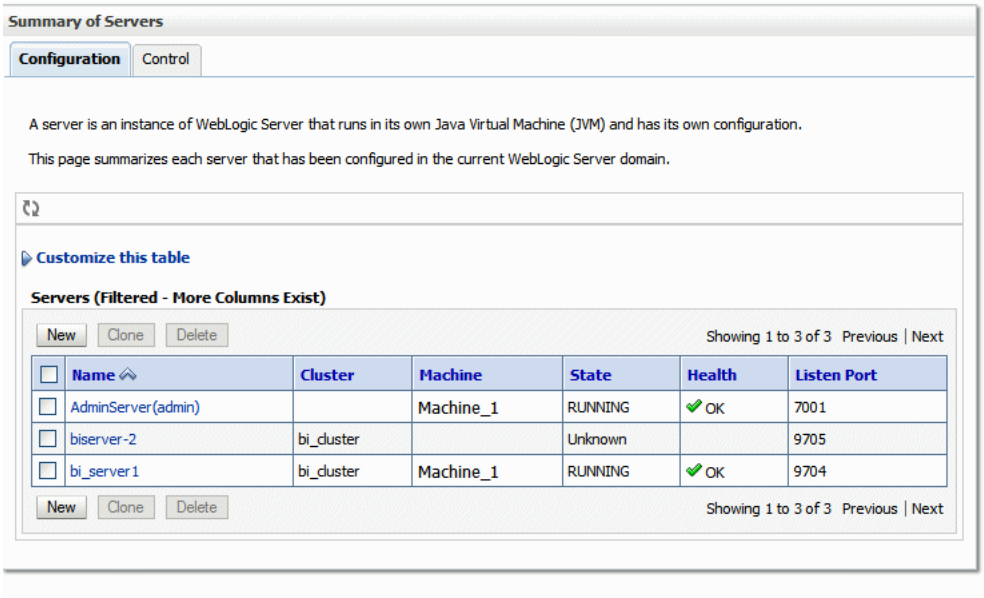
To add a managed server:

1. Access the Oracle WebLogic Administration Console using one of the following methods:
2. Click **Lock & Edit**.
3. Under **Domain Structure**, expand **Environment** and click **Servers**.
4. On the **Servers** table, click **New**.
5. On the Create a New Server: Server Properties page:
   - Enter the name of the server in the Name field.
   - In **Listen Port**, enter the port number from which you want to access the server instance.
   - Select **Yes, make this server a member of an existing cluster.**

     Select the bi_cluster from the list.
   - Click **Next**.
6. Review the configuration options that you have chosen.
7. Click **Finish**.

   The new server displays in the **Servers** table, as shown in Figure 7–5.

Figure 7–5 Servers Table

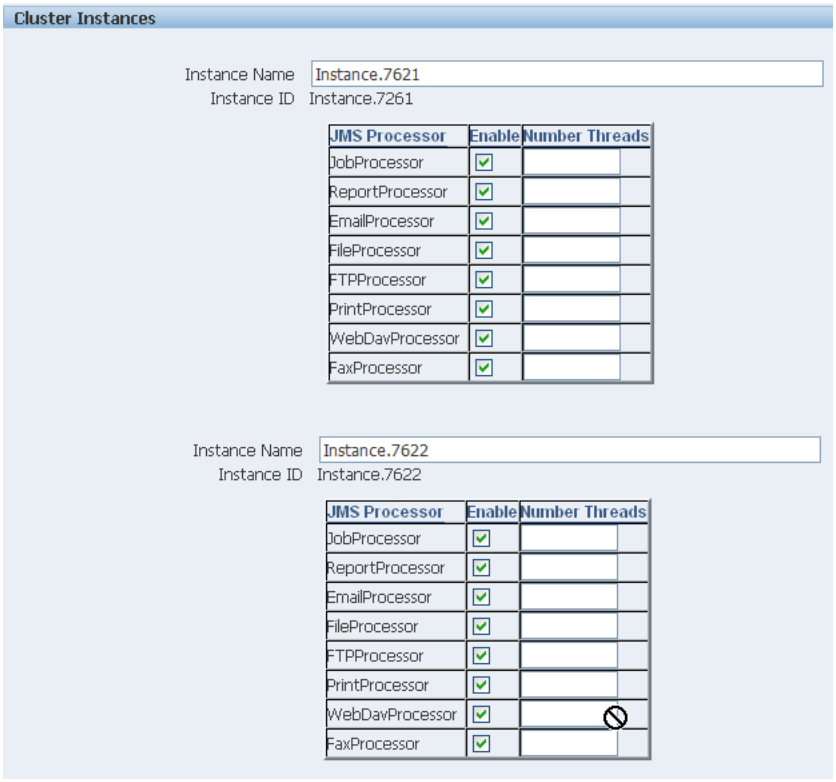


8. Click the server name to open the **Settings** page.
9. Select a **Machine** for the new server.
10. Click **Save**.
11. Click **Activate Changes**.
12. Start the new server.

7.5.2 Configure the Processors in BI Publisher

After the new managed server has been started, the set of processors for that server displays in BI Publisher, as shown in Figure 7–6.

Figure 7–6 Processors in BI Publisher



You can now configure the threads appropriately for your system load.

7.6 Scheduler Diagnostics

The Scheduler diagnostics page provides the runtime status of the scheduler. It provides status of its JMS configuration, JMS queues, Cluster instance status, Scheduler Database status, Toplink status, and Scheduler (Quartz) status.

The Diagnostics page displays how many scheduled report requests have been received by the JMS queues, how many of them have failed and how many are still running. The JMS status can be viewed at the cluster-instance level enabling you to decide whether to add more instances to scale up by one or more of these JMS processors.

For example, if there are too many requests queued up for the e-mail processor in one instance, you can consider adding another instance and enabling it to handle e-mail processing. Similarly, if there are very large reports being processed and showing in the Report Process queue in running status, then you can add another instance to scale up the Report Process capability.

Also, the Scheduler Diagnostics page reflects the status of each component to show if any component is down. You can see the connection string or JNDI name to the database, which cluster instance associates to which managed server instance, Toplink connection pool configuration, and so on.

If an instance shows a failed status, then you can recover the instance and with the failover mechanism of the JMS set up in the cluster, no jobs submitted are lost. When the server instance is brought back, it is immediately available in the cluster for service. The instance removal and addition reflects dynamically on the diagnostic page.

When an instance is added to the cluster, the Scheduler Diagnostics page immediately recognizes the new instance and displays the status of the new instances and all the threads running on that instance. This provides a powerful monitoring capability to the administrator to trace and resolve issues in any instance or any component of the scheduler.

The Scheduler Diagnostics page provides information on the following components:

- JMS
- Cluster
- Database
- Scheduler Engine

The **JMS** section provides information on the following:

- **JMS Cluster Config**: This section provides configuration information for JMS setup:
  - Provider type (Weblogic / ActiveMQ)
  - WebLogic version
  - WebLogic JNDI Factory
  - JNDI URL for JMS
  - Queue names
  - Temporary directory
- **JMS Runtime**: This provides runtime status of all JMS queues and topics, as shown in Figure 7–7

Figure 7–7 JMS Runtime Section

----**JMS Runtime**		Passed	
------Topic - BIP.System.T		Passed	
------Queue - BIP.Burst.Job.Q	0 pending	Passed	
------Queue - BIP.Burst.Report.Q	0 pending	Passed	
------Queue - BIP.Delivery.Email.Q	0 pending	Passed	
------Queue - BIP.Delivery.File.Q	0 pending	Passed	
------Queue - BIP.Delivery.FTP.Q	0 pending	Passed	
------Queue - BIP.Delivery.Print.Q	0 pending	Passed	
------Queue - BIP.Delivery.WebDAV.Q	0 pending	Passed	
------Queue - BIP.Delivery.Fax.Q	0 pending	Passed	

The **Cluster** section provides details on the cluster instance, as shown in Figure 7–8. Use this information to understand the load on each processor.

Figure 7–8 Cluster Section

--Cluster		Passed	
----Instance - Cluster369.127028		Passed	
------JMS Instance Config	/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_user/xmlpserver/war//WEB-INF/jms_config.xml	Passed	
------JMSWrapper	Started (Thu Jul 01 07:10:18 UTC 2010)	Passed	
--------JMSClient - system	Started; BIP.System.T: 3458 sent, 0 failed	Passed	
----------JMSProcessor - ClusterMessageListener	Started; BIP.System.T; 1 threads; 3458 received, 0 failed, 0 running	Passed	
--------JMSClient - jmsclient_producer	Started; BIP.Burst.Job.Q: 39 sent, 0 failed; BIP.Burst.Report.Q: 95 sent, 0 failed; BIP.Delivery.Email.Q: 82 sent, 0 failed	Passed	
--------JMSClient - jmsclient_schedule	Started	Passed	
----------JMSProcessor - JobProcessor	Started; BIP.Burst.Job.Q; 5 threads; 39 received, 0 failed, 0 running	Passed	
----------JMSProcessor - ReportProcessor	Started; BIP.Burst.Report.Q; 5 threads; 95 received, 0 failed, 0 running	Passed	
--------JMSClient - jmsclient_delivery	Started	Passed	
----------JMSProcessor - EmailProcessor	Started; BIP.Delivery.Email.Q; 5 threads; 82 received, 0 failed, 0 running	Passed	
----------JMSProcessor - FileProcessor	Started; BIP.Delivery.File.Q; 5 threads; 0 received, 0 failed, 0 running	Passed	
----------JMSProcessor - FTPProcessor	Started; BIP.Delivery.FTP.Q; 5 threads; 0 received, 0 failed, 0 running	Passed	
----------JMSProcessor - PrintProcessor	Started; BIP.Delivery.Print.Q; 5 threads; 0 received, 0 failed, 0 running	Passed	
----------JMSProcessor - WebDavProcessor	Started; BIP.Delivery.WebDAV.Q; 5 threads; 0 received, 0 failed, 0 running	Passed	
----------JMSProcessor - FaxProcessor	Started; BIP.Delivery.Fax.Q; 5 threads; 0 received, 0 failed, 0 running	Passed	

- JMS instance config
- JMS Wrapper
- JMS Client - System — Provides status of the BIP System topic. The scheduler diagnostic page is a subscriber to this topic.
- JMS Client_producer — Not used in this release.
- JMS Client_schedule — Provides status of the job processor and report processor, each processor showing number of active threads, number of messages received, number of messages failed, and number of messages running.
- JMS Client_delivery — Provides status of different delivery processors as listeners, each delivery processor showing number of active threads, number of messages received, number of messages failed, and number of messages running.

The **Database** section provides information on these components, as shown in Figure 7–9:

- Database Config — Connection type, JNDI Name, or connection string
- Toplink Config — Connection pooling, logging level
- Database Schema

Figure 7–9 Database Section

--Database		Passed	
----Database Config	/scratch/apphome/xmlpserver/repository/Admin/Scheduler/quartz-config.properties	Passed	
------Connection Type	jdbc	Info	
------Database Type	oracle.toplink.platform.database.oracle.Oracle11Platform	Info	
------Connection String	jdbc:oracle:thin:@10.144.177.30:1521:orcl	Info	
------User Name	BIPUSER2	Info	
------Database Driver	oracle.jdbc.OracleDriver	Info	
----Toplink Config	/scratch/apphome/xmlpserver/repository/Admin/Scheduler/quartz-config.properties	Passed	
------Toplink Mapping File	META-INF/toplink_mappings.xml	Info	
------Toplink Logging	severe	Info	
------Toplink Connection Policy Lazy	false	Info	
------Toplink Read Connection Pool	read-connection-pool, name: read-pool, max-connections: 20, min-connections: 10	Info	
------Toplink Write Connection Pool	write-connection-pool, name: default, max-connections: 20, min-connections: 10	Info	
----Database Schema		Passed	

The **Quartz** section provides information on these components, as shown in Figure 7–10:

- Quartz Configuration
- Quartz Initialization

Figure 7–10 Quartz Section

--Quartz		Passed	
----Quartz Config	/scratch/apphome/xmlpserver/repository/Admin/Scheduler/quartz-config.properties	Passed	
------org.quartz.dataSource.myDS.maxConnections	5	Info	
------org.quartz.scheduler.instanceId	AUTO	Info	
------org.quartz.scheduler.instanceName	BIPublisherScheduler	Info	
------org.quartz.dataSource.myDS.user	BIPUSER2	Info	
------org.quartz.jobStore.tablePrefix	QRTZ_	Info	
------org.quartz.jobStore.class	org.quartz.impl.jdbcjobstore.JobStoreTX	Info	
------org.quartz.dataSource.myDS.URL	jdbc:oracle:thin:@10.144.177.30:1521:orcl	Info	
------org.quartz.threadPool.class	org.quartz.simpl.SimpleThreadPool	Info	
------org.quartz.jobStore.useProperties	false	Info	
------org.quartz.threadPool.threadPriority	5	Info	
------org.quartz.jobStore.isClustered	false	Info	
------org.quartz.jobStore.misfireThreshold	60000	Info	
------org.quartz.threadPool.threadCount	3	Info	
------org.quartz.threadPool.threadsInheritContextClassLoaderOfInitializingThread	true	Info	
------org.quartz.jobStore.driverDelegateClass	org.quartz.impl.jdbcjobstore.oracle.OracleDelegate	Info	
------org.quartz.dataSource.myDS.driver	oracle.jdbc.OracleDriver	Info	
------org.quartz.jobStore.dataSource	myDS	Info	
----Quartz Initilization		Passed	

7.6.1 Resolving Quartz Configuration Errors

The following is a common Quartz configuration error in the Scheduler Diagnostics page:

Error Description and Resolution

During the BI Publisher start up (when the WebLogic Managed server or Admin server are started) if the JNDI data source configured as `jdbc/bip_datasource` is unavailable, then the Quartz initialization will fail. The Scheduler Diagnostics page displays an error for Quartz Configuration.

If this occurs, perform the following:

1. Verify that the data source configured as `jdbc/bip_datasource` is available. On the **Scheduler Configuration** page, click **Test Connection** to ensure the connection is working.
2. On the **Scheduler Diagnostics** page locate the "Database Schema" diagnostics item and ensure it passed.
3. Go back to the **Scheduler Configuration** page and change the **Scheduler Selection** from "Quartz" to "None" and click **Apply**. Now change it back to "Quartz" and click **Apply** again.
4. On the **Scheduler Diagnostics** verify that the Quartz error has cleared.

8

Configuring Server Properties

This chapter describes how to configure BI Publisher server properties such as caching specifications, monitoring and auditing, and catalog properties.

It covers the following topics:

8.1 Setting the Path to the Configuration Folder

The Configuration folder stores the files that contain your server configuration settings, for example, the data source connections, delivery server definitions, and scheduler settings.

The path to the Configuration folder is stored in the xmlp-server-config.xml configuration file.

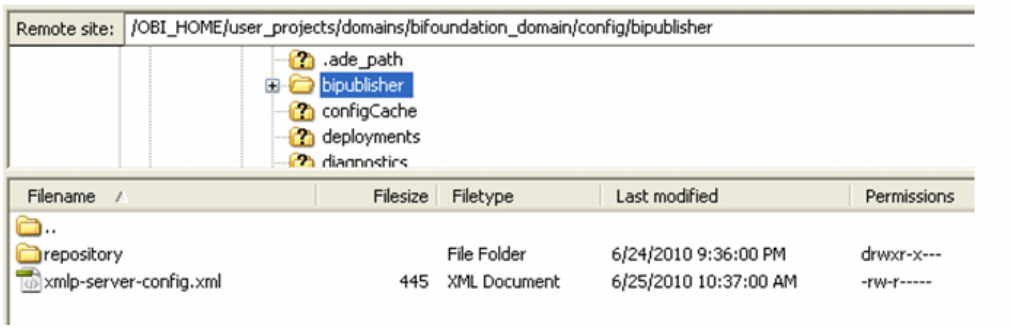
When you install BI Publisher, this is automatically configured to

```
${xdo.server.config.dir}/repository
```

The environment variable ${xdo.server.config.dir} is used to store the path to the location of the xmlp-server-config.xml configuration file. By default both the BI Publisher configuration folder and the xmlp-server-config.xml file are installed to <DOMAIN_HOME>/config/bipublisher.

For example: /OBI_HOME/user_projects/domains/bifoundation_domain/config/bipublisher, as shown in Figure 8–1.

Figure 8–1 Path for Configuration Folder



8.2 Configuring the Catalog

BI Publisher supports the following catalog types:

- Section 8.2.1, "Configuring the Oracle BI Publisher File System Catalog"

  The Oracle BI Publisher file system option is for installations of BI Publisher that are not integrated with Oracle Business Intelligence Enterprise Edition.

- Section 8.2.2, "Configuring BI Publisher to Use the Oracle BI EE Catalog"

  If you install BI Publisher with the Oracle Business Intelligence Enterprise Edition, the catalog is automatically configured to use the shared Oracle BI Presentation Catalog.

8.2.1 Configuring the Oracle BI Publisher File System Catalog

> **Note:** When using file systems such as NFS, Windows, or NAS for the repository, ensure that the file system is secured.

When you install BI Publisher, the catalog is configured by default to:

```
${xdo.server.config.dir}/repository
```

To change the location for the repository:

1. Under **Catalog**, select **Oracle BI Publisher - File System** as the **Catalog Type**.
2. Enter the absolute **Path**.
3. Apply your changes and restart your BI Publisher application.

> **Note:** Because the repository is in the file system, the case sensitivity of folder and report names is determined by the platform on which you run BI Publisher. For Windows-based environments, the repository object names are not case-sensitive. For UNIX-based environments, the repository object names are case-sensitive.

8.2.2 Configuring BI Publisher to Use the Oracle BI EE Catalog

If you installed BI Publisher as part of the Oracle Business Intelligence Enterprise Edition, then BI Publisher is automatically configured to use the shared Oracle BI EE (Oracle BI Presentation) catalog.

For prerequisites and steps for manually integrating BI Publisher with Oracle BI Enterprise Edition, see Section B.1, "About Integration."

8.2.2.1 Configuring the BI Search Fields

If you have configured Oracle Business Intelligence with Oracle Secure Enterprise Search (Oracle SES), configure the following fields to enable the full text search for BI Publisher objects.

Prerequisites

Before configuring the fields in BI Publisher, you must first perform the following:

1. Set up Oracle Secure Enterprise Search (Oracle SES).
2. Integrate Oracle SES with Oracle Business Intelligence Presentation Services.

For the procedures for completing the prerequisites, see "Configuring for Full-Text Catalog Search" in the *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

Enter the following fields in BI Publisher:

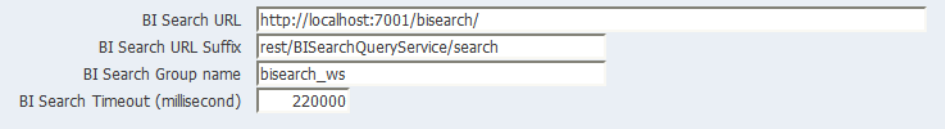
- BI Search URL - enter the basic URL for Oracle Business Intelligence, adding the search context name; takes the format:

  `http://computer_name:port/bisearch`

  For example: http://localhost:7001/bisearch
- BI Search URL Suffix - this field defaults to "rest/BISearchQueryService/search". Do not edit this field.
- BI Search Group name -enter the name of the search group that you created in Oracle SES, for example: bisearch_ws
- BI Search Timeout (millisecond) - enter the maximum number of milliseconds that Oracle BI Publisher waits for a response to return with search results. This field defaults to 22000.

Figure 8–2 shows the BI search fields.

Figure 8–2 BI Search Fields



8.3 Setting General Properties

The general properties region includes the following settings:

- Section 8.3.1, "System Temporary Directory"
- Section 8.3.2, "Report Scalable Threshold"

8.3.1 System Temporary Directory

This section includes the following topics about setting a system temporary directory:

- Section 8.3.1.1, "About Temporary Files"

8.3.1.1 About Temporary Files

BI Publisher creates the following types of temporary and cache files:

Temporary files:

- Temporary files created by the formatting engines (FO processor, PDF Form Processor, PDF generators and so on)
- Data Files

These files are removed after the reports generate successfully.

Dynamic image files for HTML output:

- Dynamic charts
- Embedded images in RTF templates

Cache files:

- Data cache
- LOV (List of Values) cache
- Document Cache
- XSL Cache from RTF templates

8.3.1.2 Setting the System Temporary Directory

If you do not specify a temporary directory here, temporary files and dynamic image files are generated under {*bip_deployment_directory*}/xdo/tmp. Cache files are generated under {*bip_deployment_directory*}/xdo/cache.

When you configure a System Temporary Directory using this field, for example: "/disk1/BIP_Temp", the BI Publisher server automatically creates the following directories:

- /disk1/BIP_Temp/xdo
- /disk1/BIP_Temp/xdo/tmp
- /disk1/BIP_Temp/xdo/cache

Temporary files are generated under /disk1/BIP_Temp/xdo/tmp.

Cache files are generated under /disk1/BIP_Temp/xdo/cache.

Dynamic image files are still created in the {*bip_deployment_directory*}/xdo/tmp directory and are not affected by this configuration.

Whenever the BI Publisher server is restarted, any files under /disk1/BIP_Temp/xdo are removed.

Note: When using the BI Publisher web services `uploadReportDataChunk()` or `downloadReportDataChunk()` in a clustered environment, you must set the **System Temporary Directory** to be a shared directory accessible to all servers within the cluster.

You must enter the absolute path to the directory. For example, the directory can exist under `${xdo.server.config.dir}/temp` but you must enter the absolute path, such as `/net/subfoldera/scratch/subfolderb/11gcat/temp`

Repeat this procedure for all servers in the cluster, entering the same value for **System Temporary Directory**.

8.3.1.3 Sizing the System Temporary Directory

Sizing requirements depend on how large the generated data files and reports are, how many reports enabled cache, and the number of concurrent users. If you must process 1 GB of data and then to generate a report that is 1 GB, then the temp disk should have more than 2 GB of disk space for a single report run. If you require ten concurrent report runs of similarly sized reports, then more than 20 GB of disk space is required. In addition, if you must cache the data and reports for these ten users, you need additional 20 GB of disk space. Note that cache is per user.

8.3.2 Report Scalable Threshold

This property specifies the threshold at which data is cached on the disk. When the data volume is large, caching the data saves memory, but results in slower processing. Enter a value in bytes. The default and general recommendation for this property is 1000000 (1 megabyte).

8.4 Setting Server Caching Specifications

When BI Publisher processes a report, the data and the report document are stored in cache. Each item creates a separate cache file. Set the following properties to configure the size and expiration of this cache:

- **Cache Expiration** — Enter the expiration period for the cache in minutes. The default is 30.
- **Cache Size Limit** — Enter the maximum number of cached items to maintain regardless of the size of these items. The default is 1000.

When BI Publisher processes a report it stores the report definition in memory so that for subsequent requests for the same report the report definition can be retrieved from memory rather than from disk. Set the following property to configure this cache:

- **Maximum Cached Report Definitions** — Enter the maximum number of report definitions to maintain in cache. The default is 50. This cache does not expire.

Note: Report-specific caching of data sets can be set as a report property. See the section "Configuring Report Properties" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

8.5 Setting Retry Properties for Database Failover

If BI Publisher fails to connect to a data source through the defined JDBC or JNDI connection, then these properties control the number of retries that are attempted before switching to the backup connection for the database.

- Number of Retries

  Default value is 6. Enter the number of times to attempt to make a connection before switching to the backup database.

- Retry Interval (seconds)

  Default value is 10 seconds. Enter the number of seconds to wait before retrying the connection.

8.6 Enabling Monitor and Audit

This setting enables user auditing and monitoring in BI Publisher. Performance monitoring enables you to monitor the performance of queries, reports and document generation and to analyze the provided details.

Selecting the **Enable Monitor and Audit** check box on the **Server Configuration** page is the first step required for enabling performance monitoring and user auditing in your system.

For the complete steps, see Section 12.5, "About Performance Monitoring and User Auditing."

8.7 Setting Report Viewer Properties

The **Report Viewer Configuration** tab enables you to set the report viewer property **Show Apply Button**.

When set to True, reports with parameter options display the **Apply** button in the report viewer. When a user changes the parameter values, he must click **Apply** to render the report with the new values.

When set to False, the report viewer does not display the **Apply** button. Instead, when a user enters a new parameter value, BI Publisher automatically renders the report after the new value is selected or entered.

This property can also be set at the report level to override the system setting. For information on setting the property at the report level, see "Configuring Parameter Settings for the Report" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

9

Setting Up Data Sources

This chapter describes how to set up data sources for BI Publisher including JDBC and JNDI connections, LDAP server connections, OLAP data sources, and file data sources.

It covers the following topics:

9.1 Overview of Setting Up Data Sources

BI Publisher supports a variety of data sources. The data can come from a database, a HTTP XML feed, a Web Service, an Oracle BI Analysis, an OLAP cube, a LDAP server, or a previously generated XML file or Microsoft Excel file.

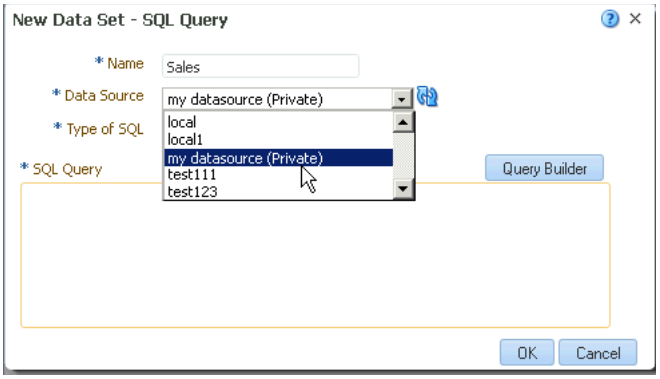
This section describes how to set up connections to the data sources that are described in the following sections:

9.1.1 About Private Data Source Connections

Private connections for OLAP, JDBC, Web Service, and HTTP data sources are supported in BI Publisher and can be created by users with data model creation privileges.

When a user creates a private data source connection, it displays only for that user in the data model editor data source menus. For example, a user creates a private data source connection called "my datasource." When the user creates a data set, the private data source connection displays in the Data Source selection menu as shown in Figure 9–1.

Figure 9–1 Selecting a Private Data Source Connection



Administrators have access to the private data source connections created by users. All private data source connections are displayed to Administrators when they view the list of OLAP, JDBC, Web Service, and HTTP data sources from the BI Publisher Administration page.

Private data source connections are distinguished by an **Allowed User** value on the Data Source Administration page as shown in Figure 9–2. Administrators can extend access to other users to a private data source connection by assigning additional user roles to it.

For more information on assigning roles to data sources, see Section 9.1.2, "Granting Access to Data Sources Using the Security Region.".

Figure 9–2 Private Data Source Connection Allowed User

For more information about creating private data source connections, see "Managing Private Data Sources" in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

9.1.2 Granting Access to Data Sources Using the Security Region

When you set up data sources, you can also define security for the data source by selecting which user roles can access the data source.

You must grant access to users for the following:

- A report consumer must have access to the data source to view reports that retrieve data from the data source
- A report designer must have access to the data source to create or edit a data model against the data source

By default, a role with administrator privileges can access all data sources.

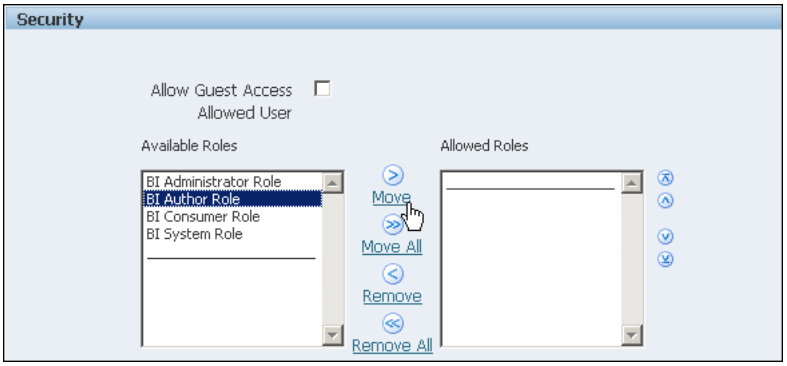
The configuration page for the data source includes a **Security** region that lists all the available roles. You can grant roles access from this page, or you can also assign the data sources to roles from the roles and permissions page.

See Section 3.8, "Configuring Users, Roles, and Data Access" for information.

If this data source must be used in guest reports, then you must also enable guest access here. For more information about guest access see Section 4.2, "Enabling a Guest User."

Figure 9–3 shows the Security region of the data source configuration page.

Figure 9–3 Security Region



9.1.3 About Proxy Authentication

BI Publisher supports proxy authentication for connections to the following data sources:

- Oracle 10*g* database
- Oracle 11*g* database
- Oracle BI Server

For direct data source connections through JDBC and connections through a JNDI connection pool, BI Publisher enables you to select "Use Proxy Authentication". When you select Use Proxy Authentication, BI Publisher passes the user name of the individual user (as logged into BI Publisher) to the data source and thus preserves the client identity and privileges when the BI Publisher server connects to the data source.

Note: Enabling this feature requires additional setup on the database. The database must have Virtual Private Database (VPD) enabled for row-level security.

For more information on Proxy Authentication in Oracle databases, see *Oracle Database Security Guide 10g* or *Oracle Database Security Guide 11g*.

For connections to the Oracle BI Server, Proxy Authentication is required. In this case, proxy authentication is handled by the Oracle BI Server, therefore the underlying database can be any database that is supported by the Oracle BI Server.

9.1.4 Choosing JDBC or JNDI Connection Type

In general, a JNDI connection pool is recommended because it provides the most efficient use of your resources. For example, if a report contains chained parameters, then each time the report is executed, the parameters initiate to open a database session every time.

9.1.5 About Backup Databases

When you configure a JDBC connection to a database, you can also configure a backup database. A backup database can be used in two ways:

- As a true backup when the connection to the primary database is unavailable
- As the reporting database for the primary. To improve performance you can configure your report data models to execute against the backup database only.

To use the backup database in either of these ways, you must also configure the report data model to use it.

See the section "Setting Data Model Properties" in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher* for information on configuring a report data model to use the backup data source.

9.1.6 About Pre Process Functions and Post Process Functions

You can define PL/SQL functions for BI Publisher to execute when a connection to a JDBC data source is created (preprocess function) or closed (postprocess function). The function must return a boolean value. This feature is supported for Oracle databases only.

These two fields enable the administrator to set a user's context attributes before a connection is made to a database and then to dismiss the attributes after the connection is broken by the extraction engine.

The system variable :xdo_user_name can be used as a bind variable to pass the login username to the PL/SQL function calls. Setting the login user context in this way enables you to secure data at the data source level (rather than at the SQL query level).

For example, assume you have defined the following sample function:

```
FUNCTION set_per_process_username (username_in IN VARCHAR2)
   RETURN BOOLEAN IS
     BEGIN
     SETUSERCONTEXT(username_in);
     return TRUE;
   END set_per_process_username
```

To call this function every time a connection is made to the database, enter the following in the **Pre Process Function** field: set_per_process_username(:xdo_user_name)

Another sample usage might be to insert a row to the LOGTAB table every time a user connects or disconnects:

```
CREATE OR REPLACE FUNCTION BIP_LOG (user_name_in IN VARCHAR2, smode IN VARCHAR2)
RETURN BOOLEAN AS
   BEGIN
   INSERT INTO LOGTAB VALUES(user_name_in, sysdate,smode);
   RETURN true;
   END BIP_LOG;
```

In the **Pre Process Function** field enter: BIP_LOG(:xdo_user_name)

As a new connection is made to the database, it is logged in the LOGTAB table. The SMODE value specifies the activity as an entry or an exit. Calling this function as a **Post Process Function** as well returns results such as those shown in Figure 9–4.

Figure 9–4 LOTGTAB Table

NAME	UPDATE_DATE	S_FLAG
oracle	14-MAY-10 09.51.34.000000000	AMStart
oracle	14-MAY-10 10.23.57.000000000	AMFinish
administrator	14-MAY-10 09.51.38.000000000	AMStart
administrator	14-MAY-10 09.51.38.000000000	AMFinish
oracle	14-MAY-10 09.51.42.000000000	AMStart
oracle	14-MAY-10 09.51.42.000000000	AMFinish

9.2 Setting Up a JDBC Connection to the Data Source

The following list shows prerequisites for setting up a JDBC connection to a data source:

- The JDBC driver for the selected database must be available to BI Publisher. If you are using an Oracle database or one of the DataDirect drivers provided by WebLogic Server, then the drivers must be installed in the correct location and there is no further setup required.
- If you plan to use a different version of any of the drivers installed with WebLogic Server, then you can replace the driver file in WL_HOME\server\lib with an updated version of the file or add the new file to the front of your CLASSPATH.

  If you plan to use a third-party JDBC driver that is not installed with WebLogic Server, then you must update the WebLogic Server classpath to include the location of the JDBC driver classes. Edit the commEnv.cmd/sh script in WL_HOME/common/bin and prefix your classes as described in "Modifying the Classpath" in *Oracle Fusion Middleware Command Reference for Oracle WebLogic Server*.

  For more information, see *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server.*

Important: When the JDBC connection is defined, the administrator defines the user that BI Publisher uses to connect to the database. It is the responsibility of the administrator to establish security on the database to allow or disallow actions this user can take on the database schema.

For report consumer access to data that is returned in a report, the administrator and data model developer can establish security, if needed, that can limit the data viewed by a particular BI Publisher user. One method for securing data returned is to use pre-process and post-process function calls to pass the xdo_username. For more information see Section 9.1.6, "About Pre Process Functions and Post Process Functions."

To set up a JDBC connection to a data source:

1. From the Administration page click **JDBC Connection** to display the list of existing JDBC connections.

   Private JDBC connections are also supported. For more information, see "Managing Private Data Sources" in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

2. Click **Add Data Source**.

3. Enter the following fields for the new connection:

   - **Data Source Name** — Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
   - **Driver Type** — Select the database type from the list. When you select a driver type, BI Publisher automatically displays the appropriate Database Driver Class and provides the appropriate Connection String format for your selected database.
   - **Database Driver Class** — This is automatically entered based on your selection for Driver Type. You can update this field if desired.

     For example: oracle.jdbc.OracleDriver or

     hyperion.jdbc.sqlserver.SQLServerDriver
   - **Connection String** — Enter the database connection string.

     Example connection strings:

     – Oracle database

       For an Oracle database (non-RAC) the connect string must have the following format:

       jdbc:oracle:thin:@[host]:[port]:[sid]

       For example: jdbc:oracle:thin:@myhost.us.example.com:1521:prod

     – Oracle RAC database

       To connect to an Oracle RAC database, use the following format:

       jdbc:oracle:thin:@//<host>[:<port>]/<service_name>

       For example: jdbc:oracle:thin:@//myhost.example.com:1521/my_service

     – Microsoft SQL Server

For a Microsoft SQL Server, the connect string must have the following format:

jdbc:hyperion:sqlserver://[hostname]:[port];DatabaseName=[Database-name]

For example:

jdbc:hyperion:sqlserver://myhost.us.example.com:7777;Database-Name=mydatabase

- **Use System User** — This is reserved for connections to the Oracle BI Server.

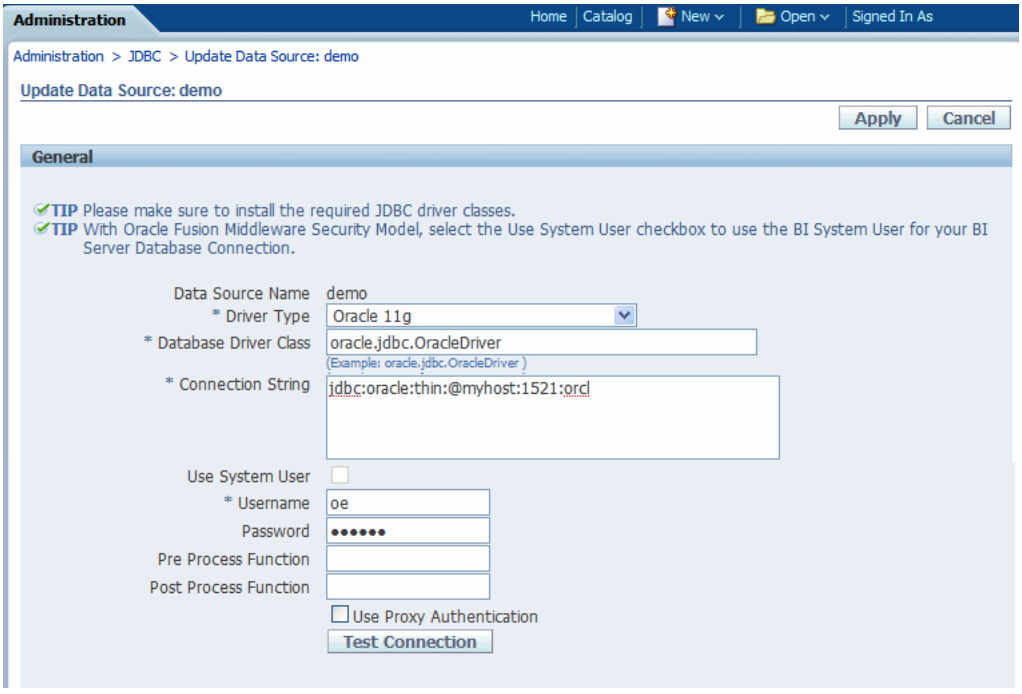
  See Section B.4, "Setting Up a JDBC Connection to the Oracle BI Server."

- **User Name** — Enter the user name required to access the data source on the database.
- **Password** — Enter the password associated with the user name for access to the data source on the database.
- **Pre Process Function** and **Post Process Function** — (Optional) Enter a PL/SQL function to execute when a connection is created (Pre Process) or closed (Post Process). For more information see Section 9.1.6, "About Pre Process Functions and Post Process Functions."
- **Use Proxy Authentication** — Select this box to enable Proxy Authentication. See Section 9.1.3, "About Proxy Authentication" for more information.

4. Click **Test Connection**. A confirmation is displayed.

   Figure 9–5 shows the general settings of the JDBC connection page.

Figure 9–5 JDBC Connection Page



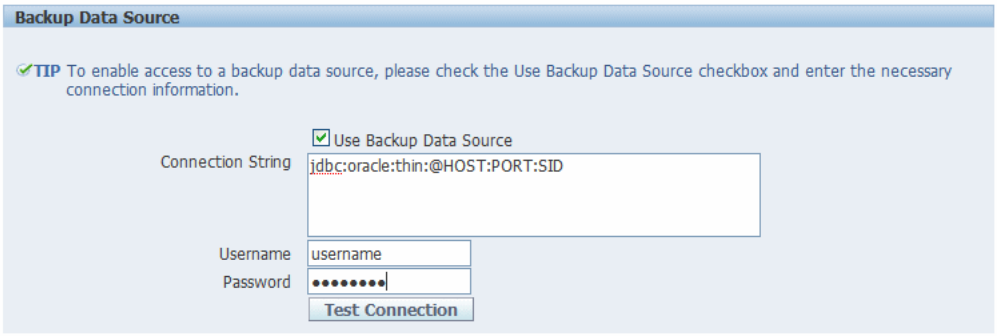
5. (Optional) Enable a backup database for this connection by entering the following:

   - **Use Backup Data Source** — Select this box.

- **Connection String** — Enter the connection string for the backup database.
- **Username** / **Password** — Enter the username and password for this database.
- Click **Test Connection**. A confirmation is displayed.

Figure 9–6 shows the Backup Data Source region of the page.

Figure 9–6 Backup Data Source Region



6. Define security for this data source. Use the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create or view reports from this data source.

   The settings defined here are passed down to the backup data source, if one is defined.

9.3 Setting Up a Database Connection Using a JNDI Connection Pool

BI Publisher supports connecting to the JDBC data source through a connection pool. Using a connection pool increases efficiency by maintaining a cache of physical connections that can be reused. When a client closes a connection, the connection gets placed back into the pool so that another client can use it. A connection pool improves performance and scalability by allowing multiple clients to share a small number of physical connections. You set up the connection pool in your application server and access it through Java Naming and Directory Interface (JNDI).

After you set up the connection pool in your application server, enter the required fields in this page so that BI Publisher can use the pool to establish connections. For information on setting up a connection pool in WebLogic Server, see the chapter "Configuring JDBC Data Sources" in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

To set up a database connection using a JNDI connection pool:

1. From the Administration page click **JNDI Connection** to display the list of existing JNDI connections.
2. Click **Add Data Source**.
3. Enter the following fields for the new connection:
   - **Data Source Name** — Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
   - **JNDI Name** — Enter the JNDI location for the pool. For example, jdbc/BIP11gSource.

- **Use Proxy Authentication** — Select this box to enable Proxy Authentication. See Section 9.1.3, "About Proxy Authentication" for more information.

4. Click **Test Connection**. A confirmation message is displayed.
5. Define security for this data source. Use the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create or view reports from this the data source.

9.4 Setting Up a Connection to an LDAP Server Data Source

To set up a connection to an LDAP data source:

1. From the Administration page select **LDAP Connection** to display the list of existing LDAP connections.
2. Click **Add Data Source**.
3. Enter the following fields for the new connection:
   - Enter the **Data Source Name** — This is the display name that is displayed in the Data Source selection list in the Data Model Editor.
   - Enter the **LDAP Connection URL** for the LDAP server in the format: ldap://hostname:port.
   - Enter the **Username** (for example: cn=admin,cn=users,dc=us,dc=company,dc=com).
   - **Password** — Enter the password if required.
   - Enter the **JNDI Context Factor Class** (for example: com.sun.jndi.ldap.LdapCtxFactory).
4. Click **Test Connection**.
5. Define security for this data source. Use the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create data models from this the data source or view reports that run against this data source.

9.5 Setting Up a Connection to an OLAP Data Source

BI Publisher supports connecting to several types of OLAP databases. Note that to connect to Microsoft SQL Server Analysis Services, BI Publisher must be installed on a supported Windows operating system. For the complete list of supported OLAP databases, see "System Requirements and Certification."

To set up a connection to an OLAP data source:

1. From the Administration page click **OLAP Connection** to display the list of existing OLAP connections.

   Private OLAP connections are also supported. For more information, see "Managing Private Data Sources" in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.
2. Click **Add Data Source**.
3. Enter the following fields for the new connection:
   - **Data Source Name** — Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.

OLAP Type — Select from the list of supported OLAP databases. When you select the type, the OLAP Connection String field is updated with the appropriate connection string format for your selection.

- **OLAP Connection String** — Enter the connection string for the OLAP database. Following are examples for each of the supported OLAP types:
  - Oracle's Hyperion Essbase

    Format: [server]

    Example: myServer.us.example.com
  - Microsoft SQL Server 2000 Analysis Services

    Format: Data Source=[server];Provider=msolap;Initial Catalog=[catalog]

    Example: Data Source=myServer;Provider=msolap;Initial Catalog=VideoStore
  - Microsoft SQL Server 2005 Analysis Services

    Format: Data Source=[server];Provider=msolap.3;Initial Catalog=[catalog]

    Example: Data Source=myServer;Provider=msolap.3;Initial Catalog=VideoStore
  - SAP BW

    Format: ASHOST=[server] SYSNR=[system number] CLIENT=[client] LANG=[language]

    Example: ASHOST=172.16.57.44 SYSNR=01 CLIENT=800 LANG=EN
- **Username** and **Password** for the OLAP database.

4. Click **Test Connection**. A confirmation message is displayed.
5. Define security for this data source. Use the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create or view reports from this the data source.

9.6 Setting Up a Connection to a File Data Source

BI Publisher enables you to use existing XML or Microsoft Excel files created from other sources as input to your BI Publisher reports. To use a file as a data source, it must reside in a directory that BI Publisher can connect to. Set up the connection details to the file data source directory using this page.

To set up a connection to a file data source:

1. From the Administration page click **File** to display the list of existing file sources.
2. Click **Add Data Source**.
3. Enter the following fields for the new data source:
   - **Data Source Name** — Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
   - **Path** — Enter the full path to the top-level directory on your server. Users can access files in this directory and any subdirectories.
4. Define security for this data source. Use the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create or view reports from this data source.

9.7 Setting Up a Connection to a Web Service

BI Publisher supports Web service data sources that return valid XML data.

> **Important:** Additional configuration may be required to access external Web services depending on your system's security. If the WSDL URL is outside the company firewall, see "Section 4.4, "Configuring Proxy Settings."

BI Publisher supports Web services that return both simple and complex data types. You must make the distinction between simple and complex when you define the Web service connection. For more information about each Web service connection type, see Section 9.7.1, "Adding a Simple Web Service" and Section 9.7.2, "Adding a Complex Web Service."

If the Web service is protected by Secure Sockets Layer (SSL), see "Section 4.3, "Configuring BI Publisher for Secure Socket Layer (SSL) Communication."

Private Web Service connections are also supported. For more information, see "Managing Private Data Sources" in *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

Only Basic and Digest authentication is supported for Web service data sources.

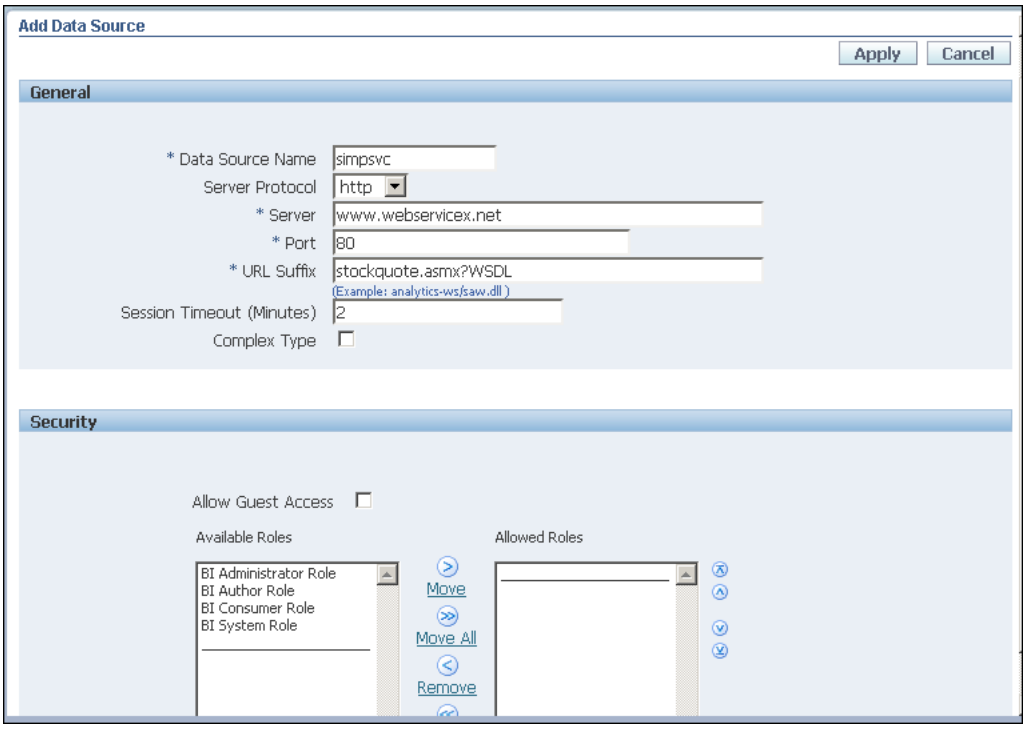
Only document/literal Web services are supported.

9.7.1 Adding a Simple Web Service

To add a Web service as a data source:

1. From the Administration page, click **Web Service Connection** to display the list of existing Web service connections.
2. On the Web Services tab, click **Add Data Source** to display the Add Data Source page as shown in figure Figure 9–7.

Figure 9–7 Creating a Simple Web Service Data Source



3. Enter the following fields for the new connection:
   - **Data Source Name** — Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
   - **Server Protocol** — Select the server protocol.
   - **Server** — Enter the server name.
   - **Port** — Enter the server port.
   - **URL Suffix** — Enter the URL suffix for the web service connection.

     For example, stockquote.asmx?WSDL
   - **(Optional) Session Timeout (Minutes)** — Enter the timeout in minutes. If the BI Publisher server cannot establish a connection to the Web service, the connection attempt will time out after the specified time out period has elapsed.
   - **Complex Type** — Deselect the check box to designate the connection as a simple Web service.
4. Define security for this data source by using the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create or view reports from this data source.
5. Click **Apply** to save the data source connection.

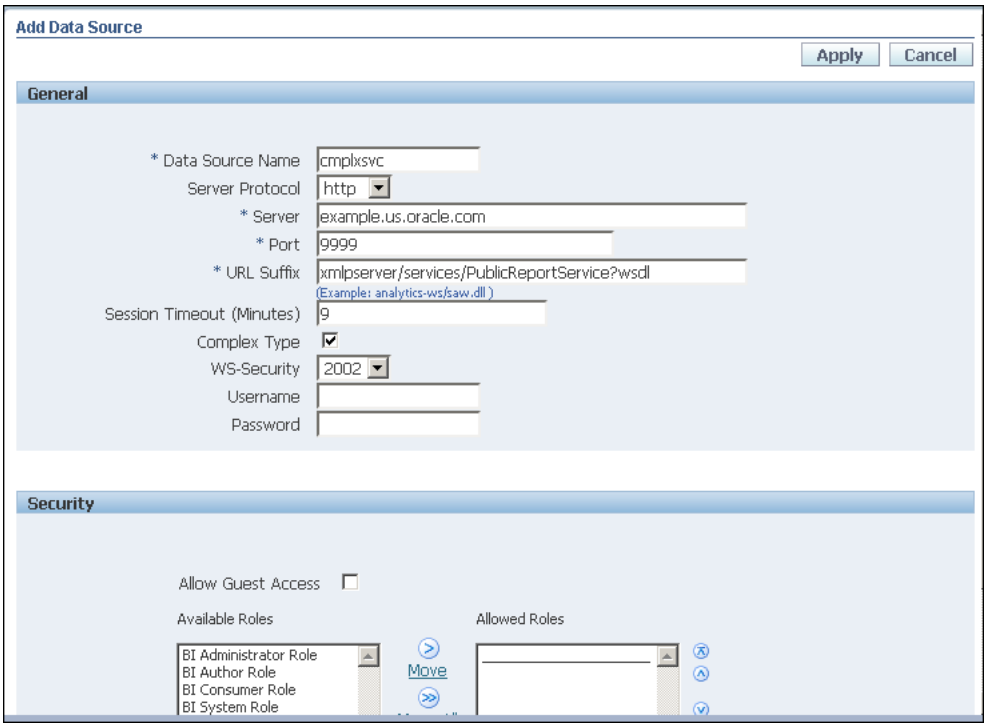
9.7.2 Adding a Complex Web Service

To add a complex Web service as a data source:

1. From the Administration page, click **Web Service Connection** to display the list of existing Web service connections.

2. Click **Add Data Source** to display the Add Data Source page as shown in figure Figure 9–8.

Figure 9–8 Creating a Complex Web Service Data Source



3. Enter the following fields for the new connection:

   - **Data Source Name** — Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
   - **Server Protocol** — Select the server protocol.
   - **Server** — Enter the server name.
   - **Port** — Enter the server port.
   - **URL Suffix** — Enter the URL for the Web service connection.
   - **(Optional) Session Timeout (Minutes)** — Enter the timeout in minutes. If the BI Publisher server cannot establish a connection to the web service, the connection attempt times out after the specified time out period has elapsed.
   - **Complex Type** — Select the check box to designate the connection as a complex Web service.
   - **WS-Security** — Select the security header.
     - 2002 — Enables the "WS-Security" Username Token with the 2002 namespace:

       http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
     - 2004 — Enables the "WS-Security" Username Token with the 2004 namespace:

       http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText

- **Authentication Type** — BI Publisher supports HTTP and SOAP authentication types. SOAP is the default. When HTTP is selected, the user name and password information are passed through HTTP headers. When soap is selected, the user name and password information are passed through XML SOAP envelope headers.
- **Username** — Enter the user name for the web service, if required.
- **Password** — Enter the password for the web service, if required.
- **WSDL protected by HTTP basic auth** — select if access to the WSDL is protected. When the WSDL is protected by user name and password, BI Publisher executes an HTTP call with the username and password to access the WSDL URL. The WSDL can then be downloaded and parsed by BI Publisher.

4. Define security for this data source. Use the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create or view reports from this data source.

   The settings defined here are passed down to the backup data source, if one is defined.

5. Click **Apply**.

9.8 Setting Up a Connection to a HTTP XML Feed

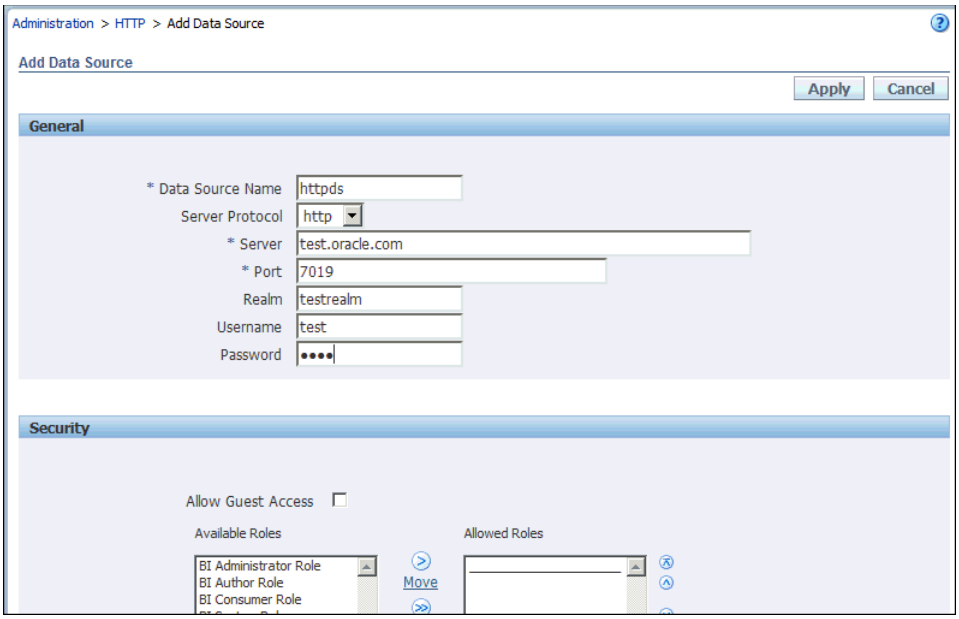
HTTP (XML Feed) data sources enable your data model designers to build data models from RSS and XML feeds over the Web by retrieving data through the HTTP GET method.

Private HTTP XML connections are also supported. For more information, see "Managing Private Data Sources" in *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

To add a HTTP XML as a data source:

1. From the Administration page, click **HTTP Connection** to display the list of existing HTTP connections.
2. Click **Add Data Source** to display the Add Data Source page as shown in Figure 9–9.

Figure 9–9 HTTP Connection Page



3. Enter the following fields for the new connection:
   - **Data Source Name** — Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
   - **Server Protocol** — Select the server protocol.
   - **Server** — Enter the server name.
   - **Port** — Enter the server port.
   - **Realm** — Enter the URL for the web service connection.

     For example:

     xmlpserver/services/v2/SecurityService?wsdl
   - **Username** — Enter the user name required to access the data source on the database.
   - **Password** — Enter the password associated with the user name for access to the data source on the database.
4. Define security for this data source. Use the shuttle buttons to move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles on the Allowed Roles list can create or view reports from this data source.

   The settings defined here are passed down to the backup data source, if one is defined.

9.9 Viewing or Updating a Data Source

To view or update a data source:

1. From the Administration page, select the Data Source type to update.
2. Select the name of the connection to view or update. All fields are updatable. See the appropriate section for setting up the data source type for information on the required fields.

3. Select **Apply** to apply any changes or **Cancel** to exit the update page.

10

Setting Up Delivery Destinations

This chapter describes the setup required to deliver BI Publisher reports to printers, e-mail servers, FTP servers, and WebDav servers. It also describes how to set up the HTTP notification server.

It covers the following topics:

10.1 Configuring Delivery Options

Use the **Delivery Configuration Options** page to set general properties for e-mail deliveries and notifications from BI Publisher and for defining the SSL certificate file.

To configure delivery options:

1. From the Admin page, select **Delivery Configuration**, as shown in Figure 10–1.

Figure 10–1 Delivery Configuration Page

2. Enter the following properties:

   - **SSL Certificate File** — If SSL is enabled for your installation, then you can leave this field empty if you want to use the default certificates built-in with BI Publisher. SSL works with the default certificate if the server uses the certificate signed by a trusted certificate authority such as Verisign. This field is mandatory only if the user uses the SSL with a self-signed certificate. The self-signed certificate means the certificate is signed by a non-trusted certificate authority (usually the user).
   - **E-mail From Address** — Enter the From address to appear on e-mail report deliveries from the BI Publisher server. The default value is bipublisher-report@oracle.com.
   - **Delivery Notification E-mail From Address** — Enter the From address to appear on notifications delivered from the BI Publisher server. The default value is bipublisher-notification@oracle.com.
   - **Success Notification Subject** — Enter the subject line to display for e-mail notification recipients when the report status is Success.
   - **Warning Notification Subject** — Enter the subject line to display for e-mail notification recipients when the report status is Warning.
   - **Failure Notification Subject** — Enter the subject line to display for e-mail notification recipients when the report status is Failed.
   - **Skipped Notification Subject** — Enter the subject line to display for e-mail notification recipients when the report status is Skipped.

10.2 Adding a Printer

Regardless of whether BI Publisher is running on Linux, Unix, or Windows, the printer destination can be any IPP server. The IPP server can be the printer itself, which is the easiest option, but if the printer does not natively support IPP, you can set up a print server that does support IPP (such as CUPS) and connect BI Publisher to the print server and then the print server to the printer. In this print server scenario, the print server can run on any operating system.

To send fax from BI Publisher, you must set up Common Unix Printing Service (CUPS) and the fax4CUPS extension, to enable connection to your fax server from BI Publisher.

The fax set up requires this plugin to the CUPS server on the operating system. Note that the Administration page makes the distinction between a fax and printer server in the UI so that users can pick one or the other or both at runtime. Even though the fax and printer server that the users see can both use a single CUPS server.

For information on setting up CUPS or Windows IPP print servers and how to connect network printers to them, refer to the CUPS or Windows IPP software vendor documentation.

About Printing PDF

PDF is a popular output format for business reports and is printable from viewer software such as Adobe Reader. However, some reports require printing directly from the report server. For example, paychecks and invoices are usually printed as scheduled batch jobs. Some newer printers with PostScript Level 3 compliant Raster Image Processing can natively support PDF documents, but there are still many printers in business use that only support PostScript Level 2 that cannot print PDF documents directly.

To print PDF documents directly from the BI Publisher server if your printer or print server does not support printing PDF, you have the following options:

- Select one of BI Publisher's filters: PDF to PostScript or PDF to PCL.
- Configure a custom, or third-party filter.

After completing all other required fields for the print server, you can schedule reports to print directly from the BI Publisher server to any printer in your system that supports PostScript Level 2.

To set up a printer:

1. From the Admin page select **Printer**. Select **Add Server**.
2. Enter the following required fields:
   - **Server Name** — Enter a unique name. Example: Localprinter
   - **URI** — Enter the Uniform Resource Identifier for the printer. Example: ipp://myhost:631/printers/myprinter
3. Enter a **Filter** (optional).

   A filter enables you to call a conversion utility to convert the PDF generated by BI Publisher to a file format supported by your specific printer type. BI Publisher provides the following filters:

   - PDF to PostScript

     BI Publisher includes a PDF to PostScript filter. This filter converts PDF to PostScript Level 2. Select **PDF to PostScript** from the list to use BI Publisher's predefined filter.

   - PDF to PCL

     To convert PDF to PCL, select **PDF to PCL**. This automatically populates the **Filter Command** field.

     BI Publisher supports the PDF to PCL conversion only for font selection requirements for check printing. For generic printing requirements, use the PDF to PostScript filter.

     You can embed PCL commands into RTF templates to invoke the PCL commands at a specific position on the PCL page; for example, to use a font installed on the printer for routing and account numbers on a check. For more

information, see "Embedding Printer Control Language (PCL) Commands for Check Printing" in *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

You can also call a custom filter using operating system commands.

About Custom Filters

To specify a custom filter, pass the native OS command string with the two placeholders for the input and output filename, {infile} and {outfile}.

This is useful especially if you are trying to call IPP printers directly or IPP printers on Microsoft Internet Information Service (IIS). Unlike CUPS, those print servers do not translate the print file to a format the printer can understand, therefore only limited document formats are supported. With the filter functionality, you can call any of the native OS commands to transform the document to the format that the target printer can understand.

For example, to transform a PDF document to a PostScript format, enter the following PDF to PS command in the **Filter** field:

pdftops {infile} {outfile}

To call an HP LaserJet printer setup on a Microsoft IIS from Linux, you can set Ghostscript as a filter to transform the PDF document into the format that the HP LaserJet can understand. To do this, enter the following Ghostscript command in the **Filter** field:

```
gs -q -dNOPAUSE -dBATCH -sDEVICE=laserjet -sOutputFile={outfile} {infile}
```

For fax servers, you can use the filter to transform the file to Tag Image File Format (TIFF).

4. Optionally enter the following fields if appropriate:
   - Security fields — Username and Password, Authentication Type (None, Basic, Digest) and Encryption Type (None, SSL).
   - Proxy Server fields — Host, Port, User Name, Password, Authentication Type (None, Basic, Digest)

10.3 Adding a Fax Server

To send fax from BI Publisher, you must set up Common Unix Printing Service (CUPS) and the fax4CUPS extension, to enable fax transmissions from BI Publisher.

See the following resources for information about setting up CUPS and the fax4CUPS extension:

To set up fax delivery:

1. From the Admin page select **Fax**. Select **Add Server**.
2. Enter the following required fields:
   - **Server Name** — Enter a unique name. Example: Localprinter
   - **URI** — Enter the Uniform Resource Identifier for the printer. Example: ipp://myhost:631/printers/myprinter
3. Enter a **Filter** (optional).

   A filter enables you to call a conversion utility to convert the PDF generated by BI Publisher to a file format supported by your specific printer type. BI Publisher provides the following filters:

- PDF to PostScript

  BI Publisher includes a PDF to PostScript filter. This filter converts PDF to PostScript Level 2. Select **PDF to PostScript** from the list to use BI Publisher's predefined filter.

- PDF to PCL

  To convert PDF to PCL, select **PDF to PCL**. This automatically populates the **Filter Command** field.

  BI Publisher supports the PDF to PCL conversion only for font selection requirements for check printing. For generic printing requirements, use the PDF to PostScript filter.

  You can embed PCL commands into RTF templates to invoke the PCL commands at a specific position on the PCL page; for example, to use a font installed on the printer for routing and account numbers on a check. For more information, see "Embedding Printer Control Language (PCL) Commands for Check Printing" in *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

You can also call a custom filter using operating system commands.

About Custom Filters

To specify a custom filter, pass the native OS command string with the two placeholders for the input and output filename, {infile} and {outfile}.

This is useful especially if you are trying to call IPP printers directly or IPP printers on Microsoft Internet Information Service (IIS). Unlike CUPS, those print servers do not translate the print file to a format the printer can understand, therefore only limited document formats are supported. With the filter functionality, you can call any of the native OS commands to transform the document to the format that the target printer can understand.

For example, to transform a PDF document to a PostScript format, enter the following PDF to PS command in the **Filter** field:

pdftops {infile} {outfile}

To call an HP LaserJet printer setup on a Microsoft IIS from Linux, you can set Ghostscript as a filter to transform the PDF document into the format that the HP LaserJet can understand. To do this, enter the following Ghostscript command in the **Filter** field:

```
gs -q -dNOPAUSE -dBATCH -sDEVICE=laserjet -sOutputFile={outfile} {infile}
```

For fax servers, you can use the filter to transform the file to Tag Image File Format (TIFF).

4. Optionally enter the following fields if appropriate:

   - Security fields — Username and Password, Authentication Type (None, Basic, Digest) and Encryption Type (None, SSL).
   - Proxy Server fields — Host, Port, User Name, Password, Authentication Type (None, Basic, Digest)

10.4 Adding an E-Mail Server

To add an e-mail server:

1. From the Admin page select Email. This displays the list of servers that have been added. Select **Add Server**.
2. Enter the **Server Name, Host**, and **Port** for the e-mail server.
3. Select a **Secure Connection** method to use for connections with the e-mail server. The options are:
   - None
   - SSL — Use Secure Socket Layer.
   - TLS (Transport Layer Security) — Use TLS when the server supports the protocol; SSL is accepted in the response.
   - TLS Required — If the server does not support TLS, then the connection is not made.
4. Optionally enter the following fields if appropriate:
   - General fields — Port
   - Security fields — Username and Password.

10.5 Adding a WebDAV Server

To add a WebDAV server:

1. From the Admin page select **WebDAV** to display the list of servers that have been added. Select **Add Server**.
2. Enter the **Name** and **Host** for the new server.
3. Optionally enter the following fields if appropriate:
   - General fields — Port
   - Security fields — Authentication Type (None, Basic, Digest) and Encryption Type (None, SSL).
   - Proxy Server fields — Host, Port, User Name, Password, Authentication Type (None, Basic, Digest)

10.6 Adding an HTTP Server

You can register an application URL or postprocess HTTP URL as an HTTP server to send a notification request to after the report has completed. The HTTP notification sent by BI Publisher posts a form data for Job ID, report URL and Job Status to the HTTP Server URL page.

For more information about setting up an HTTP notification to integrate with a third-party application, see the chapter "Setting Up After-Report Triggers" in the *Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Publisher*. For information on enabling an HTTP notification for a scheduled report, see the section "Configuring Notifications" in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

To add an HTTP server:

1. From the Admin page select HTTP to display the list of servers that have been added. Select **Add Server**.
2. Enter a name for the server, and enter the URL. When the report finishes processing, BI Publisher posts form data for Job ID, report URL and Job Status.

3. Enter the Security information, if required. If your server is password protected, enter the Username and Password. Select the Authentication Type: None, Basic, or Digest; and Encryption Type: None or SSL.
4. If the notification is to be sent through a proxy server, enter the fully qualified Host name, the Port, the Username and Password, and Authentication type of the proxy server.

10.7 Adding an FTP Server

Important: If the destination file name supplied to the BI Publisher scheduler contains non-ascii characters, BI Publisher will use UTF-8 encoding to specify the file name to the destination FTP server. Your FTP server must support UTF-8 encoding or the job delivery will fail with "Delivery Failed" error message.

To add an FTP server:

1. From the **Administration** page, under **Delivery**, click **FTP** to display the list of servers that have been added. Click **Add Server**.
2. Enter the following fields for the FTP server:
   - **Server Name** - example: myFTPserver
   - **Host** - example: myhost.company.com
   - **Port** - example: 21
   - **Create files with Part extension when copy is in process** - select this box if you want BI Publisher to create the file on the FTP sever with a .part extension while the file is transferring. The .part extension indicates that the file transfer is not complete. When the file transfer is complete, the file is renamed without the .part extension. If the file transfer does not complete, the file with the .part extension remains on the server.
3. Select the **Use Secure FTP** box to enable Secure FTP (SFTP)
4. Enter a username and Password for the server if required.

10.8 Adding a Content Server

You can deliver documents generated by BI Publisher to your Oracle WebContent Server. BI Publisher's integration with the content server provides the following features:

- At run time, the report consumer can tag the report with Security Group and Account metadata (if applicable) to ensure that the appropriate access rights are applied to the document when delivered.

  See "Adding Destinations" in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Publisher*.

- For documents that require specific custom metadata fields (such as invoice number, customer name, order date), BI Publisher enables the report author to map the custom metadata fields defined in Content Profile Rule Sets to data fields in the data model.

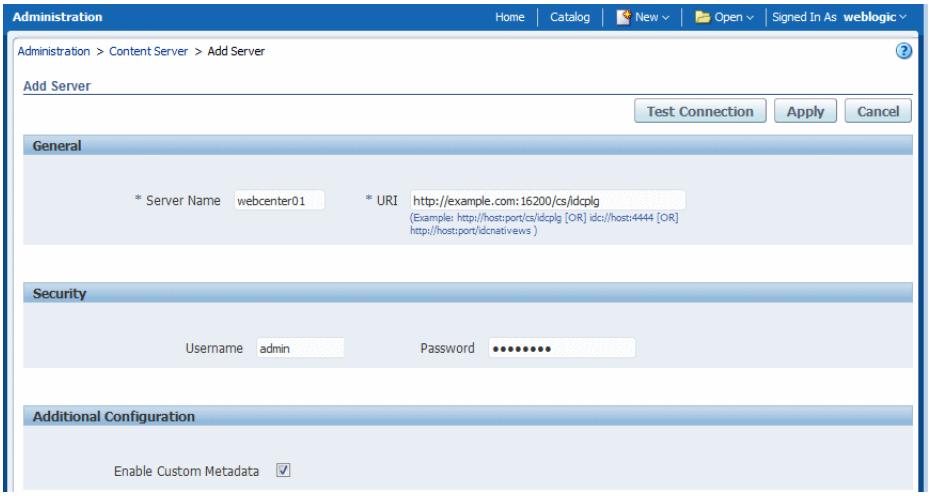
See "Adding Custom Metadata for Oracle WebCenter Content Server" in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

BI Publisher communicates with Oracle WebCenter Content Server using the Remote Intradoc Client (RIDC). The connection protocols therefore follow the standards required by the RIDC. The protocols supported are:

- Intradoc: The Intradoc protocol communicates to the Content Server over the over the Intradoc socket port (typically 4444). This protocol requires a trusted connection between the client and Content Server and will not perform any password validation. Clients that use this protocol are expected to perform any required authentication themselves before making RIDC calls. The Intradoc communication can also be configured to run over SSL.
- HTTP and HTTPS: The HTTP protocol connection requires valid user name and password authentication credentials for each request. You supply the credentials to use for requests in the BI Publisher Administration page.
- JAX-WS: The JAX-WS protocol is supported only in Oracle WebCenter Content 11*g* with a properly configured Content Server instance and the RIDC client installed. JAX-WS is not supported outside this environment.

  For more information about these protocols, see "Using RIDC to Access Content Server" in *Oracle Fusion Middleware Developing with Oracle WebCenter Content*.

Figure 10–2 Content Server Setup Page



To set up a connection to a content server as a delivery destination:

1. From the **Administration** page, under **Delivery**, click **Content Server** to display the list of servers that have been added. Click **Add Server**.
2. Enter the **Server Name**, for example: contentserver01.
3. Enter the connection **URI** for your content server. The URI can take any of the following supported protocols:
   - HTTP/HTTPS - Specifies the URL to the Content Server CGI path.

     For example:

     – http://localhost:16200/cs/idcplg

     – https://localhost:16200/cs/idcplg

- Intradoc: - The Intradoc protocol communicates to the Content Server over the Intradoc socket port (typically 4444). The IDC protocol also supports communication over SSL. For example:
  - idc://host:4444
  - idcs://host:4443
- JAX-WS - Uses the JAX-WS protocol to connect to the Content Server.

  For example:
  - http://wlsserver:16200/idcnativews

4. To enable the inclusion of custom metadata with your report documents delivered to the content server, select the **Enable Custom Metadata** box. This option must be selected to enable the custom metadata options in the Data Model Editor and the Scheduler.

10.9 Adding a Common UNIX Printing System (CUPS) Server

See Section 10.2, "Adding a Printer" for information about when you must configure CUPS.

To add a CUPS server:

1. From the Admin page, select CUPS to display the list of servers that have been added. Select **Add Server**.
2. Enter the **Server Name** and **Host** and **Port** for the CUPS server.

11

Defining Run-Time Configurations

This chapter describes processing properties for BI Publisher including PDF document security, FO processing, font mapping, and specific properties for each output type.

It covers the following topics:

11.1 Setting Run-Time Properties

The Runtime Configuration page enables you to set run-time properties at the server level. These same properties can also be set at the report level, from the report editor's Properties dialog. (See the "Defining Report Properties" section in *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.) If different values are set for a property at each level, then report level takes precedence.

11.2 PDF Output Properties

Table 11–1 describes the properties that are available for PDF output.

Table 11–1 PDF Output Properties

Property Name	Description
Compress PDF output	**Default:** true **Description**: Specify "true" or "false" to control compression of the output PDF file. **Internal Name**: `pdf-compression`
Hide PDF viewer's menu bars	**Default**: false **Description**: Specify "true" to hide the viewer application's menu bar when the document is active. The menu bar option is only effective when using the Export button, which displays the output in a standalone Acrobat Reader application outside of the browser. **Internal Name**: `pdf-hide-menubar`
Hide PDF viewer's tool bars	**Default**: false **Description**: Specify "true" to hide the viewer application's toolbar when the document is active. **Internal Name**: `pdf-hide-toolbar`
Replace smart quotes	**Default**: true **Description**: Set to "false" if you do not want curly quotes replaced with straight quotes in the PDF output. **Internal Name**: `pdf-replace-smartquotes`
Use only one shared resources object for all pages	**Default**: true **Description**: The default mode of BI Publisher creates one shared resources object for all pages in a PDF file. This mode has the advantage of creating an overall smaller file size. However, the disadvantages are the following: ■ Viewing may take longer for a large file with many SVG objects ■ If you choose to break up the file by using Adobe Acrobat to extract or delete portions, then the edited PDF files are larger because the single shared resource object (that contains all of the SVG objects for the entire file) is included with each extracted portion. Setting this property to "false" creates a resource object for each page. The file size is larger, but the PDF viewing is faster and the PDF can be broken up into smaller files more easily. **Internal Name**: `pdf-use-one-resources`
PDF Navigation Panel Initial View	**Default**: Bookmarks Open **Description**: Controls the navigation panel view that is presented when a user first opens a PDF report. The following options are supported: ■ Panels Collapsed - displays the PDF document with the navigation panel collapsed. ■ Bookmarks Open (default) - displays the bookmark links for easy navigation. ■ Pages Open - displays a clickable thumbnail view of each page of the PDF. **Internal Name**: `pdf-pagemode`

11.3 PDF Security Properties

Table 11–2 describes the properties that control the security settings for the output PDF documents.

Table 11–2 PDF Security Properties

Property Name	Description
Enable PDF Security	**Default**: false **Description**: If you specify "true," then the output PDF file is encrypted. You can then also specify the following properties: ▪ Open document password ▪ Modify permissions password ▪ Encryption Level **Internal Name**: `pdf-security`
Open document password	**Default**: N/A **Description**: This password is required for opening the document. It enables users to open the document only. This property is enabled only when "Enable PDF Security" is set to "true". Note that BI Publisher follows Adobe's password restrictions. The password must contain only Latin 1 characters and must be no more than 32 bytes long. **Internal Name**: `pdf-open-password`
Modify permissions password	**Default**: N/A **Description**: This password enables users to override the security setting. This property is effective only when "Enable PDF Security" is set to "true". Note that BI Publisher follows the Adobe's password restrictions. The password must contain only Latin 1 characters and must be no more than 32 bytes long. **Internal Name**: `pdf-permissions-password`

Table 11–2 (Cont.) PDF Security Properties

Property Name	Description
Encryption level	**Default**: 2 - high **Description**: Specify the encryption level for the output PDF file. The possible values are: ■ 0: Low (40-bit RC4, Acrobat 3.0 or later) ■ 1: Medium (128-bit RC4, Acrobat 5.0 or later) ■ 2: High (128-bit AES, Acrobat 7.0 or later) This property is effective only when "Enable PDF Security" is set to "true". When Encryption level is set to 0, you can also set the following properties: ■ Disable printing ■ Disable document modification ■ Disable context copying, extraction, and accessibility ■ Disable adding or changing comments and form fields When Encryption level is set to 1 or higher, the following properties are available: ■ Enable text access for screen readers ■ Enable copying of text, images, and other content ■ Allowed change level ■ Allowed printing level **Internal Name**: `pdf-encryption-level`
Disable document modification	**Default**: false **Description**: Permission available when "Encryption level" is set to 0. When set to "true", the PDF file cannot be edited. **Internal Name**: `pdf-no-changing-the-document`
Disable printing	**Default**: false **Description**: Permission available when "Encryption level" is set to 0. When set to "true", printing is disabled for the PDF file. **Internal Name**: `pdf-no-printing`
Disable adding or changing comments and form fields	**Default**: false **Description**: Permission available when "Encryption level" is set to 0. When set to "true", the ability to add or change comments and form fields is disabled. **Internal Name**: `pdf-no-accff`
Disable context copying, extraction, and accessibility	**Default**: false **Description**: Permission available when "Encryption level" is set to 0. When set to "true", the context copying, extraction, and accessibility features are disabled. **Internal Name**: `pdf-no-cceda`
Enable text access for screen readers	**Default**: true **Description**: Permission available when "Encryption level" is set to 1 or higher. When set to "true", text access for screen reader devices is enabled. **Internal Name**: `pdf-enable-accessibility`

Table 11–2 (Cont.) PDF Security Properties

Property Name	Description
Enable copying of text, images, and other content	**Default**: false **Description**: Permission available when "Encryption level" is set to 1 or higher. When set to "true", copying of text, images, and other content is enabled. **Internal Name**: `pdf-enable-copying`
Allowed change level	**Default**: 0 **Description**: Permission available when "Encryption level" is set to 1 or higher. Valid Values are: ■ 0: none ■ 1: Allows inserting, deleting, and rotating pages ■ 2: Allows filling in form fields and signing ■ 3: Allows commenting, filling in form fields, and signing ■ 4: Allows all changes except extracting pages **Internal Name**: `pdf-changes-allowed`
Allowed printing level	**Default**: 0 **Description**: Permission available when "Encryption level" is set to 1 or higher. Valid values are: ■ 0: None ■ 1: Low resolution (150 dpi) ■ 2: High resolution **Internal Name**: `pdf-printing-allowed`

11.4 PDF Digital Signature Properties

Table 11–3 describes the properties that should only be set at the report level to enable digital signature for a report and to define the placement of the signature in the output PDF document. For more information on how to enable digital signature for your output PDF documents, see Chapter 6, "Implementing a Digital Signature."

Note that to implement digital signature for a report based on a PDF layout template or an RTF layout template, you must set the property **Enable Digital Signature** to "True" for the report.

You also must set the appropriate properties to place the digital signature in the desired location on your output report. Your choices for placement of the digital signature depend on the template type. The choices are as follows:

- (PDF only) Place the digital signature in a specific field by setting the **Existing signature field name** property.
- (RTF and PDF) Place the digital signature in a general location of the page (top left, top center, or top right) by setting the **Signature field location** property.
- (RTF and PDF) Place the digital signature in a specific location designated by x and y coordinates by setting the **Signature field x coordinate** and **Signature field y coordinate** properties.

  If you choose this option, you can also set **Signature field width** and **Signature field height** to define the size of the field in your document.

Table 11–3 PDF Digital Signature Properties

Property Name	Description
Enable Digital Signature	**Default**: false **Description**: Set this to "true" to enable digital signature for the report. **Internal Name**: `signature-enable`
Existing signature field name	**Default**: N/A **Description**: This property applies to PDF layout templates only. If the report is based on a PDF template, then you can enter a field from the PDF template in which to place the digital signature. For more information about defining a field for the signature in a PDF template, see "Adding or Designating a Field for a Digital Signature" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*. **Internal Name**: `signature-field-name`
Signature field location	**Default**: N/A **Description**: This property can apply to RTF or PDF layout templates. This property provides a list that contains the following values: Top Left, Top Center, Top Right. Choose one of these general locations and BI Publisher inserts the digital signature to the output document, sized and positioned appropriately. If you choose to set this property, do not enter X and Y coordinates or width and height properties. **Internal Name**: `signature-field-location`
Signature field X coordinate	**Default**: 0 **Description**: This property can apply to RTF or PDF layout templates. Using the left edge of the document as the zero point of the X axis, enter the position in points that you want the digital signature to be placed from the left. For example, if you want the digital signature to be placed horizontally in the middle of an 8.5 inch by 11 inch document (that is, 612 points in width and 792 points in height), enter 306. **Internal Name**: `signature-field-pos-x`
Signature field Y coordinate	**Default**: 0 **Description**: This property can apply to RTF or PDF layout templates. Using the bottom edge of the document as the zero point of the Y axis, enter the position in points that you want the digital signature to be placed from the bottom. For example, if you want the digital signature to be placed vertically in the middle of an 8.5 inch by 11 inch document (that is, 612 points in width and 792 points in height), enter 396. **Internal Name**: `signature-field-pos-y`
Signature field width	**Default**: 0 **Description**: Enter in points (72 points equal one inch) the desired width of the inserted digital signature field. This applies only if you are also setting the properties **Signature field x coordinate** and **Signature field Y coordinate**. **Internal Name**: `signature-field-width`

Table 11–3 (Cont.) PDF Digital Signature Properties

Property Name	Description
Signature field height	**Default**: 0 **Description**: Enter in points (72 points equal one inch) the desired height of the inserted digital signature field. This applies only if you are also setting the properties **Signature field x coordinate** and **Signature field Y coordinate**. **Internal Name**: `signature-field-height`

11.5 PDF/A Output Properties

Set the properties described in Table 11–4 to configure PDF/A output. For more information on PDF/A output see the section "Generating PDF/A and PDF/X Output" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

Table 11–4 PDF/A Output Properties

Property Name	Description
PDF/A ICC Profile Data	**Default**: Default profile data provided by JVM **Description**: The name of the ICC profile data file, for example: CoatedFOGRA27.icc The ICC (International Color Consortium) profile is a binary file describing the color characteristics of the environment where this PDF/A file is intended to be displayed. (For more information, see the article at http://en.wikipedia.org/wiki/ICC_profile). The ICC profile that you select must have a major version below 4. To use a specific profile data file other than the default settings in the JVM, obtain the file and place it under `<bi publisher repository>/Admin/Configuration`. When you set this property, you must also set a value for PDF/A ICC Profile Info (`pdfa-icc-profile-info`). **Internal Name**: `pdfa-icc-profile-data`
PDF/A ICC Profile Info	**Default**: sRGB IEC61966-2.1 **Description**: ICC profile information (required when pdfa-icc-profile-data is specified) **Internal Name**: `pdfa-icc-profile-info`
PDF/A file identifier	**Default**: Automatically generated file identifier **Description**: One or more valid file identifiers set in the xmpMM:Identifier field of the metadata dictionary. To specify more than one identifier, separate values with a comma (,). **Internal Name**: `pdfa-file-identifier`
PDF/A document ID	**Default:** None **Description**: Valid document ID. The value is set in the xmpMM:DocumentID field of the metadata dictionary. **Internal Name**: `pdfa-document-id`
PDF/A version ID	**Default**: None **Description**: Valid version ID. The value is set in the xmpMM:VersionID field of the metadata dictionary. **Internal Name**: `pdfa-version-id`

Table 11–4 (Cont.) PDF/A Output Properties

Property Name	Description
PDF/A rendition class	**Default**: None **Description**: Valid rendition class. The value is set in the xmpMM:RenditionClass field of the metadata dictionary. **Internal Name**: `pdfa-rendition-class`

11.6 PDF/X Output Properties

Set the properties described in Table 11–5 to configure PDF/X output. The values that you set for these properties will depend on the printing device. Note the following restrictions on other PDF properties:

- `pdf-version` - value above 1.4 is not allowed for PDF/X-1a output
- `pdf-security` - must be set to False
- `pdf-encryption-level` - must be set to 0
- `pdf-font-embedding` - must be set to true

For more information on PDF/X output see the section "Generating PDF/A and PDF/X Output" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

Table 11–5 PDF/X Output Properties

Property Name	Description
PDF/X ICC Profile Data	**Default**: None **Description**: (Required) The name of the ICC profile data file, for example: CoatedFOGRA27.icc. The ICC (International Color Consortium) profile is a binary file describing the color characteristics of the intended output device. (For more information, see the article at http://en.wikipedia.org/wiki/ICC_profile). For production environments, the color profile may be provided by your print vendor or by the printing company that prints the generated PDF/X file. The file must be placed under `<bi publisher repository>/Admin/Configuration`. Profile data is also available from Adobe (http://www.adobe.com/support) or colormanagement.org (http://www.colormanagement.org/). **Internal Name**: `pdfx-dest-output-profile-data`
PDF/X output condition identifier	**Default**: None **Description**: (Required) The name of one of the standard printing conditions registered with ICC (International Color Consortium). The list of standard CMYK printing conditions to use with PDF/X-1a is provided on the following ICC website: http://www.color.org/chardata/drsection1.xalter. The value that you enter for this property is a valid "Reference name," for example: FOGRA43. Choose the appropriate value for the intended printing environment. This name is often used to guide automatic processing of the file by the consumer of the PDF/X document, or to inform the default settings in interactive applications. **Internal Name**: `pdfx-output-condition-identifier`

Table 11–5 (Cont.) PDF/X Output Properties

Property Name	Description
PDF/X output condition	**Default**: None **Description**: A string describing the intended printing condition in a form that will be meaningful to a human operator at the site receiving the exchanged file. The value is set in OutputCondition field of OutputIntents dictionary. **Internal Name**: `pdfx-output-condition`
PDF/X registry name	**Default**: http://www.color.org **Description**: A registry name. Set this property when the `pdfx-output-condition-identifier` is set to a characterization name that is registered in a registry other than the ICC registry. **Internal Name**: `pdfx-registry-name`
PDF/X version	**Default**: PDF/X-1a:2003 **Description**: The PDF/X version set in GTS_PDFXVersion and GTS_PDFXConformance fields of Info dictionary. PDF/X-1a:2003 is the only value currently supported. **Internal Name**: `pdfx-version`

11.7 RTF Output Properties

Table 11–6 describes the properties that control RTF output files.

Table 11–6 RTF Output Properties

Property Name	Description
Enable change tracking	**Default**: false **Description**: Set to "true" to enable change tracking in the output RTF document. **Internal Name**: `rtf-track-changes`
Protect document for tracked changes	**Default**: false **Description**: Set to "true" to protect the document for tracked changes. **Internal Name**: `rtf-protect-document-for-tracked-changes`
Default font	**Default**: Arial:12 **Description**: Use this property to define the font style and size in RTF output when no other font has been defined. This is particularly useful to control the sizing of empty table cells in generated reports. Enter the font name and size in the following format <FontName>:<size> for example: Arial:12. Note that the font you choose must be available to the BI Publisher processing engine at runtime. See Section 11.17, "Defining Font Mappings" for information about installing fonts for the BI Publisher server and also for the list of fonts predefined for BI Publisher. **Internal Name**: `rtf-output-default-font`

11.8 HTML Output Properties

Table 11–7 describes the properties that control HTML output files.

Table 11–7 HTML Output Properties

Property Name	Description
Show header	**Default**: true **Description**: Set to "false" to suppress the template header in HTML output. **Internal Name**: `html-show-header`
Show footer	**Default**: true **Description**: Set to "false" to suppress the template footer in HTML output. **Internal Name**: `html-show-footer`
Replace smart quotes	**Default**: true **Description**: Set to "false" if you do not want curly quotes replaced with straight quotes in the HTML output. **Internal Name**: `html-replace-smartquotes`
Character set	**Default**: UTF-8 **Description**: Specifies the output HTML character set. **Internal Name**: `html-output-charset`
Make HTML output accessible	**Default**: false **Description**: Specify true if you want to make the HTML output accessible. **Internal Name**: `make-accessible`
Use percentage width for table columns	**Default**: true **Description**: Set this property to true to render table columns according to a percentage value of the total width of the table rather than as a value in points. This property is especially useful if the browser renders tables with extremely wide columns. Setting this property to true improves the readability of the tables. **Internal Name**: `html-output-width-in-percentage`
View Paginated	**Default**: false **Description**: When set to true, HTML output will render in the report viewer with pagination features. These features include: ▪ Generated table of contents ▪ Navigation links at the top and bottom of the page ▪ Ability to skip to a specific page within the HTML document ▪ Search for strings within the HTML document using the browser's search capability ▪ Zoom in and out on the HTML document using the browser's zoom capability Note that these features are supported for online viewing through the report viewer only.

11.9 FO Processing Properties

Table 11–8 describes the properties that control FO processing.

Table 11–8 FO Processing Properties

Property Name	Description
Use BI Publisher's XSLT processor	**Default**: true **Description**: Controls BI Publisher's parser usage. If set to false, then XSLT is not parsed. **Internal Name**: `xslt-xdoparser`
Enable scalable feature of XSLT processor	**Default**: false **Description**: Controls the scalable feature of the XDO parser. The property "Use BI Publisher's XSLT processor" must be set to "true" for this property to be effective. **Internal Name**: `xslt-scalable`
Enable XSLT runtime optimization	**Default**: true **Description**: When set to "true", the overall performance of the FO processor is increased and the size of the temporary FO files generated in the temp directory is significantly decreased. Note that for small reports (for example 1-2 pages) the increase in performance is not as marked. To further enhance performance when you set this property to true, it is recommended that you set the property **Extract attribute sets** to "false". See Section 11.10, "RTF Template Properties." **Internal Name**: `xslt-runtime-optimization`
Enable XPath Optimization	**Default**: false **Description**: When set to "true", the XML data file is analyzed for element frequency. The information is then used to optimize XPath in XSL. **Internal Name**: `xslt-xpath-optimization`
Pages cached during processing	**Default**: 50 **Description**: This property is enabled only when you have specified a Temporary Directory (under General properties). During table of contents generation, the FO Processor caches the pages until the number of pages exceeds the value specified for this property. It then writes the pages to a file in the Temporary Directory. **Internal Name**: `system-cache-page-size`
Bidi language digit substitution type	**Default**: National **Description**: Valid values are "None" and "National". When set to "None", Eastern European numbers are used. When set to "National", Hindi format (Arabic-Indic digits) is used. This setting is effective only when the locale is Arabic, otherwise it is ignored. **Internal Name**: `digit-substitution`
Disable variable header support	**Default**: false **Description**: If "true", prevents variable header support. Variable header support automatically extends the size of the header to accommodate the contents. **Internal Name**: `fo-prevent-variable-header`

Table 11–8 (Cont.) FO Processing Properties

Property Name	Description
Add prefix to IDs when merging FO	**Default**: false **Description**: When merging multiple XSL-FO inputs, the FO Processor automatically adds random prefixes to resolve conflicting IDs. Setting this property to "true" disables this feature. **Internal Name**: `fo-merge-conflict-resolution`
Enable multithreading	**Default**: false **Description**: If you have a multiprocessor machine or a machine with a dual-core single processor, you may be able to achieve faster document generation by setting this option to True. **Internal Name**: `fo-multi-threads`
Disable external references	**Default**: true **Description**: A "true" setting (default) disallows the importing of secondary files such as subtemplates or other XML documents during XSL processing and XML parsing. This increases the security of the system. Set this to "false" if the report or template calls external files. **Internal Name**: `xdk-secure-io-mode`
FO Parsing Buffer Size	**Default**: 1000000 **Description**: Sets the size of the buffer for the FO Processor. When the buffer is full, the elements from the buffer are rendered in the report. Reports with large tables or pivot tables that require complex formatting and calculations may require a larger buffer to properly render those objects in the report. Increase the size of the buffer at the report level for these reports. Note that increasing this value affects the memory consumption of the system. **Internal Name**: `fo-chunk-size`
Enable XSLT runtime optimization for sub-template	**Default**: true **Note:** The default is true on the BI Publisher server. If you call the FOProcessor directly, the default is false. **Description**: Provides an option to perform XSL import in FOProcessor before passing only one XSL to XDK for further processing. This allows xslt-optimization to be applied to the entire main XSL template which already includes all its subtemplates. **Internal Name**: `xslt-do-import`
Enable PPTX native chart support	**Default**: false **Description**: This property applies to PowerPoint 2007 output. When set to true, charts in PowerPoint 2007 output are rendered as native PowerPoint (PPTX) charts. When set to false, the chart is rendered as an embedded PNG image. **Internal Name**: `pptx-native-chart`

Table 11–8 (Cont.) FO Processing Properties

Property Name	Description
Report Timezone	**Default**: User **Description**: Valid values: User or JVM. When set to User, BI Publisher uses the User-level Report Time Zone setting for reports. The User Report Time Zone is set in the user's Account Settings. When set to JVM, BI Publisher uses the server JVM timezone setting for all users' reports. All reports therefore display the same time regardless of individual user settings. This setting can be overridden at the report level. **Internal Name**: `fo-report-timezone`

11.10 RTF Template Properties

Table 11–9 describes the properties that control RTF templates.

Table 11–9 RTF Template Properties

Property Name	Description
Extract attribute sets	**Default**: Auto **Description**: The RTF processor automatically extracts attribute sets within the generated XSL-FO. The extracted sets are placed in an extra FO block, which can be referenced. This improves processing performance and reduces file size. Valid values are: ▪ Enable - extract attribute sets for all templates and subtemplates ▪ Auto - extract attribute sets for templates, but not subtemplates ▪ Disable - do not extract attribute sets **Internal Name**: `rtf-extract-attribute-sets`
Enable XPath rewriting	**Default**: true **Description**: When converting an RTF template to XSL-FO, the RTF processor automatically rewrites the XML tag names to represent the full XPath notations. Set this property to "false" to disable this feature. **Internal Name**: `rtf-rewrite-path`
Characters used for checkbox	**Default**: Albany WT J;9746;9747/A **Description**: The BI Publisher default PDF output font does not include a glyph to represent a checkbox. If the template contains a checkbox, use this property to define a Unicode font for the representation of checkboxes in the PDF output. You must define the Unicode font number for the "checked" state and the Unicode font number for the "unchecked" state using the following syntax: `fontname;<unicode font number for true value's glyph >;<unicode font number for false value's glyph>` Example: Albany WT J;9746;9747/A Note that the font that you specify must be made available to BI Publisher at runtime. **Internal Name**: `rtf-checkbox-glyph`

11.11 PDF Template Properties

Table 11–10 describes the properties that control PDF templates.

Table 11–10 PDF Template Properties

Property Name	Description
Remove PDF fields from output	**Default**: false **Description**: Specify "true" to remove PDF fields from the output. When PDF fields are removed, data entered in the fields cannot be extracted. For more information, see "Setting Fields as Updatable or Read Only" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* **Internal Name**: `remove-pdf-fields`
Set all fields as read only in output	**Default**: true **Description**: By default, BI Publisher sets all fields in the output PDF of a PDF template to be read only. If you want to set all fields to be updatable, set this property to "false". For more information, see "Creating PDF Templates" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*. **Internal Name**: `all-field-readonly`
Maintain each field's read only setting	**Default**: false **Description**: Set this property to "true" if you want to maintain the "Read Only" setting of each field as defined in the PDF template. This property overrides the settings of "Set all fields as read only in output." For more information, see "Creating PDF Templates" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*. **Internal Name**: `all-fields-readonly-asis`

11.12 Flash Template Properties

Table 11–11 describes the properties that control Flash templates.

Table 11–11 Flash Template Properties

Property Name	Description
Page width of wrapper document	**Default**: 792 **Description**: Specify in points the width of the output PDF document. The default is 792, or 11 inches. **Internal Name**: `flash-page-width`
Page height of wrapper document	**Default**: 612 **Description**: Specify in points the height of the output PDF document. The default is 612, or 8.5 inches. **Internal Name**: `flash-page-height`

Table 11–11 (Cont.) Flash Template Properties

Property Name	Description
Start x position of Flash area in PDF	**Default**: 18 **Description**: Using the left edge of the document as the 0 axis point, specify in points the beginning horizontal position of the Flash object in the PDF document. The default is 18, or .25 inch. **Internal Name**: `flash-startx`
Start y position of Flash area in PDF	**Default**: 18 **Description**: Using the upper left corner of the document as the 0 axis point, specify in points the beginning vertical position of the Flash object in the PDF document. The default is 18, or .25 inch. **Internal Name**: `flash-starty`
Width of Flash area	**Default**: Same as flash width in points in swf **Description**: Enter in points the width of the area in the document for the Flash object to occupy. The default is the width of the SWF object. **Internal Name**: `flash-width`
Height of Flash area	**Default**: Same as flash height in points in swf **Description**: Enter in points the height of the area in the document for the Flash object to occupy. The default is the height of the SWF object. **Internal Name**: `flash-height`

11.13 CSV Output Properties

Table 11–12 describes the properties that control comma-delimited value output.

Table 11–12 CSV Output Properties

Property Name	Description
CSV delimiter	**Default**: , **Description**: Specifies the character used to delimit the data in comma-separated value output. Other options are: Semicolon (;), Tab (\t) and Pipe (|).
Remove leading and trailing white space	**Default**: false **Description**: Specify "True" to remove leading and trailing white space between data elements and the delimiter.

11.14 Excel 2007 Output Properties

Table 11–13 describes the properties that control Excel 2007 output.

Table 11–13 Excel 2007 Output Properties

Property Name	Description
Show grid lines	**Default**: false **Description**: Set to true to show the Excel table grid lines in the report output.

Table 11–13 (Cont.) Excel 2007 Output Properties

Property Name	Description
Page break as a new sheet	**Default**: true **Description**: When set to "True" a page break that is specified in the report template generates a new sheet in the Excel workbook.
Minimum column width	**Default**: 3 (in points, 0.04 inch) **Description**: When the column width is less than the specified minimum and it contains no data, the column is merged with the preceding column. The value must be set in points. The valid range for this property is 0.5 to 20 points.
Minimum row height	**Default**: 1 (in points, 0.01 inch) **Description**: When the row height is less than the specified minimum and it contains no data, the row is removed. The value must be set in points. The valid range for this property is .001 to 5 points.

11.15 All Outputs Properties

The properties in Table 11–14 applies to all outputs.

Table 11–14 All Outputs

Property Name	Description
Hide version number in output	**Default**: false **Description**: Some report output documents display Oracle BI Publisher in the document properties. For example, PDF documents identify Oracle BI Publisher as the PDF Producer in the properties for the document. If you do not want to include the version of BI Publisher that generated the document (for example, Oracle BI Publisher 11.1.1.4.0), then set this property to true.
Use 11.1.1.5 compatibility mode	Reserved. Do not update unless instructed by Oracle.

11.16 Memory Guard & Data Model Properties

These properties are described in Appendix F, "Enabling Memory Guard Features."

11.17 Defining Font Mappings

BI Publisher's Font Mapping feature enables you to map base fonts in RTF or PDF templates to target fonts to be used in the published document. Font Mappings can be specified at the site or report level. Font mapping is performed only for PDF output and PowerPoint output.

There are two types of font mappings:

- RTF Templates - for mapping fonts from RTF templates and XSL-FO templates to PDF and PowerPoint output fonts
- PDF Templates - for mapping fonts from PDF templates to different PDF output fonts.

11.17.1 Making Fonts Available to BI Publisher

BI Publisher provides a set of Type1 fonts and a set of TrueType fonts. You can select any of the fonts in these sets as a target font with no additional setup required. For a

list of the predefined fonts see Section 11.17.4, "BI Publisher's Predefined Fonts."

The predefined fonts are located in $JAVA_HOME/jre/lib/fonts. To map to another font, place the font in this directory to make it available to BI Publisher at run time. If the environment is clustered, then you must place the font on every server.

11.17.2 Setting Font Mapping at the Site Level or Report Level

A font mapping can be defined at the site level or the report level:

- To set a mapping at the site level, select the Font Mappings link from the Administration page.
- To set a mapping at the report level, view the Properties for the report, then select the Font Mappings tab. These settings apply to the selected report only.

The report-level settings take precedence over the site-level settings.

11.17.3 Creating a Font Mapping

From the **Administration** page, under **Runtime Configuration**, select **Font Mappings**.

To create a Font Mapping:

- Under RTF Templates or PDF Templates, select **Add Font Mapping**.
- Enter the following on the **Add Font Mapping** page:
  - Base Font - enter the font family to map to a new font. Example: Arial
  - Select the **Style**: Normal or Italic (Not applicable to PDF Template font mappings)
  - Select the **Weight**: Normal or Bold (Not applicable to PDF Template font mappings)
  - Select the **Target Font Type**: Type 1 or TrueType
  - Enter the **Target Font**

    If you selected TrueType, you can enter a specific numbered font in the collection. Enter the **TrueType Collection (TTC) Number** of the desired font.

    For a list of the predefined fonts see Section 11.17.4, "BI Publisher's Predefined Fonts."

11.17.4 BI Publisher's Predefined Fonts

The following Type1 fonts are built-in to Adobe Acrobat and BI Publisher provides a mapping for these fonts by default. You can select any of these fonts as a target font with no additional setup required.

The Type1 fonts are listed in Table 11–15.

Table 11–15 Type 1 Fonts

Number	Font Family	Style	Weight	Font Name
1	serif	normal	normal	Time-Roman
1	serif	normal	bold	Times-Bold
1	serif	italic	normal	Times-Italic
1	serif	italic	bold	Times-BoldItalic

Table 11–15 (Cont.) Type 1 Fonts

Number	Font Family	Style	Weight	Font Name
2	sans-serif	normal	normal	Helvetica
2	sans-serif	normal	bold	Helvetica-Bold
2	sans-serif	italic	normal	Helvetica-Oblique
2	sans-serif	italic	bold	Helvetica-BoldOblique
3	monospace	normal	normal	Courier
3	monospace	normal	bold	Courier-Bold
3	monospace	italic	normal	Courier-Oblique
3	monospace	italic	bold	Courier-BoldOblique
4	Courier	normal	normal	Courier
4	Courier	normal	bold	Courier-Bold
4	Courier	italic	normal	Courier-Oblique
4	Courier	italic	bold	Courier-BoldOblique
5	Helvetica	normal	normal	Helvetica
5	Helvetica	normal	bold	Helvetica-Bold
5	Helvetica	italic	normal	Helvetica-Oblique
5	Helvetica	italic	bold	Helvetica-BoldOblique
6	Times	normal	normal	Times
6	Times	normal	bold	Times-Bold
6	Times	italic	normal	Times-Italic
6	Times	italic	bold	Times-BoldItalic
7	Symbol	normal	normal	Symbol
8	ZapfDingbats	normal	normal	ZapfDingbats

The TrueType fonts are listed in Table 11–16. All TrueType fonts are subset and embedded into PDF.

Table 11–16 TrueType Fonts

Number	Font Family Name	Style	Weight	Actual Font	Actual Font Type
1	Albany WT	normal	normal	ALBANYWT.ttf	TrueType (Latin1 only)
2	Albany WT J	normal	normal	ALBANWTJ.ttf	TrueType (Japanese flavor)
3	Albany WT K	normal	normal	ALBANWTK.ttf	TrueType (Korean flavor)
4	Albany WT SC	normal	normal	ALBANWTS.ttf	TrueType (Simplified Chinese flavor)
5	Albany WT TC	normal	normal	ALBANWTT.ttf	TrueType (Traditional Chinese flavor)

Table 11–16 (Cont.) TrueType Fonts

Number	Font Family Name	Style	Weight	Actual Font	Actual Font Type
6	Andale Duospace WT	normal	normal	ADUO.ttf	TrueType (Latin1 only, Fixed width)
6	Andale Duospace WT	bold	bold	ADUOB.ttf	TrueType (Latin1 only, Fixed width)
7	Andale Duospace WT J	normal	normal	ADUOJ.ttf	TrueType (Japanese flavor, Fixed width)
7	Andale Duospace WT J	bold	bold	ADUOJB.ttf	TrueType (Japanese flavor, Fixed width)
8	Andale Duospace WT K	normal	normal	ADUOK.ttf	TrueType (Korean flavor, Fixed width)
8	Andale Duospace WT K	bold	bold	ADUOKB.ttf	TrueType (Korean flavor, Fixed width)
9	Andale Duospace WT SC	normal	normal	ADUOSC.ttf	TrueType (Simplified Chinese flavor, Fixed width)
9	Andale Duospace WT SC	bold	bold	ADUOSCB.ttf	TrueType (Simplified Chinese flavor, Fixed width)
10	Andale Duospace WT TC	normal	normal	ADUOTC.ttf	TrueType (Traditional Chinese flavor, Fixed width)
10	Andale Duospace WT TC	bold	bold	ADUOTCB.ttf	TrueType (Traditional Chinese flavor, Fixed width)

11.18 Defining Currency Formats

Currency formats defined in the Administration Runtime Configuration page are applied at the system level. Currency formats can also be applied at the report level. The report-level settings take precedence over the system-level settings here. For information on setting a report-level currency format, see the section "Configuring Currency Formats" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

11.18.1 Understanding Currency Formats

The Currency Formats tab enables you to map a number format mask to a specific currency so that your reports can display multiple currencies with their own corresponding formatting. Currency formatting is only supported for RTF and XSL-FO templates.

To apply these currency formats in the RTF template, use the format-currency function. See the section "Formatting Currencies" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* for detailed procedures.

To add a currency format:

1. Click the Add icon.
2. Enter the ISO currency code, for example: USD, JPY, EUR, GBP, INR.
3. Enter the format mask to apply for this currency.

   The Format Mask must be in the Oracle number format. The Oracle number format uses the components "9", "0", "D", and "G" to compose the format, for example: 9G999D00

   where

   9 represents a displayed number only if present in data

   G represents the group separator

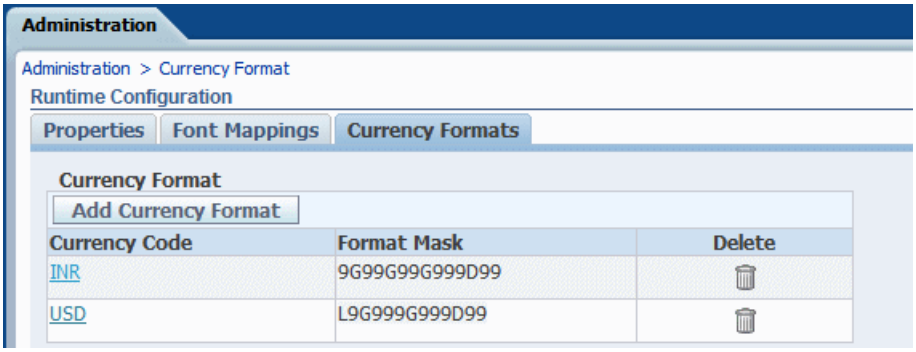
   D represents the decimal separator

   0 represents an explicitly displayed number regardless of incoming data

   See the section "Using the Oracle Format Mask" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* for more information about these format mask components.

Figure 11–1 shows sample currency formats.

Figure 11–1 Sample Currency Formats



12

Diagnostics and Performance Monitoring

This chapter describes configuring log files for diagnosing issues in BI Publisher and configuring user auditing to capture metrics on user activity and system performance.

It covers the following topics:

12.1 Diagnosing and Resolving Issues in Oracle BI Publisher

System administrators are typically responsible for supporting end users when they experience issues with the use of Oracle BI Publisher and for interacting with Oracle Support to understand the cause of issues and apply fixes.

Issues may be reported in response to end users receiving error messages, experiencing poor performance, or lack of availability.

The principal activities administrators perform to support issue resolution include:

- Examination of error and diagnostic log information. For more information, see:
  - Section 12.2, "About Diagnostic Log Files"
  - Section 12.3, "Configuring Log Files"
  - Section 12.4, "Viewing Log Messages"
- Examination of system and process metrics to understand availability and performance issues. For more information, see:
  - Section 12.5, "About Performance Monitoring and User Auditing"
  - Section 12.6, "Enabling Monitoring and Auditing"
  - Section 12.7, "Viewing the Audit Log"

- Section 12.8, "Configuring an Audit Repository"
- Section 12.9, "Using BI Publisher to Create Audit Reports"
- Section 12.10, "Viewing Performance Statistics in the MBean Browser"

12.2 About Diagnostic Log Files

BI Publisher writes diagnostic log files in the Oracle Diagnostic Logging (ODL) format. Log file naming and the format of the contents of log files conforms to an Oracle standard. You can view log files by using the WLST displayLogs command, or you can download log files to your local client and view them using another tool (for example a text editor, or another file viewing utility).

Log files are created and edited using Oracle Fusion Middleware Control. By default, after installation, the bipublisher-handler log is created. You can configure this log file or create a new logger.

12.2.1 About Log File Message Categories and Levels

Each log file message category is set to a specific default value between 1-32, and only messages with a level less or equal to the log level are logged. Various log file message categories exist, as described in Table 12–1.

Table 12–1 Log File Message Category Levels

Level	Description
IncidentError:1	A serious problem caused by unknown reasons. You can only fix the problem by contacting Oracle support. Examples are errors from which you cannot recover or serious problems.
Error:1	A problem requiring attention from the system administrator has occurred, and is not caused by a bug in the product. No performance impact.
Warning:1	A potential problem that should be reviewed by the administrator. Examples are invalid parameter values or a specified file does not exist.
Notification:1	A major lifecycle event such as the activation or deactivation of a primary sub-component or feature. This is the default level for NOTIFICATION.
NOTIFICATION:16	A finer level of granularity for reporting normal events.
TRACE:1	Trace or debug information for events that are meaningful to administrators, such as public API entry or exit points.
TRACE:16	Detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.
TRACE:32	Very detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.

12.2.2 About Log File Formats

A log file must contain a consistent format. However, since there can be multiple formats, you can change the format used in a log file. When you change the format used in a log file, and the new format differs from the current log file's format, a new log file is created. For example, a log file that contains ODL-XML, always contains XML, and is never mixed with text.

Configure the log file format in the Edit Log File dialog. See Section 12.3, "Configuring Log Files." The format can by Text or XML.

12.2.3 About Log File Rotation

Log file rotation can be file size based or time based. Whenever a log file exceeds the rotation criterion, the existing log file is renamed, and a new log file is created.

The file naming looks like this:

- log.xml
- log.xml.1 (oldest log file)
- log.xml.n

12.3 Configuring Log Files

This section describes using Oracle Fusion Middleware Control to configure BI Publisher log files. It includes the following topics:

- Section 12.3.1, "Setting the Log Level"
- Section 12.3.2, "Configuring Other Log File Options"

12.3.1 Setting the Log Level

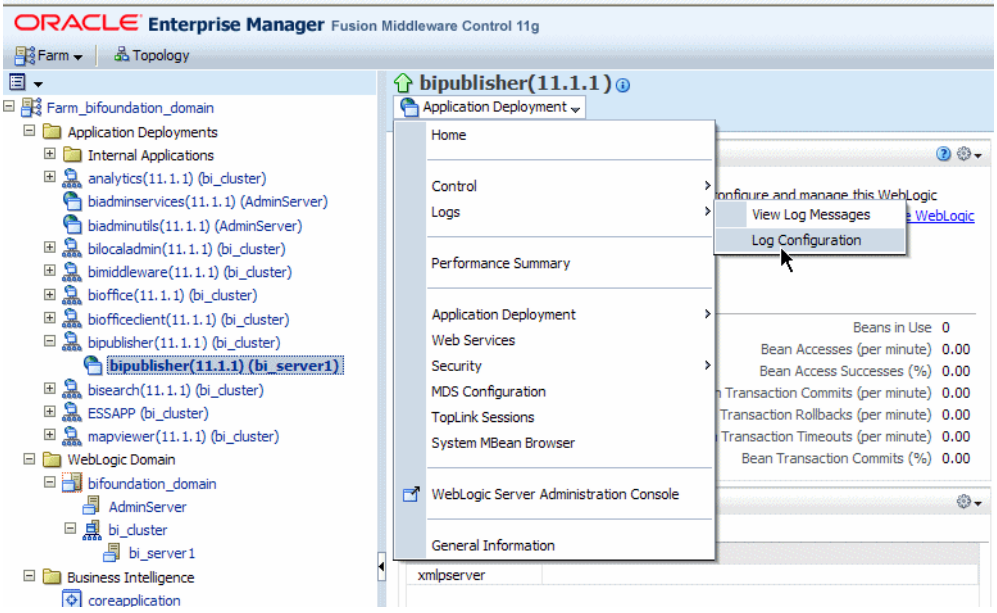
To set the log level in Oracle Fusion Middleware Control:

1. In Oracle Fusion Middleware Control, locate the BI Publisher server. For example:

   Under Application Deployments, expand bipublisher (11.1.1.) (bi_cluster), and then right-click bipublisher (11.1.1)(bi_server1)

2. From the menu, click **Logs** and then **Log Configuration** as shown in Figure 12–1:

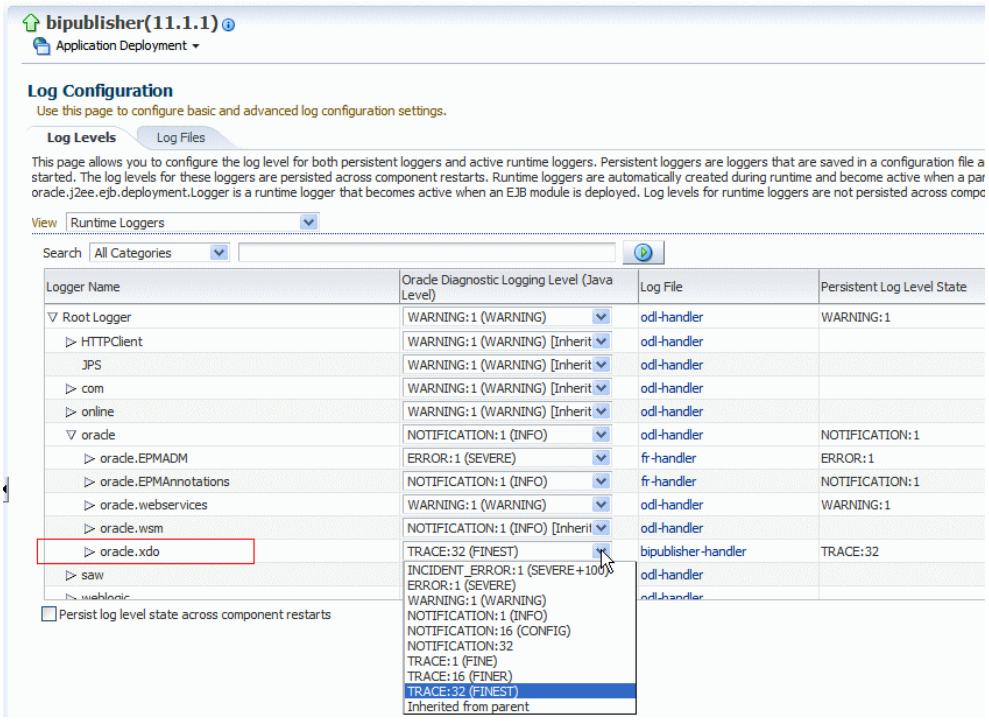
Figure 12–1 Navigating to Log Configuration



3. In the **Log Levels** tab, under **Logger Name**, expand **Root Logger**, then expand **oracle**.

   Locate **oracle.xdo** and select the log level from the drop-down list as shown in Figure 12–2.

Figure 12–2 Setting Log Level



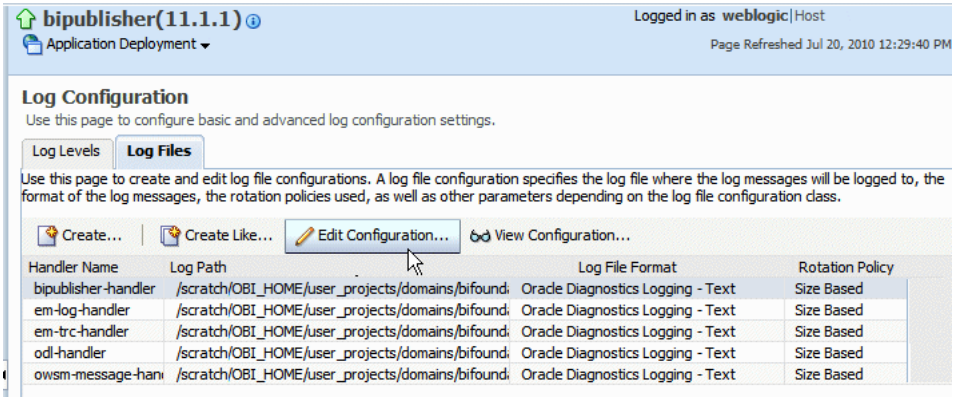
4. Click **Apply**.

12.3.2 Configuring Other Log File Options

To configure log files in Oracle Fusion Middleware Control:

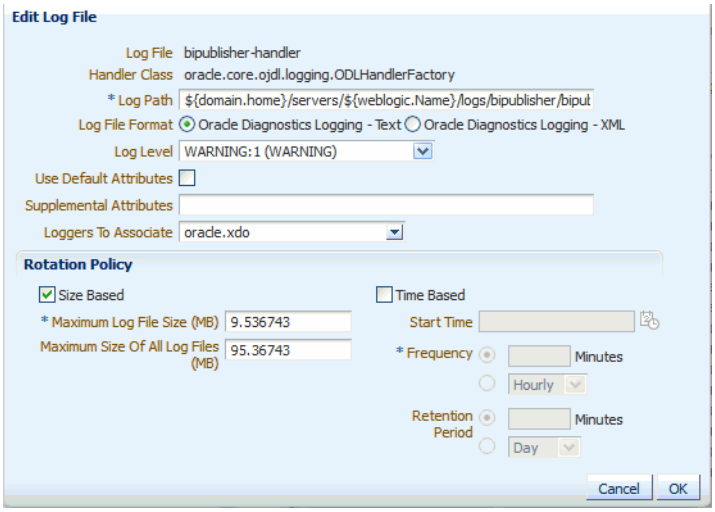
1. Navigate to the **Log Configuration** page as described in Section 12.3.1, "Setting the Log Level."
2. Click the **Log Files** tab.
3. Select bipublisher-handler in the table and click **Edit Configuration**, as shown in Figure 12–3.

Figure 12–3 Edit Log Configuration Files



4. In the **Edit Log File** dialog configure the bipublisher-handler log file options. A sample is shown in Figure 12–4.

Figure 12–4 Edit Log File Dialog



12.4 Viewing Log Messages

You can view log messages using Oracle Fusion Middleware Control or you can view the log files directly.

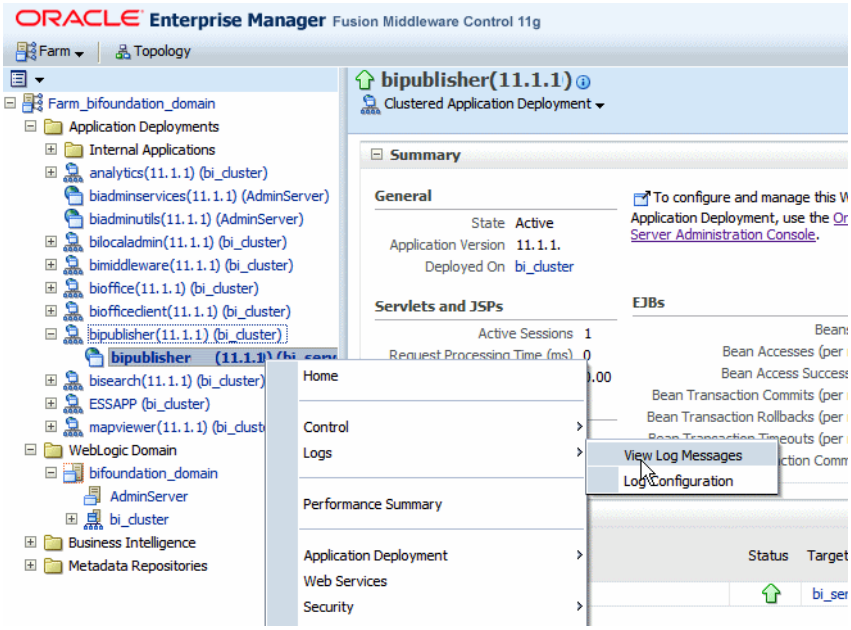
To view log messages in Oracle Fusion Middleware Control:

1. In Oracle Fusion Middleware Control, locate the BI Publisher server. For example:

   Under Application Deployments, right-click **bipublisher (11.1.1)**.

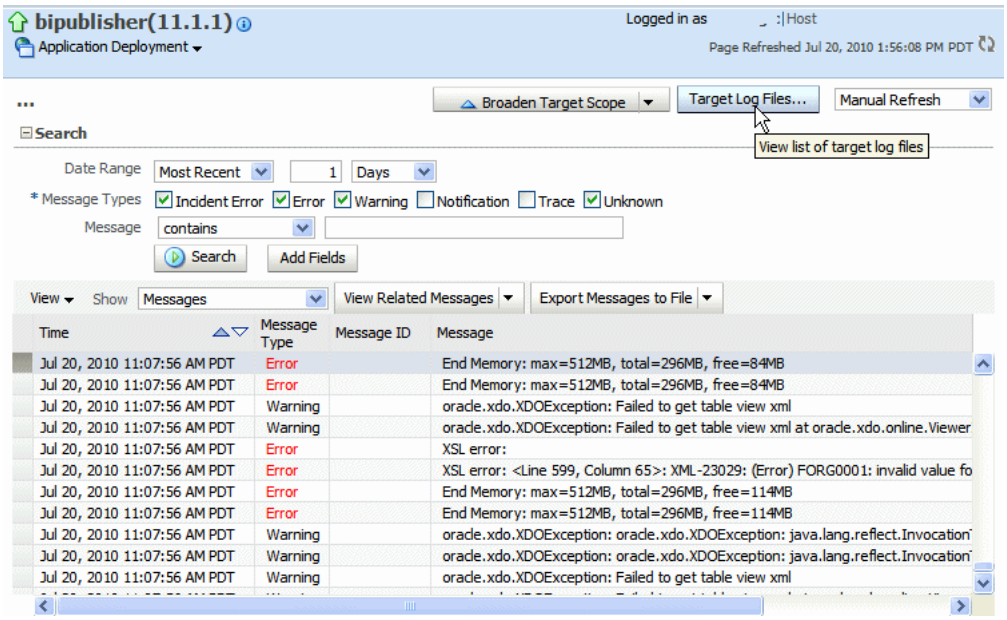
2. From the menu, click **Logs** and **View Log Messages** as shown in Figure 12–5

Figure 12–5 Navigating to View Log Messages



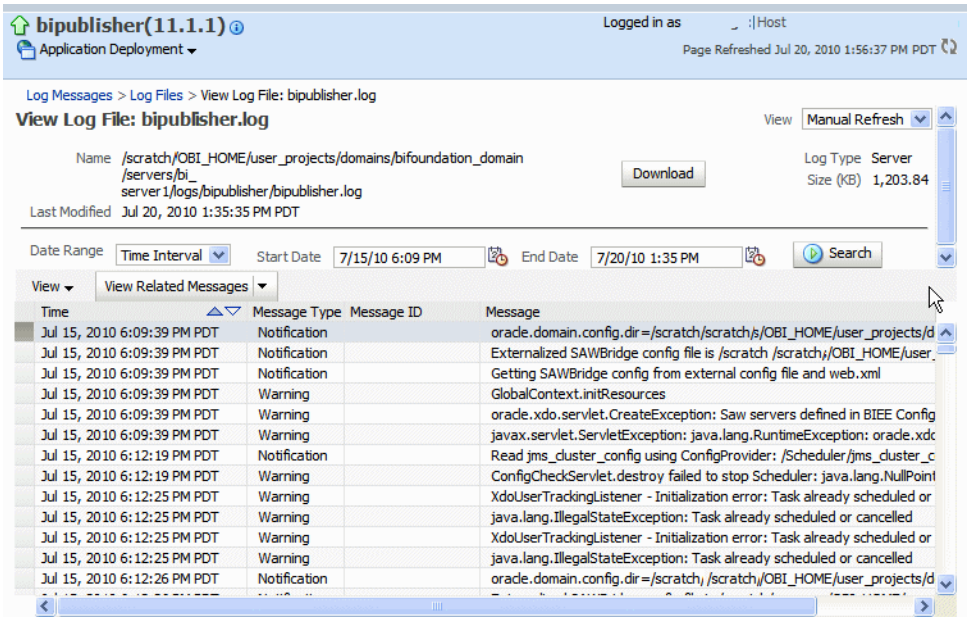
3. To view a specific log file, click **Target Log Files**, as shown in Figure 12–6.

Figure 12–6 Target Log Files



4. From the **Log Files** page, select a specific log to view messages or download the log file.
5. Click **View Log File** to view the messages, as shown in Figure 12–7.

Figure 12–7 Viewing Log Files



12.4.1 Viewing Messages by Reading the Log File

The log file is located in the directory that is specified in the Log Path in the Edit Log File dialog. Navigate to the directory on the server to view the log file.

The following example shows an ODL format error message:

```
<msg time="2009-07-30T16:00:03.150-07:00" comp_id="xdo" type="ERROR" level="1"
host_id="MyBIPHost" host_addr="122.22.222.22" module="oracle.xdo" tid="11"
user="Administrator">
<txt>Variable 'G_dept' is missing...</txt>
</msg>
```

Table 12–2 describes the message attributes displayed in the log file:

Table 12–2 Log File Message Attributes

Attribute Name	Description
time	The date and time when the message was generated. This reflects the local time zone.
comp_id	The ID of the component that originated the message.
type	The type of message. Possible values are: INCIDENT_ERROR, ERROR, WARNING, NOTIFICATION, TRACE, and UNKNOWN. See Table 12–1 for information about the message types.
level	The message level, represented by an integer value that qualifies the message type. Possible values are from 1 (highest severity) through 32 (lowest severity).
host_id	The name of the host where the message originated.
host_addr	The network address of the host where the message originated.
module	The ID of the module that originated the message. If the component is a single module, the component ID is listed for this attribute.
tid	The ID of the thread that generated the message.

Table 12–2 (Cont.) Log File Message Attributes

Attribute Name	Description
user	The name of the user whose execution context generated the message.

12.5 About Performance Monitoring and User Auditing

Performance monitoring enables you to monitor the performance of queries, reports and document generation and to analyze the provided details.

BI Publisher collects performance statistics through JMX Management Beans or Mbeans. Each MBean reveals attributes, operations, and relevant statistics gathered by the Oracle Dynamic Monitoring Service (DMS). Table 12–3 summarizes the beans that are provided.

Table 12–3 Management Beans

Management Bean	Description
ReportEventMonitor	Creates an Mbean per report and displays detailed monitoring data for the report.
ServerEventMonitor	Exists per server and displays user and server activity summaries.
UserEventMonitor	Creates an Mbean per user and displays detailed monitoring data for the user.

12.6 Enabling Monitoring and Auditing

To enable monitoring, complete the following tasks:

1. Enable Monitor and Audit on the Administration Server Configuration page. See Section 12.6.1, "Enable Monitor and Audit on the Server Configuration Page."
2. Configure the Audit Policy Settings with Fusion Middleware Control (Enterprise Manager). See Section 12.6.2, "Configure the Audit Policy Settings."
3. (Optional) For non-English installations of Oracle Fusion Middleware Control that require a translated user interface, copy the translation files to the appropriate location. See Section 12.6.3, "(Optional) Copy Translation Files."
4. Restart WebLogic Server.

12.6.1 Enable Monitor and Audit on the Server Configuration Page

To turn on monitoring and auditing for the BI Publisher application:

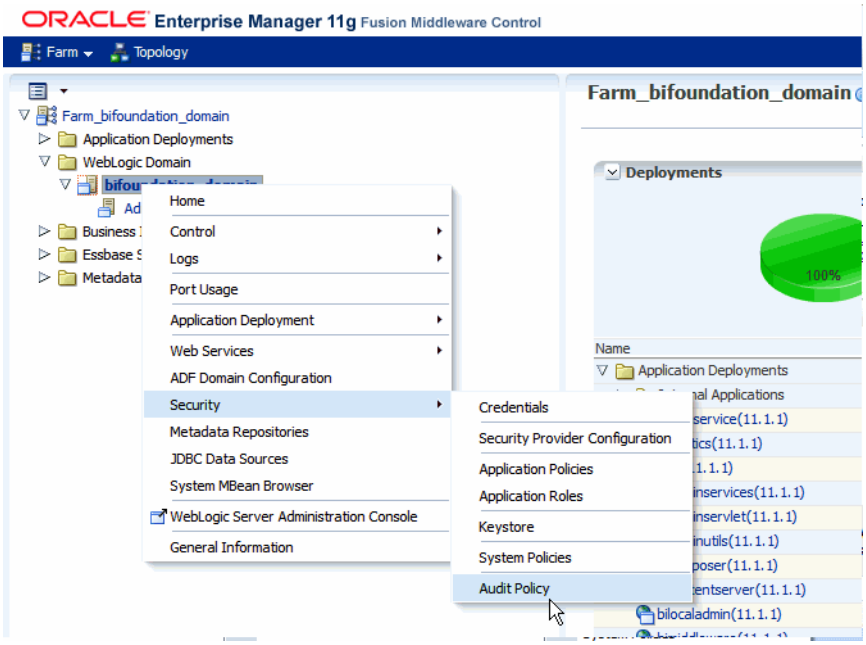
1. Click the Administration link.
2. Under **System Maintenance**, click **Server Configuration**.
3. Under the Monitor and Audit region, select the **Enable Monitor and Audit** check box.

12.6.2 Configure the Audit Policy Settings

To configure the audit policy settings:

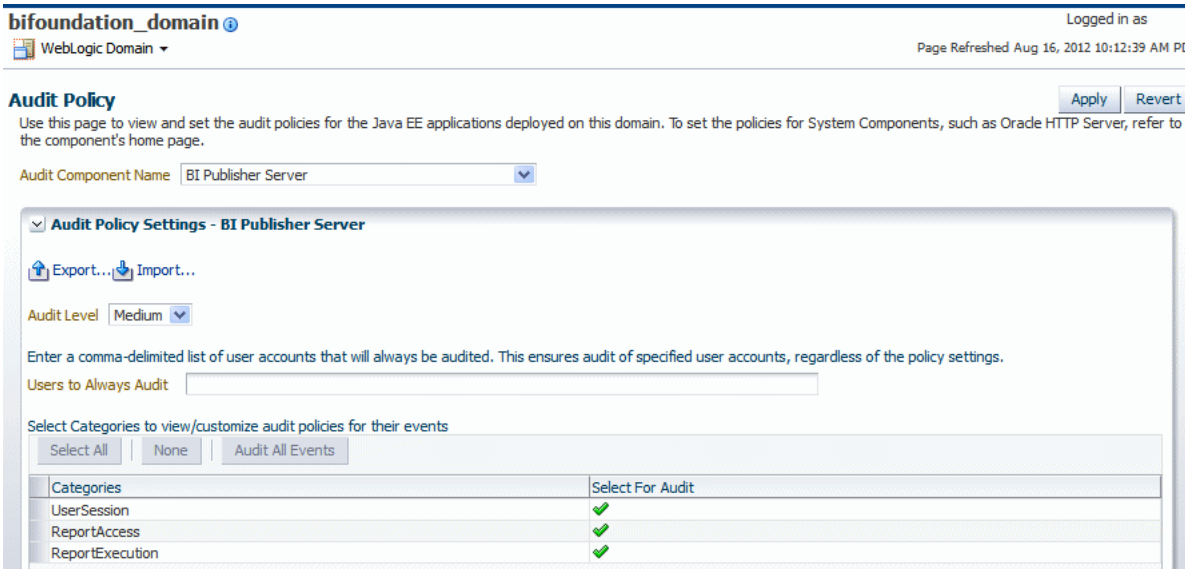
1. In Oracle Fusion Middleware Control, under **WebLogic Domain**, right-click **bifoundation_domain**. From the menu, click **Security** and click **Audit Policy**, as shown in Figure 12–8.

Figure 12–8 Navigating to the Audit Policy Settings



2. The **Audit Policy** page displays the audited applications under the bifoundation_ domain. From the **Audit Component Name** list, select BI Publisher Server. Set the **Audit Level** to enable auditing for BI Publisher. An example is shown in Figure 12–9.

Figure 12–9 Audit Policy Page



Typically set the **Audit Level** to **Medium**.

To customize the audit level for each event, select **Custom** from the **Audit Level** list. This setting enables you to set the audit level for each event and apply filters. Select a category (UserSession, ReportAccess, or ReportExecution) to view the available events.

The events that are audited for the BI Publisher server are:

- User Login
- User Logout
- Report Request
- Scheduled Report Request
- Report Republish
- Report Data Download
- Report Download
- Report Data Process
- Report Rendering
- Report Delivery

12.6.3 (Optional) Copy Translation Files

If you require that your Oracle Fusion Middleware Control (Enterprise Manager) user interface displays in a non-English language, you must copy the BI Publisher translation files from the BI Publisher repository to the appropriate location in the Oracle BI Enterprise Edition installation directory for the BI Publisher audit event user interface components to display in the correct language when viewed in Enterprise Manager.

1. Navigate to the Audit directory located under the BI Publisher repository: *<BI Publisher deployment directory>*/repository/Admin/Audit
2. Copy the *.xlf files from the Audit directory to the Oracle BI Enterprise Edition installation directory located: /oracle_common/modules/oracle.iau_11.1.1/components/xmlpserver

12.6.4 Restart WebLogic Server

Restart the WebLogic Serve instance. You can do this using Oracle Fusion Middleware Control, or if you are running Windows, you can select "Stop BI Servers" and then "Start BI Servers" from the Start menu.

12.7 Viewing the Audit Log

If you set the property AUDIT_JPS_INTEGRATION to true, then the audit log can be viewed under the xmlpserver folder under the WebLogic Server AdminServer directory: /AdminServer/logs/auditlogs/xmlpserver/audit.log

Alternatively, you can configure an audit repository in the database to store audit data in database tables instead of the log file (the file is not generated in this case). The collected data can be analyzed using reports provided by Audit Framework, or you can create your own reports using BI Publisher.

For more information on the reports provided by Audit Framework, see "Using Audit Analysis and Reporting" in *Oracle Fusion Middleware Application Security Guide*.

The following section describes how to set up the audit repository in your database to store the auditing data.

12.8 Configuring an Audit Repository

Perform this procedure to configure an audit repository to store your auditing data collected by the Fusion Middleware Auditing Framework in database tables rather than a log file.

To set up the audit database in WebLogic Server:

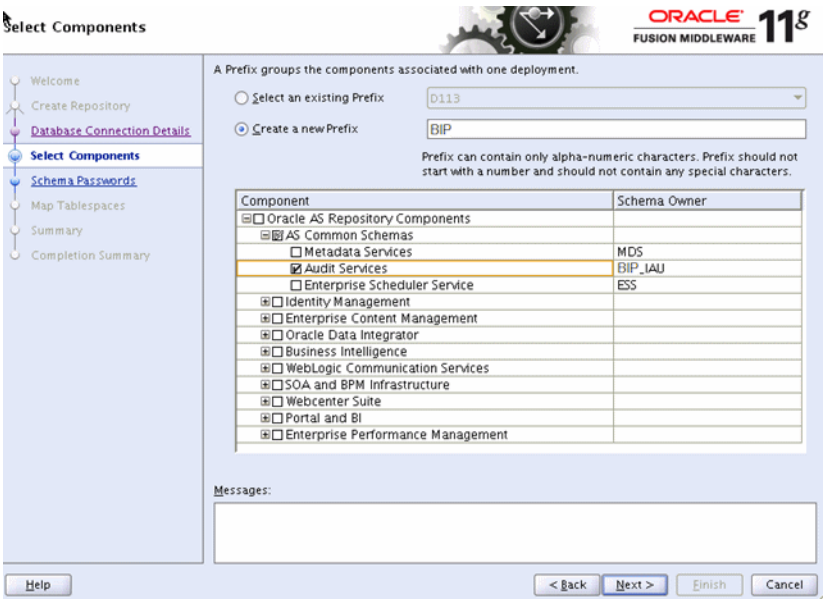
1. Create the audit schema using RCU.
2. Create a data source in your WebLogic server.
3. Register the audit database to your domain.

12.8.1 Creating the Audit Schema Using RCU

To create the audit schema:

1. Go to $RCU_HOME/bin and execute the rcu command.
2. Choose **Create** at the starting screen and click **Next.**
3. Enter your database details and click **Next**.
4. Choose the option to create a prefix, and enter a prefix; for example: BIP
5. Select **Audit Services** from the list of schemas (shown in Figure 12–10).

Figure 12–10 List of Schemas



6. Click **Next** and accept the tablespace creation.
7. Click **Finish** to start the process.

When the Repository Creation Utility process finishes, the following audit-related schemas are created in your database:

- <prefix>_IAU (for example: BIP_IAU)
- <prefix>_IAU_APPEND (for example: BIP_IAU_APPEND)
- <prefix>_IAU_VIEWER (for example: BIP_IAU_VIEWER)

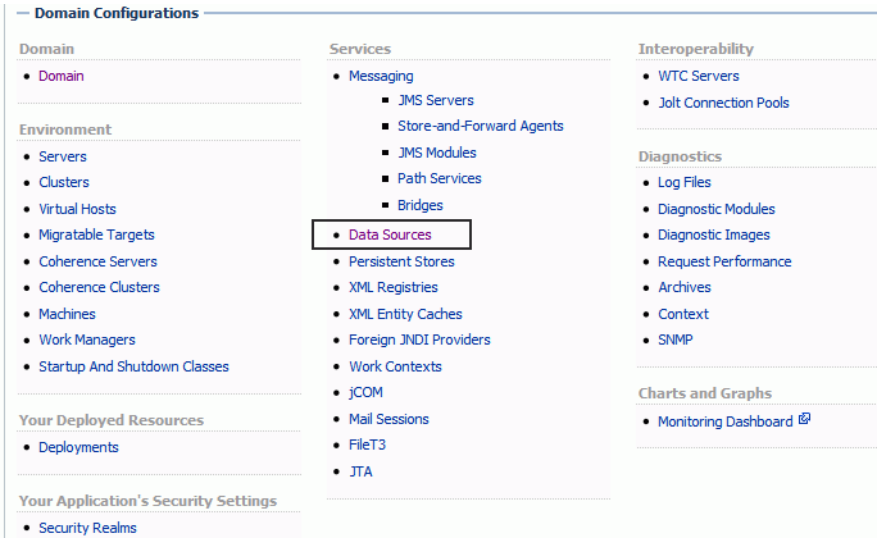
12.8.2 Creating the Data Source in WebLogic Server

After you create a database schema for your auditing data, next create a JDBC connection on your WebLogic Server so the Audit Framework can access the database schema that was created with the RCU in the previous step.

To create the JDBC connection:

1. Connect to the Oracle WebLogic Server administration console: http://hostname:port/console (e.g. http://example.com:7001/console)
2. Under **Services**, click the **Data Sources** link, as shown in Figure 12–11:

Figure 12–11 Navigating to the Data Sources Link



3. Click **Lock and Edit**.
4. On the **Summary of JDBC Data Sources** page, click **New** and then click **Generic Data Source.**
5. Enter the following details for the new data source:
   - **Name**

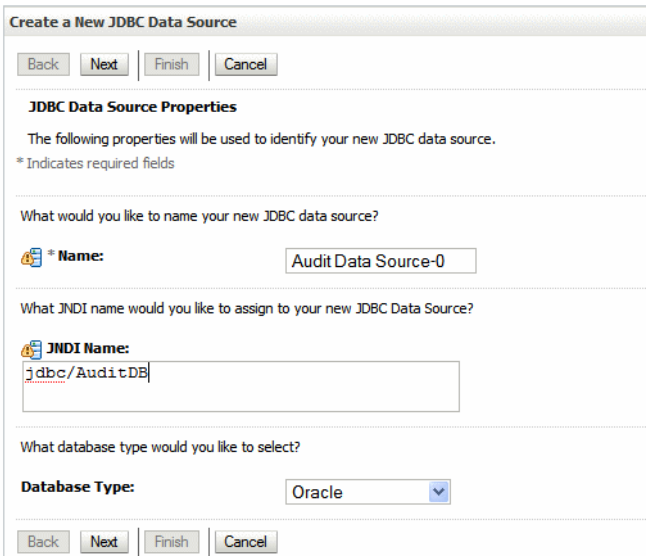
     Example: Audit Data Source-0
   - **JNDI Name**

     Example: jdbc/AuditDB
   - **Database Type**

     Example: Oracle

   Figure 12–12 shows the example entries:

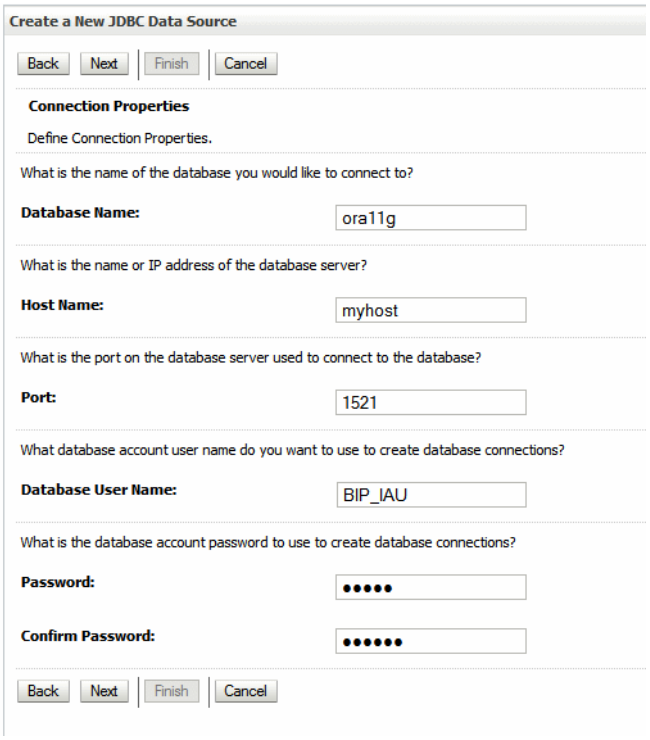
Figure 12–12 Create JDBC Data Source



6. Click **Next** and select the database driver. Choose "Oracle's Driver (Thin XA) Versions: 9.0.1 or later" if you are using Oracle database, and click **Next**.
7. In the **Connections Properties** page, enter the following:
   - **Database Name**: Enter the name of the database (SID) to which to connect.
   - **Host Name**: Enter the hostname of the database.
   - **Port**: Enter the database port.
   - **Database User Name**: Enter the name of the audit schema that you created in RCU. The suffix is always _IAU for the audit schema. For example, if you supplied the prefix as "BIP"", then the schema name would be "BIP_IAU".
   - **Password**: Enter the password for the audit schema that you created in RCU.

   Figure 12–13 shows the **Connection Properties** page:

Figure 12–13 Connection Properties Page



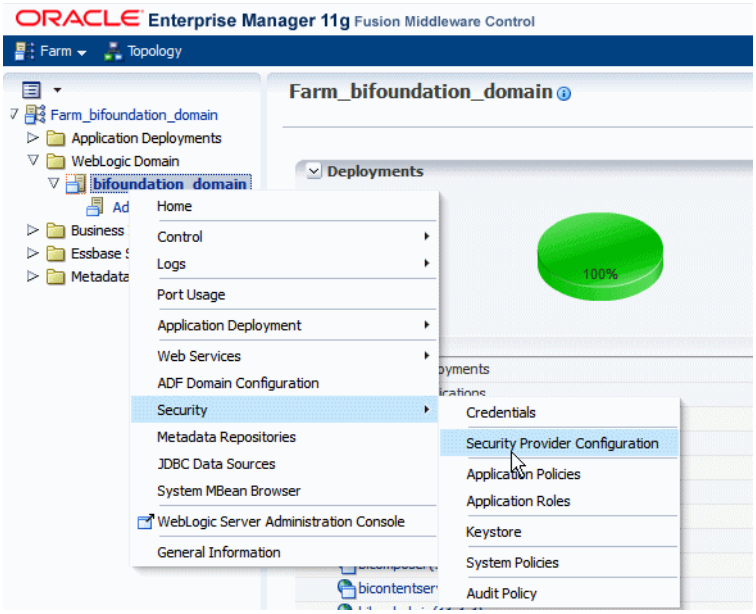
8. Click **Next**. Accept the defaults, and click **Test Configuration** to verify the connection.
9. Click **Next**. Select the listed servers where you want to make this JDBC connection available.
10. Click **Finish** and then click **Activate Changes** in the **Change Center.**

12.8.3 Registering the Audit Storage Database to Your Domain

To register the JNDI/JDBC data source as your auditing data storage with Fusion Middleware Control (Enterprise Manager):

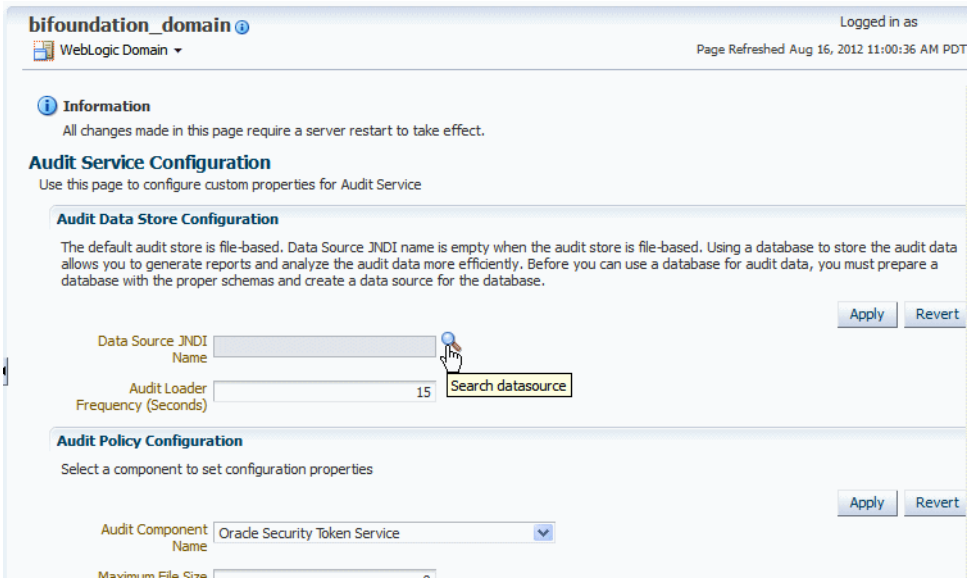
1. Log in to Fusion Middleware Control.
2. Navigate to the WebLogic Domain, right click **bifoundation_domain**, then select **Security**, then **Security Provider Configuration**. The navigation path is shown in Figure 12–14.

Figure 12–14 Navigating to the Audit Store



3. On the **Security Provider Configuration** page, under **Audit Service**, click **Configure.**
4. On the **Audit Service Configuration** page, locate your **Data Source JNDI Name** by clicking **Search datasource**, as shown Figure 12–15.

Figure 12–15 Audit Service Configuration Page



5. From the **Select Data Source** dialog, select the data source you created and click **OK**.
6. Click **Apply** to continue.
7. Restart WebLogic Server.

When the WebLogic Server is restarted, BI Publisher stores all auditing data in the database table called "IAU_BASE". To verify this procedure, try logging in to BI Publisher and opening a few reports. You should see the activity audited in the "IAU_BASE" table. If not, check the log file for possible errors. The log file is located under the Oracle BI Domain Home, for example:

MIDDLEWARE_HOME/user_projects/domains/bifoundation_domain/servers/AdminServer/logs/AdminServer-diagnostic.log

Once the data is successfully loading to the database tables, you can design your own auditing reports using BI Publisher.

12.9 Using BI Publisher to Create Audit Reports

Once you have the auditing repository set up, you can use BI Publisher to create your own reports to visualize your auditing data. To create a report on your auditing data in BI Publisher:

1. Register the data source in BI Publisher.
2. Create a data model.
3. Create the report.

12.9.1 Registering the Data Source in BI Publisher

Register the audit data source (JNDI/JDBC connection) that you created in the previous procedure as a JNDI data source in BI Publisher. Because you created a JDBC connection registered as JNDI, you do not need to create a new JDBC connection by typing the connection URL, username/password, and so on. You can just register it using the JNDI name (for example: jdbc/AuditDB).

1. Log in to BI Publisher with administrator privileges and click the **Administration** link.
2. Under **Data Sources**, click **JNDI Connection**, then click **Add Data Source**.
3. Enter the **Data Source Name** and **JNDI Name**. The **JNDI Name** is the name you provided in the WebLogic Console as the auditing data source (for example: jdbc/AuditDB).
4. Click **Test Connection** to ensure that the data source connection works.
5. Add the appropriate roles to the data source so that the report developers and consumers can view the reports built on this data source.
6. Click **Apply** to save.

12.9.2 Creating a Data Model

To create a data model from your auditing data source:

Note: Note: For the complete guidelines for developing data models in BI Publisher, see the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

---

1. On the global header, click **New** and then click **Data Model**.
2. Set the **Default Data Source** to the audit JNDI data source.

3. Click **Data Sets** and from the **Create New** menu select new **SQL Query** data set.
4. Use the Query Builder to build a query or just type a SQL query against the IAU_BASE table. The IAU_BASE table contains all the auditing data for other products running on the WebLogic Server such as JPS, OID, and so on. To create a data model that contains only the BI Publisher data, you can filter the data based on the value of the IAU_COMPONENTTYPE column that contains the product name. For BI Publisher, the value is "xmlpserver".

   The following sample SQL query returns only BI Publisher data:

```
select     "IAU_BASE"."IAU_COMPONENTTYPE" as "IAU_COMPONENTTYPE",
     "IAU_BASE"."IAU_EVENTTYPE" as "IAU_EVENTTYPE",
     "IAU_BASE"."IAU_EVENTCATEGORY" as "IAU_EVENTCATEGORY",
     "IAU_BASE"."IAU_TSTZORIGINATING" as "IAU_TSTZORIGINATING",
   to_char("IAU_TSTZORIGINATING", 'YYYY-MM-DD') IAU_DATE,
   to_char("IAU_TSTZORIGINATING", 'DAY') as IAU_DAY,
   to_char("IAU_TSTZORIGINATING", 'HH24') as IAU_HH24,
   to_char("IAU_TSTZORIGINATING", 'WW') as IAU_WEEK_OF_YEAR,
     "IAU_BASE"."IAU_INITIATOR" as "IAU_INITIATOR",
     "IAU_BASE"."IAU_RESOURCE" as "IAU_RESOURCE",
     "IAU_BASE"."IAU_TARGET" as "IAU_TARGET",
     "IAU_BASE"."IAU_MESSAGETEXT" as "IAU_MESSAGETEXT",
     "IAU_BASE"."IAU_FAILURECODE" as "IAU_FAILURECODE",
     "IAU_BASE"."IAU_REMOTEIP" as "IAU_REMOTEIP"
from    "BIP_IAU"."IAU_BASE" "IAU_BASE"
where "IAU_BASE"."IAU_COMPONENTTYPE" = 'xmlpserver'
```

5. To test your data model, click **Get XML Output**. Select a sample size, and run your data model. Save the sample XML to your data model.
6. Save your data model.

12.9.3 Creating the Report

Now you can use one of the BI Publisher's layout options to design the report layout and visualize the auditing data. To create a report using the BI Publisher layout editor:

1. On the global header, click **New** and then click **Report**.
2. Select the data model you created in the previous procedure.
3. To use the layout editor, click **Add New Layout**, and then click one of the Basic Templates to get started.

   For complete instructions on using the layout editor, see the topic: "Creating BI Publisher Layout Templates" in the Help or in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

   Figure 12–16 shows using the layout editor to design a report based on the auditing data:

Figure 12–16 Using the Layout Editor to Create an Auditing Report

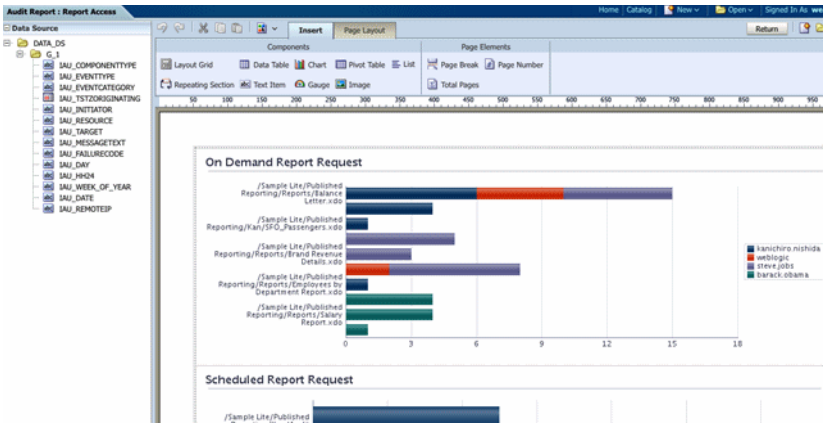
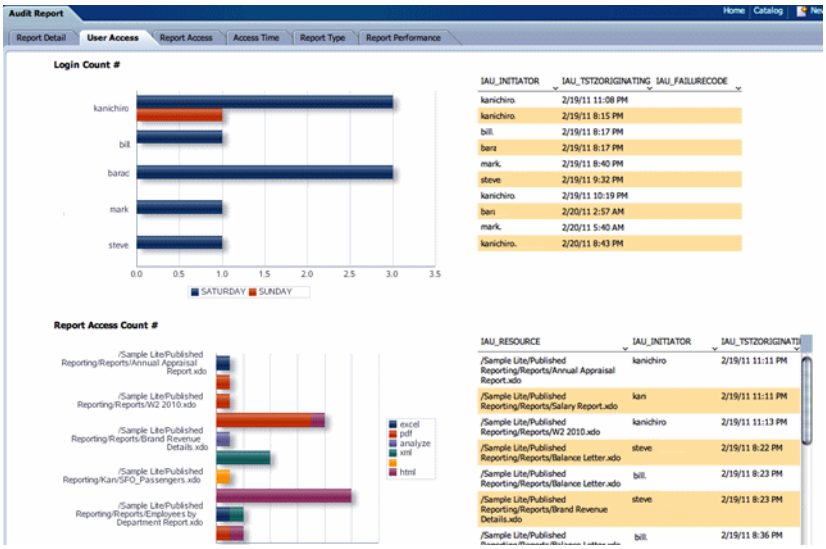


Figure 12–17 shows a sample completed auditing report displayed in the report viewer:

Figure 12–17 Sample Auditing Report



12.10 Viewing Performance Statistics in the MBean Browser

To view the performance statistics collected by the Report Event Monitor, Service Event Monitor, and User Event Monitor, you use the System MBean browser.

To view performance statistics:

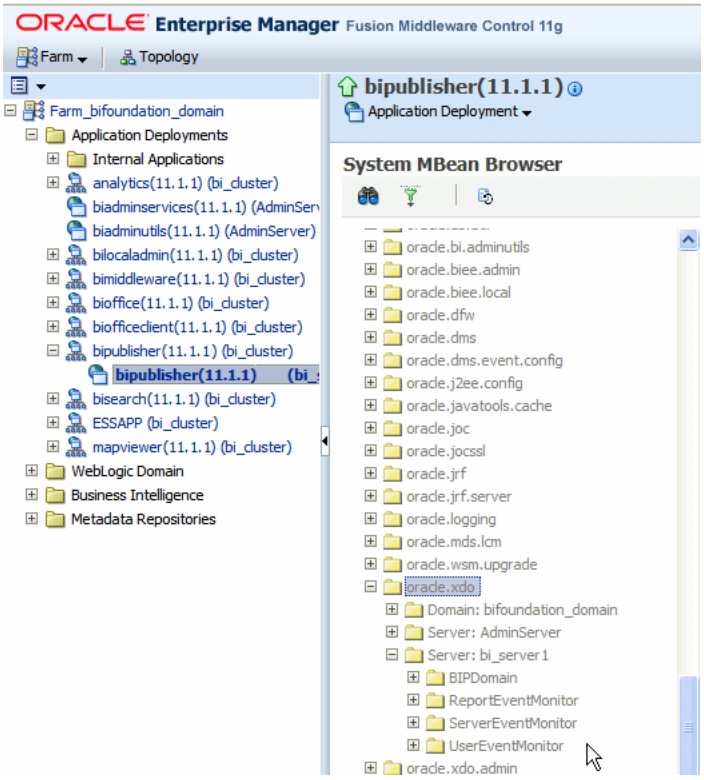
1. In Oracle Fusion Middleware Control, locate the BI Publisher server. For example:

   Under Application Deployments, expand bipublisher (11.1.1) (bi_cluster), and then right-click bipublisher (11.1.1)(bi_server1)

2. From the menu, click **System MBean Browser**.

3. In the System MBean Browser, under the Application Defined MBeans, expand the oracle.xdo folder to view the BI Publisher MBeans. Expand the list and select the bean to view the details, as shown in Figure 12–18.

Figure 12–18 Viewing Performance Statistics



13

Adding Translations for the BI Publisher Catalog and Reports

This chapter describes how to export and import translation files both for the catalog and for individual report layouts.

It covers the following topics:

13.1 Introduction

BI Publisher supports two types of translation:

- Catalog Translation
- Template (or layout) Translation

Catalog translation enables the extraction of translatable strings from all objects contained in a selected catalog folder into a single translation file; this file can then be translated and uploaded back to BI Publisher and assigned the appropriate language code.

Catalog translation extracts not only translatable strings from the report layouts, but also the user interface strings that are displayed to users, such as catalog object descriptions, report parameter names, and data display names.

Users viewing the catalog see the item translations appropriate for the UI Language they selected in their My Account preferences. Users see report translations appropriate for the Report Locale that they selected in their My Account preferences.

Template translation enables the extraction of the translatable strings from a single RTF-based template (including sub templates and style templates) or a single BI Publisher layout template (.xpt file). Use this option when you only need the final report documents translated. For example, your enterprise requires translated invoices to send to German and Japanese customers.

This chapter describes the process of downloading and uploading translation files. For more information on the concepts and processes of translating the files see the part "Translating Reports and Catalog Objects" in *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

13.1.1 Limitations of Catalog Translation

If you have existing XLIFF file translations for specific reports and then you import a catalog translation file for the folder in which the existing translations reside, the existing XLIFF files are overwritten.

13.2 Exporting and Importing a Catalog Translation File

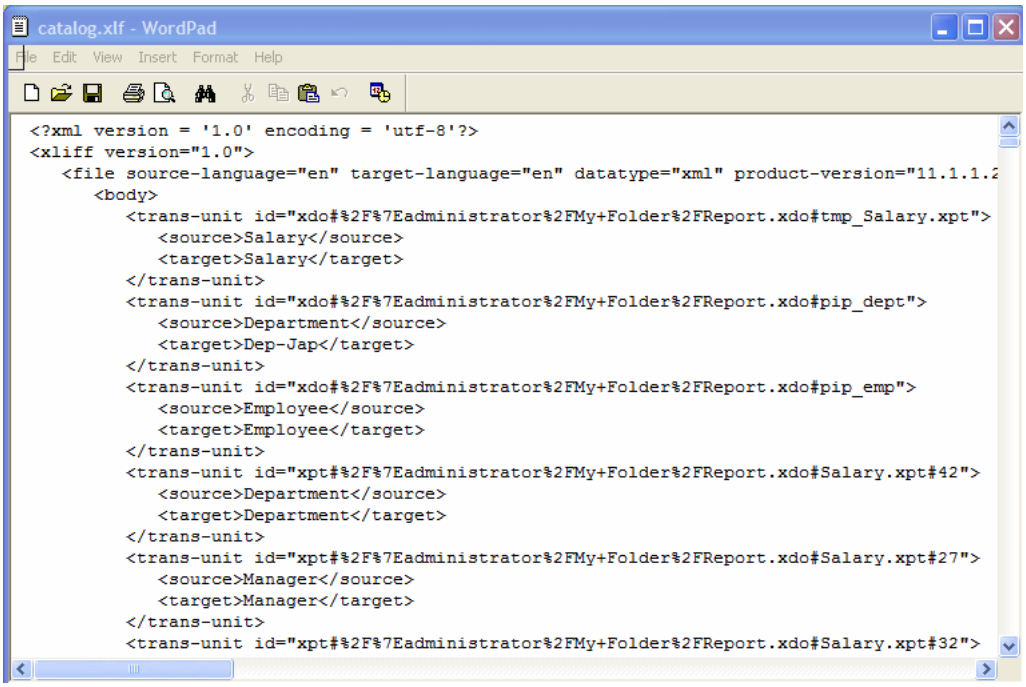
This procedure describes the process of exporting an XLIFF file from the catalog, importing the translated file back to the catalog, and testing the translation. Importing and exporting XLIFF files can only be performed by an Administrator.

To import and export an XLIFF file:

1. Select the folder in the catalog, click the **Translation** toolbar button, and then click **Export XLIFF**.
2. Save the XLIFF file to a local directory.
3. Open the Translation file (catalog.xlf) and apply translations to the Boilerplate text, as shown in Figure 13–1.

   See the "What Is an XLIFF?" section in *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* for details on how to translate an XLIFF file.

Figure 13–1 Applying Translations



4. After the file is translated, upload the XLIFF file to the BI Publisher server: Click the **Translation** toolbar button, then click **Import XLIFF**. Upload the translated XLIFF to the server.
5. To test the translation, select **My Account** from **Signed In As** in the global header.
6. On the General tab of the My Account dialog, change the Report Locale and the UI Language preferences to the appropriate language and click **OK**.

7. View the objects in the translated folder.

13.3 Template Translation

Template translation includes:

- RTF templates
- RTF sub templates
- Style templates
- BI Publisher templates (.xpt)

RTF and BI Publisher (.xpt) templates can be translated from the **Properties** page. To access the **Properties** page, click the Properties link for the layout in the Report Editor, as shown in Figure 13–2.

Figure 13–2 Accessing the Properties Page

From the **Properties** page you can generate an XLIFF file for a single template. Click **Extract Translation** to generate the XLIFF file.

13.3.1 Generating the XLIFF File from the Layout Properties Page

To generate the XLIFF file for report layout templates:

1. Navigate to the report in the catalog and click **Edit** to open it for editing.
2. From the thumbnail view of the report layouts, click the **Properties** link of the layout (RTF or XPT) to open the **Layout Properties** page.
3. In the **Translations** region, click **Extract Translation**.

   BI Publisher extracts the translatable strings from the template and exports them to an XLIFF (.xlf file).
4. Save the XLIFF to a local directory.

To generate the XLIFF file for style templates and sub templates:

1. Navigate to the style template or sub template in the catalog and click **Edit** to open the Template Manager.
2. In the **Translations** region, click **Extract Translation**.

   BI Publisher extracts the translatable strings from the template and exports them to an XLIFF (.xlf file).
3. Save the XLIFF to a local directory.

13.3.2 Translating the XLIFF File

When you have downloaded the XLIFF file, it can be sent to a translation provider, or using a text editor, you can enter the translation for each string. See Structure of the XLIFF File, *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher* for instructions on how to edit the XLIFF file.

A "translatable string" is any text in the template that is intended for display in the published report, such as table headers and field labels. Text supplied at runtime from the data is not translatable, nor is any text that you supply in the Microsoft Word form fields.

You can translate the template XLIFF file into as many languages as desired and then associate these translations to the original template.

13.3.3 Uploading the Translated XLIFF File to BI Publisher

To upload the translated XLIFF file:

1. Navigate to the report, sub template, or style template in the catalog and click **Edit** to open it for editing.

   For reports only:

   From the thumbnail view of the report layouts, click the **Properties** link of the layout to open the Template Manager.
2. In the **Translations** region, click the **Upload** toolbar button.
3. In the **Upload Translation File** dialog, locate the file in the local directory and select the **Locale** for this translation.
4. Click OK to upload the file and view it in the **Translations** table.

13.4 Using the Localized Template Option

If you need to design a different layout for the reports that you present for different localizations, then you can create new RTF file that is designed and translated for the locale and upload this file to the Template Manager.

> **Note:** The localized template option is not supported for XPT templates.

The process overview for using the localized template option is described in the following sections:

- Section 13.4.1, "Designing the Localized Template File"
- Section 13.4.2, "Uploading the Localized Template to BI Publisher"

13.4.1 Designing the Localized Template File

Use the same tools that you used to create the base template file, translating the strings and customizing the layout as desired for the locale.

13.4.2 Uploading the Localized Template to BI Publisher

To upload the localized template:

1. Navigate to the report, subtemplate, or style template in the catalog and click **Edit** to open it for editing.

   For reports only:

   From the thumbnail view of the report layouts, click the **Properties** link of the layout to open the Template Manager.

2. In the **Templates** region, click the **Upload** toolbar button.
3. In the **Upload Template File** dialog, locate the file in the local directory, select rtf as the **Template Type** and select the **Locale** for this template file.
4. Click **OK** to upload the file and view it in the **Templates** table.

14

Moving Catalog Objects Between Environments

This chapter describes how to move objects between test, production, and development environments using the BI Publisher catalog utility.

It covers the following topics:

14.1 Overview

The BI Publisher catalog utility enables administrators and report developers to export the reporting object-related files from the catalog where all BI Publisher reports are stored, and to import them to a different catalog. Use this tool to manage BI Publisher reports using a third party tool as a source control or to move a specific set of reports from a development environment to a quality assurance or production environment. The catalog utility can also be used to help manage translations of reporting objects.

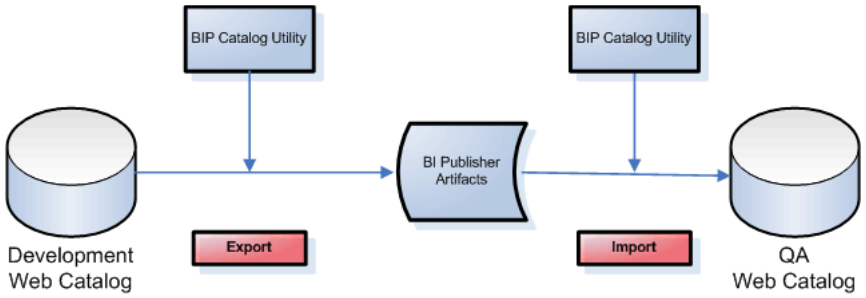
Use the BI Publisher catalog utility to perform the following tasks:

- Export BI Publisher reports from the catalog
- Import BI Publisher reports into the catalog
- Extract translatable strings and generate a translation file (XLIFF)
- Generate a security.xml file that contains the reporting object-level permission settings

14.1.1 When to Use the Catalog Utility

Use the catalog utility to move BI Publisher report artifacts from one environment to another. For example, use the catalog utility to move reports from a development environment to a quality assurance environment. This process is illustrated in the Figure 14–1.

Figure 14–1 Using the Catalog Utility



14.1.2 Other Options for Moving Catalog Objects

To download or upload a small number of objects, the download feature of the BI Publisher catalog enables you to bundle and download multicomponent objects (such as reports) in an archive file. You can then use the upload feature to unarchive the data to another location in the catalog. For more information about this feature, see the section "Downloading and Uploading Catalog Objects" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

Note: Do not manually edit the BI Publisher files in the file system. BI Publisher uses metadata files to maintain information about catalog objects. Manually editing objects in the file system can result in the corruption of the metadata files. If the metadata file becomes corrupt, then you can restore it by deleting the corrupt file and restarting BI Publisher.

14.1.3 What Files Are Moved

Table 14–1 lists the files that are included when you export an object from the catalog.

Table 14–1 Files Included In Catalog Export

Object	Files
Report Example: Balance+Letter.xdo	▪ _report.xdo — The report definition file ▪ xdo.cfg — The configuration file that contains the report property settings ▪ ~metadata.meta — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. ▪ security.xml file — Specifies the object level permissions defined for the report ▪ template files — All template files loaded to the report definition. The file names include the language suffix, for example: My_RTF_template_en_us.rtf, My_BIP_layout_en_us.xpt ▪ translation files — All translation files (.xlf), for example: My_RTF_template_jp_jp.xlf

Table 14–1 (Cont.) Files Included In Catalog Export

Object	Files
Data Model Example: myDataModel.xdm	■ _datamodel.xdm — The report definition file ■ ~metadata.meta — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. ■ security.xml file — Specifies the object level permissions defined for the data model
Subtemplate Example: mysubtempate.xsb	■ _template_en_us.rtf — The subtemplate file with locale designation ■ ~metadata.meta — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. ■ security.xml file — Specifies the object level permissions defined for the subtemplate ■ translation files — Any translations, when present; for example: _template_jp_jp.rtf
Style Template Example: myStyleTemplate.xss	■ _template_en_us.rtf — The style template file with locale designation ■ ~metadata.meta — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. ■ security.xml file — Specifies the object level permissions defined for the style template ■ translation files — Any translations, when present; for example: _template_jp_jp.rtf

14.1.4 Maintaining Identical Folder Names and Structure Across Environments

A BI Publisher report references the following components using the physical path to the component in the catalog: data models, subtemplates, and style templates. When you move a report between environments the report maintains the physical mappings to the referenced components. Therefore if you move a data model into a different folder location under Shared Folders in the new environment, the report cannot find the data model and the report does not run. In the case of style templates or subtemplates, the report may run, but the referenced component is not applied.

For example, assume in your test environment Report A references Data Model B that is located in Shared Folders/Test/Data Models. When you move this report and its data model to the production environment you place Data Model B under the different path Shared Folders/Data Models. When you run the report in the new environment it still expects the data model to be located under Shared Folders/Test/Data Models. The report cannot find the data model and does not run.

You can correct the mapping in the new environment by opening the report in the report editor, selecting the data model in its new location, and saving the report.

To avoid manual steps, Oracle recommends that you maintain the same folder names and structure in the environments across which you intend to move reports.

14.2 Preparing to Use the BI Publisher Catalog Utility

The BI Publisher catalog utility is installed in the following location:

```
ORACLE_HOME/clients/bipublisher
```

14.2.1 Configuring the Environment

You must configure each environment in which you run the catalog utility.

To configure the environment for the catalog utility:

1. Set the environment variables to the values in the following list:
   - path = ($HOME/BIPCatalogUtil/bin $path)
   - BIP_LIB_DIR = $HOME/BIPCatalogUtil/lib
   - BIP_CLIENT_CONFIG = $HOME/BIPCatalogUtil/config
   - JAVA_HOME = $HOME/java/jdk1.6.0_18

   The following example shows setting the environment variables for C-shell:

   ```
   % set path = ($HOME/BIPCatalogUtil/bin $path)
   % setenv BIP_LIB_DIR $HOME/BIPCatalogUtil/lib
   % setenv BIP_CLIENT_CONFIG $HOME/BIPCatalogUtil/config
   % setenv JAVA_HOME $HOME/java/jdk1.6.0_18
   ```

2. Edit xmlp-client-config.xml. This configuration file is located under the BIPCatalogUtil/config directory.

   Specify the BI Publisher instance URL ("bipurl") and the user name and password of the BI Publisher instance from which you must export or to which you must import.

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
   <properties>
           <comment>BIP Server Information</comment>
       <entry key="bipurl">http://sta00XXX.example.com:14001/xmlpserver/</entry>
           <entry key="username">OPERATIONS</entry>
       <entry key="password">welcome</entry>
   </properties>
   ```

   If you do not want to store this information in the configuration file, then at the time of import/export you can also set the bipurl, username, and password as parameters in the command line to overwrite values defined in xmlp-client-config.xml.

14.3 Exporting BI Publisher Reporting Objects

Use the export command to export either a single reporting object or a set of BI Publisher reporting objects under a specified folder. There are two export commands:

- -export — Use this command to export a single report object
- -exportfolder — Use this command to export a folder and its contents

Table 14–2 describes the supported parameters for the -export and -exportfolder commands.

Table 14–2 Parameters for Export Commands

Parameter	Used With	Sample	Description
catalogpath	-export -exportfolder	/Samples/Financials /Balance+Letter.xdo	The path to the object in the catalog. If there are spaces in any of the names, use the '+' sign to substitute.

Table 14–2 (Cont.) Parameters for Export Commands

Parameter	Used With	Sample	Description
target	-export	/tmp/Financials/BalanceLetter	The destination directory in which to place the extracted reporting objects.
basedir	-exportfolder	/home/bipub/samples	The base directory into which to place subfolders of extracted reporting objects. When present, data models are saved to {basedir}/datamodels; reports are saved to {basedir}/reports; style and subtemplates are saved to {basedir}/templates.
extract	-export -exportfolder	true/false	The default is false, which means that the utility exports the reporting object in a zip format that contains all the related files such as '.xdo', '.rtf', '.cfg', and so on. If the value is set to 'true', then the utility exports the reporting object-related files under the specified target folder.
subfolders	-exportfolder	true/false	When you specify a folder as the "catalogpath" parameter you can use this "subfolders"' parameter to control whether to download all subfolder content. If you specify true, then all reporting objects in all subfolders are downloaded. If you specify false, then subfolder contents are not downloaded.
overwrite	-export -exportfolder	true/false	Specify true to overwrite existing objects in the target area.

14.3.1 Example Export Command Lines

The following examples show how to use the utility to export the reporting objects:

- Section 14.3.1.1, "Exporting a Single Report in Archive Format"
- Section 14.3.1.2, "Exporting a Single Report with Files Extracted"
- Section 14.3.1.3, "Exporting a Set of Reports to a Specified Folder"

14.3.1.1 Exporting a Single Report in Archive Format

The following example exports the reporting object in a zip format. The zip file contains all the reporting object related files such as .xdo, .rtf, .cfg, and so on. To extract a report in archived format use the ".xdoz" extension for the target. To extract a data model, use the ".xdmz" extension.

```
$ BIPCatalogUtil.sh -export catalogpath=/Samples/Financials/Balance+Letter.xdo
target=/home/bipub/reports/BalanceLetter.xdoz extract=false
```

14.3.1.2 Exporting a Single Report with Files Extracted

The following example extracts the reporting object-related files to a directory named "/home/bipub/reports/BalanceLetter". Existing files are overwritten.

```
$ BIPCatalogUtil.sh -export catalogpath=/Samples/Financials/Balance+Letter.xdo
```

```
target=/home/bipub/reports/BalanceLetter extract=true overwrite=true
```

14.3.1.3 Exporting a Set of Reports to a Specified Folder

The following example extracts all the reporting objects under the "/Samples" folder and its subfolders in the catalog. Data models are saved under {basedir}/datamodels. Reports are saved into {basedir}/reports. Style and subtemplates are saved into {basedir}/templates.

```
$ BIPCatalogUtil.sh -exportfolder catalogpath=/Samples basedir=/home/bipub/samples
subfolders=true extract=true overwrite=true
```

14.4 Importing BI Publisher Reporting Objects

Use the import command to import either a single BI Publisher reporting object or a set of BI Publisher reporting objects under a specified folder. Table 14–3 describes the supported parameters for the import command.

Table 14–3 Parameters for Import Command

Parameter	Sample	Description
catalogpath	/Samples/Financials/Balance+Letter.xdo	Specify the catalog path to where you want to import the reporting object only when you want to override the default information. If you do not specify this parameter, then the reporting object is imported to the same location where it was originally exported from.
source	/tmp/Financials/BalanceLetter	The directory where the reporting object is located. Use this parameter when you are importing a single report.
basedir	/home/bipub/samples	The directory that contains multiple reports or data models to be imported. Specify this parameter when importing a set of reports or data models.
overwrite	true/false	Specify 'true' to overwrite existing objects in the target area.

Typically, you import the reporting object to where it was originally exported from. When you export the reporting object with the utility, it generates a metafile (.meta) that contains the catalog path information. The utility uses this information to import the reporting object to the original location. To import the objects into a different location, you can override the original catalog path location by specifying the catalogpath parameter.

14.4.1 Example Import Command Lines

The following examples show how to use the utility to import reports:

- Section 14.4.1.1, "Importing a Report to an Original Location"
- Section 14.4.1.2, "Importing a Report to a New Location"
- Section 14.4.1.3, "Importing a Zipped Report"
- Section 14.4.1.4, "Importing a set of BI Publisher Reporting Objects Under a Specified Folder"

14.4.1.1 Importing a Report to an Original Location

The following example imports a report to a catalog path saved in its metafile (.meta). Existing reports are overwritten.

```
$ BIPCatalogUtil.sh -import source=/tmp/Financials/BalanceLetter overwrite=true
```

14.4.1.2 Importing a Report to a New Location

The following example imports a report into a new location in the catalog.

```
$ BIPCatalogUtil.sh -import source=/home/bipub/reports/BalanceLetter
catalogpath=/Production/Financials/Balance+Letter+Report.xdo
```

14.4.1.3 Importing a Zipped Report

The following example imports a zipped reporting object to an original location in the catalog.

```
$ BIPCatalogUtil.sh -import source=/home/bipub/reports/BalanceLetter.xdoz
overwrite=true
```

14.4.1.4 Importing a set of BI Publisher Reporting Objects Under a Specified Folder

The following example imports all the reports under the base directory (basedir) into the original locations in the catalog.

```
$ BIPCatalogUtil.sh -import basedir=/Users/bipub subfolders=true overwrite=true
```

14.5 Generating Translation Files and Checking for Translatability

The catalog utility supports the -xliff command to generate a translatable XLIFF file for a specific file. Table 14–4 describes the supported parameters for generating XLIFF files.

The source file can be the report definition (.xdo) file, an RTF template file (.rtf), or a BI Publisher layout template file (.xpt). When the source is the .xdo file, the generated XLIFF file includes all user-entered strings from the report definition interface, for example: description, layout names, parameter names.

Table 14–4 Parameters for Generating XLIFF Files

Parameter	Sample	Description
source	/Samples/Financials/Balance+Letter.xdo	The path to the report or template file (RTF or XPT) for which to generate the XLIFF file.
target	/home/bipub/reports/Balance+Letter/Balance+Letter.xlf	The location to save the generated .xlf document.
basedir	/home/bipub/reports/Balance+Letter/	The directory to place the generated .xlf files into.

The following examples show how to generate translation files:

- Section 14.5.1, "Generating a Translation File for a Report Definition File (.xdo)"
- Section 14.5.2, "Generating a Translation File for an RTF Template"

14.5.1 Generating a Translation File for a Report Definition File (.xdo)

The following example generates an XLIFF file for a single report definition file:

```
$ BIPCatalogUtil.sh -xliff
source=/home/bipub/reports/Balance+Letter/Balance+Letter.xdo
target=/home/bipub/reports/Balance+Letter/Balance+Letter.xlf
```

To save the XLIFF to a base directory:

```
$ BIPCatalogUtil.sh -xliff source=/home/bipub/reports/Balance/Balance+Letter.xdo
basedir=/home/bipub/reports/Balance+Letter/
```

14.5.2 Generating a Translation File for an RTF Template

The following example generates an XLIFF file for a single RTF template file:

```
$ BIPCatalogUtil.sh -xliff
source=/home/bipub/reports/Balance+Letter/Balance+Letter+Template.rtf
target=/home/bipub/reports/Balance+Letter/Balance+Letter+Template.xlf
```

To save the XLIFF to a base directory:

```
$ BIPCatalogUtil.sh -xliff
source=/home/bipub/reports/Balance/Balance+Letter+Template.rtf
basedir=/home/bipub/reports/Balance+Letter/
```

15

Customizing the BI Publisher User Interface

The user interface in BI Publisher is generated by using scripts and is therefore highly customizable. The look-and-feel is controlled by skins and styles. BI Publisher is shipped with the Skyros (default style), and blafplus (browser look-and-feel plus), styles.

The following sections provide information about how to customize the BI Publisher user interface:

- Section 15.1, "What are Skins and Styles?"
- Section 15.2, "About Style Customizations"
- Section 15.3, "Modifying the User Interface Styles for BI Publisher"
- Section 15.4, "Customizing the Style"

> **Note:** Customizing the BI Publisher user interface applies to Oracle BI Publisher 11.1.1.7.10 and later versions, and might not be available in earlier versions. For more information about Oracle BI Publisher 11.1.1.7.10, see "New Features for 11.1.1.7.10."

15.1 What are Skins and Styles?

Styles and skins are organized into folders that contain Cascading Style Sheets (CSS) and images. Skins and styles are typically used to customize the look and feel of the BI Publisher user interface by providing logos, color schemes, fonts, table borders, and other elements. Skins and styles can also be used to control the position and justification of various elements by including specialized style tags in the relevant style sheet file. For more information, see Section 15.2, "About Style Customizations."

15.2 About Style Customizations

To customize the look-and-feel of BI Publisher, Oracle strongly recommends that you use the custom style provided in the bicustom-template.ear file as your starting point. This custom style is a copy of the Skyros style.

For more information, see Section 15.3, "Modifying the User Interface Styles for BI Publisher."

Most of the common Skyros styles and image files, including the style sheet (master.css), are contained in the master directory. For more information about the master directory and its structure, see "Customizing Your Style" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

Within the master.css style sheet, each element (or class) that is available for update is documented in the comments.

Other style sheets are also contained within the Skyros style and skin folders. You are not likely to need to update these files unless you are creating an advanced custom skin that provides styles for each detail of the user interface.

Note: The Skyros style does not apply to Administration pages in BI Publisher.

15.3 Modifying the User Interface Styles for BI Publisher

To change the skin for BI Publisher, modify the xmlp-server-config.xml configuration file located at CATALOG_
DIRECTORY/Admin/Configuration/xmlp-server-config.xml as follows:

To change the skin to blafplus, set the THEME property as follows:

```
<property name="THEME" value="blafplus"/>
```

To change the skin back to the default skin, Skyros, set the THEME property as follows:

```
<property name="THEME" value="skyros"/>
```

Note: The THEME property must be either "blafplus" or "skyros".

15.4 Customizing the Style

Enterprise Archive (EAR) files are archive (ZIP) files composed of a specific folder and file structure. You can create an EAR file using any ZIP tool (for example, 7-zip) and then rename the ZIP extension to EAR. Oracle provides the bicustom-template.ear file as a starting point.

The bicustom-template.ear file contains a bicustom.war file. Web Archive (WAR) files are also ZIP files composed of a specific folder and file structure. You must update the bicustom.war file within the bicustom-template.ear file to include your custom skin files. The bicustom.war file that is shipped with BI Publisher contains an example folder structure to help you get started.

When creating styles and skins for BI Publisher, you must create CSS and image files, and make them available to BI Publisher. Only the CSS defined in master.css and images defined in the master folder can be customized for BI Publisher (bundled in the bicustom.ear file.)

15.4.1 Customizing the Style for BI Publisher Standalone

To create a custom style for BI Publisher when BI Publisher is not integrated with the Oracle BI Enterprise Edition:

1. Copy ORACLE_HOME\bifoundation\jee\bicustom-template.ear to ORACLE_HOME\bifoundation\jee\bicustom.ear.

Note: The patch or upgrade process may overwrite the bicustom-template.ear file, but it does not overwrite the bicustom.ear file.

2. Extract the bicustom.war file from the bicustom.ear file to the machine where BI Publisher is deployed.
3. Extract the files from the bicustom.war file.
4. Edit the master.css and images files in the unzipped directory to create the custom style, and save the changes.
5. Update the bicustom.war file with the changes.
6. Update the bicustom.ear file with the new bicustom.war file.
7. Deploy the new bicustom.ear file to the application server.
8. Update the xmlp-server-config.xml file and save the changes.

   The following example configurations assume that bicustom.ear file is deployed with application name "custom" on the same application server where BI Publisher is running:

```
<!-- required: this is the base skin to use for styles not defined inside
custom css -->
<property name="THEME" value="skyros"/>
<!-- required: this is the custom css http url -->
<property name="THEME_CUSTOM_MASTER_CSS_URL" value="/custom/res/s_
Custom/master/master.css"/>
<!-- required: this is the custom image http url prefix -->
<property name="THEME_CUSTOM_MASTER_IMAGE_URL_PREFIX" value="/custom/res/s_
Custom/master"/>
<!-- required: this is the file system path where custom images are located -->
<property name="THEME_CUSTOM_MASTER_IMAGE_PATH"
value="/scratch/aime1/custom/res/s_Custom/master"/>
```

   Note that when a web page displays an image, the image is fetched through HTTP. Therefore an image must be available through an HTTP URL no matter where it is stored in the local directory. If you deploy bicustom.ear but place a custom image in a unrelated local directory, the result is that the HTTP URL is serving one image while the local directory is serving another image. To ensure that the and HTTP URL and the local path are pointing to the same image file, unpack bicustom.ear into the local directory (for example, path_A), make changes to the css/images, and then install a "custom" application from the unpacked local directory path_A.
9. Restart BI Publisher.

For information on redeploying the bicustom.ear file, see "Approach 1: Redeploying the "bicustom.ear" File" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

15.4.2 Customizing the Style for BI Publisher Integrated with the Oracle BI Enterprise Edition

To create a custom style for BI Publisher integrated with Oracle BI Enterprise Edition:

1. Copy ORACLE_HOME\bifoundation\jee\bicustom-template.ear to ORACLE_HOME\bifoundation\jee\bicustom.ear.

Note: The patch or upgrade process may overwrite the bicustom-template.ear file, but it does not overwrite the bicustom.ear file.

2. Extract the bicustom.war file from the bicustom.ear file to the machine where BI Publisher is deployed.
3. Extract the files from the bicustom.war file.
4. Edit the master.css and images files in the unzipped directory to create the custom style, and save the changes.
5. Update the bicustom.war file with the changes.
6. Update the bicustom.ear file with the new bicustom.war file.
7. Deploy the new bicustom.ear file to the application server.
8. Update the xmlp-server-config.xml file and save the changes.

```
<!-- required: http url of OBIEE master css -->
<property name="THEME_MASTER_CSS_URL" value="/custom/s_
skyros/master/master.css"/>
<!-- required: this is the master css http url -->
<property name="THEME_IMAGE_URL_PREFIX" value="/custom/s_skyros/master"/>
<!-- required: this is the file system path where master images are located -->
<property name="THEME_MASTER_IMAGE_PATH" value="/scratch/aime1/bip/res/s_
Custom/master"/>
```

9. Restart BI Publisher.

Note: The custom configuration properties override the master configuration properties; therefore the value of THEME_CUSTOM_MASTER_CSS_URL takes precedence over the value of THEME_MASTER_CSS_URL. The same rule applies for images.

15.4.3 Fallback Mechanism for Custom Styles

When creating custom styles for BI Publisher (standalone and integrated Oracle BI Enterprise), Oracle recommends that you copy only what you want to change in the customization. Anything not copied "falls back" to the style specified in the base skin for BI Publisher, which is the Skyros theme.

15.4.3.1 Custom Style Sheets

For custom style sheets (css), if THEME_CUSTOM_MASTER_CSS_URL is provided, BI Publisher references those styles and ignores any others. If THEME_MASTER_CSS_URL is provided, BI Publisher uses those styles. If neither are provided, BI Publisher uses the styles defined in the base skin.

15.4.3.2 Images

For images, if THEME_CUSTOM_MASTER_IMAGE_PATH is provided, and the requested image exists in the directory, BI Publisher uses the value of THEME_CUSTOM_MASTER_IMAGE_URL_PREFIX to construct the image URL.

If THEME_MASTER_IMAGE_PATH is provided and the requested image exists in the directory, BI Publisher uses the value of THEME_MASTER_IMAGE_URL_PREFIX to

construct the image URL. If neither are provided, BI Publisher uses the images defined in the base skin.

A

Scheduler Configuration Reference

This appendix describes how to configure the BI Publisher scheduler for each supported database and how to configure ActiveMQ as the JMS provider.

It covers the following topics:

- Section A.1, "Introduction"
- Section A.2, "Configuring BI Publisher for ActiveMQ"
- Section A.3, "Manually Configuring the Quartz Scheduler"

A.1 Introduction

The Oracle Business Intelligence Platform Installer configures the connection to the scheduler and installs the scheduler schema to your selected scheduler database. The WebLogic JMS queues are set up and the scheduler is up and running after installation is complete and the servers have been started.

This information in this appendix is provided for reference for manually configuring the scheduler and for setting up ActiveMQ as an alternative JMS provider.

For conceptual information about the scheduler, information for installing and configuring additional managed servers, and a description of the scheduler diagnostics page, see Chapter 7, "Configuring the Scheduler."

A.2 Configuring BI Publisher for ActiveMQ

The scheduler is configured by default to use WebLogic JMS. The scheduler also supports ActiveMQ as an alternative JMS provider. Use these guidelines with the ActiveMQ documentation to configure BI Publisher if you choose to use ActiveMQ as the JMS provider.

A.2.1 Install ActiveMQ

It is recommended that you install ActiveMQ version 5.2.0 or later. This can be installed in Windows, UNIX or Linux. Follow the installation steps documented at the following location:

`http://activemq.apache.org`

A.2.2 Register ActiveMQ as a JNDI Service

When you start ActiveMQ, the queues can be accessed using JNDI service.

The default URL to access this service is:

failover://tcp://localhost:61616

To change this configuration, update the activemq.xml configuration file found in apache-activemq-x.x.x\conf for example: apache-activemq-5.2.0\conf.

A.2.3 Update the BI Publisher Scheduler Configuration Page

To update the Scheduler Configuration page:

1. On the BI Publisher Administration page, under **System Maintenance**, click **Scheduler Configuration**.
2. Under the JMS Configuration region, select ActiveMQ.
3. Enter the ActiveMQ JNDI URL. For example: failover://tcp://localhost:61616
4. Enter the threads per processor (for example: 5).
5. Enter the path to a shared temporary directory.
6. Click **Test JMS** to test the connection.
7. Click **Apply** to apply the changes to this page.

The ActiveMQ URL is dynamically applied. The queues and topics are automatically created in ActiveMQ and are ready for scheduling. You can confirm the queues by checking them in the Scheduler Diagnostics page. Alternatively, you can check the status in the ActiveMQ Web console: http://localhost:8161/admin.

A.3 Manually Configuring the Quartz Scheduler

BI Publisher includes the Hyperion-branded DataDirect Connect for JDBC drivers to setup a connection to install and use the scheduler tables in your database. These drivers can be used as an alternative to the native JDBC drivers provided by your database vendor. When you choose a database for which a DataDirect driver is available, BI Publisher automatically enters the database driver class information in the setup screen for you. There is no additional setup required for the driver files.

If you choose to use a data direct driver not provided by the BI Platform Installer, then you must download, install, and configure the driver manually.

A.3.1 Recommendations for Using DataDirect Connect or Native Database Drivers

DataDirect Connect for JDBC drivers are provided for the following databases:

- IBM DB2 v8.1, v9.1
- Microsoft SQL Server 2000, 2005
- Sybase Adaptive Server Enterprise
- Oracle 9*i*, Oracle 10*g*, Oracle 11*g*,

> **Note:** Some database options listed here and in the Scheduler page might not be supported in this release. See "System Requirements and Certification" for the most up-to-date information on supported hardware and software.

Table A–1 displays the driver recommendations for the supported scheduler databases.

Table A–1 Driver Recommendations

Database	Native JDBC Driver	DataDirect JDBC Driver
Oracle 10*g*, Oracle 11*g*	Recommended	Supported
IBM DB2 v8.1, v9.1	Supported	Recommended
Microsoft SQL Server 2000, 2005	Supported	Recommended
Sybase Adaptive Server Enterprise	Supported	Recommended
MySQL 4.1.10a-NT, 5.0	Supported	Not Supplied

A.3.2 Set Up a User on Your Scheduler Database

To set up the connection to the scheduler database, you must ensure that you have created a user on the selected database. BI Publisher uses this user to connect to the database. Depending on the database type, this user might require specific privileges. These are detailed in the database-specific sections later in this appendix.

A.3.3 Connecting to Your Scheduler Database and Installing the Schema

Following are the general steps for setting up the Scheduler database. Also refer to the subsequent section that is specific to your database.

To set up the Scheduler database:

1. Log in to BI Publisher with Administrator credentials and select the **Administration** tab.
2. Under **System Maintenance**, click **Scheduler Configuration**.
3. In the **Scheduler Selection** region, select Quartz.

> **Note:** The option "Enterprise Scheduler Services" is reserved for Oracle Fusion Applications.

4. Enter the following fields for the Database Connection:
   - **Database Type** — Select the database from the list. After you make a selection, the Database Driver Class field automatically updates with the recommended driver class.
   - **Connection String** — Enter the connection string for your selected database. Sample strings are provided in the database-specific sections that follow.
   - **Username** and **Password** — Enter the scheduler user you set up for your database. The user must have permissions to connect to the database and create tables. Other permissions might be required depending on the database type. See the appropriate database-specific section later in this chapter.
   - **Database Driver Class** — When you select the database type this field is automatically updated with the recommended driver. If you want to use another driver, then specify it in this field.

Note: **Note:** The Oracle database drivers and the DataDirect drivers are installed with BI Publisher and no further setup is required. Note that for other databases, even though the recommended native drivers are automatically populated in this field, additional setup is required to make the drivers available to BI Publisher.

5. Click **Test Connection** to ensure that BI Publisher can connect to the database. If the connection fails, ensure that you have entered the fields as shown and set up your database appropriately.
6. Click **Install Schema** to install the BI Publisher scheduler schema to your database.

A.3.4 Connecting to Oracle Database

Prerequisite: Ensure that the database user you enter has "connect" or "create session" and "create table" privileges and that the user has been assigned a quota (otherwise the quota is 0).

For example, the following sample creates the user "bipuser":

```
SQL> CREATE USER bipuser
   2  IDENTIFIED BY welcome
   3  DEFAULT TABLESPACE USERS
   4  TEMPORARY TABLESPACE TEMP
   5  QUOTA 20G ON USERS
   6  QUOTA 1M ON TEMP;

User created.

 SQL> GRANT CREATE SESSION TO bipuser; -- or "GRANT CONNECT TO bipuser;"

 Grant succeeded.

 SQL> grant create table to bipuser;

 Grant succeeded.
```

Table A–2 describes the fields for the Oracle native driver to connect to the Oracle Database.

Table A–2 Oracle Native Driver Fields

Field	Description
Database Type:	Select Oracle 11*g* or Oracle 10*g* from the list.
Connection String:	Enter the following connection string parameters: jdbc:oracle:thin:@<hostname>:<port>:<oracle SID> For example: jdbc:oracle:thin:@mydatabaseserver.com:1521:bipscheduler
Database Driver Class:	oracle.jdbc.driver.OracleDriver

A.3.5 Connecting to IBM DB2

Prerequisite: Ensure that the user that you enter to configure the scheduler has been set up with a 32 K page size tablespace. If not, create the table and assign it to the user. The user must also have "Connect to database" and "Create tables" privileges.

Table A–3 describes the fields for the DataDirect driver to connect to an IBM DB2 v8 or IBM DB2 v9 database.

Table A–3 DataDirect Driver Fields for IBM Databases

Field	Entry
Database Type:	Select IBM DB2 v9 or IBM DB2 v8 from the list.
Connection String:	Enter the following connection string parameters:jdbc:hyperion:db2://<hostname>:<port>;DatabaseName=<DATABASENAME> For example: jdbc:hyperion:db2://mydatabaseserver.com:1433;DatabaseName=bipscheduler
Database Driver Class:	hyperion.jdbc.db2.DB2Driver

A.3.6 Connecting to Microsoft SQL Server

Prerequisite: Ensure that the Microsoft SQL Server is set up with mixed mode authentication. Also ensure that the user that you enter to configure the scheduler has the "db_owner" role.

Table A–4 describes the fields for the DataDirect driver to connect to a Microsoft SQL Server 2000 or 2005 database.

Table A–4 DataDirect Driver Fields for SQL Server Databases

Field	Entry
Database Type:	Select Microsoft SQL Server 2000 or Microsoft SQL Server 2005 from the list.
Connection String:	Enter the following connection string parameters: jdbc:hyperion:sqlserver://<hostname>:<port>;DatabaseName=<DATABASENAME> For example: jdbc:hyperion:sqlserver://mydatabaseserver.com:1433;DatabaseName=bipscheduler
Database Driver Class:	hyperion.jdbc.sqlserver.SQLServerDriver

A.3.7 Connecting to Sybase Adaptive Server Enterprise Database

Prerequisite: Ensure that you set the "ddl in tran" mode to true in the database. Consult the Sybase documentation or contact your database administrator for instruction how to enable this option.

Table A–5 describes the fields for the DataDirect driver to connect to a Sybase Adaptive Server Enterprise database.

Table A–5 DataDirect Driver Fields for Sybase Database

Field	Entry
Database Type:	Select Sybase Adaptive Server Enterprise from the list.
Connection String:	Enter the following connection string parameters: jdbc:hyperion:sybase://<hostname>:<port>;DatabaseName=<DATABASENAME> For example: jdbc:hyperion:sybase://mydatabaseserver.com:4100;DatabaseName=bipscheduler
Database Driver Class:	hyperion.jdbc.sybase.SybaseDriver

B

Integration Reference for Oracle BI Enterprise Edition

This appendix describes configuration details for integrating BI Publisher with Oracle BI Presentation Services and Oracle BI Server.

It covers the following topics:

B.1 About Integration

The information in this chapter is for reference to highlight the integration points between BI Publisher and the Oracle BI Enterprise Edition.

You might need to reference this information in the following scenarios:

- You are upgrading from a 10*g* release to the 11*g* release
- You run a separate installation of BI Publisher and want to integrate it
- You need to modify the installed configuration

The points of integration discussed in this chapter are:

- Connecting to Oracle BI Server as a data source
- Configuring BI Publisher to use the Oracle BI Presentation Catalog
- Configuring integration with Oracle BI Presentation Services

B.1.1 Prerequisites

Oracle BI Publisher must be installed on the same server with the other components of Oracle BI Enterprise Edition.

The security configuration must be either Oracle Fusion Middleware security or Oracle BI Server security.

B.2 Configuring BI Publisher to Use the Oracle BI Presentation Catalog

To manually configure BI Publisher to use the Oracle BI Presentation Catalog:

1. On the **Server Configuration** page in the **Catalog** region, select **Catalog Type**: Oracle BI EE Catalog
2. Enter the following:
   - **Server Version** - v7
   - **System Username**

     System Password

     Leave these fields blank.

     BI Publisher uses the BISystemUser and password to connect to the BI Presentation Catalog. The BISystemUser is a fixed user created during installation for trusted communication between components. Typically, these fields are configured at installation and are no longer visible from this configuration page. If you are manually configuring the integration between BI Publisher and BI Presentation Catalog and these fields are visible, you can leave them blank, as BI Publisher automatically reads the values from the security store.
   - **Connection Protocol** — TCP

   If you are manually configuring BI Publisher to use the BI Presentation Catalog, you may see the fields **System Username** and **System Password**. These fields can be left blank, as these values are automatically read by BI Publisher from the security store.
3. Click **Test Connection** to ensure BI Publisher can connect to Oracle BI Presentation Services.
4. Enter the path of the current **BI Publisher repository**.
5. Click **Upload to BI Presentation Catalog**. This action uploads the contents of the BI Publisher catalog to the BI Presentation catalog.
6. Restart the BI Publisher application.

B.3 Configuring Integration with Oracle BI Presentation Services

When you install Oracle BI Enterprise Edition the integration with BI Publisher is automatically configured. This means that the Oracle BI Platform installer sets the Presentation Services host name, port, and URL suffix values. Furthermore, the username and password fields are hidden, because both products are configured to use Oracle Fusion Middleware security.

To configure integration with Presentation Services:

1. From the Administration page, under **Integration**, click **Oracle BI Presentation Services**.
2. Enter the following information about your BI Presentation Services server:
   - Server Protocol — Select http or https
   - Server Version — Select v6
   - Server — Enter the server host name. For example: BIEEServer
   - Port for the server where the BI Presentation Services plug-in is running. For example: 9704

- Administrator Username and Password — These fields are hidden when using Oracle Fusion Middleware Security. If you are manually configuring the integration, enter the BISystemUser username and password.
- URL Suffix — Default value is: analytics/saw.dll

> **Note:** If your deployment is configured for SSO, then the suffix must be entered as "analytics-ws/saw.dll" to allow the Web services between BI Publisher and BI Presentation Services. For more information on configuring SSO for Oracle BI Enterprise Edition, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

- Session time out in minutes

B.4 Setting Up a JDBC Connection to the Oracle BI Server

> **Note:** If you installed BI Publisher with the Oracle BI Enterprise Edition, then this data source is automatically configured.

To add the Oracle BI Enterprise Edition server as a JDBC data source, follow the guidelines in Section 9.2, "Setting Up a JDBC Connection to the Data Source" with these specific guidelines.

Note that if your Oracle BI Server is SSL-enabled, then you must copy the keystore to the BI Publisher server and provide it in the connection string.

The entries for Database Driver Class and Connection String must be as follows:

Database Driver Class — oracle.bi.jdbc.AnaJdbcDriver

Connection String — The appropriate connection string depends on your specific deployment. Clustered and SSL-enabled deployments require specific parameters to construct the URL. For example, if the Oracle BI Server is SSL-enabled, then you must copy the keystore to the BI Publisher server and provide it in the connection string. For more information on SSL, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

The URL for the connection string requires the following format:

```
<URL>:= <Prefix>: [//<Host>:<Port>/][<Property Name>=<Property Value>;]*
```

where

<Prefix> — The string jdbc:oraclebi

<Host> — The hostname of the analytics server. It can be an IP Address or hostname. The default is localhost.

<Port> — The port number that the server is listening on. The default is 9703.

```
<Property Name>:= <Catalog>|<User>|<Password>|<SSL>|<SSLKeyStoreFileName>
|<SSLKeyStorePassword>|<TrustAnyServer>|<TrustStoreFileName
>|<TrustStorePassword>|<LogLevel>|<LogFilePath>|<PrimaryCCS>|<PrimaryCCSPort>|
<SecondaryCCS>|<SecondaryCCSPort>
```

Valid property values are:

<Catalog> — Any catalog name that is available on the server. If the catalog is not specified, then it defaults to the default catalog specified by the server. If the catalog name is not found in the server, then it still uses the default catalog and issues a warning during connect.

<User> — Specifies the user name for the BI Server. The default is "Administrator".

<Password> — Specifies the password for the BI Server for the user name. The password is encrypted using 3DES.

<SSL> True | False — Default is False. Specifies if the JDBC driver uses SSL or not. If true, then driver checks whether SSLKeyStoreFileName is readable; if not, it issues an error message.

<SSLKeyStoreFileName> — Specifies the name of the file that store the SSL Keys. This file must exist in the local file system and be readable by the driver.

<SSLKeyStorePassword> — Specifies the password to open the file pointed to by SSLKeyStoreFileName.

<TrustAnyServer> - True | False — The default is False. If SSL is set to "True" the property specifies whether to check the trust store for the server. If TrustAnyServer is set to "False", the driver verifies that TrustStoreFileName is readable.

<TrustStoreFileName> — If TrustAnyServer is set to false, this property is required to specify the trust store file name.

<TrustStorePassword> — If TrustAnyServer and TrustStoreFileName are specified, this property specifies the password to open up the file specified by TrustStoreFileName.

<LogLevel> — Specifies the log level. Valid values are

SEVERE | WARNING | INFO | CONFIG | FINE | FINER | FINEST

<LogFilePath> — Specifies the file path of the desired logging destination. Default is %TEMP% on windows, $TMP on UNIX. Driver needs to have write permission on the file. It creates a new entry marked as _0, _1 if the same file name exists.

<PrimaryCCS> — (For clustered configurations) specifies the primary CCS machine name instead of using the "host" to connect. If this property is specified, the "host" property value is ignored. The jdbc driver tries to connect to the CCS to obtain the load-balanced machine. Default is localhost.

<PrimaryCCSPort> — Specifies the primary CCS port number running on the PrimaryCCS machine. Default is 9706.

<SecondaryCCS> — Specifies the secondary CCS machine name instead of using the "host" to connect. If this property is specified, then the jdbc driver trues to connect to the CCS to obtain the load-balanced machine. Default is localhost.

<SecondaryCCSPort> — Specifies the secondary CCS port number running on the secondary machine. Default is 9706.

Following is an example connection string for a clustered deployment with SSL enabled:

```
jdbc:oraclebi://machine01.domain:9706/PrimaryCCS=machine01;PrimaryCCSPort=9706;Sec
ondaryCCS=machine02;SecondaryCCSPort=9706;user=admin;password=welcome;ssl=true;ssl
Keystorefilename=c:\mycompany\OracleBI\sslc\javahost.keystore;sslKeystorepassword=
welcome;trustanyserver=true;
```

Use System User - you must select this box to use the BISystem User. When you select this box, BI Publisher will use the BISystem Username and password to connect to the BI Server. The Username and Password fields are no longer editable.

Username - leave blank

Password- leave blank

Use Proxy Authentication — (Required) select this box. Proxy authentication is required.

C

Configuration File Reference

This appendix describes the BI Publisher run-time configuration file.

It covers the following topics:

C.1 BI Publisher Configuration Files

This appendix contains reference information about the following BI Publisher configuration file:

- Runtime Configuration Properties File

The properties in the Runtime Configuration file are set through the Runtime Configuration Properties, Currency Formats, and Font Mappings pages. (For information, see Section 11.1, "Setting Run-Time Properties.")

C.2 Setting Properties in the Runtime Configuration File

The runtime properties and font mappings are set through the Runtime Configuration Properties page and the Font Mappings page in the Administration interface.

If you do not use the Administration page to set the properties, then BI Publisher falls back to the properties set in this file.

It is important to note that the Administration interface does not update this file. Any settings in the Administration pages take precedence over the settings in the xdo.cfg file.

C.2.1 File Name and Location

The configuration file is named xdo.cfg.

The file is located under the <BI Publisher Repository>/Admin/Configuration.

C.2.2 Namespace

The namespace for this configuration file is:

http://xmlns.oracle.com/oxp/config/

C.2.3 Configuration File Example

Following is a sample configuration file:

```
<config version="1.0.0"
  xmlns="http://xmlns.oracle.com/oxp/config/">

   <!-- Properties -->
   <properties>
      <!-- System level properties -->
      <property name="system-temp-dir">/tmp</property>

      <!--  PDF compression -->
      <property name="pdf-compression">true</property>

     <!--  PDF Security -->
     <property name="pdf-security">true</property>
     <property name="pdf-open-password">user</property>
     <property name="pdf-permissions-password">owner</property>
     <property name="pdf-no-printing">true</property>
     <property name="pdf-no-changing-the-document">true</property>
    </properties>

   <!-- Font setting -->
   <fonts>
      <!-- Font setting (for FO to PDF etc...) -->
      <font family="Arial" style="normal" weight="normal">
         <truetype path="/fonts/Arial.ttf" />
      </font>
      <font family="Default" style="normal" weight="normal">
         <truetype path="/fonts/ALBANWTJ.ttf" />
      </font>

      <!--Font substitute setting (for PDFForm filling etc...) -->
      <font-substitute name="MSGothic">
         <truetype path="/fonts/msgothic.ttc" ttcno="0" />
      </font-substitute>
   </fonts>
</config>
```

C.2.4 Understanding the Element Specifications

The following is an example of an element specification:

```
<Element Name Attribute1="value"
              Attribute2="value"
                                        AttributeN="value"
   <Subelement Name1/>[occurrence-spec]
   <Subelement Name2>...</Subelement Name2>
   <Subelement NameN>...</Subelement NameN>
</Element Name>
```

The [occurrence-spec] describes the cardinality of the element, and corresponds to the following set of patterns:

- [0..1] — Indicates the element is optional, and might occur only once.
- [0..n] — Indicates the element is optional, and might occur multiple times.

C.3 Structure of the Root Element

The <config> element is the root element. It has the following structure:

```
<config version="cdata" xmlns="http://xmlns.oracle.com/oxp/config/">
   <fonts> ... </fonts> [0..n]
   <properties> ... </properties> [0..n]
</config>
```

C.3.1 Attributes of Root Element

The <config> element has the attributes described in Table C–1.

Table C–1 config Element Attributes

Attribute	Description
version	The version number of the configuration file format. Specify 1.0.0.
xmlns	The namespace for BI Publisher's configuration file. Must be http://xmlns.oracle.com/oxp/config/

C.3.2 Description of Root Element

The root element of the configuration file. The configuration file consists of two parts:

- Properties (<properties> elements)
- Font definitions (<fonts> elements)

The <fonts> and <properties> elements can appear multiple times. If conflicting definitions are set up, the last occurrence prevails.

C.4 Properties and Property Elements

This section describes the <properties> element and the <property> element.

C.4.1 <properties> Element

The <properties> element is structured as follows:

```
<properties locales="cdata">
   <property>...
   </property> [0..n]
</properties>
```

C.4.1.1 Description of <properties> Element

The <properties> element defines a set of properties. You can specify the locales attribute to define locale-specific properties. Following is an example:

```
<!-- Properties for all locales -->
<properties>
 ...Property definitions here...
</properties>
```

```
<!--Korean specific properties-->
<properties locales="ko-KR">
 ...Korean-specific property definitions here...
</properties>
```

C.4.2 <property> Element

The <property> element has the following structure:

```
<property name="cdata">
  ...pcdata...
</property>
```

C.4.2.1 Attribute of <property> Element

The <property> element has a single attribute, name, which specifies the property name.

C.4.2.2 Description of <property> Element

Property is a name-value pair. Specify the internal property name (key) to the name attribute and the value to the element value. The internal property names used in the configuration file are listed in the property descriptions in Chapter 11, "Defining Run-Time Configurations."

```
<properties>
  <property name="system-temp-dir">d:\tmp</property>
  <property name="system-cache-page-size">50</property>
  <property name="pdf-replace-smart-quotes">false</property>
</properties>
```

C.5 Font Definitions

Font definitions include the following elements:

- <fonts>
- <font>
- <font-substitute>
- <truetype>
- <type1>

For the list of Truetype and Type1 fonts, see Section C.6, "Predefined Fonts."

C.5.1 <fonts> Element

The <fonts> element is structured as follows:

```
<fonts locales="cdata">
   <font> ... </font> [0..n]
   <font-substitute> ... </font-substitute> [0..n]
</fonts>
```

C.5.1.1 Attribute of <fonts> Element

The <fonts> element has a single optional attribute, locales, which specifies the locales for this font definition.

C.5.1.2 Description of <fonts> Element

The <fonts> element defines a set of fonts. Specify the locales attribute to define locale-specific fonts.

```
<!-- Font definitions for all locales -->
<fonts>
  ..Font definitions here...
</fonts>

<!-- Korean-specific font definitions -->
<fonts locales="ko-KR">
... Korean Font definitions here...
</fonts>
```

C.5.2 <font> Element

Following is the structure of the <font> element:

```
<font family="cdata" style="normalitalic"
weight="normalbold">
   <truetype>...</truetype>
or <type1> ... <type1>
</font>
```

C.5.2.1 Attributes of <font> Element

The <font> element has the attributes described in Table C–2.

Table C–2 font Element Attributes

Attribute	Description
family	Specify any family name for the font. If you specify "Default" for this attribute, then you can define a default fallback font. The **family** attribute is case-insensitive.
style	Specify "normal" or "italic" for the font style.
weight	Specify "normal" or "bold" for the font weight.

C.5.2.2 Description of <font> Element

Defines a BI Publisher font. This element is primarily used to define fonts for FO-to-PDF processing (RTF to PDF). The PDF Form Processor (used for PDF templates) does not refer to this element.

```
<!-- Define "Arial" font -->
<font family="Arial" style="normal" weight="normal">
  <truetype path="/fonts/Arial.ttf"/>
</font>
```

C.5.3 <font-substitute> Element

Following is the structure of the <font-substitute> element:

```
<font-substitute name="cdata">
   <truetype>...</truetype>
or <type1>...</type1>
</font-substitute>
```

C.5.3.1 Attributes of <font-substitute> Element

The <font-substitute> element has a single attribute, name, which specifies the name of the font to be substituted.

C.5.3.2 Description of <font-substitute> Element

Defines a font substitution. This element is used to define fonts for the PDF Form Processor.

```
<font-substitute name="MSGothic">
   <truetype path="/fonts/msgothic.ttc" ttccno=0"/>
</font-substitute>
```

C.5.4 <type1> element

Following is the structure of the <type1> element:

```
<type1 name="cdata"/>
```

C.5.4.1 Attribute of <type1> Element

The <type1> element has a single attribute, name, which specifies one of the Adobe standard Latin1 fonts, such as "Courier".

C.5.4.2 Description of <type1> Element

The <type1> element defines an Adobe Type1 font.

```
<!--Define "Helvetica" font as "Serif" -->
<font family="serif" style="normal" weight="normal">
  <type1 name="Helvetica"/>
</font>
```

C.6 Predefined Fonts

The following Type1 fonts are built-in to Adobe Acrobat and BI Publisher provides a mapping for these fonts by default. You can select any of these fonts as a target font with no additional setup required.

The Type1 fonts are listed in Table C–3.

Table C–3 Type 1 Fonts

Number	Font Family	Style	Weight	Font Name
1	serif	normal	normal	Time-Roman
1	serif	normal	bold	Times-Bold
1	serif	italic	normal	Times-Italic
1	serif	italic	bold	Times-BoldItalic

Table C–3 (Cont.) Type 1 Fonts

Number	Font Family	Style	Weight	Font Name
2	sans-serif	normal	normal	Helvetica
2	sans-serif	normal	bold	Helvetica-Bold
2	sans-serif	italic	normal	Helvetica-Oblique
2	sans-serif	italic	bold	Helvetica-BoldOblique
3	monospace	normal	normal	Courier
3	monospace	normal	bold	Courier-Bold
3	monospace	italic	normal	Courier-Oblique
3	monospace	italic	bold	Courier-BoldOblique
4	Courier	normal	normal	Courier
4	Courier	normal	bold	Courier-Bold
4	Courier	italic	normal	Courier-Oblique
4	Courier	italic	bold	Courier-BoldOblique
5	Helvetica	normal	normal	Helvetica
5	Helvetica	normal	bold	Helvetica-Bold
5	Helvetica	italic	normal	Helvetica-Oblique
5	Helvetica	italic	bold	Helvetica-BoldOblique
6	Times	normal	normal	Times
6	Times	normal	bold	Times-Bold
6	Times	italic	normal	Times-Italic
6	Times	italic	bold	Times-BoldItalic
7	Symbol	normal	normal	Symbol
8	ZapfDingbats	normal	normal	ZapfDingbats

The TrueType fonts are listed in Table C–4. All TrueType fonts are subsetted and embedded into PDF.

Table C–4 TrueType Fonts

Number	Font Family Name	Style	Weight	Actual Font	Actual Font Type
1	Albany WT	normal	normal	ALBANYWT.ttf	TrueType (Latin1 only)
2	Albany WT J	normal	normal	ALBANWTJ.ttf	TrueType (Japanese flavor)
3	Albany WT K	normal	normal	ALBANWTK.ttf	TrueType (Korean flavor)

Table C–4 (Cont.) TrueType Fonts

Number	Font Family Name	Style	Weight	Actual Font	Actual Font Type
4	Albany WT SC	normal	normal	ALBANWTS.ttf	TrueType (Simplified Chinese flavor)
5	Albany WT TC	normal	normal	ALBANWTT.ttf	TrueType (Traditional Chinese flavor)
6	Andale Duospace WT	normal	normal	ADUO.ttf	TrueType (Latin1 only, Fixed width)
6	Andale Duospace WT	bold	bold	ADUOB.ttf	TrueType (Latin1 only, Fixed width)
7	Andale Duospace WT J	normal	normal	ADUOJ.ttf	TrueType (Japanese flavor, Fixed width)
7	Andale Duospace WT J	bold	bold	ADUOJB.ttf	TrueType (Japanese flavor, Fixed width)
8	Andale Duospace WT K	normal	normal	ADUOK.ttf	TrueType (Korean flavor, Fixed width)
8	Andale Duospace WT K	bold	bold	ADUOKB.ttf	TrueType (Korean flavor, Fixed width)
9	Andale Duospace WT SC	normal	normal	ADUOSC.ttf	TrueType (Simplified Chinese flavor, Fixed width)
9	Andale Duospace WT SC	bold	bold	ADUOSCB.ttf	TrueType (Simplified Chinese flavor, Fixed width)
10	Andale Duospace WT TC	normal	normal	ADUOTC.ttf	TrueType (Traditional Chinese flavor, Fixed width)
10	Andale Duospace WT TC	bold	bold	ADUOTCB.ttf	TrueType (Traditional Chinese flavor, Fixed width)

C.6.1 Included Barcode Fonts

BI Publisher also includes the barcode fonts that are described in Table C–5.

Table C–5 Barcode Fonts

Font File	Supported Algorithm
128R00.TTF	code128a, code128b, and code128c
B39R00.TTF	code39, code39mod43
UPCR00.TTF	upca, upce

For information on using barcode fonts in an RTF template, see Using the Barcodes Shipped with BI Publisher, *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

D

Audit Reference for Oracle Business Intelligence Publisher

This appendix provides reference information for auditing in Oracle Business Intelligence Publisher.

This appendix contains these sections:

- Section D.1, "About Custom and Standard Audit Reports"
- Section D.2, "Audit Events in Oracle Business Intelligence Publisher"

D.1 About Custom and Standard Audit Reports

The Common Audit Framework in Oracle Fusion Middleware provides a set of standard reports based on your audit records. It also enables you to modify the standard reports and create your own custom audit reports.

This appendix provides details about events that can be audited in Oracle Business Intelligence Publisher. Use this information to understand the structure of each event record to develop custom reports.

The following documents provide more information to help you write custom reports:

- Attributes of Audit Reports in Oracle Business Intelligence Publisher in the *Oracle Fusion Middleware Application Security Guide*
- Customizing Audit Reports in the *Oracle Fusion Middleware Application Security Guide*

The following documents provide additional information about how to configure auditing and view standard reports:

- Configuring auditing for Oracle Business Intelligence Publisher - See Section 12.6, "Enabling Monitoring and Auditing."
- List of events audited for Oracle Business Intelligence Publisher - See Section 12.7, "Viewing the Audit Log."

D.2 Audit Events in Oracle Business Intelligence Publisher

Table D–1 lists the audit events and their attributes:

Table D–1 Oracle Business Intelligence Publisher Audit Events

Event Category	Event	Attributes used by Event
UserSession	UserLogin	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, JobId, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData
	UserLogout	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, JobId, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData
ReportAccess	ReportRequest	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, Format, Template, IsScheduled, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData
	ScheduledReportRequest	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, JobId, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData, UserJobName, UserJobDescription, StartDate, EndDate, Bursting, JobGroup, RunType, OutputInfo, DeliveryInfo
	ReportRepublish	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, IsScheduled, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData, RepublishId
	ReportDataDownload	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, OutputId, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData
	ReportDownload	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, Format, OutputId, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData

Table D–1 (Cont.) Oracle Business Intelligence Publisher Audit Events

Event Category	Event	Attributes used by Event
ReportExecution	ReportDataProcess	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, FreeMemory, TotalMemory, ProcessTime, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData
	ReportRendering	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, Format, Template, Locale, FreeMemory, TotalMemory, DataSize, ProcessTime, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData
	ReportDelivery	TargetComponentType, ApplicationName, EventStatus, Initiator, MessageText, FailureCode, Resource, JobId, FreeMemory, TotalMemory, DataSize, ProcessTime, ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, EventType, EventCategory, TstzOriginating, ComponentName, DomainName, ComponentData, OutputName, DeliveryMethod, DeliveryProperties

E

Converting Oracle Reports

Oracle BI Publisher provides Oracle Reports to BI Publisher Conversion Assistant as a tool to convert reports from the Oracle Reports format to the Oracle BI Publisher format. Oracle Reports to BI Publisher Conversion Assistant can be used for BI Publisher stand alone implementations or when it is integrated with Oracle Business Intelligence Enterprise Edition.

This appendix includes the following sections:

E.1 Overview

In Oracle Reports, the data model (SQL query or extraction logic) and report layout specifications are contained in a single file. In BI Publisher, the data model and the layout are separate objects. Oracle Reports to BI Publisher Conversion Assistant therefore generates several files from a single Oracle Report file that will make up your report in BI Publisher.

Oracle Reports to BI Publisher Conversion Assistant generates a report definition file, a data model file, and a layout template file. These are the BI Publisher objects that are uploaded to the BI Publisher repository. When the Oracle Report includes calls to package functions, Oracle Reports to BI Publisher Conversion Assistant also creates the PL/SQL specification and body.

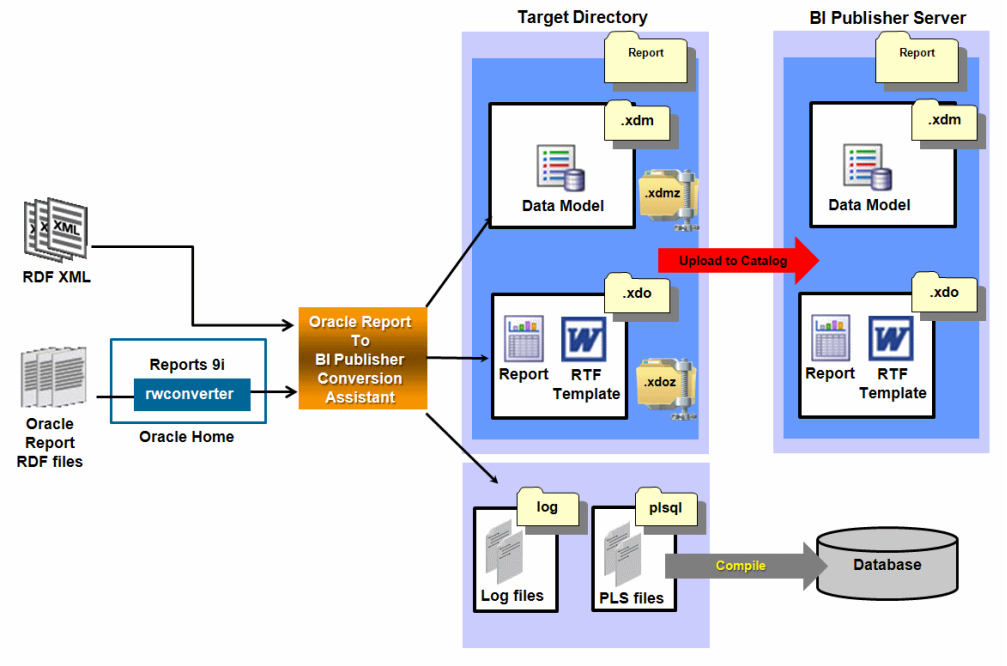
After Oracle Reports to BI Publisher Conversion Assistant completes the report conversion, compile the .pls files on the database. After compilation, test the report to ensure that the output is as expected. If the report fails to display the expected results, review the log files (conversion log, upload log, BI Publisher Server log).

In some cases, the data model and layout template may require manual adjustments to improve report output.

Some reports contain structures that Oracle Reports to BI Publisher Conversion Assistant cannot convert. These structures must be manually implemented in the converted reports.

Figure E–1 describes the conversion process.

Figure E–1 Oracle Report to BI Publisher Report Conversion Process



The overall flow for the conversion process is as follows:

1. In BI Publisher, create a JDBC or JNDI data source connection that connects to the same database as the Oracle Report.
2. Run Oracle Reports to BI Publisher Conversion Assistant.
3. (Conditional) Upload the report to BI Publisher Catalog if you did not choose Oracle Reports to BI Publisher Conversion Assistant's automatic upload option.
4. (Conditional) Compile the PL/SQL package on the database.
5. Test the report and check the conversion log files to identify any manual modifications needed to complete the conversion.

E.2 Obtaining Oracle Reports to BI Publisher Conversion Assistant

To obtain Oracle Reports to BI Publisher Conversion Assistant:

1. Go to the following website: http://www.oracle.com/technetwork/middleware/bi-publisher/downloads/index.html
2. Download OR2BIPConvAssist.zip for your Windows or Linux environment and unzip the installation zip files to any directory on your machine. The following folders are created in the directory: bin, config, and lib.
3. Under the bin folder, use OR2BIPConvAssist.bat or OR2BIPConvAssist.sh to launch Oracle Reports to BI Publisher Conversion Assistant.

E.3 Prerequisites and Limitations of Oracle Reports to BI Publisher Conversion Assistant

The following are prerequisites for running Oracle Reports to BI Publisher Conversion Assistant:

- You must be running JDK version 1.1.8 or later.
- If your source Oracle Reports are not in XML format, then you must have Oracle Reports rwconverter 9i in your Oracle Home.
- You must have a BI Publisher role with Write access to Shared Folders in BI Publisher Catalog.

E.3.1 Converting Oracle Reports to XML Format

Oracle Reports to BI Publisher Conversion Assistant uses Oracle Report XML (or RDF XML) to convert Oracle Reports into BI Publisher reports. The conversion from RDF binary to RDF XML report formatting is supported for Oracle Reports 9*i* and above.

Oracle Reports to BI Publisher Conversion Assistant allows source reports to be in both formats - RDF binary and RDF XML.

The following requirements are necessary in order for Oracle Reports to BI Publisher Conversion Assistant to convert reports:

- Oracle Reports Designer 9*i* or later installed on the same machine as Oracle Reports to BI Publisher Conversion Assistant.
- You must enter the Oracle Reports Home Path in Oracle Reports to BI Publisher Conversion Assistant in order to call the rwconverter executable and convert the reports into Oracle Reports XML format.

Oracle Reports Designer is part of the Oracle Developer Suite 10*g* (10.1.2.0.2) and is available from:
http://www.oracle.com/technology/software/products/ids/index.html

E.3.2 Limitations

Oracle Reports to BI Publisher Conversion Assistant has the following limitations:

- Format triggers cannot be converted, but the code is written to a log file for manual implementation. For more information on PL/SQL format triggers in RTF layout templates, see Section E.7.2, "PL/SQL Format Trigger Logic Not Supported in RTF Layout Templates."
- Charts cannot be converted. If you choose to create a RTF layout, use the Template Builder for Word to create charts. If you choose to create a BI Publisher layout template, use the Layout Editor to create charts. For more information about the layout design tools, see *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.
- Matrix reports cannot be converted. If you choose to create a RTF layout, use the Template Builder for Word's Pivot Table Wizard. If you choose to create a BI Publisher layout template, use the Layout Editor to create pivot tables. For more information about the layout design tools, see *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

E.4 Running Oracle Reports to BI Publisher Conversion Assistant

This section describes how to run the conversion assistant and includes:

- Running the Assistant When the Source Report Is an Oracle Reports RDF File
- Running the Assistant When the Source Report Is in Oracle Reports XML Format
- Output Files

E.4.1 Running the Assistant When the Source Report Is an Oracle Reports RDF File

If the source report is an Oracle Reports RDF File, perform the following steps:

1. Create the JDBC/JNDI Connection data source on the BI Publisher Administration page. This data source connects to the same database as the Oracle Report. You must have the Administrator role to perform this step.
2. Launch Oracle Reports to BI Publisher Conversion Assistant.
3. Enter the following fields:
   - **Source Path** - source directory for Oracle Reports files.
   - **Oracle Reports Home Path** - Oracle Reports directory. For example, C:\Oracle_1\Middleware\OracleFRHome1.

     Oracle Reports to BI Publisher Conversion Assistant requires rwconverter from Oracle Reports in order to convert the report from RDF format to XML format. Specify the Oracle Reports home path where Oracle Reports Designer (9*i* or later version) is installed. Oracle Reports to BI Publisher Conversion Assistant assumes that rwconverter is contained in the bin directory beneath the Oracle Reports home path specified.
   - **Target Path** - location to place the output files. For each converted report, Oracle Reports to BI Publisher Conversion Assistant creates the following separate output objects in the **Target Path**: Oracle BI Publisher Report folder (.xdo) containing the report definition (.xdo) and layout template (.rtf) files, the data model file (.xdm), zipped report definition file (.xdoz) and zipped data model file (.xdmz), the plsql_pkgs folder containing the PL/SQL packages, and bipconvert.log and bipupload.log files.

Note: You can use the zipped report definition file (.xdoz) and the zipped data model file (.xdmz) to manually upload reports to BI Publisher Catalog.

   - **Data Source Name** - specifies the data source that you created in step 1.
   - **Debug** - specifies if Oracle Reports to BI Publisher Conversion Assistant will run in debug mode, and write debug statements to the log files.

     Select **No** to write conversion process details and any statement level error messages to a log file at the report level in the target directory.

     Select **Yes** to write the details of the conversion process, including the complete error stack in case there are exceptions, to a log file at the component level in the target directory.
   - **Upload to Catalog** - specifies if converted reports are automatically loaded to BI Publisher Catalog.

Select **No** to create converted reports only in the **Target Path**. This is the default option.

Select **Yes** to automatically upload the reports to the BI Publisher Server. If you select this option, you must provide the BI Publisher Server URL (http://<server>:<port>/xmlpserver,) username and password, and the BI Publisher Catalog folder (under the Shared folder) where the converted reports will be stored. If you specify a folder that does not yet exist in BI Publisher Catalog, it will automatically be created during the conversion process.

Note: If you choose to automatically upload the converted reports to the BI Publisher Server, the bipupload.log file is populated with the details in the **Target Path**. If you do not automatically upload the converted reports, this log file is empty.

---

4. Click **Run**. Oracle Reports to BI Publisher Conversion Assistant converts the Oracle Reports and creates a folder for each of the Oracle Reports in the **Target Path**. If you selected **Upload to Catalog**, a folder is created for each report in the specified folder under the Shared folder. Each report folder contains the data model and report. The PL/SQL packages are created only in the **Target Path**.
5. Compile the PL/SQL packages (.pls files) to the database. The PL/SQL packages are created in the **Target Path** in Oracle Reports to BI Publisher Conversion Assistant. For more information, see Section E.5, "Compiling the PL/SQL Package to the Database."

Note: Oracle Reports to BI Publisher Conversion Assistant populates the **Oracle DB Default Package** field in the data model with the package name defined in the report.

---

6. (Conditional) If you did not select **Upload to Catalog**, use the **Upload Resource** feature in BI Publisher Catalog to upload each zipped data model and report file (.xdmz and .xdoz). Ensure that you use the BI Publisher UI (/xmlpserver) to upload both objects to the same folder in BI Publisher Catalog. For more information, see "*Downloading and Uploading Catalog Objects*" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*
7. View the converted reports in the Report Viewer.
8. If the reports were not converted as expected, review the log files. For more information, see Section E.8, "Troubleshooting Oracle Reports to BI Publisher Conversion Assistant."

E.4.2 Running the Assistant When the Source Report Is in Oracle Reports XML Format

If the source report is in Oracle Reports XML format, follow these steps:

1. Create the JDBC/JNDI Connection data source on the BI Publisher Administration page. This data source connects to the same database as the Oracle Report. You must have the Administrator role to perform this step.
2. Launch Oracle Reports to BI Publisher Conversion Assistant.
3. Enter the following fields:
   - **Source Path** - source directory for the Oracle Reports files in XML format.

- **Oracle Reports Home Path** - this is not applicable for Oracle Reports XML format files.
- **Target Path** - location to place the output files. For each converted report, Oracle Reports to BI Publisher Conversion Assistant creates the following separate output objects in the **Target Path**: Oracle BI Publisher Report folder (.xdo) containing the report definition (.xdo) and layout template (.rtf) files, the data model file (.xdm), zipped report definition file (.xdoz) and zipped data model file (.xdmz), the plsql_pkgs folder containing the PL/SQL packages, and bipconvert.log and bipupload.log files.

Note: You can use the zipped report definition file (.xdoz) and the zipped data model file (.xdmz) to manually upload reports to BI Publisher Catalog.

- **Data Source Name** - the data source that you created in step 1.
- **Debug** - specifies if Oracle Reports to BI Publisher Conversion Assistant will run in debug mode, and write debug statements to the log files.

  Select **No** to write conversion process details and any statement level error messages to a log file at the report level in the target directory.

  Select **Yes** to write the details of the conversion process, including the complete error stack in case there are exceptions, to a log file at the component level in the target directory.
- **Upload to Catalog** - specifies if converted reports are automatically loaded to BI Publisher Catalog.

  Select **No** to create converted reports only in the **Target Path**. This is the default option.

  Select **Yes** to automatically upload the reports to the BI Publisher Server. If you select this option, you must provide the BI Publisher Server URL (http://<server>:<port>/xmlpserver,) username and password, and the BI Publisher Catalog folder (under the Shared folder) where the converted reports will be stored. If you specify a folder that does not yet exist in BI Publisher Catalog, it will automatically be created during the conversion process.

Note: If you choose to automatically upload the converted reports to the BI Publisher Server, the bipupload.log file is populated with the details in the **Target Path**. If you do not automatically upload the converted reports, this log file is empty.

---

4. Click **Run**. Oracle Reports to BI Publisher Conversion Assistant converts the Oracle Reports and creates a folder for each of the Oracle Reports in the **Target Path**. If you selected **Upload to Catalog**, a folder is created for each report in the specified folder under the Shared folder. Each report folder contains the data model and report. The PL/SQL packages are created only in the **Target Path**.
5. Compile the PL/SQL packages (.pls files) to the database. The PL/SQL packages are created in the **Target Path** in Oracle Reports to BI Publisher Conversion Assistant. For more information, see Section E.5, "Compiling the PL/SQL Package to the Database."

Note: Oracle Reports to BI Publisher Conversion Assistant populates the **Oracle DB Default Package** field in the data model with the package name defined in the report.

6. (Conditional) If you did not select **Upload to Catalog**, use the **Upload Resource** feature in BI Publisher Catalog to upload each zipped data model and report file (.xdmz and .xdoz). Ensure that you use the BI Publisher UI (/xmlpserver) to upload both objects to the same folder in BI Publisher Catalog. For more information, see "*Downloading and Uploading Catalog Objects*" in *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.
7. View the converted reports in the Report Viewer.
8. If the reports were not converted as expected, review the log files. For more information, see Section E.8, "Troubleshooting Oracle Reports to BI Publisher Conversion Assistant."

E.4.3 Output Files

Oracle Reports to BI Publisher Conversion Assistant generates the following output files in your target directory for each converted report:

- Report definition file (format: REPORT.xdo).
- Zipped report definition file (for example: REPORT.xdoz).
- Data model file that defines the data model (format: REPORT.xdm).
- Zipped data model file (for example: REPORT.xdmz).
- RTF Layout Template (for example: REPORT.rtf).
- Default PL/SQL package specification (format: REPORTS.pls).
- Default PL/SQL package body (format: REPORTB.pls).
- Log files (format: REPORT.log).

For example, assume that you want to convert a report called invoice.rdf located in the source directory D:\reports\pay and you use Oracle Reports to BI Publisher Conversion Assistant to enter the target directory D:\BIPublisher_reports\invoice.

The following output files are generated:

- Report definition: C:\BIPublisher_reports\invoice\invoice.xdo_report.xdo
- Zipped report definition: C:\BIPublisher_reports\invoice\invoice.xdo_report.xdoz
- RTF layout template: C:\BIPublisher_reports\invoice\invoice.xdo\invoice.rtf
- Data model: C:\BIPublisher_reports\invoice\invoice_template.xdm_datamodel.xdm
- Zipped data model C:\BIPublisher_reports\invoice\invoice_template.xdm_datamodel.xdmz
- PL/SQL package specification: C:\BIPublisher_reports\plsql_packages\invoice\invoiceS.pls
- PL/SQL package body: C:\BIPublisher_reports\plsql_packages\invoice\invoiceB.pls
- Report conversion log file C:\BIPublisher_reports\bipconvert.log

- Report upload log file C:\BIPublisher_reports\bipupload.log

E.5 Compiling the PL/SQL Package to the Database

Many converted Oracle Reports will generate a PL/SQL package specification file and a PL/SQL package body file as follows:

- <report_name>S.pls
- <report_name>B.pls

Run the PL/SQL package files against your Oracle Database using the following commands to create the PL/SQL package specification and body.

```
SQL> @C:\BIPublisher_reports\invoice\invoiceS.pls
SQL> @C:\BIPublisher_reports\invoice\invoiceB.pls
```

E.6 Moving Converted Reports to the Oracle BI Publisher Repository

This step is only required if you did not specify for Oracle Reports to BI Publisher Conversion Assistant to automatically upload the converted reports to Oracle BI Publisher Catalog. You can manually upload the converted report components using one of the following methods:

- Method 1: Move the .xdo and .xdm folder files to the BI Publisher Repository Path in the File System
- Method 2: Upload the .xdoz and .xdmz zip files using the BI Publisher UI

E.6.1 Method 1: Move the .xdo and .xdm folder files to the BI Publisher Repository Path in the File System

To use this method, you must have access to the server file system. Oracle recommends that you use this method only when you have a large number of converted reports.

After you move the .xdo and .xdm folder files to the BI Publisher repository path in the file system, the next step depends on your implementation:

If BI Publisher is deployed standalone (not part of Oracle BI Enterprise Edition), move the .xdo and .xdm folder files to the BI Publisher repository path in the file system. Ensure that you place each report definition (.xdo) and data model (.xdm) pair in the same folder. You do not need to restart the BI Publisher Server.

If BI Publisher is integrated with BI Presentation Services Catalog, perform the following steps:

1. Move the .xdo and .xdm folder files to the BI Publisher repository path in the file system. Ensure that you place each report definition (.xdo) and data model (.xdm) pair in the same folder.
2. Navigate to the BI Publisher Server Administration page and select **Server Configuration**.
3. On the Server Configuration page, in the Catalog region, click **Upload to BI Presentation Catalog**. You do not need to restart the BI Publisher Server.

E.6.2 Method 2: Upload the .xdoz and .xdmz zip files using the BI Publisher UI

To use this method, you must have Write permissions on the folder in the catalog to which you are uploading and you must have one of the following roles: BI Publisher Administrator, BI Publisher Template Designer, BI Publisher Developer, or BI Author.

1. Log in to the BI Publisher server (for example, http://www.<host>:<port>/xmlpserver)
2. Select the folder in the catalog to which you want to upload the report.
3. In the Tasks region, click **Upload**. Click **Browse** to locate the report component (.xdoz or .xdmz) to upload. Repeat this step for each object, ensuring that you upload each report definition (.xdoz) and data model (.xdmz) pair to the same folder.

E.7 Testing and Editing Converted Reports

After you have successfully converted your reports, created the required PL/SQL packages, and moved the report definition and data model files into Oracle BI Publisher Catalog, test your reports. Most converted reports will run as expected without further modification of the data model; however, it is likely that the layout will require additional modifications to achieve the desired appearance. The following sections are some common issues that you may encounter with converted reports.

E.7.1 Summary Columns Moved to the Select Clause

Occasionally, when you convert a more complex Oracle Reports report, the data model or PL/SQL may contain minor errors and require manual corrections. Oracle Reports to BI Publisher Conversion Assistant will move all formula columns to the select clause of the SQL query in the data model. In most cases, this will not cause a problem. However, if any argument to the formula is a summary column, this will cause errors because the summary column will not be calculated at the same time as the query is executed.

To correct this problem, remove this formula from the select clause and implement the formula in the data model. For more information on formula columns, see *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.

E.7.2 PL/SQL Format Trigger Logic Not Supported in RTF Layout Templates

Many Oracle Reports use simple "if" formatting logic. Oracle Reports to BI Publisher Conversion Assistant automatically converts this logic into equivalent XSL-FO and inserts the code into form fields in the RTF layout template. However, there is no support for PL/SQL in RTF layout templates. Oracle Reports to BI Publisher Conversion Assistant does not convert any PL/SQL format trigger logic present in the report. Instead, Oracle Reports to BI Publisher Conversion Assistant writes all the format trigger code to a log file. You will need to implement any corresponding PL/SQL logic as XSL code.

To aid in this process, the resulting RTF template will contain form fields that hold the format trigger names that are called; these fields will be highlighted in red. You can then refer to the log to find the actual PL/SQL code used in the original Oracle Report.

You must rewrite the PL/SQL triggers using BI Publisher code syntax or XSL in the RTF template form fields.

E.8 Troubleshooting Oracle Reports to BI Publisher Conversion Assistant

Oracle Reports to BI Publisher Conversion Assistant produces two log files (bipconvert.log and bipupload.log) to assist you with troubleshooting the report conversion process. These log files capture exceptions at the statement level or in debug mode depending on which options you selected during report conversion.

After the you upload the reports to BI Publisher Catalog, you can view the reports in the Report Viewer. If the report is not displayed as expected in the Report Viewer, review the errors captured on the screen or review the following BI Publisher log file: (MWHOME\user_projects\domains\bifoundation_domain\servers\AdminServer\logs\bipublisher\bipublisher.log).

Also, if you encounter problems with a converted report, test the data model as follows:

- If the data model is successful, verify the data in the XML output. If the data contains special characters, the report may fail while parsing the data.
- If the data model fails, check the data source connection. Next, verify if the .pls files were compiled successfully on the database to create the package. Finally, verify that the result data is accurate.

F

Enabling Memory Guard Features

This section describes property settings available to safeguard your system against memory failures caused by report requests that generate excessive data.

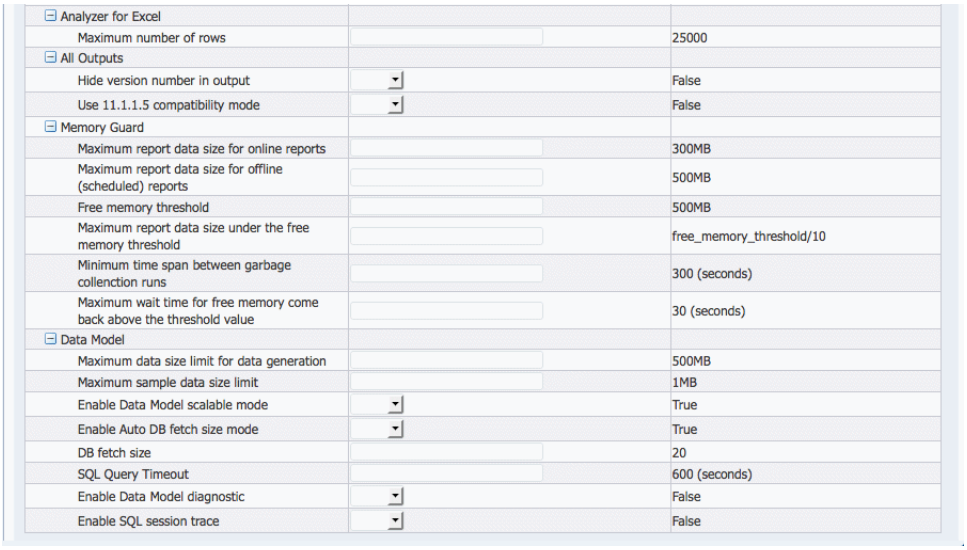
It includes the following sections:

- Section F.1, "What Are Memory Guard Features?"
- Section F.2, "Key Features"
- Section F.3, "Configuring Memory Guard & Data Model Properties"
- Section F.4, "Configuring a Maximum Threads Constraint to Avoid Out of Memory Errors"

F.1 What Are Memory Guard Features?

BI Publisher provides a set of features to protect against out-of-memory errors by blocking report requests that generate excessive amounts of data. These "memory guard" features consist of a set of properties. The properties enable you to configure conditions and processing points at which data size is inspected to determine whether the system continues to process a report request or terminates processing. When processing terminates due to data size, an error message is returned.

Figure F–1 Memory Guard and Data Model Property Settings



F.2 Key Features

The full set of properties are listed in Table F–1 and Table F–2. The properties enable you to protect against out of memory errors and enhance data processing by setting controls such as:

Maximum data size for reports

Maximum data size for scheduled reports

Minimum free memory size

SQL pruning for unused data set columns

Time out for SQL queries

The following sections highlight some of the properties and provide detail on how the system responds to the settings:

- Restricting Maximum Data Sizes for Report Processing
- Configuring Free Memory Threshold
- Setting Data Engine Properties

F.2.1 Restricting Maximum Data Sizes for Report Processing

By restricting the data size allowed for report processing you can prevent out of memory errors when a query returns more data than the system can handle.

- Specify a Maximum Data Size Allowed for Online Processing
- Specify a Maximum Data Size Allowed for Offline (Scheduled Report) Processing

Specify a Maximum Data Size Allowed for Online Processing

Property: **Maximum report data size for online reports**.

This property enables you to specify a maximum data size allowed for online report viewing. When you set a maximum data size, the following occurs when a user opens a report for online viewing:

1. A user submits a report to view online in the browser.
2. The data engine generates the data for the report.
3. Before document generation, the size of the data (in bytes) is inspected.
4. If the data generated is larger than the maximum setting, the report processing is ended. The user gets the following message:

   The report you are trying to run exceeds the data limit set for this server. Either re-run with parameters that reduce the data or schedule this report. Contact your Administrator if you have questions.

   The user can then either set parameters (if available for the report) to limit the data and resubmit online; or use the BI Publisher scheduler to submit the report.

The default value for this property is 300 MB.

Specify a Maximum Data Size Allowed for Offline (Scheduled Report) Processing

Property: **Maximum report data size for offline (scheduled) reports**.

This feature enables you to specify a maximum data size allowed for scheduled reports. When you set a maximum data size, the following occurs when a scheduled report job executes:

1. The scheduler commences processing of a report job.
2. The data engine generates the data for the report.
3. If the data generated is larger than the maximum setting, the report processing is ended. The scheduled report job fails with the following status message:

   Report data size exceeds the maximum limit (<nnn> bytes). Stopped processing.

   The user can then set parameters (if available for the report) to limit the data.

The default value for this property is 500 MB.

F.2.2 Configuring Free Memory Threshold

This set of properties helps you to protect against out of memory conditions by establishing a minimum available free memory space. This set of properties enables your system to automatically protect free memory availability and intelligently process reports with large data sets based on this availability.

- Specify A Minimum Free Memory Threshold for Report Processing
- Specify Maximum Report Data Size Under the Free Memory Threshold
- Set Minimum Time Span Between Garbage Collection Runs
- Set Maximum Wait Time for Free Memory to Come Back Above the Threshold

Specify A Minimum Free Memory Threshold for Report Processing

Property: **Free memory threshold**

This setting enables you to specify a minimum value for free JVM space. This enables you to control whether to run a report based on two factors: current usage and the size of the report data. This feature requires the setting of several properties that work together. You specify the threshold JVM space, the report maximum report size that will be allowed when the JVM falls below the threshold, and the maximum wait time to pause the report to wait for more JVM free space to become available.

When you set these properties, the following occurs when a user opens a report for online viewing:

1. A user submits a report to view online in the browser.
2. The data engine generates the data for the report.
3. JVM memory is inspected. If the available JVM memory is above the **Free memory threshold** property value, the report processes as usual and there is no system intervention.

   If the available JVM memory is below the threshold value, the size of the report data is inspected and compared to the property setting for **Maximum report data size under the free memory threshold**. If the report data is below this threshold, then the report continues processing.

   If the report data size exceeds the threshold, then the report is paused to wait for free memory to become available. The report will wait for the time specified in the property Maximum Wait Time for Free Memory to Come Back Above Threshold Value. If the free memory does not rise back above the minimum in the wait period specified, the report request is rejected.

The default value for this property is 500 MB.

Specify Maximum Report Data Size Under the Free Memory Threshold

Property: **Maximum report data size under the free memory threshold**

Default value: (value of Free Memory Threshold)/10

Maximum single report data size allowed when free JVM memory is under the specified threshold value set in **Free memory threshold**. For example (assuming the default setting), if the data generated for a single report exceeds one-tenth of the value set for **Free memory threshold**, then processing is terminated. Therefore if the Free memory threshold is set to 100 MB and a single report data extract exceeds 10 MB, then the report processing is terminated.

This property takes effect only when **Free memory threshold** is set to be a positive value.

Set Minimum Time Span Between Garbage Collection Runs

Minimum time span in seconds between any two subsequent garbage collection runs. Set this value to avoid overrunning JVM garbage collection. The server enforces the minimum of 120 seconds, which means the value will be reset to 120 seconds if it falls below the minimum.

The default is 300 seconds.

Set Maximum Wait Time for Free Memory to Come Back Above the Threshold

The maximum time in seconds that a run-report request will wait for free JVM memory to come back above the threshold value. This property value takes effect only when a positive value for Free memory threshold is specified.

If the free memory becomes available within the time specified, the request will proceed immediately to generate the document. If free memory is still below the threshold value after the time specified, the request is rejected. For online requests, the larger this property value, the longer the browser will wait for a request to run.

The default for this property is 30 seconds.

F.2.3 Setting Data Engine Properties

The data engine property settings provide additional points to protect your system against out of memory errors. These include:

- Set Maximum Data Size That Can Be Generated by the Data Engine
- Set Maximum Sample Data Size
- Set Automatic Database Fetch Size

Set Maximum Data Size That Can Be Generated by the Data Engine

This setting sets an absolute limit to the data that can be generated from the execution of a data model. This setting applies to both online report requests and to requests submitted through the scheduler. When the size of the file generated by the data engine exceeds the limit, the data engine terminates execution of the data model and throws the exception:

"oracle.xdo.dataengine.diagnostic.XMLSizeLimitException: XML Output (NNNNNNbytes) generated exceeds specified file size limit (NNNNNNbytes)..!!!!!!!".

If the report request was submitted through the scheduler, the job will show as failed in the Report Job History page. The exception error noted above displays when you rest your cursor over the status. If the report request was submitted online, the user will get the error "Unable to retrieve the data XML."

Set Maximum Sample Data Size

A sample data set is required for all data models. The sample data is used during template design. Sample data can be generated by the data model editor or uploaded to the data model. Large sample data sets can impact the performance of the design tools.

Set this property to limit the size of the sample data file that can be uploaded to the data model.

Set Automatic Database Fetch Size

This setting calculates and sets database fetch size at run time depending on total number of data set columns and total number of query columns. Setting this property will override the server-level and data model-level database fetch size properties. When set, this property takes effect for all data models and can significantly slow processing time. This setting is recommended for implementations of BI Publisher that frequently process complex queries of hundreds of columns, such as Oracle Fusion Applications implementations. This setting is not recommended for most general implementations of BI Publisher.

F.3 Configuring Memory Guard & Data Model Properties

Implement the memory guard features by setting the properties in the Administration Runtime properties page.

The Memory Guard property settings are described in Table 1.

Table F–1 Memory Guard Properties and Descriptions

Property	Description
Maximum report data size for online reports	Default value: 300MB Sets the maximum allowed online report data size. You can set the value in GB, MB, or KB. For example: 1GB, 200MB, or 1500KB. An online report request will be rejected immediately when the report data size returned from the execution of the data model exceeds the value of this property. To turn off this property, enter 0 or a negative number.
Maximum report data size for offline (scheduled) reports	Default value: 500 MB Set the maximum allowed offline (or scheduled) report data size. You can set the value in GB, MB, or KB. For example: 1GB, 200MB, or 1500KB. A scheduled report request will be rejected immediately when the report data size returned from the execution of the data model exceeds the value of this property. To turn off this property, enter 0 or a negative number.
Free memory threshold	Default value: 500MB Threshold value of free JVM memory to trigger possible rejection of report requests. You can set the value in GB, MB, or KB. For example: 1GB, 200MB, or 1500KB. When JVM free memory returned from run time is below the value of this property, the server will check the report data size to decide if a request should be accepted or rejected. This property works together with the three properties: ■ **Maximum report data size under the free memory threshold** ■ **Minimum Time Span Between Garbage Collection Runs** ■ **Maximum Wait Time for Free Memory to Come Back Above Threshold Value** If this property value is to 0 or a negative number, this condition will be ignored. This property is for online report requests only.

Table F–1 (Cont.) Memory Guard Properties and Descriptions

Property	Description
Maximum report data size under the free memory threshold	Default value: (value of Free memory threshold/10) Maximum report data size allowed when free JVM memory is under the specified threshold value set in **Free memory threshold**. A request will be rejected when its report data size exceeds the value of this property. This property takes effect only if **Free memory threshold** is set to be a positive value. This property is for online report requests only. You can set the value in GB, MB, or KB. For example: 1GB, 10MB, or 1500KB. If you do not explicitly set the value, the default value is calculated by dividing the value you set for **Free memory threshold** by 10. So if you set **Free memory threshold** to 100MB, the default value for this property is 10MB.
Minimum time span between garbage collection runs	Default value: 300 (seconds) Minimum time span in seconds between any two subsequent garbage collection runs. Set this value to avoid overrunning JVM garbage collection. Note that the server automatically enforces a minimum value of 120 seconds, so if you enter a value less than 120 seconds, the server overrides it.
Maximum wait time for free memory to come back above the threshold value	Default value: 30 (seconds) The maximum time in seconds that a run-report request will wait for free JVM memory to come back above the threshold value. This property value takes effect only when a positive value of **Free memory threshold** is specified. If the free memory comes back in the time less than the value of **Maximum wait time for free memory to come back above the threshold value**, the request will proceed immediately to generate the document. If free memory is still below the threshold value after the time set for this property, the request will be rejected. The larger this property value, the longer the browser will wait for a request to run.

The Data Model property settings are described in Table F–2.

Table F–2 Data Model Properties and Descriptions

Property	Description
Maximum data size limit for data generation	Default value: 500MB Maximum XML data size in that can be generated from the execution of a data model. This setting applies to both online report requests and to requests submitted through the scheduler. When the size of the file generated by the data engine exceeds the value set for this property, the data engine terminates execution of the data model and throws an exception. You can set the value in GB, MB, or KB. For example: 1GB, 200MB, or 1500KB. To turn this property off, enter 0 or a negative number.
Maximum sample data size limit	Default value: 1MB Maximum file size of a sample data file that can be uploaded to the data model editor.
Enable Data Model scalable mode	Default: True Processing large data sets requires the use of large amounts of RAM. To prevent running out of memory, activate scalable mode for the data engine. In scalable mode, the data engine takes advantage of disk space when it processes the data. You can also set this property for specific data models. The data model setting overrides the system setting here. See "Setting Data Model Properties" in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher*.
Enable Auto DB fetch size mode	Default value: True When set to True, BI Publisher calculates and sets database fetch size at run time according to the total number of data set columns and total number of query columns. This setting avoids out of memory conditions, but can significantly slow processing times. **IMPORTANT**: When set to True, any other **DB fetch size** settings are ignored. This setting is recommended for implementations of BI Publisher that frequently process complex queries of hundreds of columns, such as Oracle Fusion Applications implementations. This setting is not recommended for most general implementations of BI Publisher. Setting this property will override the data model- level database fetch size properties. When set, this property takes effect for all data models and can significantly slow processing time.
DB fetch size	Default value: 20 (rows) The maximum database fetch size for a data model. This property value takes effect only when **Enable Auto DB fetch size mode** is set to False. When the fetch size is met, the rows are written to a temp file and another fetch is executed; this process is repeated until all the rows are returned to the temp file. A smaller fetch size requires more round trips from BI Publisher to the database and can impact overall processing time; however, the smaller data chunks ensure against excessive memory usage. This property can also be set at the data model level. The data model setting overrides the server property.

Table F–2 (Cont.) Data Model Properties and Descriptions

Property	Description
SQL Query Timeout	Default: 600 seconds Time out for SQL query-based data models. If the SQL query is still processing when the time out value is met, the error "Failed to retrieve data xml." is returned. This property can also be set at the data model level. The data model setting overrides the server property here.
Enable Data Model diagnostic	Default value: False Setting this property to true will write data set details, memory, and SQL execution time information to the log file. Oracle recommends setting this property to true only for debugging purposes. When set to true, processing time is increased.
Enable SQL Session Trace	Default value: False Setting this property to True writes a SQL session trace log to the database for every SQL query that is executed. The log can then be examined by a database administrator. Only set this to True for debugging purposes.

F.4 Configuring a Maximum Threads Constraint to Avoid Out of Memory Errors

During the processing of large BI Publisher reports Oracle WebLogic Server can use multiple concurrent threads to generate the report. If the threads are not constrained, out of memory errors can occur when Oracle WebLogic Server allots too many threads to report generation. To avoid this error, you can create a Work Manager to enforce the maximum number of threads that Oracle WebLogic Server can allot to BI Publisher report processing.

To configure a maximum threads constraint perform the following procedures:

1. Create the Maximum Threads Constraint in Oracle WebLogic Server
2. Create the Work Manager (XdoWorkManager)
3. Redeploy the xmlpserver.ear File

> **Note:** This procedure describes redeploying the xmlpserver.ear file to activate the new Work Manager. Alternatively, you can perform one of the following instead of step 3:
>
> - Restart (stop & start) the bipublisher application
> - Restart the Oracle WebLogic Server instances (for example, bi_server1, bi_server2)

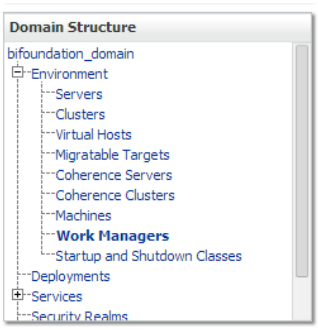
Once this initial setup procedure is completed, changing the value of the maximum threads count (for example from 10 to 20) takes effect immediately; no restart or redeployment operations are required.

F.4.1 Create the Maximum Threads Constraint in Oracle WebLogic Server

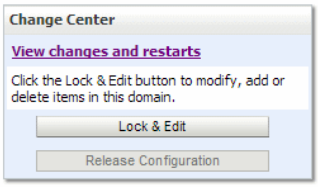
To create the maximum threads constraint component:

1. Log in to Oracle WebLogic Console.

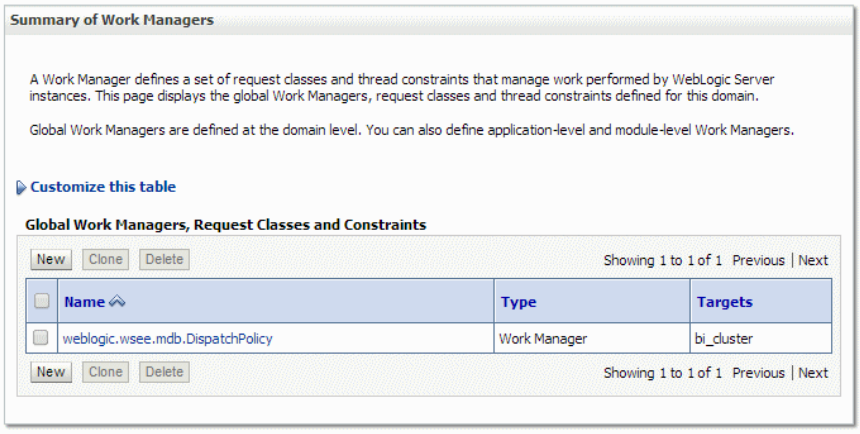
2. Under the **Domain Structure** pane, click **Work Managers**.



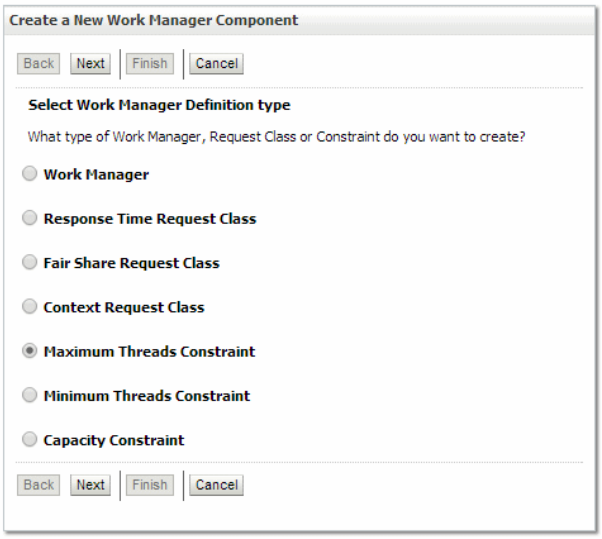
3. In the **Change Center** pane, click **Lock & Edit**.



4. In the **Global Workers, Request Classes and Constraints** table, click **New**.



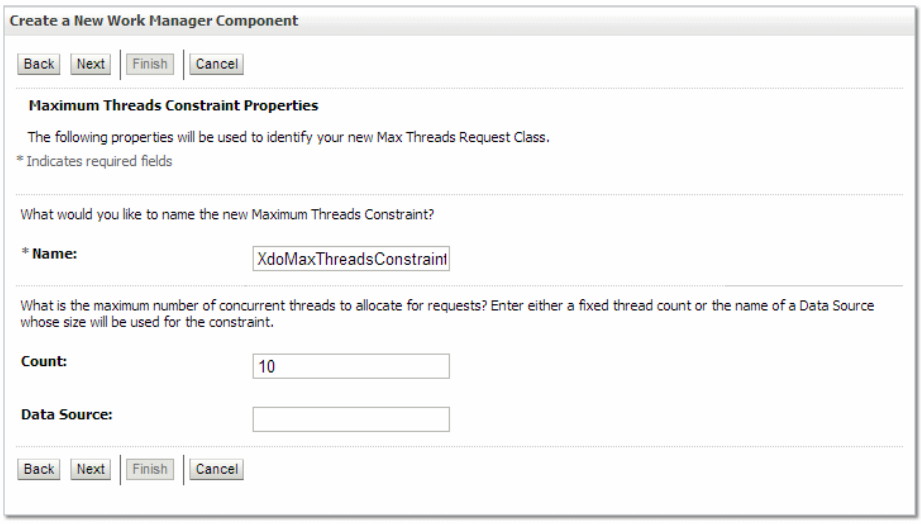
5. In the **Create a New Work Manager Component** dialog, select **Maximum Threads Constraint** and click **Next**.



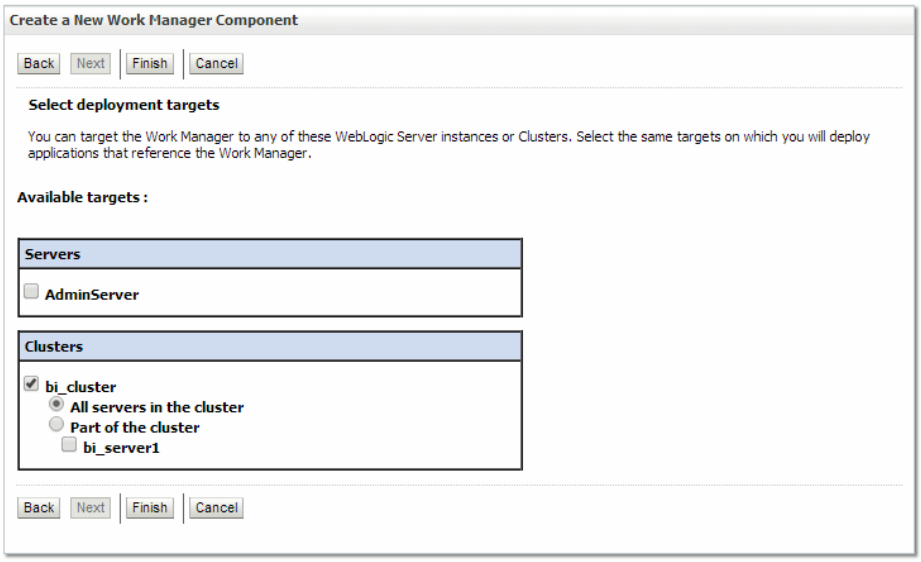
6. Under **Maximum Threads Constraint Properties,** enter the following property values:

   - **Name** - enter XdoMaxThreadsConstraint
   - **Count** - enter the maximum number of threads to allot for BI Publisher report generation, for example, 10

   Click **Next**.



7. Under **Select deployment targets**, select "bi_cluster" and then click **Finish**.

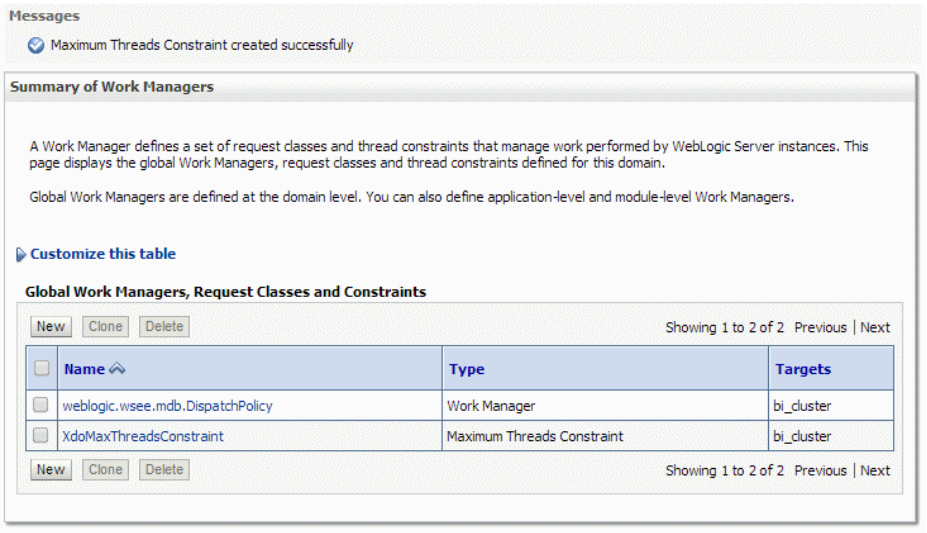


F.4.2 Create the Work Manager (XdoWorkManager)

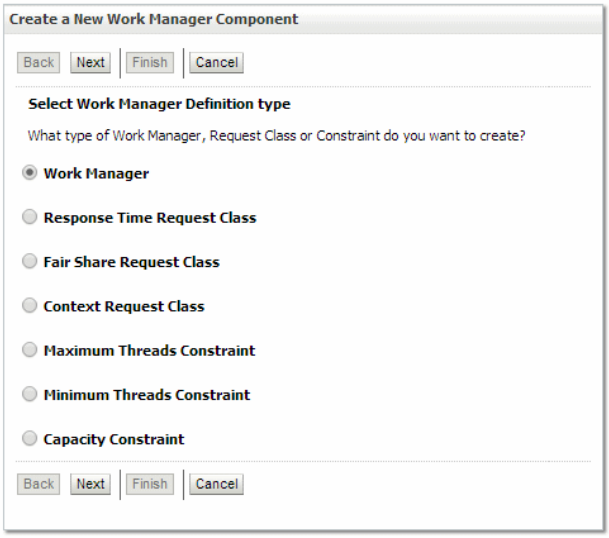
Now that you have created the Maximum Threads Constraint component and named it "XdoMaxThreadsConstraint"; next create the work manager and associate it to the XdoMaxThreadsConstraint component.

To create the work manager:

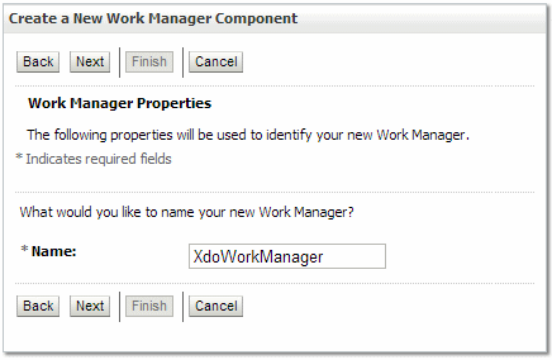
1. While still on the **Summary of Work Managers** page, click **New** again.



2. In the **Create a New Work Manager Component** dialog, select **Work Manager** and click **Next**.

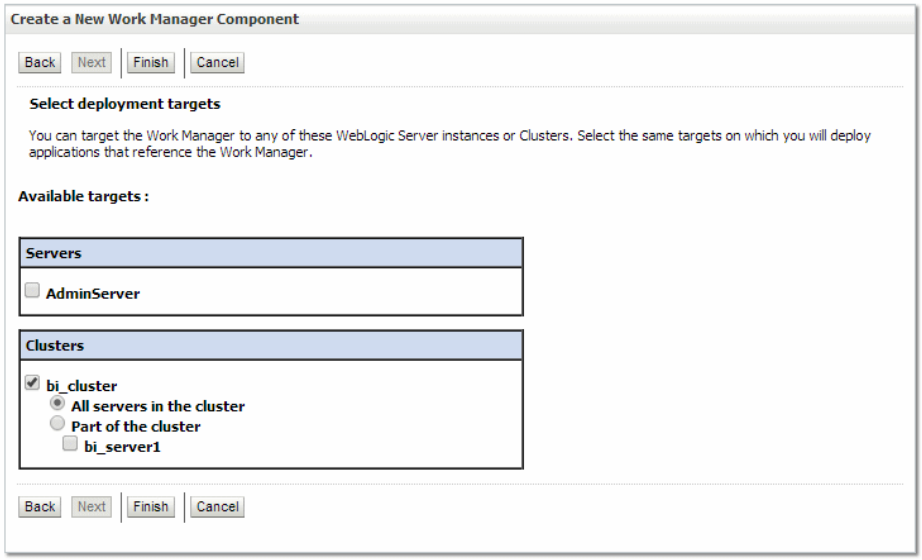


3. Under **Work Manager Properties** enter the **Name** property as: XdoWorkManager.

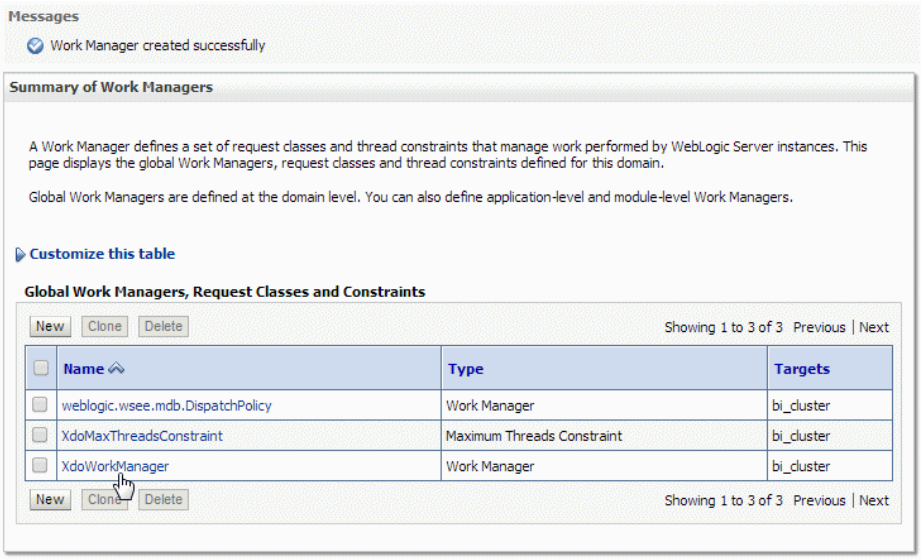


Click **Next**.

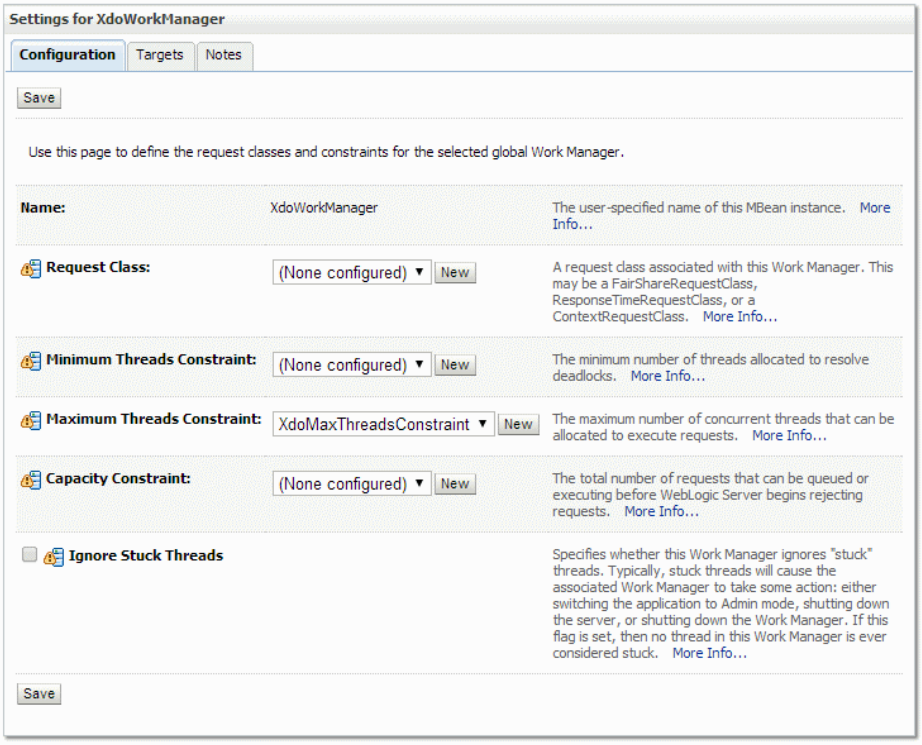
4. Under **Select deployment targets**, select "bi_cluster" and then click **Finish**.



5. Back on the **Summary of Work Managers** page, click your newly created XdoWorkManager link.



6. On the **Settings for XdoWorkManager** page, on the **Configuration** tab, specify the **Maximum Threads Constraint** as XdoMaxThreadsConstraint and click **Save**.

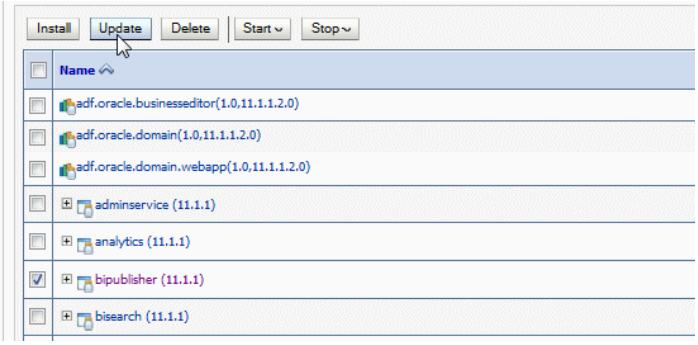


F.4.3 Redeploy the xmlpserver.ear File

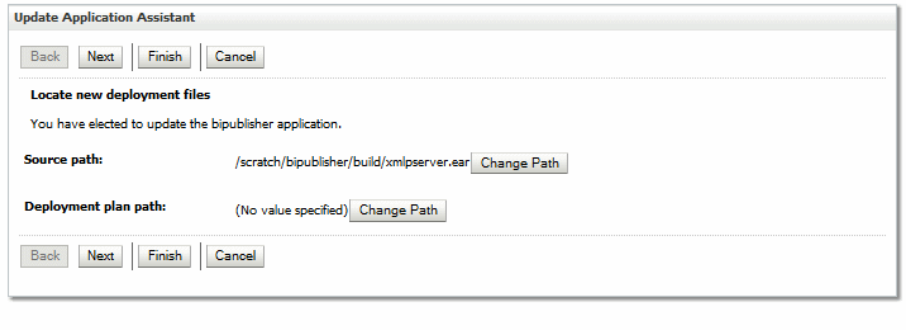
To redeploy the xmlpserver.ear file:

1. In the left pane of the Console, select **Deployments**. A table in the right pane displays all deployed applications and modules.
2. In the table, select the bipublisher application.

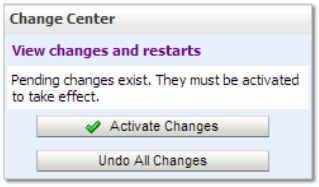
3. Click **Update**.



4. In the **Upgrade Application Assistant** click Next.



5. Click **Finish**.
6. Click **Activate Changes** in the **Change Center** pane.



Index

R

S

T

U

W

X