

Oracle® Analytics

Administering Oracle Analytics Publisher in Oracle Analytics Server



5.9.0

F24231-07

October 2021

ORACLE®

Oracle Analytics Administering Oracle Analytics Publisher in Oracle Analytics Server, 5.9.0

F24231-07

Copyright © 2020, 2021, Oracle and/or its affiliates.

Primary Author: Hemala Vivek

Contributing Authors: Pete Brownbridge, Nick Fry, Suzanne Gill, Rosie Harvey, Christine Jacobs, Stefanie Rhone, Jenny Smalling

Contributors: Oracle Analytics Server development, product management, and quality assurance teams

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xii
Documentation Accessibility	xii
Diversity and Inclusion	xii
Related Resources	xiii
Conventions	xiii

1 Introduction to Publisher Administration

Introduction	1-1
Configurations Performed by the Installer	1-2
Flow of Tasks for First Time Setup of Publisher	1-2
Start and Stop Publisher	1-2
Use Oracle WebLogic Server Administration Console	1-3
About the Administration Page	1-4
Navigate to the Administration Pages for Pixel-Perfect Reporting	1-4
About Integration with Oracle Analytics Server	1-4
About the Security Model Options	1-5
About the Data Source Connections	1-5
About Report Delivery Destinations	1-6
About Setting Runtime Configuration Properties	1-6
About the Server Configuration Settings	1-7

2 Configure Oracle Fusion Middleware Security Model

Understand the Security Model	2-1
Key Security Elements	2-2
Permission Grants and Inheritance	2-3
Default Security Configuration	2-5
Default Users and Groups	2-6
Default Application Roles and Permissions	2-7
Grant the BIServiceAdministrator Role Catalog Permissions	2-8
Manage Authentication	2-8

Access Oracle WebLogic Server Administration Console	2-9
Manage Users and Groups Using the Default Authentication Provider	2-10
Manage Authorization	2-14
Access Oracle Enterprise Manager Fusion Middleware Control	2-15
Manage the Policy Store Using Fusion Middleware Control	2-16
Modify Application Roles Using Fusion Middleware Control	2-16
Modify Membership in an Application Role	2-16
Manage Credentials	2-17
Manage BI System User Credentials	2-18
Customize the Default Security Configuration	2-18
Configure a New Authentication Provider	2-18
Configure a New Policy Store and Credential Store Provider	2-19
Reassociate the Policy Store and Credential Store	2-19
Customize the Policy Store	2-19
Create Application Roles Using Fusion Middleware Control	2-20
Create Application Policies Using Fusion Middleware Control	2-21
Change Permission Grants for an Application Policy	2-22

3 Alternative Security Options

About Alternative Security Options	3-1
Authentication and Authorization Options	3-2
Understand Publisher Users, Roles, and Permissions	3-2
Options for Configuring Users and Roles	3-3
About Privileges to Use Functionality	3-3
About Catalog Permissions	3-4
How Functional Privileges and Permissions Work Together	3-4
A Role Must Be Assigned Catalog Permissions	3-5
A Role Can Be Granted Catalog Permissions Only	3-5
Inherited Permissions	3-5
About Access to Data Sources	3-5
Configure Users, Roles, and Data Access	3-5
Create Roles	3-6
Create Users and Assign Roles to a User	3-6
Grant Catalog Permissions	3-6
Grant Data Access	3-9
Security and Catalog Organization	3-9
Use LDAP with Publisher	3-12
Configure Publisher to Use an LDAP Provider for Authentication Only	3-12
Configure Publisher to Use an LDAP Provider for Authentication and Authorization	3-13
Set Up Users and Roles in the LDAP Provider	3-14

Configure Publisher to Recognize the LDAP Server	3-14
Assign Data Access and Catalog Permissions to Roles	3-17
Disable Users Without Publisher-Specific Roles from Logging In	3-17
Integrate with Microsoft Active Directory	3-17
Configure the Active Directory	3-18
Configure Publisher	3-18
Log In to Publisher Using the Active Directory Credentials	3-19
Assign Data Access and Catalog Permissions to Roles	3-20
Configure Publisher with Single Sign-on (SSO)	3-20
How Publisher Operates with SSO Authentication	3-21
Tasks for Setting Up SSO Authentication with Publisher	3-21
Configure SSO in an Oracle Access Manager Environment	3-21
Configure a New Authenticator for Oracle WebLogic Server	3-22
Configure OAM as a New Identity Asserter for Oracle WebLogic Server	3-23
Configure Publisher for Oracle Fusion Middleware Security	3-24
Set Up Oracle Single Sign-On	3-24
Setup Procedure	3-25

4 Other Security Topics

Enable a Local Superuser	4-1
Enable a Guest User	4-1
Configure Publisher for Secure Socket Layer (SSL) Communication	4-2
Import Certificates for Web Services Protected by SSL	4-3
Add the Virtualize Property to the Identity Store Configuration	4-3
Update the JDBC Connection String to the Data Source	4-4
Update the JMS Configuration	4-4
Configure the Delivery Manager	4-5
Enable Secure Cookies	4-5
Configure Proxy Settings	4-6
Restrict Embedding of Publisher in iframes	4-7

5 Integrate with Other Oracle Security Models

Integrate with Other Oracle Security Models	5-1
Before You Begin: Create a Local Superuser	5-1
Integrate with Oracle BI Server Security	5-2
Configure Publisher for Oracle BI Server Security	5-2
Add Data Sources to BI Server Roles	5-2
Integrate with Oracle E-Business Suite	5-3
Features of the Integration with E-Business Suite Security	5-3

Configure Publisher to Use E-Business Suite Security	5-4
Add Data Sources to the E-Business Suite Roles	5-5
Grant Catalog Permissions to the E-Business Suite Roles	5-5
Integrate with Oracle Database Security	5-5
Define the Publisher Functional Roles in the Oracle Database	5-6
Add Data Sources to Roles	5-7
Grant Catalog Permissions to Roles	5-7
Integrate with Oracle Siebel CRM Security	5-7
Set Up Publisher Roles as Siebel CRM Responsibilities	5-8
Configure Publisher to Use Siebel Security	5-8
Add Data Sources to Roles	5-8
Grant Catalog Permissions to Roles	5-9

6 Configure System Maintenance Properties

Set Server Caching Specifications	6-1
Set Retry Properties for Database Failover	6-1
Set Report Viewer Properties	6-2
Clear Report Objects from the Server Cache	6-2
Clear the Subject Area Metadata Cache	6-2
Enable Diagnostics	6-2
Enable Diagnostics for Scheduler Jobs	6-3
Enable Diagnostics for Online Reports	6-3
Purge Job Diagnostic Logs	6-4
Purge Job History	6-4

7 Configure the Scheduler

Understand the Scheduler	7-1
Architecture	7-1
About Clustering	7-3
How Failover Works	7-4
About Prioritizing Jobs	7-4
About the Scheduler Configuration	7-4
Configure Processors and Processor Threads	7-5
Scheduler Diagnostics	7-5

8 Set Up Data Sources

About Private Data Source Connections	8-1
Grant Access to Data Sources Using the Security Region	8-1
About Proxy Authentication	8-2

About Pre Process Functions and Post Process Functions	8-2
About Backup Databases	8-3
Choose JDBC or JNDI Connection Type	8-4
Set Up a JDBC Connection to a Data Source	8-4
Create a Secure JDBC Connection to Oracle Autonomous Data Warehouse	8-6
Set Up a Database Connection Using a JNDI Connection Pool	8-6
Set Up a Connection to a File Data Source	8-7
Set Up a Connection to an LDAP Server Data Source	8-7
Set Up a Connection to an OLAP Data Source	8-8
Set Up a Connection to an HTTP Data Source	8-9
Set Up a Connection to a Content Server	8-9
Set Up a Connection to a Web Service	8-10
View or Update a Data Source	8-10

9 Set Up Delivery Destinations

Add a WebDAV Server	9-1
Configure Delivery Options	9-1
Understand Printer and Fax Server Configuration	9-2
Add a Printer	9-3
Add a Fax Server	9-4
Add an Email Server	9-4
Deliver Reports Using Email Delivery Service on Oracle Cloud Infrastructure	9-5
Add an HTTP Server	9-6
Add an FTP or SFTP Server	9-7
SSH Options for SFTP	9-8
Add a Content Server	9-9
Add a Common UNIX Printing System (CUPS) Server	9-10
Add an Oracle Content and Experience Server	9-10

10 Define Runtime Configurations

Set Runtime Properties	10-1
PDF Output Properties	10-2
PDF Digital Signature Properties	10-5
PDF Accessibility Properties	10-7
PDF/A Output Properties	10-7
PDF/X Output Properties	10-8
DOCX Output Properties	10-10
RTF Output Properties	10-10
PPTX Output Properties	10-11

HTML Output Properties	10-11
FO Processing Properties	10-13
RTF Template Properties	10-15
XPT Template Properties	10-16
PDF Template Properties	10-17
Excel Template Properties	10-17
CSV Output Properties	10-18
Excel Output Properties	10-18
EText Output Properties	10-19
All Outputs Properties	10-20
Memory Guard Properties	10-20
Data Model Properties	10-21
Delivery Properties	10-22
Define Font Mappings	10-22
Make Fonts Available for Publishing	10-23
Set Font Mapping at the Site Level or Report Level	10-23
Create a Font Map	10-23
Define Currency Formats	10-23
Understand Currency Formats	10-24

11 Secure Reports

Use Digital Signature in a PDF Document	11-1
Prerequisites and Limitations	11-1
Obtain Digital Certificates	11-1
Create PFX Files	11-2
Apply a Digital Signature	11-2
Register Your Digital Signature and Assign Authorized Roles	11-2
Specify the Signature Display Field or Location	11-3
Specify a Template Field in a PDF Template for the Digital Signature	11-3
Specify the Location for the Digital Signature in the Report Properties	11-3
Run and Sign Reports with a Digital Signature	11-4
Use PGP Keys for Encrypted Report Delivery	11-4

12 View Audit Data of Publisher Catalog Objects

About Viewing Audit Data of Publisher Catalog Objects	12-1
Enable or Disable Viewing of Publisher Audit Data	12-1
Specify the Data Source Connection for Publisher Audit Data	12-2
View Publisher Audit Data	12-2

13 Add Translations for the Catalog and Reports

Introduction	13-1
Limitations of Catalog Translation	13-1
Export and Import a Catalog Translation File	13-1
Template Translation	13-2
Generate the XLIFF File from the Layout Properties Page	13-3
Translate the XLIFF File	13-3
Upload the Translated XLIFF File to Publisher	13-4
Use a Localized Template	13-4
Design the Localized Template File	13-4
Upload the Localized Template to Publisher	13-4

14 Move Catalog Objects Between Environments

Overview	14-1
When to Use the Catalog Utility	14-1
Other Options for Moving Catalog Objects	14-2
What Files Are Moved	14-2
Maintaining Identical Folder Names and Structure Across Environments	14-3
Prepare to Use the Catalog Utility	14-4
Configure the Environment	14-4
Export the Reporting Objects	14-5
Import the Reporting Objects	14-5
Generate Translation Files and Checking for Translatability	14-6
Generate a Translation File for a Report Definition File (.xdo)	14-6
Generate a Translation File for an RTF Template	14-7

15 Customize the Publisher User Interface

What are Skins and Styles?	15-1
About Style Customizations	15-1
Modify the User Interface Styles	15-2
Customize the Style	15-2
Customize the Style for Publisher Standalone	15-2
Customize the Style for Publisher Integrated with the Oracle Analytics Server	15-3
Fallback Mechanism for Custom Styles	15-4
Custom Style Sheets	15-4
Images	15-4

A Scheduler Configuration Reference

Introduction	A-1
Configure Publisher for ActiveMQ	A-1
Install ActiveMQ	A-1
Register ActiveMQ as a JNDI Service	A-1
Update the Publisher Scheduler Configuration Page	A-2
Configure the Quartz Scheduler	A-2
Recommendations for Using DataDirect Connect or Native Database Drivers	A-2
Set Up a User on Your Scheduler Database	A-3
Connect to Your Scheduler Database and Install the Schema	A-3
Connect to Oracle Databases	A-4
Connect to IBM DB2 Databases	A-5
Connect to Microsoft SQL Server Databases	A-5
Connect to Sybase Adaptive Server Enterprise Databases	A-5

B Integration Reference

About Integration	B-1
Prerequisites	B-1
Integrate with Presentation Services	B-1
Set Up a JDBC Connection to the Oracle BI Server	B-2

C Configuration File Reference

Publisher Configuration Files	C-1
Set Properties in the Runtime Configuration File	C-1
File Name and Location	C-1
Namespace	C-1
Configuration File Example	C-2
Example Element Specification	C-2
Structure of the Root Element	C-3
Attributes of Root Element	C-3
Description of Root Element	C-3
Properties and Property Elements	C-3
<properties> Element	C-3
Description of <properties> Element	C-4
<property> Element	C-4
Attribute of <property> Element	C-4
Description of <property> Element	C-4
Font Definitions	C-4
 Element	C-5

Attribute of Element	C-5
Description of Element	C-5
 Element	C-5
Attributes of Element	C-5
Description of Element	C-6
<font-substitute> Element	C-6
Attributes of <font-substitute> Element	C-6
Description of <font-substitute> Element	C-6
<type1> element	C-6
Attribute of <type1> Element	C-7
Description of <type1> Element	C-7
Predefined Fonts	C-7
Included Barcode Fonts	C-9

D Update the Publisher Context Root

Update the Publisher URL Context Root	D-1
Example	D-1
Update the xmlpserver META-INF/application.xml File	D-2
Update the xmlpserver WAR/WEB-INF/web.xml File	D-2
Update the xmlpserver WAR/WEB-INF/weblogic.xml File	D-3
Update the xmlp-server-config.xml File	D-3
Update the analytics META-INF/application.xml File	D-3
Update the instanceconfig.xml File	D-4
Update the bipublisher and analytics Applications in WebLogic Server	D-4

E Use Command-Line Utilities

Generate the Utilities	E-1
Configure Memory Guard Properties Using Utility	E-2
Memory Guard Properties	E-3

Preface

Learn to administer Publisher.

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Related Resources](#)
- [Conventions](#)

Audience

This document is intended for system administrators who are responsible for managing the data source connections, delivery servers, security, and configuration of Publisher.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

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

Related Resources

For a full list of guides, refer to the Books tab on Oracle Analytics Server Help Center.

Conventions

The following text conventions are used in this document:

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

Videos and Images

Your company can use skins and styles to customize the look of the application, dashboards, reports, and other objects. It is possible that the videos and images included in the product documentation look different than the skins and styles your company uses.

Even if your skins and styles are different than those shown in the videos and images, the product behavior and techniques shown and demonstrated are the same.

1

Introduction to Publisher Administration

This topic describes tasks required to administer Publisher.

Topics:

- [Introduction](#)
- [Configurations Performed by the Installer](#)
- [Flow of Tasks for First Time Setup of Publisher](#)
- [Start and Stop Publisher](#)
- [About the Administration Page](#)
- [About Integration with Oracle Analytics Server](#)
- [About the Security Model Options](#)
- [About the Data Source Connections](#)
- [About Report Delivery Destinations](#)
- [About Setting Runtime Configuration Properties](#)
- [About the Server Configuration Settings](#)

Introduction

You can author, manage, and deliver pixel-perfect reports such as operational reports, electronic funds transfer documents, government PDF forms, shipping labels, checks, sales and marketing letters.

The administrator sets up and maintains the following system components.

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

For other business roles, see the guides that are outlined in the table below for information about using the product.

Role	Sample Tasks
Data Model developer	Fetch and structure the data to use in reports
Application developer or integrator	Integrate Publisher into existing applications using the application programming interfaces

Role	Sample Tasks
Report consumer	View reports Schedule report jobs Manage report jobs
Report designer	Create report definitions Design layouts

Configurations Performed by the Installer

The Installer performs certain configurations after installation.

Post-installation configurations include:

- 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 Publisher catalog and repository are configured to \${xdo.server.config.dir}/repository.

Flow of Tasks for First Time Setup of Publisher

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

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.	Enable a Local Superuser
Set up the chosen security model and test	Configure Oracle Fusion Middleware Security Model Alternative Security Options Integrate with Other Oracle Security Models
Set up the data sources and test	Set Up Data Sources
Set up the delivery servers and test	Set Up Delivery Destinations
Configure server properties	Configure System Maintenance Properties
Configure system runtime properties	Define Runtime Configurations

Start and Stop Publisher

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

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

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

Use Oracle WebLogic Server Administration Console

Use the Oracle WebLogic Server Administration Console to start and stop Publisher.

1. Start the Oracle WebLogic Server Administration Console.
2. Under the Domain Structure, click **Deployments**.
3. Click **Control**.
4. In the Deployments table, select the Publisher application.
5. Click the appropriate action.

When you **Start** an application, pick one of the following options:

- **Servicing all requests:** Specifies that WebLogic Server make the application immediately available to all clients.

- **Servicing only administration requests:** Specifies that WebLogic Server make the application available in Administration Mode only.

When you **Stop** an application, pick one of the following options:

- **When work completes:** Specifies that WebLogic Server wait for the application to finish its work and for all currently connected users to disconnect.
- **Force stop now:** Specifies that WebLogic Server stop the application immediately, regardless of the work being performed and the users that are connected.
- **Stop, but continue servicing administration requests:** Specifies that WebLogic Server stop the application once all its work has finished, but to then put the application in Administration Mode so it can be accessed for administrative purposes.

About the Administration Page

Many of the tasks described in the Administration section of this guide are performed from the 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.

Navigate to the Administration Pages for Pixel-Perfect Reporting

Administrators set the options for publishing reports through the administration pages for pixel-perfect reporting.

1. On the header, click **Administration**, and then click **Manage Publisher**.
2. On the Publisher Administration page, select the required option.

About Integration with Oracle Analytics Server

If you installed Publisher with the Oracle Analytics Server, then you must perform the Administration tasks in the Publisher Administration page.

Administration tasks are described in the following table. Navigate to the Publisher Administration page as follows:

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

Task	Where to Get Information
Set up data source connections for reporting	Set Up Data Sources
Grant access to data sources for user roles defined in Oracle Analytics Server	Grant Data Access
Configure the connections to delivery servers (for example, printers, e-mail servers, FTP servers, and so on)	Set Up Delivery Destinations
Configure the scheduler processors	Configure the Scheduler

Task	Where to Get Information
Configure system runtime properties such as PDF security properties, properties specific to each output format, template type properties, font mappings, and currency formats.	Set Up Delivery Destinations
Configure server properties such as caching specifications, database failover properties, and database fetch size.	Configure System Maintenance Properties

About the Security Model Options

Publisher offers a variety of security options.

- **Oracle Fusion Middleware Security**
After installation, Publisher is configured to use Oracle Fusion Middleware Security. See [Configure Oracle Fusion Middleware Security Model](#). If you prefer to use another security model, then choose from the alternative options.
- **Publisher Security**
Use Publisher's Users and Roles paradigm to control access to reports and data sources. See [Alternative Security Options](#).
- **Integration with an LDAP server**
Set up the Publisher roles in your LDAP server then configure Publisher to integrate with it. See [Alternative Security Options](#).
- **Oracle E-Business Suite**
Upload a DBC file to recognize your Oracle E-Business Suite users. See [Integrate with Other Oracle Security Models](#)
- **Oracle BI Server**
You can still leverage the 10g legacy BI Server authentication method if you choose not to upgrade to Oracle Fusion Middleware Security. See [Integrate with Other Oracle Security Models](#).
- **Oracle Database**
Set up the Publisher roles in your Oracle Database and then configure Publisher to integrate with it. See [Integrate with Other Oracle Security Models](#).
- **Oracle Siebel CRM Security Model**
See [Integrate with Other Oracle Security Models](#).

About the Data Source Connections

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

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

- Database connections

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 Publisher can access
- Web Service connections
- HTTP XML connections
- Content Server

If you integrated your system with Oracle Analytics Server 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.

About Report Delivery Destinations

The Publisher delivery manager supports multiple delivery channels.

Supported delivery channels include:

- Printer
- Fax
- E-mail
- HTTP notification
- FTP
- Web Folder (or WebDAV)
- Content Server
- Oracle Content and Experience Cloud server
- Common UNIX Printing System (CUPS) Server

About Setting Runtime Configuration Properties

Use the Runtime Configuration page to enable configuration settings for your system.

The properties include settings that do the following:

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

About the Server Configuration Settings

Publisher administration also includes a set of system maintenance settings and tasks.

Settings and tasks include:

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

2

Configure Oracle Fusion Middleware Security Model

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

Topics:

- [Understand the Security Model](#)
- [Key Security Elements](#)
- [Permission Grants and Inheritance](#)
- [Default Security Configuration](#)
- [Manage Authentication](#)
- [Manage Authorization](#)
- [Manage Credentials](#)
- [Customize the Default Security Configuration](#)

Understand 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, Publisher can furnish uniform security and identity management across the enterprise.

After installation 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's 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. Publisher is part of the active security realm configured for the Oracle WebLogic Server domain into which it is installed.

See *Securing Applications with Oracle Platform Security Services*. For more information about managing the Oracle WebLogic Server domain and security realm, see *Understanding Security for Oracle WebLogic Server* and *Administering Security for Oracle WebLogic Server*.

Key Security Elements

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

These key elements include:

- Application policy

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 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 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 Publisher is installed into that domain. 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. 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.

Permission Grants and Inheritance

Publisher provides application-specific permissions for accessing different features.

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 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 Publisher. The default security configuration provides the following preconfigured application roles: BIServiceAdministrator (an administrator), BIContentAuthor (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 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.

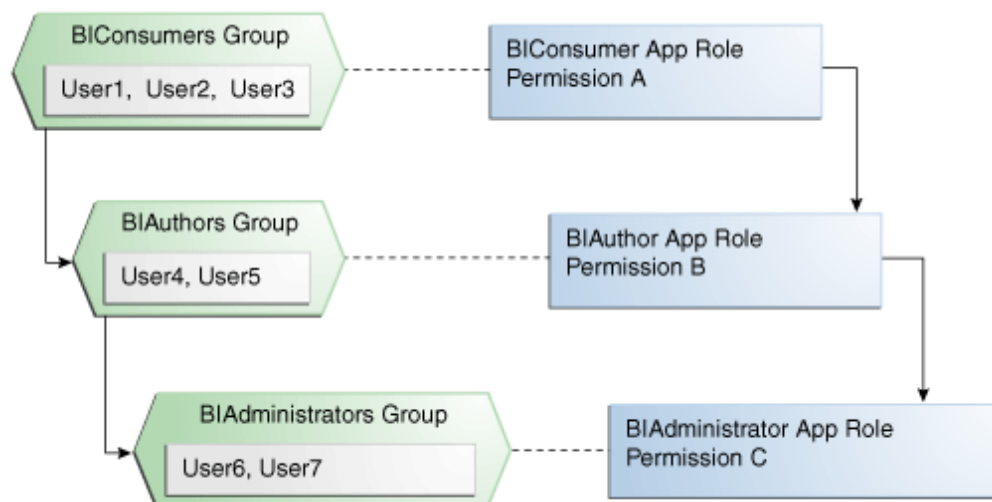
A user can also be granted permissions if they inherit other application roles. Members of application roles can include other groups and application roles. The result is a hierarchical role structure where permissions can be *inherited* in addition to being *explicitly granted*. This hierarchy provides 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 BIServiceAdministrator application role includes the BIAdministrators group, the BIContentAuthor application role includes the BIAuthors group, and the BIConsumer application role includes the BIConsumers group. The default BIServiceAdministrator application role is a member of the BIContentAuthor application role, and the BIContentAuthor 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 BIServiceAdministrator role, the BIContentAuthor 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 [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).

The figure below shows these relationships between the default groups and application roles.



The table below summarizes how permissions are granted explicitly or are inherited in the previous example and figure.

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	BIContentAuthor: Explicit BIconsumer: Inherited	Permission B: Explicit Permission A: Inherited
User6, User7	BIAAdministrators: Explicit BIAuthors: Inherited BIconsumers: Inherited	BIServiceAdministrator: Explicit BIContentAuthor: Inherited BIconsumer: Inherited	Permission C: Explicit Permission B: Inherited Permission A: Inherited

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 Publisher is installed and is configured to use the Oracle Fusion Middleware security model. 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 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 Publisher permissions.

The Publisher default security stores are configured as described in the table below during installation.

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.	Publisher can be configured to use alternative authentication providers.

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 <code>MW_HOME/user_projects/domain/your_domain/config/fmwconfig</code> - Managed with Oracle Enterprise Manager Fusion Middleware Control.	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.	Publisher can be configured to use Oracle Internet Directory as the credential store provider.

Default Users and Groups

Default user and group names can be changed to different values and new names can be added by an administrative user using Oracle WebLogic Server Administration Console.

The table below lists the default user names and passwords added to the Publisher identity store provider after installation.

Default User Name and Password	Purpose	Description
Name: <i>administrator user</i> Password: <i>user supplied</i>	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 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.

No default groups are created during the installation of Publisher.

Default Application Roles and Permissions

Permissions in Publisher are granted by specific roles. Permissions can also be inherited from group and application role hierarchies.

The table below lists the Publisher permissions and the application role that grants these permissions. This mapping exists in the default policy store.

The table also 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 [Permission Grants and Inheritance](#).

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 Grant the BIServiceAdministrator Role Catalog Permissions for additional steps required to grant the BIServiceAdministrator permissions on Shared Folders.	BIServiceAdministrator
oracle.bi.publisher.developDataModel	Grants permission to create or edit data models.	BIContentAuthor
oracle.bi.publisher.developReport	Grants permission to create or edit reports, style templates, and sub templates. This permission also enables connection to the Publisher server from the Template Builder.	BIContentAuthor
oracle.bi.publisher.runReportOnline	Grants permission to open (run) 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. 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 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 *Securing Applications with Oracle Platform Security Services*.

Grant the BIServiceAdministrator Role Catalog Permissions

The BIServiceAdministrator role is granted only Read permissions on the catalog by default.

This means that before a BIServiceAdministrator can manage Shared Folders the BIServiceAdministrator role must be granted Write and Delete permissions on the Shared Folders node. See [Grant Catalog Permissions](#) for a detailed description of granting permissions in the catalog.

Manage 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. 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 [Configure 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 WebLogic Server Administration Console Online Help*.

Access 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's configured as the default authenticator. It's 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.

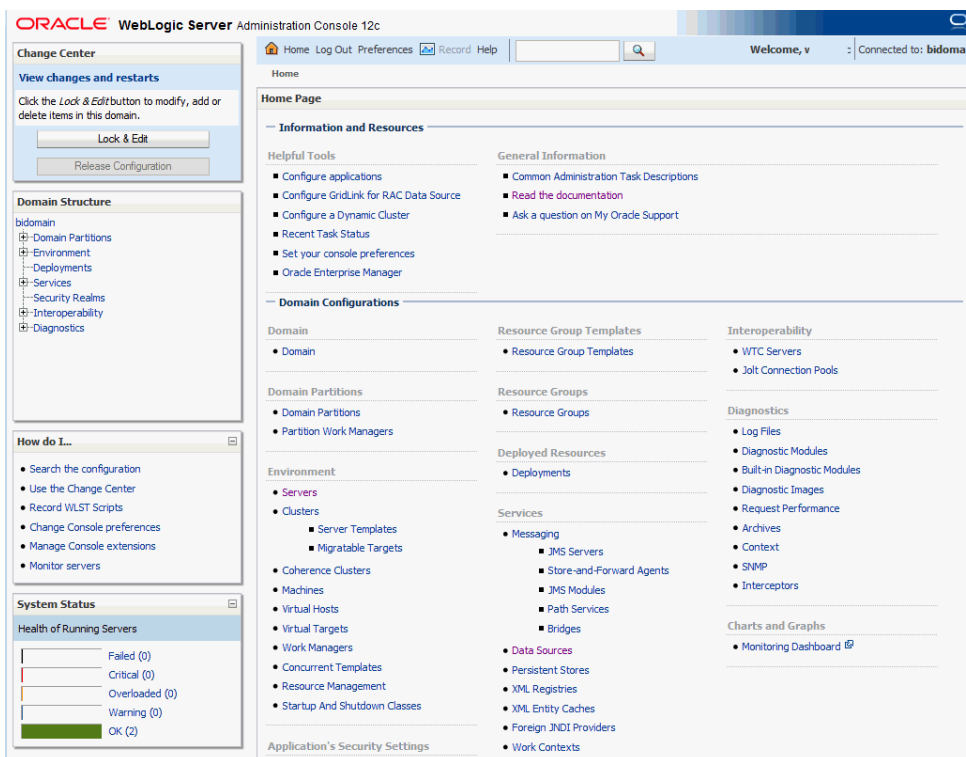
For example, `http://hostname:7001/console`. The Administration Console login page displays, as shown the figure below.



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

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

The Administration Console displays, as shown the figure below.



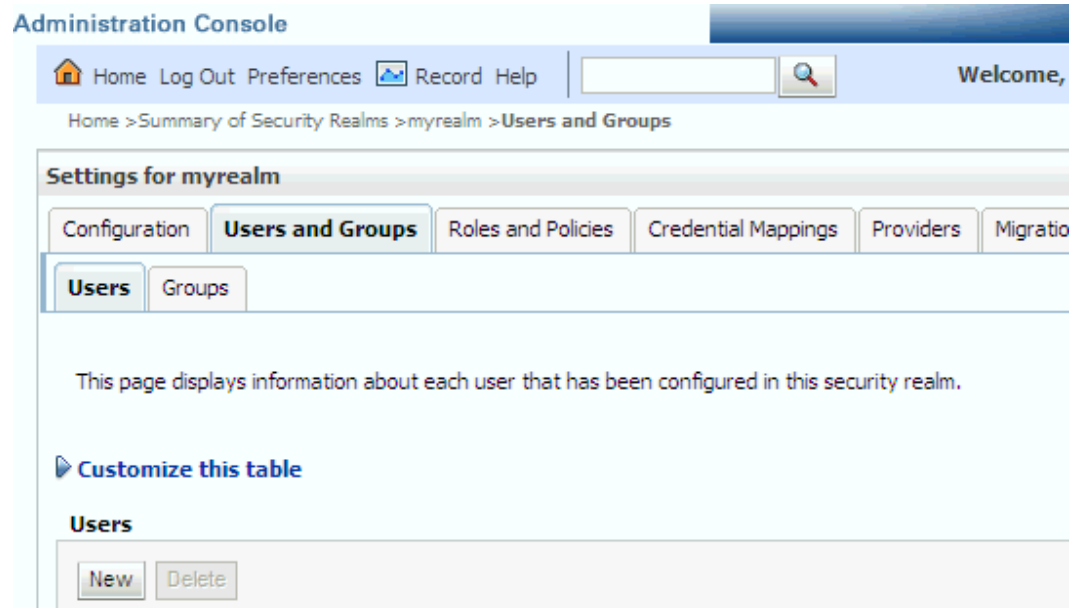
Manage Users and Groups Using the Default Authentication Provider

Managing a group is more efficient than managing a large number of users individually. Best practice is to first organize all 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 requires change, then you need only modify the permissions granted by the application roles, or create a new application role 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.
See [Access 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're configuring. For example, myrealm.
4. Select Users and Groups tab (shown below), then **Users**. Click **New**.



5. In the Create a New User page (shown below) 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 at least 8 characters long.
 - **Confirm Password:** Re-enter the user password.

The screenshot shows the 'Create a New User' form in the Oracle WebLogic Server Administration Console. The breadcrumb trail is 'Home > Summary of Security Realms > myrealm > Users and Groups'. The form includes a title bar 'Create a New User' with 'OK' and 'Cancel' buttons. Under 'User Properties', it states: 'The following properties will be used to identify your new User. * Indicates required fields'. The form fields are: 'Name' (required) with the value 'Danny Developer', 'Description' (optional) with the value 'Report Developer', 'Provider' (dropdown menu) set to 'DefaultAuthenticator', 'Password' (required, masked with dots), and 'Confirm Password' (required, masked with dots). 'OK' and 'Cancel' buttons are at the bottom.

6. Click **OK**.

The user name is added to the User table.

To create a group in the default directory server:

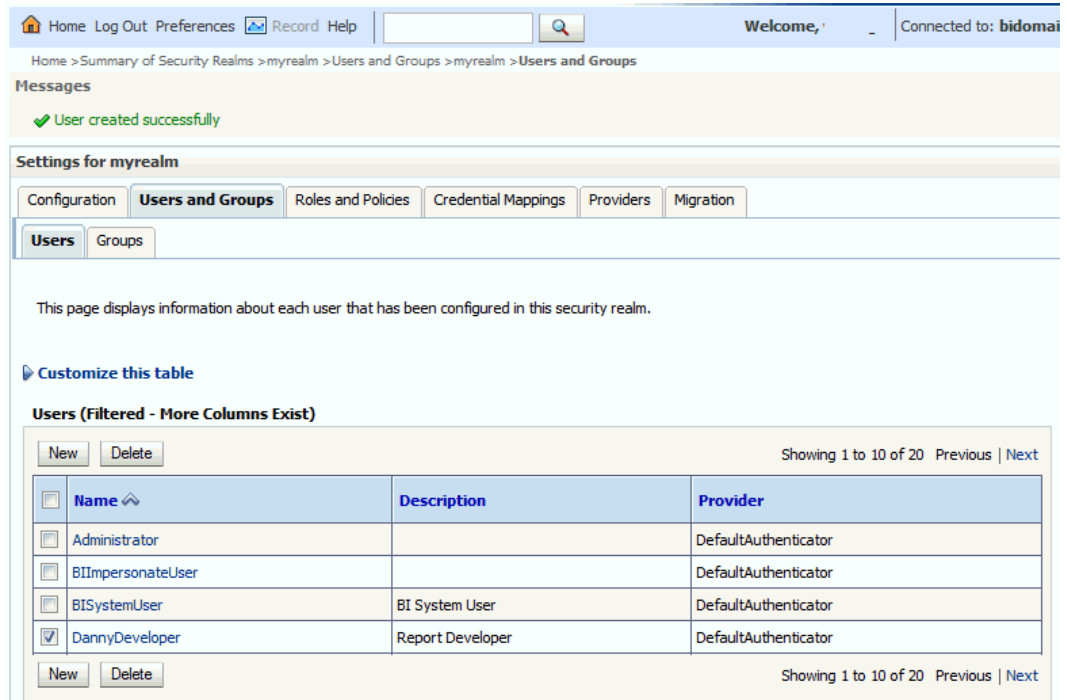
1. If needed, launch Oracle WebLogic Server Administration Console.
See [Access Oracle WebLogic Server Administration Console](#).
2. Log in as an administrative user.
3. In the Administration Console, select **Security Realm** from the left pane and click the realm you're configuring. For example, **myrealm**.
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.
See [Access 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're configuring. For example, myrealm.
4. Select **Users and Groups** tab, then **Users**, as shown in the figure below. Select the user from **Name**.



Home > Summary of Security Realms > myrealm > Users and Groups > myrealm > Users and Groups

Messages

✔ User created successfully

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

Users Groups

This page displays information about each user that has been configured in this security realm.

Customize this table

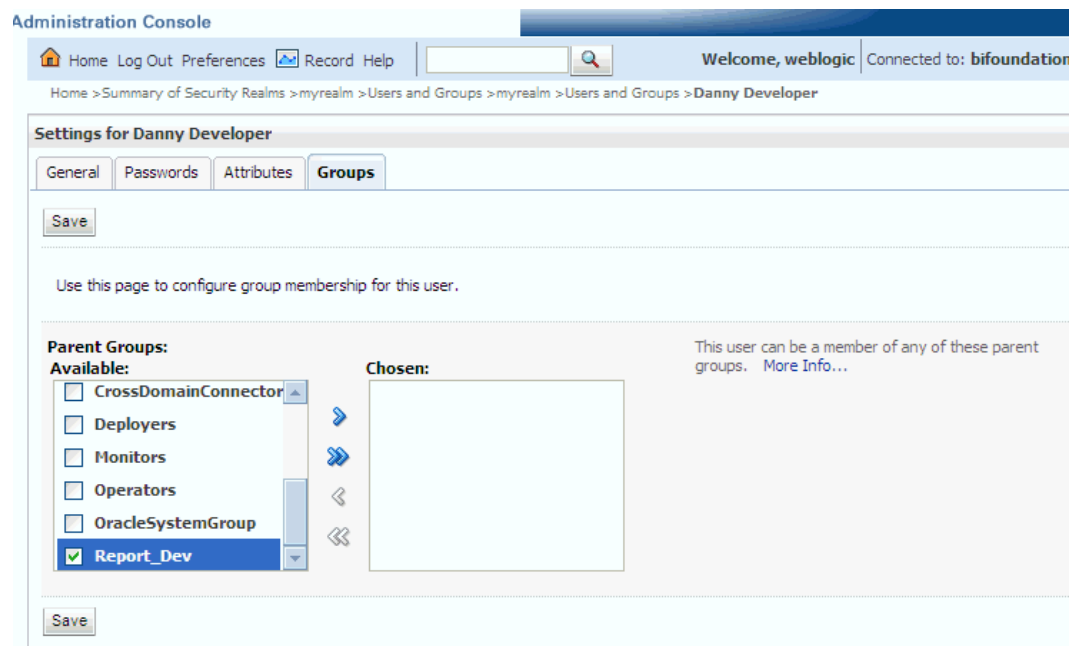
Users (Filtered - More Columns Exist)

New Delete Showing 1 to 10 of 20 Previous Next

☐	Name	Description	Provider
☐	Administrator		DefaultAuthenticator
☐	BIImpersonateUser		DefaultAuthenticator
☐	BISystemUser	BI System User	DefaultAuthenticator
☒	DannyDeveloper	Report Developer	DefaultAuthenticator

New Delete Showing 1 to 10 of 20 Previous Next

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 below.



Administration Console

Home > Summary of Security Realms > myrealm > Users and Groups > myrealm > Users and Groups > Danny Developer

Settings for Danny Developer

General Passwords Attributes **Groups**

Save

Use this page to configure group membership for this user.

Parent Groups:

Available:

- ☐ CrossDomainConnector
- ☐ Deployers
- ☐ Monitors
- ☐ Operators
- ☐ OracleSystemGroup
- ☒ Report_Dev

Chosen:

This user can be a member of any of these parent groups. More Info...

Save

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.
See [Access 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're 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 below.

The screenshot shows the Oracle WebLogic Server Administration Console interface. At the top, there's a navigation bar with links like Home, Log Out, Preferences, Record, and Help. Below this, a breadcrumb trail indicates the current path: Home > Summary of Security Realms > myrealm > Users and Groups > dnoonan > Users and Groups > ddeveloper. The main content area is titled 'Settings for ddeveloper' and has four tabs: General, Passwords, Attributes, and Groups. The 'General' tab is active, showing a 'Save' button at the top. Below it, a message states: 'Use this page to change the description for the selected user.' There are two main sections: 'Name' with the value 'ddeveloper' and a description 'The login name of this user. More Info...'; and 'Description' with a text input field containing 'Danny Developer' and a description 'A short description of this user. For example, the user's full name. More Info...'. A second 'Save' button is at the bottom of the 'Description' section.

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

Manage Authorization

After a user is authenticated, further access to 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 Publisher. A policy store can be file-based or LDAP-based and holds the mapping definitions between the default Publisher application roles, permissions, users and groups. 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.

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 Publisher understands. All valid 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 Publisher permissions grants, see [Default Application Roles and Permissions](#). For more information about customizing application roles and permission grants, see [Customize the Policy Store](#).

Access 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 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 *Administering Oracle Fusion Middleware*.

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`.

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

The password is the one you supplied during the installation of 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. Select **Security** to display a submenu.
- From the content pane, display the **WebLogic Domain** menu and select **Security**. Select **Security** to display a submenu.

Manage the Policy Store Using Fusion Middleware Control

Use Fusion Middleware Control to manage the 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 [Configure 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. See [Modify Membership in an Application Role](#).
- Modifying the permission grants for an application role. See [Change Permission Grants for an Application Policy](#).
- Creating a new application role from the beginning. See [Create Application Roles Using Fusion Middleware Control](#).
- Creating a new application role based on an existing application role. See [Create Application Roles Using Fusion Middleware Control](#).

Modify 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 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.

Modify 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 [Access 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**.

3. Select the cell next to the application role name and click **Edit** to display the Edit Application Role page.

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**.

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

Manage 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. You can configure and administer LDAP-based credential stores 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.

Publisher supports the following credential maps:

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

Publisher supports the following credential types:

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

To help you get started with your development environment, default credentials are added to the file-based credential store during installation. Note that Publisher credentials such as user

passwords are stored in the identity store and managed with its corresponding administrative interface.

Manage BI System User Credentials

If using Oracle Analytics Server as a data store, Publisher establishes system communication with it as BI system user.

Oracle Analytics Server uses BI system user for trusted system communication. To change the password of BI system user in the credential store (oracle.bi.system credential map), see *Security Guide for Oracle Business Intelligence Enterprise Edition*.

Customize the Default Security Configuration

You can customize the default security configuration in various ways.

- Configure a new authentication provider. See [Configure a New Authentication Provider](#).
- Configure new policy store and credential store providers. See [Configure a New Policy Store and Credential Store Provider](#).
- Migrate policies and credentials from one store to another. See [Reassociate the Policy Store and Credential Store](#).
- Create new application roles. See [Create Application Roles Using Fusion Middleware Control](#).
- Create new application policies. See [Create Application Policies Using Fusion Middleware Control](#).
- Modify the permission grants for an application policy. See [Change Permission Grants for an Application Policy](#).

Configure a New Authentication Provider

You can configure another supported LDAP server to be the authentication provider.

Configuring Publisher to use an alternative external identity store is performed using the Oracle WebLogic Server Administration Console. 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 [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 Publisher application roles to different groups again. For example, if the corporate LDAP server is being used as the identity store and you're unable to re-create the 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 WebLogic Server Administration Console Online Help* and *Administering Security for Oracle WebLogic Server*.

Configure a New Policy Store and Credential Store Provider

The policy store and credential store can be file-based or LDAP-based.

The pre-requisites for using an LDAP-based store are the same as for both the policy store and credential store. See *Securing Applications with Oracle Platform Security Services*.

Reassociate 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.

See *Securing Applications with Oracle Platform Security Services*.

Customize 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 *Securing Applications with Oracle Platform Security Services*.



Note:

Before creating a new application policy or application role and adding it to the default 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 [Permission Grants and Inheritance](#).

Create Application Roles Using Fusion Middleware Control

You can create a new application role or copy from an existing role using Fusion Middleware Control.

Creating Application Roles

There're two methods for creating an application role.

- **Create New** — Creates an application role. You can add members when you create the new role, or you can save the new role after naming it and later add members.
- **Copy Existing** — Creates an application role by copying an existing application role. The copy contains the same members as the original, and the new role will be Grantee of the same application policy. You can modify the copy as required when you create the new role.

Creating a New Application Role

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

For information, see [Access Oracle Enterprise Manager Fusion Middleware Control](#).

2. Choose **obi** from **Application Stripe** list. Click the search icon next to **Role Name**.

You can view the Publisher roles.

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.
4. In the Members section, Click **Add** to select the users, groups, or application roles you want to map to the applications role.
 - a. From the **Type** list, select **Application Role**, **User**, or **Role** you want to map to the application role.
 - b. Optionally, you can specify the criteria for **Principal Name** and **Display Name**.
 - c. Click the search icon next to **Display Name**.
 - d. Select the principals from the Searched Principals table.
 - e. Click **OK**.
 5. Click **OK** to return to the Application Roles page.

The table at the bottom of the page displays the new application role.

Creating an Application Role Based on an Existing role

1. Log into Fusion Middleware Control, navigate to **Security**, and select **Application Roles** to display the Application Roles page.
For information, see [Access Oracle Enterprise Manager Fusion Middleware Control](#).
2. Choose **obi** from **Application Stripe** list. Click the search icon next to **Role Name**.
You can view the Publisher roles.
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.
6. Use **Add** and **Delete** to modify the members as appropriate and click **OK**.
The table at the bottom of the page displays the newly created application role.

Create Application Policies Using Fusion Middleware Control

All Publisher permissions are provided and you cannot create new permissions. Permission grants are controlled in the Fusion Middleware Control Application Policies page.

Create Application Policies

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're two methods for creating a new application policy:

- **Create New** — Creates an application policy and adds permissions to it.
- **Copy Existing** — Creates an application policy by copying an existing application policy. You can name the copy, remove existing permissions, or add new permissions as required.

Create a New Application Policy

1. Log in to Fusion Middleware Control, navigate to **Security**, and select **Application Policies** to display the Application Policies page.
For information, see [Access Oracle Enterprise Manager Fusion Middleware Control](#).
2. Choose **obi** from **Application Stripe** list. Click the search icon next to **Principal Name**.
You can view the Publisher policies. 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.
 - a. Complete the Search section, and click the search icon next to the **Resource Name** field.

All permissions located in the **obi** application stripe are displayed. For information about the Publisher permissions, see [Default Application Roles and Permissions](#).

- b. Select the desired Publisher permissions, and click **Continue**. Selecting non-Publisher permissions has no effect in the policy.
- c. If required, customize the permission and click **Select**.

The Permissions section display the selected permissions.

5. To add an application role, user, or group to the policy being created, click **Add** in the Grantee section..

In the Add Principal dialog, do the following:

- Complete the Search section, and click the search icon next to the **Display Name** field.
 - Select the required principals from the **Searched Principals** list.
 - Click **OK**.
6. Click **OK** to return to the Application Policies page. You can view the **Principal** (Grantee) and **Permissions** of the new policy in the table.

Create an Application Policy Based on an Existing Policy

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

For information, see [Access Oracle Enterprise Manager Fusion Middleware Control](#).

2. Choose **obi** from **Application Stripe** list. Click the search icon next to **Principal Name**.

You can view the Publisher application policies. The **Principal** column displays the name of the policy **Grantee**.

3. Select an existing policy from the table.
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 role, user, or group to the policy, click **Add** in the **Grantee** area to display the Add Principal dialog.
 - Complete the Search area and click the blue search icon next to the **Display Name** field.
 - Select from the **Searched Principals** list.
 - Click **OK**.

The Application Policies page displays the Principal and Permissions of the policy.

Change 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 in to Fusion Middleware Control, navigate to Security, then select **Application Policies** to display the Application Policies page.

For information, see [Access 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 Publisher 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 Publisher, including Single Sign-on (SSO), LDAP options, Oracle Access Manager (OAM), and Microsoft Active Directory. Topics:

- [About Alternative Security Options](#)
- [Authentication and Authorization Options](#)
- [Understand Publisher Users, Roles, and Permissions](#)
- [About Privileges to Use Functionality](#)
- [About Catalog Permissions](#)
- [How Functional Privileges and Permissions Work Together](#)
- [About Access to Data Sources](#)
- [Configure Users, Roles, and Data Access](#)
- [Security and Catalog Organization](#)
- [Use LDAP with Publisher](#)
- [Integrate with Microsoft Active Directory](#)
- [Configure Publisher with Single Sign-on \(SSO\)](#)
- [Configure SSO in an Oracle Access Manager Environment](#)
- [Set Up Oracle Single Sign-On](#)

About Alternative Security Options

This chapter describes security concepts and options for a standalone implementation of Publisher that's not installed as part of the Oracle Analytics Server.

Note the following:

- If you installed the Oracle Analytics Server, then see the security guide for information about security.
- If you installed Publisher on its own and you plan to use Oracle Fusion Middleware Security, then see [Understand the Security Model](#). The following topics will be of interest in this chapter:
 - [About Catalog Permissions](#)
 - [About Access to Data Sources](#)
- To configure Publisher with these other Oracle security models:
 - Oracle BI Server security
 - Oracle E-Business Suite security
 - Oracle Database security

- Siebel CRM security

See [Integrate with Other Oracle Security Models](#).

Use the information in this chapter to configure the following:

- Publisher Security
- Integration with an LDAP provider

 Note:

Any identity store provider supported by Oracle WebLogic Server can be configured to be used with Publisher. Configuring Publisher to use an alternative external identity store is performed using the Oracle WebLogic Server Administration Console. See [Customize the Default Security Configuration](#).

- Integration with a Single Sign-On provider

Authentication and Authorization Options

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 Publisher to use a Single Sign-On provider or LDAP provider for authentication with another security model to handle authorization.

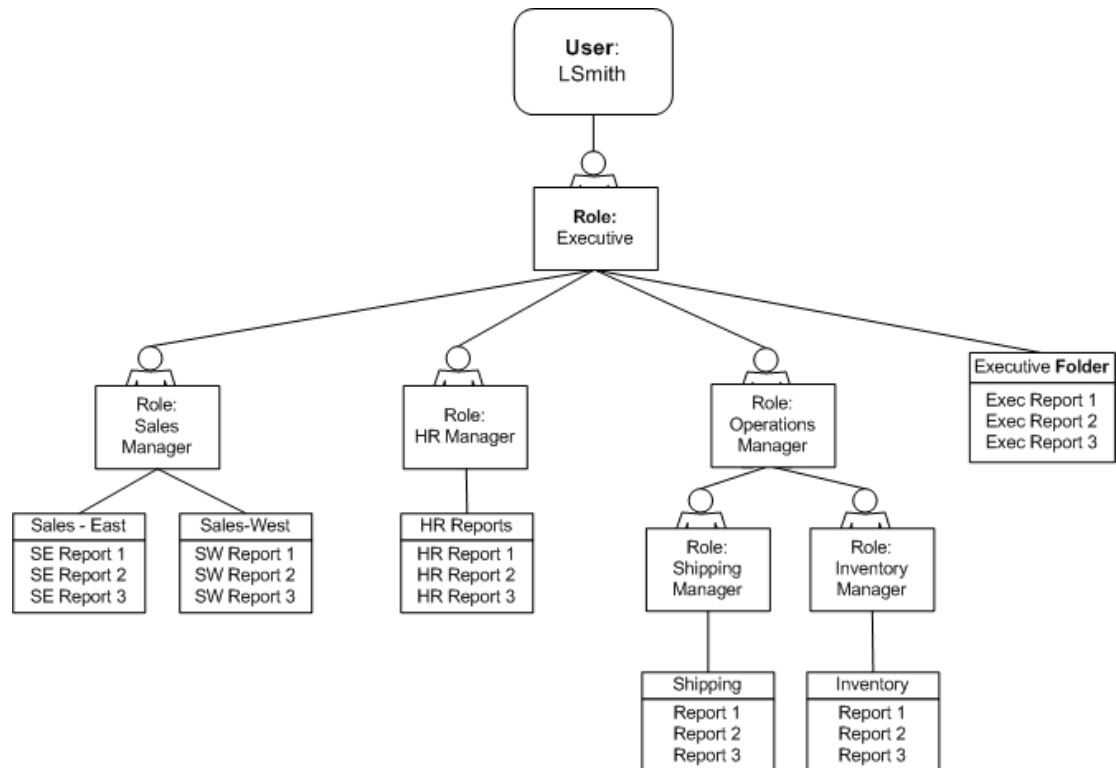
Understand Publisher Users, Roles, and Permissions

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. The figure below shows an example of the hierarchy structure of User, Role, and Folder.



Options for Configuring Users and Roles

There're three options for setting up users and roles.

- Set up users and roles in the Publisher Security Center

For this option, follow the instructions in this section.

- Configure Publisher with your LDAP server

For this option, see [Configure 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 [Integrate with Other Oracle Security Models](#).

About Privileges to Use Functionality

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.

The table below shows the privileges granted to each functional role.

Role	Privilege
Publisher Scheduler	View Export History Schedule
Publisher Template Designer	View Export History (public reports only) Enables access to Layout Editor Enables log on from Template Builder

Role	Privilege
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
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're also granted permissions on the catalog objects.

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.

The table below 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.

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 Publisher Developer role - Data Model — requires the Publisher Developer role - Sub Template and Style Template - requires the Publisher Developer Role or the 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.

For a report consumer to successfully run a report, the consumer's role must have read access to every object 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.

How Functional Privileges and Permissions Work Together

Certain rules determine 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

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 (Publisher Developer, 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.

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 Publisher except to run and view reports, then the roles for these users consist of catalog permissions only.

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 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 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 Publisher Developer role is granted Read access.

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 [Grant Data Access](#).

Configure Users, Roles, and Data Access

This chapter details the procedures to configure users, roles, and data access.

- [Create Roles](#)
- [Create Users and Assign Roles to a User](#)
- [Grant Catalog Permissions](#)
- [Grant Data Access](#)

Create Roles

You create roles on the Administration page.

To create a new role in Publisher:

1. Navigate to the 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 [Grant Data Access](#).

Create Users and Assign Roles to a User

You create users in the Administration page.

To create users and assign roles to them:

1. Navigate to the 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**.

Grant 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.

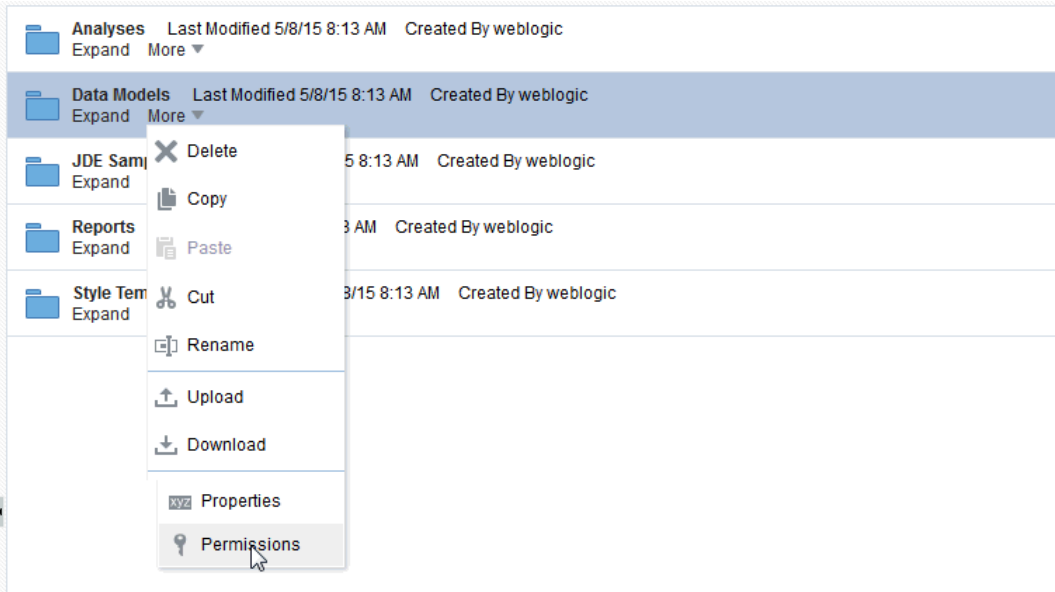
Permissions can be granted at the folder level and applied to all the objects and subfolders it contains, or applied to individual objects.

To grant catalog permissions to a role:

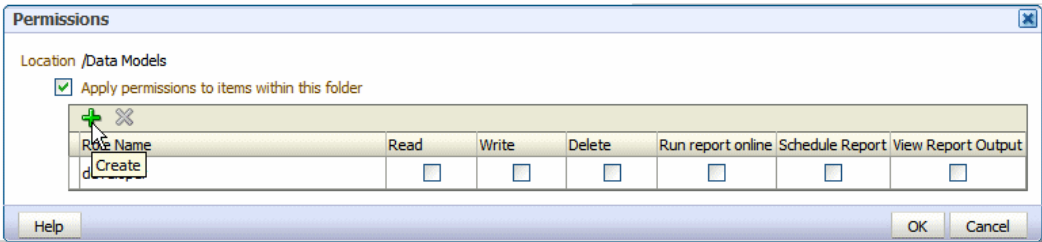
1. Navigate to the Catalog.
2. Locate the folder or object on which to grant permissions and click **More**. From the menu (shown in the figure below), select **Permissions**. Alternatively, you can select the folder and click **Permissions** in the Tasks region.

 Note:

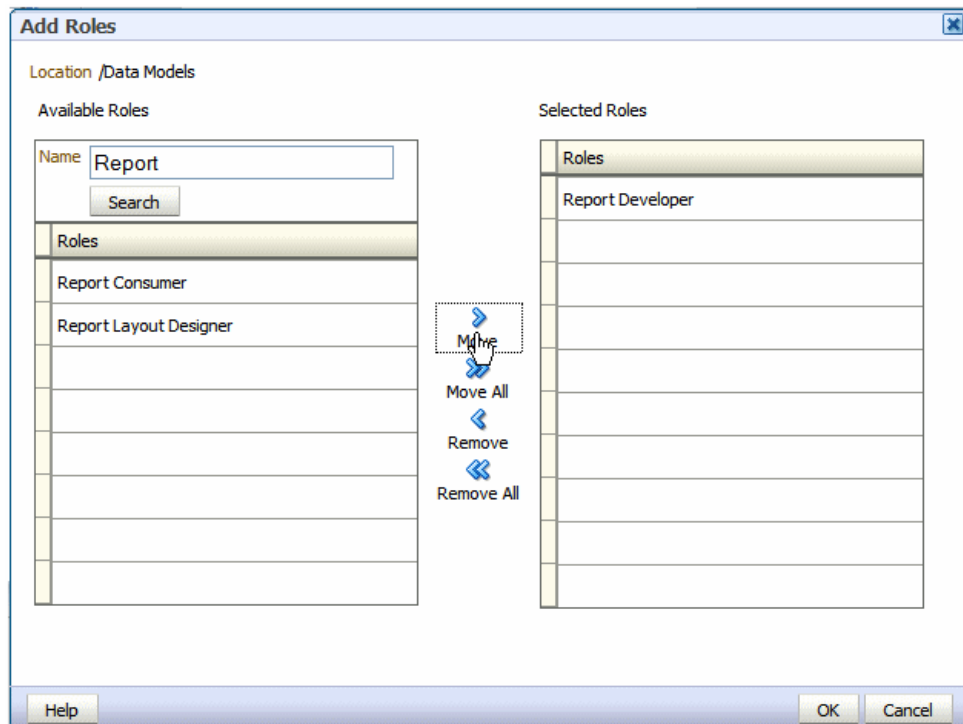
Permissions cannot be granted on the root Shared folder.



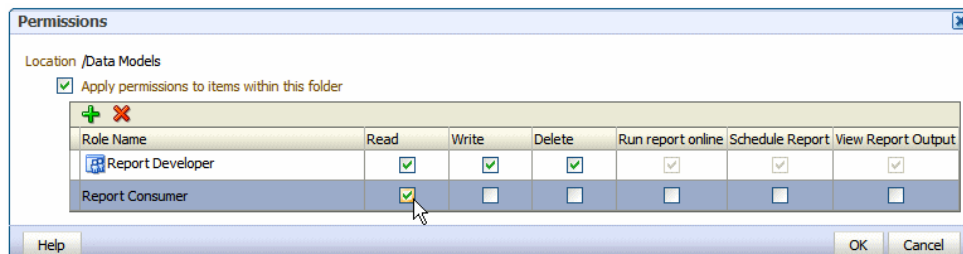
3. On the Permissions dialog, click **Create**.



4. On the Add Roles dialog, 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.



5. When finished, click **OK** to return to the Permissions dialog.
6. On the Permissions dialog, configure the permissions required by the role.



Note the following:

- The icon next to the Report Developer role indicates that this role is assigned one of the Publisher functional roles (in this case, the 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 Publisher Developer Role: Run report online, Scheduler Report, View Report Output.
7. If you're granting permissions on a Folder, select **Apply permissions to items within this folder**, if the permissions should apply to all objects.

Grant Data Access

Roles must be granted access to data sources to run or schedule certain reports or to create or edit certain data models.

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 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**.

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 the table below.

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.
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 the table below.

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 Publisher Developer Role and the 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 Publisher Developer Role, and the 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 Publisher Template Designer Role and the 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

Use LDAP with Publisher

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



Note:

By default, Publisher allows every LDAP user to log in to the system even when no 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 assigned permissions on catalog objects (such as traverse and open) the user can perform those tasks.

To prevent users from logging in to Publisher unless they have a Publisher role assigned, see [Disable Users Without Publisher-Specific Roles from Logging In](#).

- [Configure Publisher to Use an LDAP Provider for Authentication Only](#)
- [Configure Publisher to Use an LDAP Provider for Authentication and Authorization](#)

Configure Publisher to Use an LDAP Provider for Authentication Only

Configure Publisher to use an LDAP provider for authentication in conjunction with another security model for authorization.

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 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're 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, `orclguid`.
5. Click **Apply**.
6. Restart the Publisher server.

Configure Publisher to Use an LDAP Provider for Authentication and Authorization

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 Publisher server to access the LDAP server.

In the 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 Publisher with Oracle LDAP consists of three main tasks:

1. Set up users and roles in the LDAP provider
2. Configure Publisher to recognize the LDAP server
3. Assign catalog permissions and data access to roles

Set Up Users and Roles in the LDAP Provider

This procedure 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 to integrate with Publisher. See [Understand Publisher Users, Roles, and Permissions](#) for full descriptions of the required functional roles.
 - XMLP_ADMIN - The administrator role for Publisher. 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 Publisher from the Template Builder for Word and to upload and download templates. Allows users to design layouts using the 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 Publisher functional roles.
3. Assign roles to users.

 Note:

Ensure that you assign the Administrator account the XMLP_ADMIN role.

Configure Publisher to Recognize the LDAP Server

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

 Note:

Ensure that you understand your site's LDAP server configuration before entering values for the Publisher settings.

To configure Publisher 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 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're using LDAP over SSL, then note the following:
 - the protocol is "ldaps"
 - the default port is 636For 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=groupofuniquecnames)(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 Publisher server doesn't 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.

- **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

5. Click **Apply**. Restart Publisher.

The figure below shows a sample of the LDAP security model entry fields from the Security Configuration page.

The screenshot shows the 'Authorization' tab in a configuration interface. The 'Security Model' is set to 'LDAP'. The following fields are visible:

- URL:** ldap://hostname:port (Example: ldap://hostname:port)
- Administrator Username:** Admin
- Administrator Password:** [Masked]
- Distinguished Name for Users:** cn=Users,dc=example,dc=com (Example: cn=Users,dc=example,dc=com)
- Distinguished Name for Groups:** [Empty] (Example: null)
- Group Search Filter:** (&(objectclass=groupofuniquenames)(cn=*)) (Default Value: (&(objectclass=groupofuniquenames)(cn=*)))
- Group Attribute Name:** cn (Default Value: cn)
- Group Member Attribute Name:** uniquemember (Default Value: uniquemember)
- Member Of Group Attribute Name:** [Empty] (Optional) Please 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, wlsMemberOf
- Group Description Attribute Name:** description (Default Value: description)
- JNDI Context Factory Class:** com.sun.jndi.ldap.LdapCtxFactory (Default Value: com.sun.jndi.ldap.LdapCtxFactory)
- Group Retrieval Page Size:** 200 (Page size feature is not supported by all LDAP servers)
- Attribute used for RDN:** cn (Default Value: cn)
- Automatically clear LDAP cache:** ☒ (Enabled)
- Ldap Cache Interval:** 1 (Please enter value that is greater than or equal to 1)
- Ldap Cache Interval Unit:** Hour (Dropdown menu)
- Default User Group Name:** [Empty] (Optional) Please enter a user group name that is added to all authenticated users
- Attribute Names for Data Query Bind Variables:** [Empty] (Optional) Please enter ldap attribute names separated by commas that are used as bind variables for data query

If you're configuring Publisher to use LDAP over SSL, then you must also configure Java keystore to add the server certificate to JVM. See [Configure Publisher for Secure Socket Layer \(SSL\) Communication](#).

Assign Data Access and Catalog Permissions to Roles

Assign data access and catalog permissions to roles in the Administration page.

To assign data access and catalog permissions to roles:

1. Log in to 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 Publisher interface, except for adding data sources.
3. Click **Add Data Sources** to add 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 [Grant Data Access](#).
 4. Grant catalog permissions to roles. See [About Catalog Permissions](#) and [Grant Catalog Permissions](#) for details on granting catalog permissions to roles.

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

Disable Users Without Publisher-Specific Roles from Logging In

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

The xmlp-server-config.xml file is located at:

```
$DOMAIN_HOME/config/fmwconfig/biconfig/bipublisher/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"/>
```

Integrate with Microsoft Active Directory

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

- [Configure the Active Directory](#)
- [Configure Publisher](#)
- [Log In to Publisher Using the Active Directory Credentials](#)
- [Assign Data Access and Catalog Permissions to Roles](#)

Configure the Active Directory

Configure support for Active Directory by adding users and system groups.

To configure the active directory:

1. Add users who must access Publisher.
Add the users under "Users" or any other organization unit in the Domain Root.
2. Add the Publisher system groups. The Scope of the groups must be Domain Local.
 - XMLP_ADMIN - The administrator role for Publisher. 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 Publisher from the Template Builder for Word and to upload and download templates. Allows users to design layouts using the Publisher Layout Editor.
3. Grant Publisher system groups to global groups or users.
You can grant Publisher system groups directly to users or through global groups.

Example 1: Grant Users the 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 Publisher Administrator privileges.

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.

Configure Publisher

You configure Publisher on the Administration page.

To configure 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 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 [Configure Publisher 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 you're 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/`

The figure below shows an example configuration highlighting the recommendations stated above.

The screenshot shows the 'Authorization' configuration window with the 'Security Model' set to 'LDAP'. The following fields are visible:

- URL:** `ldaps://172.16.237.22:389` (Example: `ldaps://hostname:port`)
- Administrator Username:** `CN=bi_admin_user,CN=Users,DC=hostname,DC=domainname,DC=com`
- Administrator Password:** `*****`
- Distinguished Name for Users:** `DC=hostname,DC=domainname,DC=com` (Example: `cn=Users,dc=example,dc=com`)
- Distinguished Name for Groups:** `DC=hostname,DC=domainname,DC=com` (Example: `null`)
- Group Search Filter:** `(&(objectclass=group)(cn=*))` (Default Value: `(&(objectclass=groupofuniquenames)(cn=*))`)
- Group Attribute Name:** `cn` (Default Value: `cn`)
- Group Member Attribute Name:** `memberOf` (Default Value: `uniquemember`)
- Member Of Group Attribute Name:** `memberOf` (Optional) Please 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, wlsMemberOf`
- Group Description Attribute Name:** `description` (Default Value: `description`)
- JNDI Context Factory Class:** `com.sun.jndi.ldap.LdapCtxFactory` (Default Value: `com.sun.jndi.ldap.LdapCtxFactory`)
- Group Retrieval Page Size:** `1` (Page size feature is not supported by all LDAP servers)
- Attribute used for Login Username:** `sAMAccountName` (Default Value: `cn`)
- Ldap Cache Interval:** `1` (Please enter value that is greater than or equal to 1)
- Ldap Cache Interval Unit:** `Hour`
- Default User Group Name:** `memberOf,sAMAccountName,primaryGroupID,mail` (Optional) Please enter a user group name that is added to all authenticated users
- Attribute Names for Data Query Bind Variables:** `memberOf,sAMAccountName,primaryGroupID,mail` (Optional) Please enter ldap attribute names separated by commas that are used as bind variables for data query

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

If you're configuring Publisher to use LDAP over SSL, then you must also configure Java keystore to add the server certificate to JVM. For more information, see [Configure Publisher for Secure Socket Layer \(SSL\) Communication](#).

Log In to Publisher Using the Active Directory Credentials

The User login name defined in **Active Directory Users and Computers >User Properties >Account** is used for the Publisher login name.

Add the Domain to the user name to log in to 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 Publisher login username.
- User names must be unique across all organization units.

Assign Data Access and Catalog Permissions to Roles

You assign data access and catalog permissions to roles on the Administration page.

To assign data access and catalog permissions to roles:

1. Log in to 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 Publisher interface, except for adding data sources.
3. Click **Add Data Sources** to add 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 [Grant Data Access](#).
 4. Grant catalog permissions to roles. See [About Catalog Permissions](#) and [Grant Catalog Permissions](#) for details on granting catalog permissions to roles.

Configure 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. Publisher can be configured to trust incoming HTTP requests authenticated by a SSO solution configured for use with Oracle Fusion Middleware and Oracle WebLogic Server. For information about configuring SSO for Oracle Fusion Middleware, see *Securing Applications with Oracle Platform Security Services*.

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

Configuring Publisher to work with SSO authentication requires that:

- Oracle Fusion Middleware and Oracle WebLogic Server are configured to accept SSO authentication. Oracle Access Manager is recommended in production environments.
- 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.

How Publisher Operates with SSO Authentication

After SSO authorization has been implemented, 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.

Tasks for Setting Up SSO Authentication with Publisher

Refer to the table below for SSO authentication configuration tasks and links providing more information.

Task	Description	For More Information
Configure Oracle Access Manager as the SSO authentication provider.	Configure Oracle Access Manager to protect the Publisher URL entry points.	Configure SSO in an Oracle Access Manager Environment <i>See Securing Applications with Oracle Platform Security Services</i>
Configure the HTTP proxy.	Configure the web proxy to forward requests from Publisher to the SSO provider.	
Configure a new authenticator for Oracle WebLogic Server.	Configure the Oracle WebLogic Server domain in which Publisher is installed to use the new identity store.	Configure a New Authenticator for Oracle WebLogic Server <i>See Oracle WebLogic Server Administration Console Online Help</i>
Configure a new identityasserter for Oracle WebLogic Server.	Configure the Oracle WebLogic Server domain in which Publisher is installed to use the SSO provider as an assserter.	
Enable Publisher to accept SSO authentication.	Enable the SSO provider configured to work with Publisher.	Configure Publisher for Oracle Fusion Middleware Security

Configure SSO in an Oracle Access Manager Environment

Configure Oracle Access Manager as the SSO authentication provider for Oracle Fusion Middleware with WebLogic Server.

See Securing Applications with Oracle Platform Security Services .

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

- Configure the SSO provider to protect the Publisher URL entry points.
- Configure the web server to forward requests from Publisher to the SSO provider.
- Configure the new identity store as the main authentication source for the Oracle WebLogic Server domain in which Publisher has been installed. For more information, see [Configure a New Authenticator for Oracle WebLogic Server](#).

- Configure the Oracle Access Manager domain in which Publisher is installed to use an Oracle Access Manager asserter. For more information, see [Configure OAM as a New Identity Asserter for Oracle WebLogic Server](#).
- After configuration of the SSO environment is complete, enable SSO authentication for Publisher. For more information, see [Configure Publisher for Oracle Fusion Middleware Security](#).

Configure a New Authenticator for Oracle WebLogic Server

After installing 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 Publisher is installed).

For more information about configuring authentication providers in Oracle WebLogic Server, see *Administering Security for 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**, select the Configuration\Commontab, then select **SUFFICIENT** from the **Control Flag** list, and click **Save**.
 7. Display the Provider Specific tab and specify the following connection, users, groups, and general settings using appropriate values for your environment:
 - **Host** - The LDAP host name. For example, <localhost>.
 - **Port** - The LDAP host listening port number. For example, 6050.
 - **Principal** - The distinguished name (DN) of the user that connects to the LDAP server. For example, cn=orcladmin.
 - **Credential** - The password for the LDAP administrative user entered as the Principal.
 - **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.
 - **All Users Filter** - The LDAP search filter. For example, (&(uid=*)(objectclass=person)). The asterisk (*) filters for all users. Click More Info... for details.

- **User From Name Filter** - The LDAP search filter. Click More Info... for details.
 - **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're using in the authentication provider.
 - **Group Base DN** - The base distinguished name (DN) of the LDAP server tree that contains groups (same as User Base DN).
 - **GUID attribute** - The attribute used to define object GUIDs in LDAP. orclguid
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, and 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. In the **Providers** tab, click **Reorder** to display the Reorder Authentication Providers page.
 - b. 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)
 - c. Click **OK** to save your changes.
 11. In the Change Center, click **Activate Changes**.
 12. Restart Oracle WebLogic Server.

Configure OAM as a New Identity Asserter for Oracle WebLogic Server

The Oracle WebLogic Server domain in which 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 WebLogic Server Administration Console Online Help*.

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

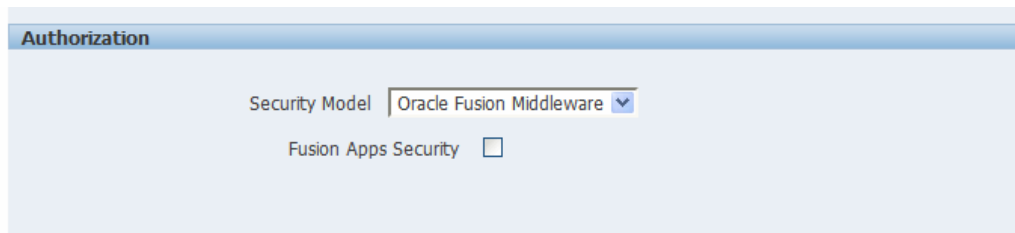
1. Log in to Oracle WebLogic Server Administration Console.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you're configuring. For example, **myrealm**. Select **Providers**.
3. Click **New**. Complete the fields as follows:
 - **Name:** *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.

Configure Publisher for Oracle Fusion Middleware Security

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



Set Up Oracle Single Sign-On

Set up Oracle Single Sign-On in the Identity Store Configuration page.

To set up Oracle Single Sign-On, first configure WebLogic Server using the instructions in *Administering Security for Oracle WebLogic Server*. Publisher must be configured to use Oracle Internet Directory as the default LDAP server.

**Note:**

When using Oracle SSO, 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 Publisher assumes the value of first cn= is the login user name (that is, "admin" in this case).

Therefore if your Osso-User-Dn doesn't 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).

Setup Procedure

You set up SSO in the `mod_osso.conf` file.

To set up SSO:

1. Modify the application server configuration file to protect the `xmlpservlet`. See *Securing Applications with Oracle Platform Security Services*.
2. In the `mod_osso.conf` add a new "Location" directive as follows:

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

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

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

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

```
Location /xmlpservlet/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 Publisher. 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 Publisher, set up the Single Sign-Off URL on the 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.

7. Create a Publisher Local Superuser to ensure access to Publisher regardless of your selected security configuration. See [Enable a Local Superuser](#) for more information.

8. Click **Apply**.

9. Restart the application through the Oracle Fusion Middleware Control page.

10. Enter the URL to access the Publisher application, and you're redirected to the SSO login page.

4

Other Security Topics

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

Topics:

- [Enable a Local Superuser](#)
- [Enable a Guest User](#)
- [Configure Publisher for Secure Socket Layer \(SSL\) Communication](#)
- [Enable Secure Cookies](#)
- [Configure Proxy Settings](#)
- [Restrict Embedding of Publisher in iframes](#)

Enable a Local Superuser

Publisher enables you to define an administration Superuser.

Using the Superuser credentials you can directly access the 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. Catalog operations are not available to a Superuser, if Publisher is configured to use Oracle Analytics Server catalog,

1. Click **Administration**.
2. Under Security Center, click **Security Configuration**.
3. Under Local Superuser, select the box and enter the credentials for the Superuser.
4. Restart the Publisher.

Enable a Guest User

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 Publisher uses a shared catalog or is installed with Oracle Analytics Server.

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.

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

The report must reference the data model 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.

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

Configure Publisher for Secure Socket Layer (SSL) Communication

It's recommended that you enable Secure Socket Layer (HTTPS) on the middle tier hosting the Web services because the trusted username/password that's passed can be intercepted.

This also applies to Web services that are used for communication between Publisher and Oracle BI Presentation Services.

- [Import Certificates for Web Services Protected by SSL](#)
- [Add the Virtualize Property to the Identity Store Configuration](#)
- [Update the JDBC Connection String to the Data Source](#)
- [Update the JMS Configuration](#)
- [Configure the Delivery Manager](#)

Import 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's running Publisher.

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 your browser type.
3. Install the Certificate into your keystore using the Java keytool, as follows:

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

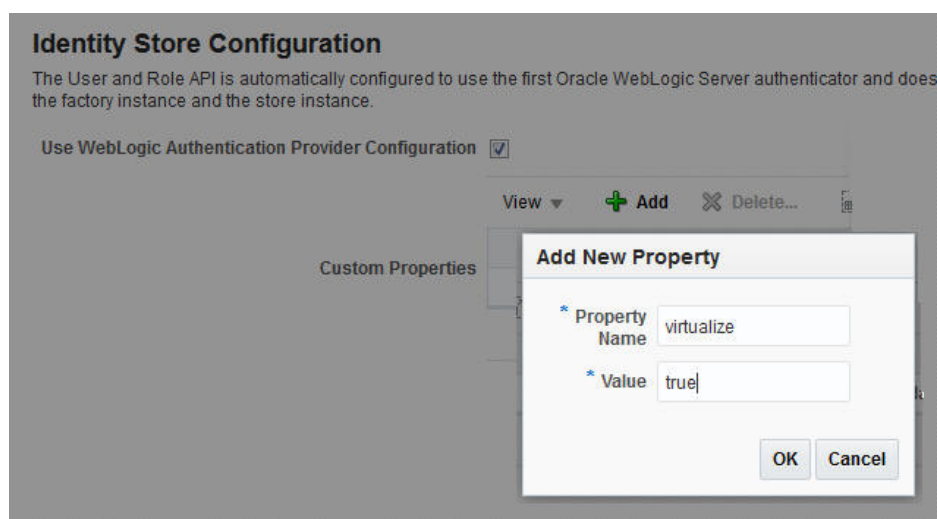
4. Restart the application server.

These steps shouldn't 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.

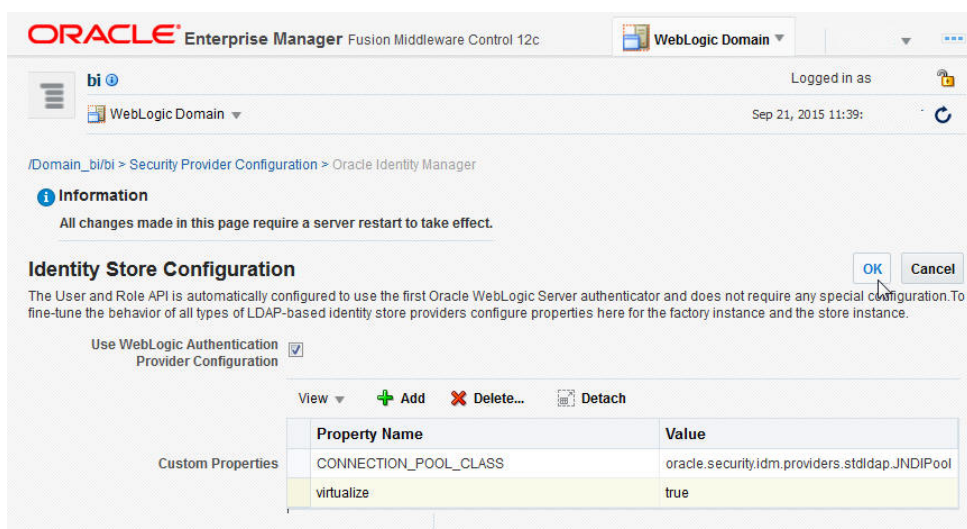
Add the Virtualize Property to the Identity Store Configuration

You must add the property "virtualize" to the Identity Store Configuration in Fusion Middleware Control to enable SSL for Publisher.

1. Log in to Fusion Middleware Control:
`https://<Host>/<SecureAdminPort>/em`
2. Select **WebLogic Domain, Security**, and then **Security Provider Configuration**.
3. Expand the **Security Store Provider** segment.
4. Expand the **Identity Store Provider** segment.
5. Click **Configure**.
 - a. Click **Add (+)** to add a new property.
 - b. In the Add New Property dialog, enter
Property Name — **virtualize**
Value — **true**



6. On the Identity Store Provide page, click **OK**.



7. Confirm that the property is added to the `jps-config.xml` file:
 - a. Open the `jps-config.xml` file located in
`<DomainHome>/config/fmwconfig/jps-config.xml`
 - b. Ensure that the file contains the line:
`<property name="virtualize" value="true"/>`

Update the JDBC Connection String to the Data Source

For Publisher to connect to Oracle Analytics Server as a data source when SSL is enabled, you must update the default connection string.

Follow the guidelines detailed in [Set Up a JDBC Connection to the Oracle BI Server](#).

Update the JMS Configuration

You update the Scheduler JMS configuration to use the SSL URL.

1. On the Publisher Administration page, under System Maintenance, click **Scheduler Configuration**.
2. Update the WebLogic JNDI URL to use SSL. For example,

Administration > Scheduler Configuration

System Maintenance

Server Configuration Scheduler Configuration Scheduler Diagnostics Report Viewer Configuration Manage Cache

Apply Cancel

JMS Configuration

JMS Provider WebLogic

WebLogic JNDI URL cluster:t3s://bi cluster

Threads Per JMS Processor 5

Shared Directory

Test JMS

3. Click **Apply**.
4. Select the **Scheduler Diagnostics** tab.
5. Verify that the connection passed diagnostics.

Configure the Delivery Manager

If you want to use the default certificates built-in with 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, as described in [Configure Delivery Options](#). A self-signed certificate means that the certificate is signed by a non-trusted certificate authority (usually the user).

Enable Secure Cookies

The cookie-secure flag tells the Web browser to only send the cookie back over an HTTPS connection.

This ensures that the cookie is transmitted only on a secure channel. HTTPS must be enabled for the URL exposed by the application.

To enable the cookie-secure flag, you update the `weblogic.xml` within the `xmlpserver.war` file (within the `xmlpserver.ear`).

1. Locate the `xmlpserver.ear` file under `ORACLE_HOME/bifoundation/jee/`
2. Unpack the `xmlpserver.ear` file.
3. Unpack the `xmlpserver.war` file.
4. Back up the `WEB-INF/weblogic.xml` file.

5. Open the `WEB-INF/weblogic.xml` file.
6. Add the following attributes to the `<wls:session-descriptor>`:

```
<wls:cookie-secure>true</wls:cookie-secure>
<wls:url-rewriting-enabled>false</wls:url-rewriting-enabled>
```

Example:

```
<?xml version = '1.0' encoding = 'US-ASCII'?>
<wls:weblogic-web-app
xmlns:wls="http://xmlns.oracle.com/weblogic/weblogic-web-app"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/ejb-jar_3_0.xsd
http://xmlns.oracle.com/weblogic/weblogic-web-app
http://xmlns.oracle.com/weblogic/weblogic-web-app/1.2/weblogic-web-
app.xsd">
  <wls:session-descriptor>
    <wls:cookie-path>/xmlpserver</wls:cookie-path>
    <wls:cookie-secure>true</wls:cookie-secure>
    <wls:url-rewriting-enabled>false</wls:url-rewriting-enabled>
  </wls:session-descriptor>
  <wls:context-root>xmlpserver</wls:context-root>
  <wls:library-ref>
...

```

7. Repack the `xmlpserver.war` file.
8. Repack the `xmlpserver.ear` file.
9. Go to your WebLogic Server console and update the bipublisher deployment.

Configure Proxy Settings

To use external Web Services or HTTP data sources when the 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 Publisher must connect to directly (not through the proxy) for example, the Oracle Analytics Server 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, you update the WebLogic `setDomainEnv` script.

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.MessageFactoryImpl  
  
-Djavax.xml.soap.SOAPFactory=oracle.j2ee.ws.saaj.SOAPFactoryImpl -  
Djavax.xml.soap.SOAPConnectionFactory=oracle.j2ee.ws.saaj.client.p2p.HttpS  
OAPConnectionFactory ${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

Restrict Embedding of Publisher in iframes

You can prevent embedding of Publisher in iframes.

Configuring Embedding of Publisher

By default, users can embed Publisher in an iframe only if the iframe and Publisher are in the same domain.

If you want to allow embedding of Publisher in an iframe belonging to another domain or you want to completely restrict embedding of Publisher in an iframe, provide appropriate values for the `X_FRAME_OPTIONS` and `FRAME_ANCESTORS` properties in the `xmlp-server-config.xml` file.



Note:

If you set `X_FRAME_OPTIONS` to `Deny` and `FRAME_ANCESTORS` to `none`, you can't access the user interface of Publisher from other products that can embed Publisher, including Oracle Analytics Server. If you specify the values for both `X_FRAME_OPTIONS` and `FRAME_ANCESTORS`, the value used depends on the browser. Make sure you provide similar values to `X_FRAME_OPTIONS` and `FRAME_ANCESTORS` to ensure consistent behavior across browsers.

X_FRAME_OPTIONS Values

Value	Specifies
False	Do not set the header option.
Deny	Do not allow users to embed Publisher in iframes.
SameOrigin	Allow users to embed Publisher in iframes of the same domain. This is the default.
Allow-From <i>url</i>	Allow users to embed Publisher only from the domain specified in the <i>url</i> parameter.

FRAME_ANCESTORS Values

Value	Specifies
False	Do not set the header option.
none	Do not allow users to embed Publisher in iframes.
self	Allow users to embed Publisher in iframes of the same domain. This is the default.
<i>url</i>	Allow users to embed Publisher only from the domain specified in the <i>url</i> parameter. The URL can be repeated and can be specified in more than one format.

5

Integrate with Other Oracle Security Models

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

Topics:

- [Integrate with Other Oracle Security Models](#)
- [Before You Begin: Create a Local Superuser](#)
- [Integrate with Oracle BI Server Security](#)
- [Integrate with Oracle E-Business Suite](#)
- [Integrate with Oracle Database Security](#)
- [Integrate with Oracle Siebel CRM Security](#)

Integrate with Other Oracle Security Models

This chapter describes how to integrate Publisher with other Oracle product security models.

In most cases you must first define the Publisher functional roles in the other Oracle product and then configure 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 [Alternative Security Options](#).

For conceptual information regarding Publisher roles and permissions, see [Understand Publisher Users, Roles, and Permissions](#).

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 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**.
3. Enter a name and password for your superuser.
4. Restart Publisher to activate the superuser in the system.

Integrate with Oracle BI Server Security

If you installed Publisher as part of the Oracle Analytics Server and you configured Oracle Analytics Server to use legacy Oracle BI Server authentication, then follow the procedures below to configure Publisher to use BI Server security.

- [Configure Publisher for Oracle BI Server Security](#)
- [Add Data Sources to BI Server Roles](#)

The Oracle BI Server security option is for customers who want to use legacy 10g authentication. This information doesn't apply to you if you configured Oracle Fusion Middleware Security.

These procedures assume that you performed the configuration required in the BI Server.

Configure Publisher for Oracle BI Server Security

You configure Publisher for BI Server Security on the Administration page.

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

 Note:

To log in directly to the Publisher server, use the login URL with the /xmlpserver suffix, for example: `http://example.com:9502/xmlpserver`

2. In the Authorization region of the page, select **Oracle BI Server** from the Security Model list. Provide the following connection information for the Oracle BI Server:
 - **JDBC Connection String** — Example: `jdbc:oraclebi://host:port/`
If you don't 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 Analytics Server server and copy the connection string. If this has not been configured, then see [Set Up a JDBC Connection to the Oracle BI Server](#).
 - **Database Driver Class** — Example: `oracle.bi.jdbc.AnaJdbcDriver`
3. Click **Apply**. Restart Publisher for the security changes to take effect.

Add Data Sources to BI Server Roles

Add data sources to BI server roles from the Administration page.

1. Log in to Oracle Business Intelligence as an administrator.
2. On the global header click **Administration**. On the Oracle Business Intelligence Administration page, click **Manage Publisher**.

3. On the Publisher Administration page, click **Roles and Permissions**. The groups to which you assigned the 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 the above steps for all roles that need access to report data sources.

Integrate with Oracle E-Business Suite

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

When users log in, they're 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 Publisher security center. You can then add Publisher catalog permissions and data access privileges to the imported roles/responsibilities. See [Understand Publisher Users, Roles, and Permissions](#).

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

- [Configure Publisher to Use E-Business Suite Security](#)
- [Add Data Sources to the E-Business Suite Roles](#)
- [Grant Catalog Permissions to the E-Business Suite Roles](#)



Note:

Users cannot access or run reports that are stored on the E-Business Suite instance. Reports must reside in the Publisher catalog. The E-Business Suite data security is enforced when Publisher connects to the E-Business Suite data tables to retrieve the report data.

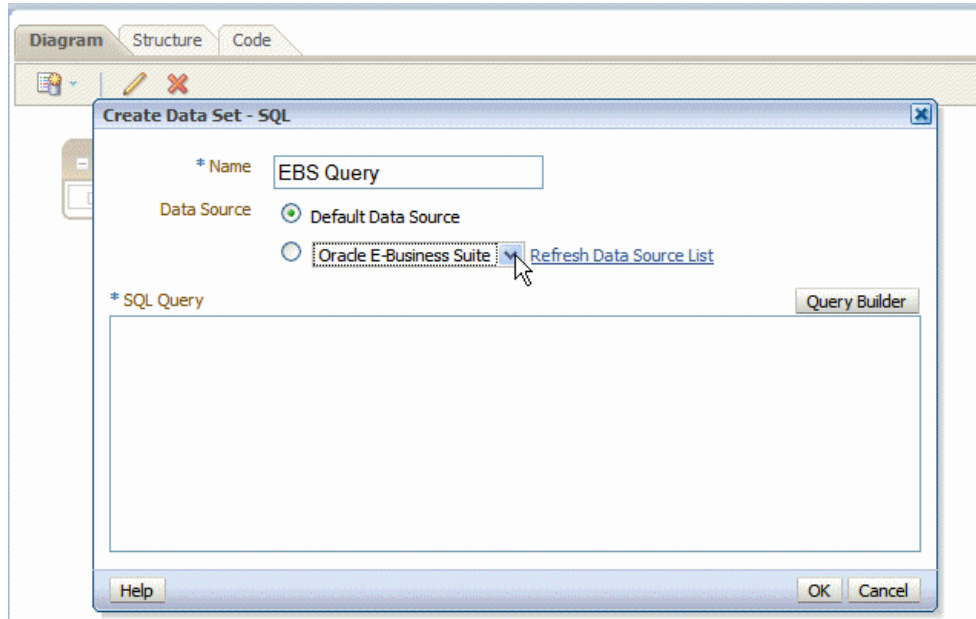
Publisher relies on information stored in the DBC file to connect to the E-Business Suite instance. Ensure that you can locate and have access to this file. The DBC file is typically located under the \$FND_SECURE directory.

Features of the Integration with E-Business Suite Security

When Publisher is integrated with E-Business Suite security, certain features are enabled.

- When users log in to Publisher using their E-Business Suite credentials, they're prompted to choose a responsibility.

- 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 below.



Configure Publisher to Use E-Business Suite Security

You configure Publisher for E-Business Suite Security on the Administration page.

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

Note:

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

3. Log in to 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 Publisher should access the E-Business Suite instance.

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

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

Add 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.

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.

Grant Catalog Permissions to the E-Business Suite Roles

For a role to access objects in a folder, you 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.

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 **Create** 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. To apply the selections to all items within a folder, select **Apply permissions to items within this folder**.

Integrate with Oracle Database Security

Publisher offers integration with Oracle Database security to enable you to administer the Publisher users with your Oracle Database users.

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

- [Define the Publisher Functional Roles in the Oracle Database](#)
- [Add Data Sources to Roles](#)
- [Grant Catalog Permissions to Roles](#)



Note:

For information on setting up Oracle Database security, see *Oracle Database Security Guide*.

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

Define the Publisher Functional Roles in the Oracle Database

You can create roles in the Oracle database that correspond to Publisher functional roles.

1. In the Oracle Database, create the following roles to correspond to the Publisher functional roles:
 - XMLP_ADMIN — Serve as the administrator role for the 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 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 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. Assign the XMLP_ADMIN role to a user with administration privileges, such as SYSTEM.
4. Log in to Publisher with Administrator privileges. On 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 Publisher for the security changes to take effect.

Add 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.

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.

Grant 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.

1. In the catalog, navigate to a catalog object that's 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. To apply the selections to all items within a folder, select **Apply permissions to items within this folder**.

Integrate with Oracle Siebel CRM Security

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

- [Set Up Publisher Roles as Siebel CRM Responsibilities](#)
- [Configure Publisher to Use Siebel Security](#)
- [Add Data Sources to Roles](#)
- [Grant Catalog Permissions to Roles](#)

Set Up Publisher Roles as Siebel CRM Responsibilities

After setting up Publisher Roles as Siebel CRM Responsibilities, 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 Publisher.

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 Publisher functional roles:
 - XMLP_ADMIN — Serves as the administrator role for the 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 Publisher 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 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.

Configure Publisher to Use Siebel Security

You configure Publisher to use Siebel Security on the Administration page.

1. Log in to Publisher with Administrator privileges. On 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 Publisher for the security changes to take effect.

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

Add 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.

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.

Grant 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.

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

Configure System Maintenance Properties

This topic describes how to configure Publisher server properties.

Topics:

- [Set Server Caching Specifications](#)
- [Set Retry Properties for Database Failover](#)
- [Set Report Viewer Properties](#)
- [Clear Report Objects from the Server Cache](#)
- [Clear the Subject Area Metadata Cache](#)
- [Purge Job Diagnostic Logs](#)
- [Purge Job History](#)

Set Server Caching Specifications

When Publisher processes a report, the data and the report document are stored in cache.

Report-specific caching of data sets can be set as a report property.

To configure caching at the server level:

1. In the Server Configuration page, set the following properties:
 - **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.
 - **Maximum Cached Report Definitions** — Enter the maximum number of report definitions to maintain in cache. The default is 50.
2. To manually purge this cache, on the Manage Cache tab, click **Clear Object Cache**.

Set Retry Properties for Database Failover

If Publisher fails to connect to a data source through the defined JDBC or JNDI connection, Publisher switches to the backup database.

The following 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.

Set Report Viewer Properties

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

If **Show Apply Button** is set to True, reports with parameter options display the **Apply** button in the report viewer. If you change the parameter values, click **Apply** to render the report with the new values.

If **Show Apply Button** is set to False, the report viewer doesn't display the **Apply** button. If you enter a new parameter value, Publisher automatically renders the report after the new value is selected or entered.

You set this property at the report level to override the system setting.

Clear Report Objects from the Server Cache

Use the Manage Cache page to clear the server cache.

The server cache stores report definitions, report data, and report output documents. If you need to manually purge this cache (for example, after patching) use the Manage Cache page.

To clear the report objects from the server cache:

1. From the Administration page, select **Manage Cache**.
2. On the Manage Cache page, click **Clear Object Cache**.

Clear the Subject Area Metadata Cache

You can clear the subject area metadata cache.

BI subject area metadata such as the dimension and measure names are cached at the server to quickly open the report in report designer. You can manually clear this cache if the BI subject area is updated through a binary repository (.RPD) file.

To clear the subject area metadata cache:

1. From the Administration page, select **Manage Cache**.
2. On the Manage Cache page, in the Clearing Subject Area Metadata Cache section, click **Clear Metadata Cache**.

Enable Diagnostics

This topic describes tasks performed by administrators and BI Authors to enable the diagnostics logs.

Topics:

- [Enable Diagnostics for Scheduler Jobs](#)
- [Enable Diagnostics for Online Reports](#)

Enable Diagnostics for Scheduler Jobs

You can enable diagnostics for a scheduler job in the **Schedule Report Job** page, and download the diagnostic logs from **Report Job History**.

You must have BI Administrator or BI Data Model Developer privileges to access the **Diagnostics** tab in the **Schedule Report Job** page. Perform the following steps to enable diagnostics.

To enable and download diagnostics for a scheduler job:

1. From the **New** menu, select **Report Job**.
2. Select the report to schedule, and click the **Diagnostics** tab.
3. Select and enable the required diagnostics.
 - Select **Enable SQL Explain Plan** to generate a diagnostic log with Explain plan/SQL monitor report information.
 - Select **Enable Data Engine Diagnostic** to generate a data processor log.
 - Select **Enable Report Processor Diagnostic** to generate FO (Formatting Options) and server related log information.
 - Select **Enable Consolidated Job Diagnostic** to generate the entire log, which includes scheduler log, data processor log, FO and server log details.
4. Submit the report.
5. After the report job runs, in the Report Job History page, select your report to view the details.
6. Under Output & Delivery, click **Diagnostic Log** to download the job diagnostic log and view the details.

Use the Manage Job Diagnostics Log page to purge the old job diagnostic logs.

Enable Diagnostics for Online Reports

In the Report Viewer, you can enable diagnostics for online reports.

Administrators and BI Authors can enable diagnostics before running the online report, and then download the diagnostic logs after the report finishes. Diagnostics are disabled by default.

If you enable diagnostics for an online report with interactive output, you can:

- Download the following diagnostic logs in a .zip file:
 - SQL logs
 - Data engine logs
 - Report Processor logs
- View the following details in the diagnostic logs:
 - Exceptions
 - Memory guard limits
 - SQL query

To enable diagnostics and download the diagnostic logs for an online report:

1. If the report is running, click **Cancel** to stop the reporting process.
2. Click **Actions** in the Report Viewer.
3. Select **Enable Diagnostics** from the **Online Diagnostics** option.
4. Submit the report.
5. To download the diagnostic logs after the report runs:
 - a. Click **Actions** in the Report Viewer.
 - b. Select **Download Diagnostics** from the **Online Diagnostics** option.

Purge Job Diagnostic Logs

You can purge old diagnostic logs to increase the available space on your system.

The retention period of job diagnostic logs is set to 30 days, by default. If you frequently enable diagnostic logs, these diagnostic logs might consume space in the database, and you might need to periodically free the space consumed by the old diagnostic logs. You can manually purge the job diagnostic logs older than the retention period .

To purge the job diagnostic logs:

1. On the Administration page, under System Maintenance, select **Manage Job Diagnostics Log**.
2. Click **Purge log beyond retention period**.

Purge Job History

Use the Manage Job Diagnostics Log page to purge old job history.

The retention period of a job history is set to 180 days, by default. You can manually purge the history of jobs that are older than the retention period. When you purge old job history, the saved output, saved XML, job delivery info, and the job status details of the old jobs are deleted.

To purge old job history:

1. On the Administration page, under System Maintenance, select **Manage Job Diagnostics Log**.
2. Click **Purge scheduler metadata**.

7

Configure the Scheduler

This topic describes the features, architecture, diagnostics, and configuration of the scheduler.

Topics:

- [Understand the Scheduler](#)
- [About the Scheduler Configuration](#)
- [Configure Processors and Processor Threads](#)
- [Scheduler Diagnostics](#)

Understand the Scheduler

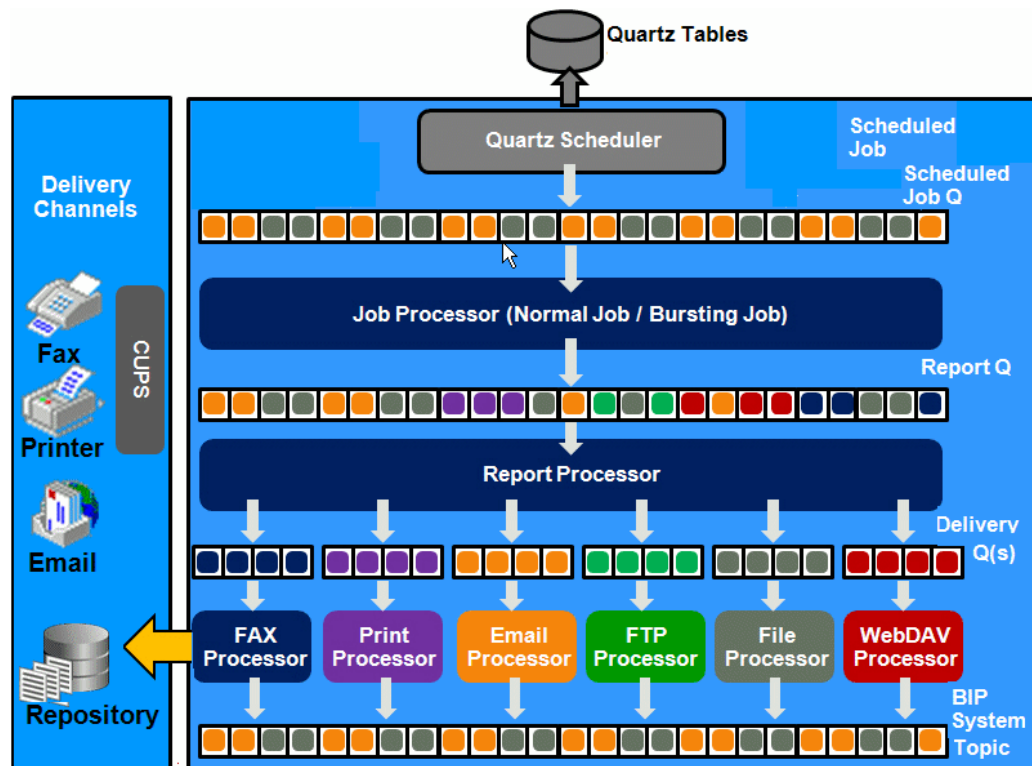
The updated architecture of the Scheduler uses the Java Messaging Service (JMS) queue technology.

This architecture enables you to add multiple publishing servers to a cluster and then dedicate each server to a particular function: report generation, document generation, or specific delivery channels.

Architecture

The architecture of the Scheduler uses JMS queues and topics to provide a highly scalable, highly performing and robust report scheduling and delivery system.

The figure below displays the 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

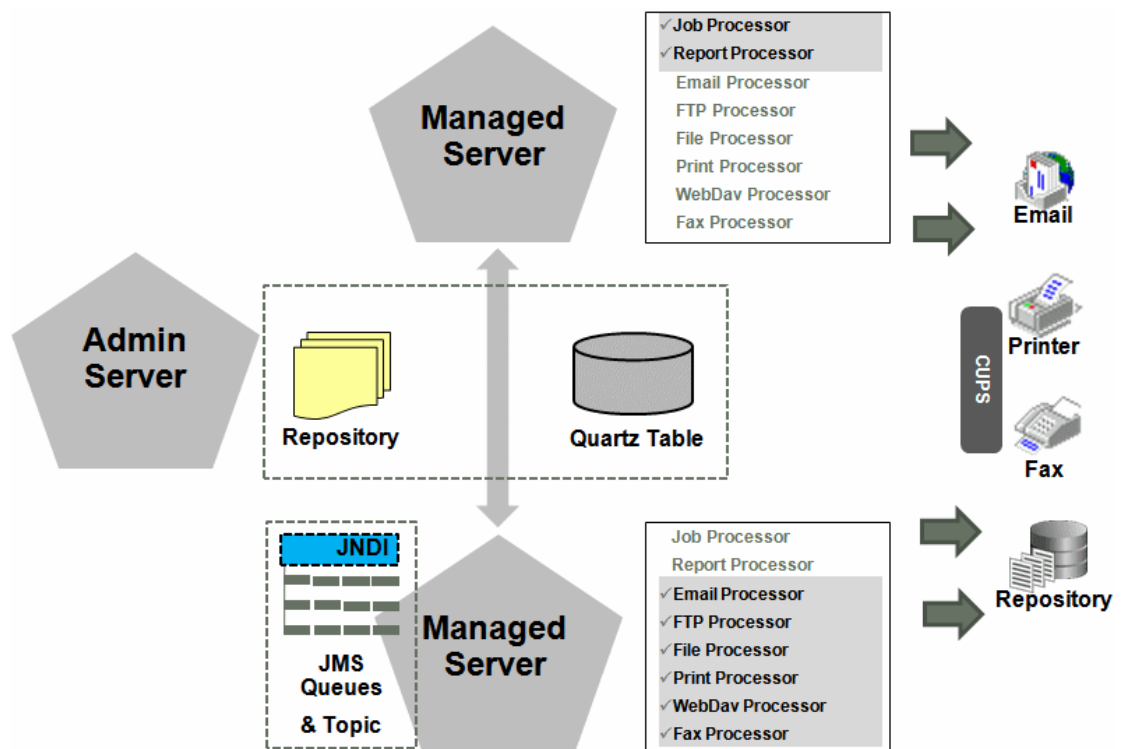
- Generates report based on metadata
 - Stores report in shared TEMP directory
 - Puts report delivery information in Delivery Queue
5. Delivery Processors
 - Listen to Delivery queue
 - Call delivery API to deliver to different channels
 6. Publisher System Topic

The Publisher 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.

About Clustering

Clustering enables you to add server instances on demand to handle processing and delivery load.

The figure below illustrates clustering. 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.



Each managed server instance points to the same report repository. In each managed server instance all the processes such as Job Processor, Report Processor, E-mail Processor, FTP Processor, Fax Processor, and Print 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.

How Failover Works

The failover mechanism ensures 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 pending jobs.

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 processing: data retrieval, data formatting, or report delivery, the job is marked as failed, and must be manually resubmitted.

About Prioritizing Jobs

You can configure the processing order of jobs.

You can prioritize jobs and ensure that the high-priority report jobs run before the non-critical jobs when multiple jobs run simultaneously. In the General tab of the Report Properties page, you can set the job priority as Critical, Normal, or Low priority. When jobs are queued, the processing of a job depends on the priority specified for the job's report. If you don't prioritize jobs, the critical jobs, non-critical jobs, and on-demand queries can compete for resources and the critical jobs might get delayed. In the Report Job History page, you can identify the critical jobs and view the status of each job.

About the Scheduler Configuration

When the scheduler starts automatically, certain configurations occur.

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

You can see the configuration in the Scheduler Configuration page under System Maintenance.

Configure Processors and Processor Threads

For each cluster instance that configure, 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 the figure below. Edit the threads for a specific processor in the Cluster Instances region by updating the **Number Threads** setting. 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. 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.

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're 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're 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 the table below.

----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 the figure below. Use this information to understand the load on each processor.

--Cluster		Passed	
----Instance - Cluster 369.127028		Passed	
-----JMS Instance Config	/user_projects/domains/base_domain/servers/AdminServer/tmp/_WL_user/xmlpservlet/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.
- 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 the figure below.

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

--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:ord	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 the figure below.

- Quartz Configuration
- Quartz Initialization

--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:ord	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 Initialization		Passed	

8

Set Up Data Sources

This topic describes how to set up data sources for Publisher.

Topics:

- [Set Up a JDBC Connection to a Data Source](#)
- [Set Up a Database Connection Using a JNDI Connection Pool](#)
- [Set Up a Connection to a File Data Source](#)
- [Set Up a Connection to an LDAP Server Data Source](#)
- [Set Up a Connection to an OLAP Data Source](#)
- [Set Up a Connection to a Web Service](#)
- [Set Up a Connection to an HTTP Data Source](#)
- [Set Up a Connection to a Content Server](#)
- [View or Update a Data Source](#)

About Private Data Source Connections

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

When you create a private data source connection, the private data source connection is available only to you in the data model editor data source menus.

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 Administration page.

Private data source connections are distinguished by an **Allowed User** value on the Data Source Administration page. 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 [Grant Access to Data Sources Using the Security Region](#).

Grant 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.

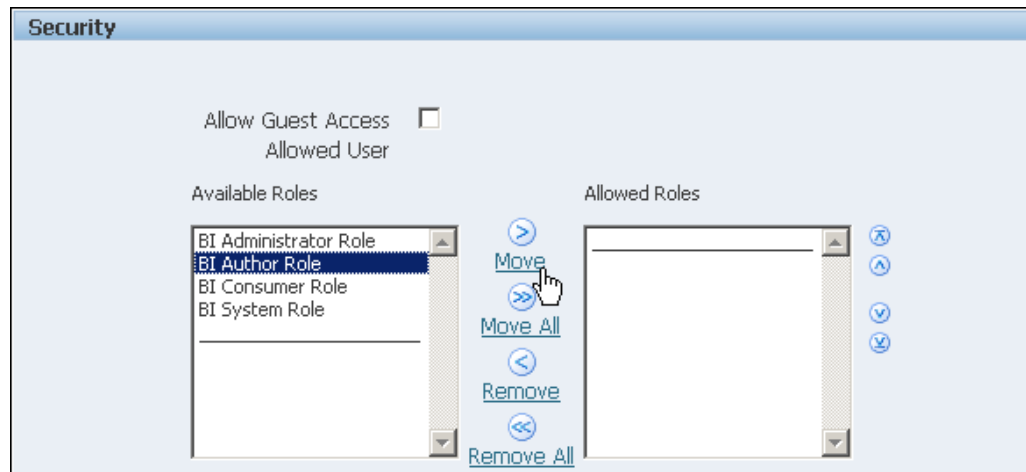
You must grant access to users for the following:

- 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.

The figure below shows the Security region of the data source configuration page.



About Proxy Authentication

Publisher supports proxy authentication for connections to various data sources

Supported data sources include:

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

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

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

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 supported by the Oracle BI Server.

About Pre Process Functions and Post Process Functions

You can define PL/SQL functions for Publisher to run 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, assuming 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 the table below.

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

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 run 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.

Choose 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 processed, the parameters initiate to open a database session every time.

Set Up a JDBC Connection to a Data Source

You can set up a JDBC connection to a data source.

Make sure all prerequisites have been met before setting up a JDBC connection to a data source:

- The JDBC driver for the selected database must be available to Publisher. If you're 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's not installed with WebLogic Server, then you must update the WebLogic Server classpath to include the location of the JDBC driver classes.

When the JDBC connection is defined, the administrator defines the user that 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's returned in a report, the administrator and data model developer can establish security, if needed, that can limit the data viewed by a particular Publisher user. One method for securing data returned is to use pre-process and post-process function calls to pass the `xdo_username`.

To set up a JDBC connection to a data source:

1. From the Administration page, click **JDBC Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source in the **Data Source Name** field. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select the driver type.
5. Select **Use Data Gateway** only if you want to connect to a remote data source.

Your administrator must enable remote data connectivity and configure Data Gateway on your target on-premise database. If you select **Use Data Gateway**,

the **Database Driver Class**, **Use System User**, **Pre Process Function**, **Post Process Function**, **Client Certificate**, and **Use Proxy Authentication** settings aren't available for selection or update.

6. You can update the **Database Driver Class** field if required.
7. Enter the database connection string.

Example connection strings:

- Oracle database

To connect to an Oracle database (non-RAC), use the following format for the connection string:

`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 for the connection string:

`jdbc:oracle:thin:@//<host>[:<port>]/<service_name>`

For example: `jdbc:oracle:thin:@//myhost.example.com:1521/my_service`

- Microsoft SQL Server

To connect to a Microsoft SQL Server, use the following format for the connection string:

`jdbc:hyperion:sqlserver://[hostname]:[port];DatabaseName=[Databasename]`

For example: `jdbc:hyperion:sqlserver://myhost.us.example.com:7777;DatabaseName=mydatabase`

8. Select **Use System User**. This is reserved for connections to the Oracle BI Server.
9. Enter the user name and password required to access the data source.
10. (Optional) Enter a PL/SQL function to run when a connection is created (Pre Process) or closed (Post Process).
11. (Optional) Select a client certificate for secured connection.
12. To enable Proxy Authentication, select **Use Proxy Authentication**.
13. Click **Test Connection**.
14. (Optional) Enable a backup database for this connection:
 - a. Select **Use Backup Data Source**.
 - b. Enter the connection string for the backup database.
 - c. Enter the user name and password for this database.
 - d. Click **Test Connection**.
15. Define security for this data source connection. Move the required roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

If you defined a backup data source, the security settings are passed to the backup data source.

Create a Secure JDBC Connection to Oracle Autonomous Data Warehouse

You can upload a JDBC client certificate and set up an SSL based JDBC connection to a database on the cloud.

To create a secure JDBC connection to Oracle Autonomous Data Warehouse:

1. From the Administration page, click **JDBC Connection**.
2. Click **Add Data Source**.
3. Specify the following details for the connection:
 - **Data Source Name:** DBaaSConnection
 - **Driver Type:** Oracle 12c
 - **Database Driver Class:** oracle.jdbc.OracleDriver
4. Enter the JDBC connection string.
Use TCPS strings. For example,
`jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)
(HOST=server_name)(PORT=port))
(CONNECT_DATA=(SERVICE_NAME=serviceName)))`
5. In the **Client Certificate** field, enter the full path of the wallet file, `cwallet.sso`.
6. Click **Test Connection**.
7. Click **Apply**.

Set Up a Database Connection Using a JNDI Connection Pool

You can connect to the JDBC data source using 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).



Note:

You can create JNDI connections to the user-defined data sources, but you can't create JNDI connections to the system-defined data sources. Only to create audit reports, you are allowed to create JNDI connections to the system-defined data sources to access the audit data source (AuditViewDataSource).

To set up a database connection using a JNDI connection pool:

1. From the Administration page, click **JNDI Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Enter the JNDI name for the connection pool. For example, jdbc/BIPSource.
5. Enter the following fields for the new connection:
6. Select **Use Proxy Authentication** to enable Proxy Authentication.
7. Click **Test Connection**. You see a confirmation message if the connection is established.
8. Define security for this data source connection. Move the required roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Connection to a File Data Source

You can use existing XML or Microsoft Excel files created from other sources as input to your reports.

To use a file as a data source, it must reside in a directory that 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. 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.

Set Up a Connection to an LDAP Server Data Source

You set up a connection to an LDAP data source from the Administration page.

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.LdapCtxFactory`).
4. Click **Test Connection**.
 5. Define security for this data source. 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.

Set Up a Connection to an OLAP Data Source

You can set up connections to several types of OLAP databases.

To set up a connection to an OLAP data source:

1. From the Administration page, click **OLAP Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select the OLAP type.
5. 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`
6. Enter the user name and password for the OLAP database.
 7. Click **Test Connection**.
 8. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Connection to an HTTP Data Source

HTTP data sources enable your data model designers to build data models from XML, JSON, and CSV data over the web by retrieving data through the HTTP GET method.

To add an HTTP data source:

1. From the Administration page, click **HTTP Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select the server protocol.
5. Enter the server name and the server port.
6. Enter the URL context for the HTTP data source connection in the **Realm** field.
For example, `xmlpserver/services/rest/v1/reports`
7. Enter the user name and password required to access the data source on the database.
8. Optional: From the **SSL Certificate** list, select the SSL certificate you want to use for the data source.
9. If you're using a proxy-enabled server, select **Use System Proxy**.
10. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Connection to a Content Server

Content server data source enables you to retrieve a text attachment content stored in Oracle WebCenter Content (earlier known as UCM) server and display it in the report of the corresponding document.

To set up a connection to a content server data source:

1. From the Administration page, select the **Content Server** link.
2. Click **Add Data Source**.
3. Enter the name in the **Data Source Name** field.
4. Enter the URL in the **URI** field.
5. Enter the user name and password in the **Username** and **Password** fields, respectively.
6. Click **Test Connection**.
7. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.
8. Click **Apply**.

Set Up a Connection to a Web Service

You can use a web service as a data source.

To add a web service as a data source:

1. From the Administration page, click **Web Service Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select the server protocol.
5. Enter the server name and the server port.
6. Enter the URL for the web service connection.
7. Optional: Enter the session timeout in minutes.
8. Select the security header from **WS-Security**.
 - 2002 — Enables the "WS-Security" Username Token with the 2002 namespace: `http://docs.oasis-open.org/wss/2002/01/oasis-200201-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`
9. Optional: Enter the user name and password for the web service data source.
10. Optional: From the **SSL Certificate** list, select the SSL certificate you want to use for the connection.
11. If you're using a proxy-enabled server, select **Use System Proxy**.
12. Click **Test Connection**.
13. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.
14. Click **Apply**.

View or Update a Data Source

You can view or update a data source from the Administration page.

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 editable. 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.

9

Set Up Delivery Destinations

This topic describes the setup required to deliver Publisher reports. It also describes how to set up the HTTP notification server.

Topics:

- [Configure Delivery Options](#)
- [Understand Printer and Fax Server Configuration](#)
- [Add a Printer](#)
- [Add a Fax Server](#)
- [Add an Email Server](#)
- [Add a WebDAV Server](#)
- [Add an HTTP Server](#)
- [Add an FTP or SFTP Server](#)
- [Add a Content Server](#)
- [Add a Common UNIX Printing System \(CUPS\) Server](#)
- [Add an Oracle Content and Experience Server](#)

Add a WebDAV Server

You add a WebDAV server from the Administration page.

To add a WebDAV server:

1. From the Administration 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)

Configure Delivery Options

You can define the SSL certificate file and set the general properties for e-mail deliveries and notifications.

To configure delivery options:

1. From the Administration page, select **Delivery Configuration**.
2. If you want to use a self-signed certificate, select a file from **SSL Certificate File**.
3. Enter the From address to appear on e-mail report deliveries. The default value is `bipublisher-report@oracle.com`.
4. Enter the From address to appear on notifications deliveries. The default value is `bipublisher-notification@oracle.com`.
5. Enter the subject line to display for e-mail notification recipients when the report status is Success, Warning, Failed, or Skipped.
6. Select **Use System Proxy Settings** if the Delivery Manager must look up the proxy server settings from the Java runtime environment.
 - Printer, Fax, WebDAV, HTTP and CUPS servers use proxy settings for HTTP protocol when SSL is not used. When SSL is used, the HTTPS proxy setting is used.
 - FTP and SFTP use proxy settings for FTP.
 - Contents servers and email servers don't support connection over a proxy, regardless of this setting.

You can override the proxy settings per delivery server, using proxy configuration fields on the individual server setup page. If a proxy server and ports are configured for a delivery server, the Delivery Manager uses the proxy server and port configured for the server instead of the one defined in the Java Runtime environment. In Cloud installations, **Use System Proxy Settings** is always selected, and cannot be turned off or overridden by individual server settings.

Understand Printer and Fax Server Configuration

Understand your printer type before you set up the printer or fax server.

Regardless of the operating system, the printer destination can be any IPP server. The IPP server can be the printer itself, but if the printer doesn't natively support IPP, you can set up a print server that does support IPP (such as CUPS), and then connect to the print server to the printer.

To send a fax, you must set up Common Unix Printing Service (CUPS) and the fax4CUPS extension. 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.

PDF is a popular output format for business reports. However, some reports require printing directly from the report server. For example, paychecks and invoices are usually printed as scheduled batch jobs. Some printers with PostScript Level 3 compliant Raster Image Processing can natively support PDF documents, but there're still many printers in business use that only support PostScript Level 2 that can't print PDF documents directly.

To print PDF documents directly, if your printer or print server doesn't support printing PDF:

- Select a filter - PDF to PostScript or PDF to PCL.
- Configure a custom, or third-party filter.

A filter enables you to call a conversion utility to convert the PDF to a file format supported by your specific printer type. You can use the PDF to PCL conversion only for font selection requirements for check printing. For generic printing requirements, use the PDF to PostScript level 2 filter.

Selection of **PDF to PCL** filter automatically populates the **Filter Command** field. 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.

You can also call a custom filter using operating system commands.

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're trying to call IPP printers directly or IPP printers on Microsoft Internet Information Service (IIS). Unlike CUPS, those print servers don't translate the print file to a format the printer can understand. 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 Command** 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 Command** 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).

Add a Printer

You can set up a printer to print reports.

To set up a printer:

1. From the Administration page, under **Delivery**, select **Printer**, and then click **Add Server**.
2. Enter the server name and URI of the printer.
3. Optional: If your printer or print server doesn't support printing PDF, enter a filter to call a conversion utility to convert the PDF to a file format supported by your specific printer type.
4. Optional: Enter the user name, password, authentication type (None, Basic, Digest), and encryption Type (None, SSL).
5. Optional: Enter the host, port, user name, password, and authentication type (None, Basic, Digest) of the proxy server.
6. In the Access Control section, deselect **Public**.
7. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.

8. Click **Apply**.

Add a Fax Server

You must set up Common Unix Printing Service (CUPS) and the fax4CUPS extension, if you want to send fax.

To set up a fax sever:

1. From the Administration page, under **Delivery**, select **Fax**, and then click **Add Server**.
2. Enter the server name and the URI (Uniform Resource Identifier) of the fax server.
3. Optional: If your fax server doesn't support printing PDF, enter a filter to call a conversion utility to convert the PDF to a file format supported by your specific fax server.
4. Optional: Enter the user name, password, authentication type (None, Basic, Digest), and encryption Type (None, SSL) of the fax server.
5. Optional: Enter the host, port, user name, password, and authentication type (None, Basic, Digest) of the proxy server.
6. In the Access Control section, deselect **Public**.
7. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.
8. Click **Apply**.

Add an Email Server

You can add an email server to deliver reports by email.

To add an email server:

1. From the Administration page, under **Delivery**, select **Email**, and then click **Add Server**.
2. Enter the **Server Name** and **Host** of the email server.
3. Optional: Select a **Secure Connection** method to use for connections with the email server.

Use TLS when the server supports the protocol; SSL is accepted in the response.
4. Optional: Enter the port number, user name, and password.
5. In the Access Control section, deselect **Public**.
6. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.
7. Click **Test Connection**.
8. Click **Apply**.

Deliver Reports Using Email Delivery Service on Oracle Cloud Infrastructure

You can use the Email Delivery service on Oracle Cloud Infrastructure to deliver reports.

If you don't have access to Oracle Cloud Infrastructure Console, ask your Oracle Cloud Infrastructure administrator to provide you access.

To deliver reports using the Email Delivery service on Oracle Cloud Infrastructure:

1. In Oracle Cloud Infrastructure Console, perform these steps.
 - a. Generate SMTP credentials.
 - i. Open the navigation menu, under **Governance and Administration** go to **Identity**, and then click **Users**.
 - ii. Locate the user in the list that has permissions to manage email, and then click the user's name to view the details.
 - iii. Click **SMTP Credentials**, click **Generate SMTP Credentials**, and then enter a Description of the SMTP Credentials in the dialog box.
 - iv. Click **Generate SMTP Credentials**, copy the user name and password for your records, and then click **Close**.
 - b. Obtain SMTP connection information for your Email Delivery service.
 - i. Open the navigation menu. Under **Solutions and Platform** go to **Email Delivery**.
 - ii. Click **Email Configuration**.
Get the following details of the email server:
 - Server name and host
 - Port number
 - SMTP credentials (user name and password)
 - Secure connection method
 - c. Verify that the "From" addresses for report delivery are registered as approved senders.

Only approved senders can send mail through Oracle Cloud Infrastructure.
 - i. Open the navigation menu. Under **Solutions and Platform**, go to **Email Delivery**.
 - ii. Click **Email Approved Senders**.
2. In Oracle Analytics Cloud, add a connection to the email server.
 - a. From the Administration page, under **Delivery**, select **Email**, and then click **Add Server**.
 - b. Enter the name of the email server (Email Delivery service hostname).
 - c. Enter the port number and SMTP credentials (user name and password).
 - d. Select the secure connection method.
 - e. In the Access Control section, deselect **Public**.

- f. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.
 - g. Click **Test Connection**.
 - h. Click **Apply**.
3. Set up delivery notification.
 - a. From the Administration page, under **Delivery**, select **Delivery Configuration**.
 - b. Enter values for **Email From Address** and **Delivery Notification Email From Address**.
 - c. Optional: Enter values for **Success Notification Subject**, **Warning Notification Subject**, **Failure Notification Subject**, and **Skipped Notification Subject**.

The completed jobs use the appropriate notification subject depending on the status of the job.
 - d. Deselect **Use System Proxy Settings**.
4. Configure the bursting jobs to deliver reports using the email server.

Update bursting queries to specify Email as the delivery channel in `DEL_CHANNEL` and provide the "From" address in `PARAMETER3`.
5. Test report delivery.
 - a. Schedule a job to email a report using the email server.
 - b. In the Job History Details page, check the status of the job.

Add 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 Publisher posts a form data for Job ID, report URL and Job Status to the HTTP Server URL page.

To add an HTTP server:

1. From the Administration page, under **Delivery**, select **HTTP**, and then click **Add Server**.
2. Enter the server name and the URL of the server.
3. Optional: Enter the host, port, user name, password, authentication type (None, Basic, Digest), and encryption type (None, SSL) of the server.
4. Optional: If the notification is to be sent through a proxy server, enter the user name, password, and the authentication type (None, Basic, Digest).
5. In the Access Control section, deselect **Public**.
6. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.
7. Click **Apply**.

Add an FTP or SFTP Server

You can add an FTP server or SFTP server as a delivery channel from the Administration page.

If the destination file name supplied to the scheduler contains non-ascii characters, UTF-8 encoding is used 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.

Publisher doesn't support FTP over TLS / SSL (FTPS). You can't use FTP over TLS or SSL for delivery. Use SFTP for secure file transfer.

1. From the Administration page, under **Delivery**, select **FTP**, and then click **Add Server**.
2. Enter the server name, host name, and port number for the FTP or SFTP server.

The default port for FTP is 21. The default port for Secure FTP (SFTP) is 22.

3. To enable Secure FTP (SFTP), select **Use Secure FTP**.
4. If the FTP server is behind a firewall, select **Use Passive Mode**.
5. In the **Host Key Fingerprint** field, enter the host key. The value must match the fingerprint calculated from server's host key at runtime. If it doesn't match, an exception error is thrown. When you connect the first time, the Delivery Manager API allows you to retrieve the server key fingerprint.
6. Optional: In the **Filter Command** field, specify a custom filter to apply a file conversion such as encryption.

To specify a custom filter, pass the native Operating System command string with the two placeholders for the input and output file name, {infile} and {outfile}.

For example, to set up encryption of the file using a Filter Command, enter the following:

```
gpg -e -r myKey -o {outfile} {infile}
```

where

myKey is the ID to gpg key (such as real name, email address, or fingerprint).

The Filter command field doesn't support quotes. Therefore you cannot use certain valid gpg formats that include spaces, for example: "myname <myemail@example.com>". You must specify the ID in a single string with no spaces.

7. Select **Create files with Part extension when copy is in process** to create a file on the FTP server with a .part extension while the file is transferring.

When the file transfer is complete, the file is renamed without the .part extension. If the file transfer doesn't complete, the file with the .part extension remains on the server.

8. Optional: Enter the security information.
 - a. If your server is password protected, enter the User name and Password.
 - b. Select the **Authentication Type**: Private Key or Password
 - c. Depending on the authentication type selection, select the private key file or specify the private password.
9. Optional: Enter the host, port, user name, password, and authentication type (None, Basic, Digest) of the proxy server.
10. Optional: To deliver PGP encrypted documents to the FTP server:

- a. From the **PGP Key** list, select the PGP keys you uploaded in Security Center.
This step updates the filter command in the **Filter Command** field.
 - b. To sign the encrypted document, select **Sign Output**.
This step adds a `-s` parameter to the existing filter command in the **Filter Command** field.
 - c. If you want to deliver PGP encrypted document in ASCII armored format, select **ASCII Armored Output**.
This step adds a `-a` parameter to the existing filter command in the **Filter Command** field.
11. In the Access Control section, deselect **Public**.
 12. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.
 13. Click **Test Connection**.
 14. Click **Apply**.

SSH Options for SFTP

Secure File Transfer Protocol (SFTP) is based on the Secure Shell technology (SSH). Publisher supports the following SSH options for SFTP delivery.

SSH Option	Supported Values
Cipher Suites	• 3des-cbc • blowfish-cbc • aes128-cbc • aes128-ctr • aes192-ctr • aes256-ctr You can use aes192-ctr and aes256-ctr cipher suites only when Publisher is running on a JVM on which the Java Cryptography Extension (JCE) unlimited strength jurisdiction policy files are installed.
Key Exchange Method	• diffie-hellman-group1-sha1 • diffie-hellman-group14-sha1 • diffie-hellman-group-exchange-sha1 • diffie-hellman-group-exchange-sha256 You can use diffie-hellman-group-exchange-sha256 key exchange methods only when Publisher is running on a JVM on which the Java Cryptography Extension (JCE) unlimited strength jurisdiction policy files are installed.
Public Key Algorithm	• ssh-dss • ssh-rsa

Add a Content Server

You can deliver documents to Oracle WebCenter Content.

When you use a content server as a delivery destination:

- At runtime, 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.
- For documents that require specific custom metadata fields (such as invoice number, customer name, order date), the report author can map the custom metadata fields defined in Content Profile Rule Sets to data fields in the data model.

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 Publisher Administration page.
- JAX-WS: The JAX-WS protocol is supported only in Oracle WebCenter Content 11g with a properly configured Content Server instance and the RIDC client installed. JAX-WS is not supported outside this environment.

To set up a content server as a delivery destination:

1. From the Administration page, under **Delivery**, select **Content Server**, and then 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. Optional: Enter the user name and password of the content server.
5. Optional: To enable the inclusion of custom metadata with your report documents delivered to the content server, select **Enable Custom Metadata**.
6. Optional: To deliver PGP encrypted documents to the content server:
 - a. From the **PGP Key** list, select the PGP keys you uploaded in Security Center.
This step updates the filter command in the **Filter Command** field.
 - b. To sign the encrypted document, select **Sign Output**.
This step adds a `-s` parameter to the existing filter command in the **Filter Command** field.
 - c. If you want to deliver PGP encrypted document in ASCII armored format, select **ASCII Armored Output**.
This step adds a `-a` parameter to the existing filter command in the **Filter Command** field.
7. In the Access Control section, deselect **Public**.
8. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.
9. Click **Test Connection**.
10. Click **Apply**.

Add a Common UNIX Printing System (CUPS) Server

You add CUPS servers from the Administration page.

You can configure Common Unix Printing Service (CUPS) for sending fax and to enable printing using a printer that doesn't natively support IPP.

To add a CUPS server:

1. From the Administration page, select **CUPS** to display the list of servers that have been added.
2. Select **Add Server**.
3. Enter the **Server Name** and **Host** and **Port** for the CUPS server.

Add an Oracle Content and Experience Server

You can deliver reports to an Oracle Content and Experience server to enable easy access and share reports on the cloud.

To add an Oracle Content and Experience server:

1. From the Administration page, under **Delivery**, select **Content and Experience**, and then click **Add Server**.
2. In the **Server Name** field, type the name of the server through which you want to deliver the reports to the cloud-based content hub.
3. In the **URI** field, type the URI of the Oracle Content and Experience server. For example, `https://host.oraclecloud.com`.

4. In the **Username** and **Password** fields, provide the credentials for accessing the Oracle Content and Experience server.
5. In the Access Control section, deselect **Public**.
6. From the **Available Roles** list, select one or more roles you want to provide access to the delivery channel, and click **Move** to add them to the **Allowed Roles** list.
7. Click **Test Connection**.
8. Click **Apply**.

10

Define Runtime Configurations

This topic describes processing properties for PDF document security, FO processing, PDF accessibility, and specific properties for each output type.

Topics:

- [Set Runtime Properties](#)
- [PDF Output Properties](#)
- [PDF Digital Signature Properties](#)
- [PDF Accessibility Properties](#)
- [PDF/A Output Properties](#)
- [PDF/X Output Properties](#)
- [DOCX Output Properties](#)
- [RTF Output Properties](#)
- [PPTX Output Properties](#)
- [HTML Output Properties](#)
- [FO Processing Properties](#)
- [RTF Template Properties](#)
- [XPT Template Properties](#)
- [PDF Template Properties](#)
- [Excel Template Properties](#)
- [CSV Output Properties](#)
- [Excel Output Properties](#)
- [EText Output Properties](#)
- [All Outputs Properties](#)
- [Memory Guard Properties](#)
- [Data Model Properties](#)
- [Delivery Properties](#)
- [Define Font Mappings](#)
- [Define Currency Formats](#)

Set Runtime Properties

The Runtime Configuration page enables you to set runtime properties at the server level.

These same properties can also be set at the report level, from the report editor's Properties dialog. If different values are set for a property at each level, then report level takes precedence.

PDF Output Properties

Generate the type of PDF files you want by setting available output properties.

Property Name	Description	Default	Configuration Name
Compress PDF output	Specify "true" or "false" to control compression of the output PDF file.	true	pdf-compression
Hide PDF viewer's menu bars	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.	false	pdf-hide-menubar
Hide PDF viewer's tool bars	Specify "true" to hide the viewer application's toolbar when the document is active.	false	pdf-hide-toolbar
Replace smart quotes	Specify "false" if you don't want curly quotes replaced with straight quotes in the PDF output.	true	pdf-replace-smartquotes
Disable opacity and gradient shading for DVT chart	Specify "true" if you don't want opacity and gradient shading for the PDF output. This reduces the size of the PostScript file.	false	pdf-dvt-no-opacity-no-gradient-shading
Enable PDF Security	Specify "true" if you want to encrypt the PDF output. You can then also specify the following properties: • Open document password • Modify permissions password • Encryption Level	false	pdf-security
Open document password	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 Adobe's password restrictions apply. The password must contain only Latin-1 characters and must be no more than 32 bytes long.	N/A	pdf-open-password

Property Name	Description	Default	Configuration Name
Modify permissions password	This password enables users to override the security setting. This property is effective only when "Enable PDF Security" is set to "true". Note that Adobe's password restrictions apply. The password must contain only Latin-1 characters and must be no more than 32 bytes long. If you set a password in the pdf-open-password property without setting a password in the pdf-permissions-password property, or if you set the same password in both the pdf-open-password and pdf-permissions-password properties, the user gets full access to the document and its features, and permission settings such as "Disable printing" are bypassed or ignored.	N/A	pdf-permissions-password
Encryption level	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	2 - high	pdf-encryption-level

Property Name	Description	Default	Configuration Name
Disable document modification	Permission available when "Encryption level" is set to 0. When set to "true", the PDF file cannot be edited.	false	pdf-no-changing-the-document
Disable printing	Permission available when "Encryption level" is set to 0. When set to "true", printing is disabled for the PDF file.	false	pdf-no-printing
Disable adding or changing comments and form fields	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.	false	pdf-no-accff
Disable context copying, extraction, and accessibility	Permission available when "Encryption level" is set to 0. When set to "true", the context copying, extraction, and accessibility features are disabled.	false	pdf-no-cceda
Enable text access for screen readers	Permission available when "Encryption level" is set to 1 or higher. When set to "true", text access for screen reader devices is enabled.	true	pdf-enable-accessibility
Enable copying of text, images, and other content	Permission available when "Encryption level" is set to 1 or higher. When set to "true", copying of text, images, and other content is enabled.	false	pdf-enable-copying
Allowed change level	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	0	pdf-changes-allowed
Allowed printing level	Permission available when "Encryption level" is set to 1 or higher. Valid values are: • 0: None • 1: Low resolution (150 dpi) • 2: High resolution	0	pdf-printing-allowed

Property Name	Description	Default	Configuration Name
Use only one shared resources object for all pages	The default mode of 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.	true	pdf-use-one-resources
PDF Navigation Panel Initial View	Controls the navigation panel view 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.	Bookmarks Open	pdf-pagemode

PDF Digital Signature Properties

You set properties at the report level to enable digital signature for a report and to define the placement of the signature in the output PDF document.

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.

Property Name	Description	Default	Configuration Name
Enable Digital Signature	Set this to "true" to enable digital signature for the report.	false	signature-enable
Existing signature field name	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.	N/A	signature-field-name
Signature field location	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 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.	N/A	signature-field-location
Signature field X coordinate	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.	0	signature-field-pos-x

Property Name	Description	Default	Configuration Name
Signature field Y coordinate	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.	0	signature-field-pos-y
Signature field width	Enter in points (72 points equal one inch) the desired width of the inserted digital signature field. This applies only if you're also setting the Signature field x coordinate and Signature field Y coordinate properties..	0	signature-field-width
Signature field height	Enter in points (72 points equal one inch) the desired height of the inserted digital signature field. This applies only if you're also setting the Signature field x coordinate and Signature field Y coordinate properties.	0	signature-field-height

PDF Accessibility Properties

Set the properties described in the table below to configure PDF accessibility.

Property Name	Description	Default
Make PDF output accessible	Set to "true" to make the PDF outputs accessible. Accessible PDF output contains the document title and PDF tags.	False
Use PDF/UA format for accessible PDF output	Set to "true" to use the PDF/UA format for the accessible PDF outputs.	False

PDF/A Output Properties

Set the properties described in the table below to configure PDF/A output.

Property Name	Description	Default	Configuration Name
PDF/A version	Set the PDF/A version.	PDF/A-1B	pdfa-version

Property Name	Description	Default	Configuration Name
PDF/A ICC Profile Data	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. 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 <code><bi_publisher_repository>/Admin/Configuration</code>. When you set this property, you must also set a value for PDF/A ICC Profile Info (pdfa-icc-profile-info).	Default profile data provided by JVM	pdfa-icc-profile-data
PDF/A ICC Profile Info	ICC profile information (required when pdfa-icc-profile-data is specified)	sRGB IEC61966-2.1	pdfa-icc-profile-info
PDF/A file identifier	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 (,).	Automatically generated file identifier	pdfa-file-identifier
PDF/A document ID	Valid document ID. The value is set in the xmpMM:DocumentID field of the metadata dictionary.	None	pdfa-document-id
PDF/A version ID	Valid version ID. The value is set in the xmpMM:VersionID field of the metadata dictionary.	None	pdfa-version-id
PDF/A rendition class	Valid rendition class. The value is set in the xmpMM:RenditionClass field of the metadata dictionary.	None	pdfa-rendition-class

PDF/X Output Properties

Configure PDF/X output by setting the properties described below. 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`.

Property Name	Description	Default	Configuration Name
PDF/X ICC Profile Data	(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 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 support or colormanagement.org.	None	pdfx-dest-output-profile-data
PDF/X output condition identifier	(Required) The name of one of the standard printing conditions registered with ICC (International Color Consortium). 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.	None	pdfx-output-condition-identifier
PDF/X output condition	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.	None	pdfx-output-condition
PDF/X registry name	A registry name. Set this property when the pdfx-output-condition-identifier is set to a characterization name registered in a registry other than the ICC registry.	http://www.color.org	pdfx-registry-name
PDF/X version	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.	PDF/X-1a:2003	pdfx-version

DOCX Output Properties

The table below describes the properties that control DOCX output files.

Property Name	Description	Default	Configuration Name
Enable change tracking	Set to "true" to enable change tracking in the output document.	false	docx-track-changes
Protect document for tracked changes	Set to "true" to protect the document for tracked changes.	false	docx-protect-document-for-tracked-changes
Default font	Use this property to define the font style and size in the 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 processing engine at runtime.	Arial:12	docx-output-default-font
Open password	Use this property to specify the password that report users must provide to open any DOCX report.	NA	docx-open-password

RTF Output Properties

Configure RTF output files by setting the properties described in the table below.

Property Name	Description	Default	Configuration Name
Enable change tracking	Set to "true" to enable change tracking in the output RTF document.	false	rtf-track-changes
Protect document for tracked changes	Set to "true" to protect the document for tracked changes.	false	rtf-protect-document-for-tracked-changes

Property Name	Description	Default	Configuration Name
Default font	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 processing engine at runtime. See Define Font Mappings for information about installing fonts and for the list of predefined fonts.	Arial:12	rtf-output-default-font
Enable widow orphan	Set to "true" to ensure that the document includes no "hanging paragraphs". Suppose the last para in a page contains an orphaned line and the remaining lines of the paragraph continue on the next page. With this setting enabled, the starting line of the paragraph moves to the next page to keep all the lines of the paragraph together for improved readability.	false	rtf-enable-widow-orphan

PPTX Output Properties

The table below describes the properties that control PPTX output files.

Property Name	Description	Default	Configuration Name
Open password	Use this property to specify the password that report users must provide to open any PPTX report.	NA	pptx-open-password

HTML Output Properties

The table below describes the properties that control HTML output files.

Property Name	Description	Default	Configuration Name
Show header	Set to "false" to suppress the template header in HTML output.	true	html-show-header
Show footer	Set to "false" to suppress the template footer in HTML output.	true	html-show-footer
Replace smart quotes	Set to "false" if you don't want curly quotes replaced with straight quotes in the HTML output.	true	html-replace-smartquotes

Property Name	Description	Default	Configuration Name
Character set	Specify the output HTML character set.	UTF-8	html-output-charset
Make HTML output accessible	Set to "true" to make the HTML output accessible.	false	make-accessible
Use percentage width for table columns	Set to "true" to display 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 display tables with extremely wide columns. Setting this property to true improves the readability of the tables.	true	html-output-width-in-percentage
View Paginated	When you set this property 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.	false	
Reduce Padding in Table-cell	When you set this property to true, cells in HTML tables are displayed without padding, which maximizes the page space available for text.	false	html-reduce-padding
Embed images and charts in HTML for offline viewing	When you set this property to true, charts and images are embedded in the HTML output, which is suitable for viewing offline.	true	html-use-data-uri
Use SVG for charts	When you set this property to true, charts display as a SVG (Scalable Vector Graphic) to provide a higher resolution in the HTML output. When you set this property to false, charts display as a raster image.	true	html-use-svg
Keep original table width	When you set this property to true, if a column in a table is deleted, the original width of the table is maintained.	true	html-keep-original-table-width

Property Name	Description	Default	Configuration Name
Enable horizontal scrollbar automatically for html table	When you set this property to true, a horizontal scroll bar is added to a table that doesn't fit within the current size of the browser window.	false	html-enable-horiz-table-scroll
Enable html table column size auto adjust	When you set this property to true, the column widths in a table are automatically adjusted to the size of the browser window.	false	html-enable-table-col-size-auto-adjust
Set zero height for empty paragraph	When you set this property to true and the output is HTML, the height of an empty paragraph (that is, a paragraph without text) is set to zero points.	true	html-set-empty-paragraph-zero-height

FO Processing Properties

The table below describes the properties that control FO processing.

Property Name	Description	Default	Configuration Name
Use BI Publisher's XSLT processor	Controls the use of parser. If set to false, then XSLT isn't parsed.	true	xslt-xdoparser
Enable scalable feature of XSLT processor	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.	false	xslt-scalable
Enable XSLT runtime optimization	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 isn't as marked. To further enhance performance when you set this property to true, set the Extract attribute sets property to "false".	true	xslt-runtime-optimization
Enable XPath Optimization	When set to "true", the XML data file is analyzed for element frequency. The information is then used to optimize XPath in XSL.	false	xslt-xpath-optimization

Property Name	Description	Default	Configuration Name
Pages cached during processing	This property is enabled only when you specify 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.	50	system-cache-page-size
Bidi language digit substitution type	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's ignored.	National	digit-substitution
Disable variable header support	When set to true, prevents variable header support. Variable header support automatically extends the size of the header to accommodate the contents.	false	fo-prevent-variable-header
Enable multithreading	If you have a multiprocessor machine or a machine with a dual-core single processor, you might achieve faster document generation by setting this option to True.	false	fo-multi-threads
Disable external references	When set to true, disallows 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.	true	xdk-secure-io-mode
FO Parsing Buffer Size	Specifies 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.	1000000	fo-chunk-size
FO extended linebreaking	When set to true, punctuation, hyphenation, and international text are handled properly when line breaking is necessary.	true	fo-extended-linebreaking

Property Name	Description	Default	Configuration Name
Enable XSLT runtime optimization for sub-template	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. The default is true. If you call the FOProcessor directly, the default is false.	true	xslt-do-import
Report Timezone	Valid values: User or JVM. When set to User, 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, 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.	User	fo-report-timezone
PDF Bidi Unicode Version	Specifies the Unicode version (3.0 or 4.1) used to display the BIDI strings in the PDF output.	4.1	pdf-bidi-unicode-version

RTF Template Properties

Configure RTF templates by setting the properties described in the table below.

Property Name	Description	Default	Configuration Name
Extract attribute sets	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	Auto	rtf-extract-attribute-sets

Property Name	Description	Default	Configuration Name
Enable XPath rewriting	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.	true	rtf-rewrite-path
Characters used for checkbox	The default PDF output font doesn't 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: <pre>fontname;<unicode font number for true value's glyph >;<unicode font number for false value's glyph></pre> Example: Albany WT J;9746;9747/A Note that the font that you specify must be made available at runtime.	Albany WT J;9746;9747/ A	rtf-checkbox-glyph

XPT Template Properties

Configure XPT templates by setting the properties described in the table below.

Property Name	Description	Default
XPT Scalable Mode	When you set this property to true, the scheduled reports that use the XPT template and include a large amount of data run without memory issues. The first 100,000 rows of data in the report are stored in memory and the remaining rows are stored in the file system. When you set this property to false, the scheduled reports that use XPT template are processed in-memory. Set this property to false for reports that contain less data.	False

Property Name	Description	Default
Enable Asynchronous Mode for Interactive Output	When you set this property to true, interactive reports that use the XPT template make asynchronous calls to Oracle WebLogic Server. When you set this property to false, interactive reports that use the XPT template make synchronous calls to Oracle WebLogic Server. Oracle WebLogic Server limits the number of synchronous calls. Any calls that are stuck expire in 600 seconds.	True

PDF Template Properties

Generate the types of PDF files you want by setting available PDF template properties.

Property Name	Description	Default	Configuration Name
Remove PDF fields from output	Specify "true" to remove PDF fields from the output. When PDF fields are removed, data entered in the fields cannot be extracted.	false	remove-pdf-fields
Set all fields as read only in output	By default, all fields in the output PDF of a PDF template is read only. If you want to set all fields to be updatable, set this property to "false".	true	all-field-readonly
Maintain each field's read only setting	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."	false	all-fields-readonly-asis

Excel Template Properties

Configure Excel templates by setting the properties described in the table below.

Property Name	Description	Default
Enable Scalable Mode	When set to true, large reports that use Excel template run without out of memory issues. Data overflows automatically into multiple sheets if a group of data in a sheet exceeds 65000 rows. This overcomes the Microsoft Excel limitation of 65000 rows per sheet. When set to false, large reports that use Excel template can cause out of memory issues.	false

CSV Output Properties

The table below describes the properties that control comma-delimited value output.

Property Name	Description	Default
CSV delimiter	Specifies the character used to delimit the data in comma-separated value output. Other options are: Semicolon (;), Tab (\t) and Pipe ().	Comma (,)
Remove leading and trailing white space	Specify "True" to remove leading and trailing white space between data elements and the delimiter.	false
Add UTF-8 BOM Signature	Specify "False" to remove the UTF-8 BOM signature from the output.	true

Excel Output Properties

You can set specific properties to control Excel output.

Property Name	Description	Default
Show grid lines	Set to true to show the Excel table grid lines in the report output.	false
Page break as a new sheet	When set to "True" a page break specified in the report template generates a new sheet in the Excel workbook.	true
Minimum column width	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.	3 (in points, 0.04 inch)
Minimum row height	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 0.001 to 5 points.	1 (in points, 0.01 inch)
Keep values in same column	Set this property to True to minimize column merging. Column width is set based on column contents using the values supplied in the Table Auto Layout property. Output may not appear as neatly laid out as when using the original layout algorithm.	False

Property Name	Description	Default
Table Auto Layout	Specify a conversion ratio in points and a maximum length in points, for example 6.5,150. See example. For this property to take effect, the property "Keep values in same column" must be set to True. This property expands the table column width to fit the contents. The column width is expanded based on the character count and conversion ratio up to the maximum specification. Example: Assume a report with two columns of Excel data -- Column 1 contains a text string that's 18 characters and Column 2 is 30 characters long. When the value of this property is set to 6.5,150, the following calculations are performed: Column 1 is 18 characters: Apply the calculation: $18 * 6.5\text{pts} = 117\text{pts}$ The column in the Excel output will be 117 pts wide. Column 2 is 30 characters: Apply the calculation: $30 * 6.5\text{pts} = 195\text{pts}$ Because 195 pts is greater than the specified maximum of 150, Column 2 will be 150 pts wide in the Excel output.	N/A
Maximum allowable nested table row count	Specify the maximum allowable row count for a nested table. Allowed values are 15000 to 999,999. During report processing, nested inner table rows cannot be flushed to the XLSX writer, therefore they stay in-memory, increasing memory consumption. Set this limit to avoid out-of-memory exceptions. When this limit is reached for the size of the inner table, generation is terminated. The incomplete XLSX output file is returned.	20,000
Open password	Use this property to specify the password that report users must provide to open any XLSX output file. Configuration name: <code>xlsx-open-password</code>	NA

EText Output Properties

The table below describes the properties that control EText output files.

Property Name	Description	Default
Add UTF-8 BOM Signature	When set to true, the Etext output is in UTF-8 Unicode with BOM format.	false
Enable bigdecimal	When set to true, you enable high-precision numeric calculation of the Etext output.	false

All Outputs Properties

The properties in the table below apply to all outputs.

Property Name	Description	Default
Use 11.1.1.5 compatibility mode	Reserved. Do not update unless instructed by Oracle.	false

Memory Guard Properties

The Runtime Configuration page lists the default values of the memory guard properties. The values of the memory guard properties depend on the compute shape used for your instance.

Property	Description	Default Value
Maximum report data size for online reports	Limits the data size for online reports.	300MB
Maximum report data size for offline (scheduled) reports	Limits the data size for scheduled reports.	500MB
Maximum report data size for bursting reports	Limits the data size for bursting reports.	500MB
Free memory threshold	Ensures a minimum available free memory space.	500MB
Maximum report data size under the free memory threshold	Limits the data size of a report when the Free memory threshold property is set to a positive value.	50MB
Minimum time span between garbage collection runs	Ensures a minimum time gap in seconds between any two subsequent garbage collection runs.	300 (seconds)
Maximum wait time for free memory to come back above the threshold value	Limits the time in seconds for a run-report request to wait for the free JVM memory to exceed the threshold value. This property value takes effect only if you specify a positive value for the Free memory threshold property. If free memory is still below the threshold value after the specified wait time, the run-report request is rejected.	30 (seconds)
Timeout for online reports	Specifies the timeout value in seconds for processing an online report (includes the time for data extraction and report generation).	535 (seconds)

Property	Description	Default Value
Maximum rows for CSV output	Limits the rows for reports in CSV format.	1000000

Data Model Properties

The Runtime Configuration page lists the values of the data model properties. The values of the data model properties depend on the compute shape used for your instance.

Property	Description	Default
Maximum data size limit for data generation	Limits the size of XML data that can be generated by executing a data model.	500MB
Maximum sample data size limit	Limits the size of a sample data file that can be uploaded from the data model editor.	1MB
Enable Data Model scalable mode	Prevents out of memory conditions. When set to true, the data engine takes advantage of the disk space while processing data.	True
Enable Auto DB fetch size mode	Avoids out of memory conditions, but can significantly increase the processing time. This setting is recommended only for frequently processing complex queries of hundreds of columns. When set to true, the database fetch size is set at runtime according to the total number of columns and the total number of query columns in the data set. Ignores the DB fetch size setting. This property overrides the data model-level database fetch size properties.	True
DB fetch size	Limits the database fetch size for a data model. This property value takes effect only when Enable Auto DB fetch size mode is set to False.	20 (rows)
SQL Query Timeout	Specifies the timeout value for SQL queries for scheduled reports. This value is based on the compute size of the instance. The value for online reports is 500 seconds and is the same for all implementations. You can't modify the value for online reports.	600 seconds
Enable Data Model diagnostic	Writes the data set details, memory, and SQL processing time information to the log file when set to true. Oracle recommends setting this property to true only for debugging purposes. If you enable this property, the processing time is increased.	False
Enable SQL Session Trace	When set to true, for every SQL query that's processed, Publisher writes a SQL session trace log to the database. A database administrator can examine the log.	False

Property	Description	Default
Enable SQL Pruning	Applies only to the Oracle Database queries that use Standard SQL. If your query returns many columns but only a subset are used by your report template, SQL pruning returns only those columns required by the template. If you enable this property, the processing time and the memory usage reduces. SQL pruning is not applicable for PDF, Excel, and E-text template types.	False
Enable Data Chunking	When set to true, you can enable XML data chunking for individual data models, reports, and report jobs. Select this property and specify an appropriate value for the Data Chunk Size property to process large and long-running reports.	False
Data Chunk Size	Specifies the data size for each data chunk. Applies only when the Enable Data Chunking property is set to true.	300MB
DV Data Row Limit	Limits the number of rows that can be retrieved from a data set.	2000000
Trim Leading and Trailing Spaces From Parameter Value	Trims the leading and trailing spaces from the parameter values of data models.	True

Delivery Properties

The properties in the table below apply to report delivery.

Property Name	Description	Default
Enable FTP/SFTP delivery retry	If a delivery through an FTP or SFTP delivery channel fails, Publisher makes another attempt to deliver, 10 seconds after the first attempt fails. This setting affects all FTP and SFTP delivery requests, and can't be configured for individual servers.	True

Define Font Mappings

Map base fonts in RTF or PDF templates to target fonts to be used in the published document.

You can specify font mapping at the site or report level. Font mapping is performed only for PDF output and PowerPoint output.

There're 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.

Make Fonts Available for Publishing

A set of Type1 fonts and a set of TrueType fonts are available for publishing. You can select any of the fonts in these sets as a target font with no additional setup required.

The predefined fonts are located in `<oracle_home>/oracle_common/internal/fonts`. To map to another font, place the font in this directory to make it available for publishing at runtime. If the environment is clustered, then you must place the font on every server.

Set 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.

Create a Font Map

Provide the base font and target font.

1. From the Administration page, under **Runtime Configuration**, select **Font Mappings**.
2. Under RTF Templates or PDF Templates, click **Add Font Mapping**.
3. Provide the details for the base font.
 - **Base Font:** Enter the font family to map to a new font. Example: Arial
 - **Style:** Normal or Italic (Not applicable to PDF Template font mappings)
 - **Weight:** Normal or Bold (Not applicable to PDF Template font mappings)
4. Provide the details of the target font.
 - **Target Font Type:** Type 1 or TrueType
 - **Target Font:** Select a 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.

Define 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.

Understand 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 currency formats in the RTF template, use the format-currency function.

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

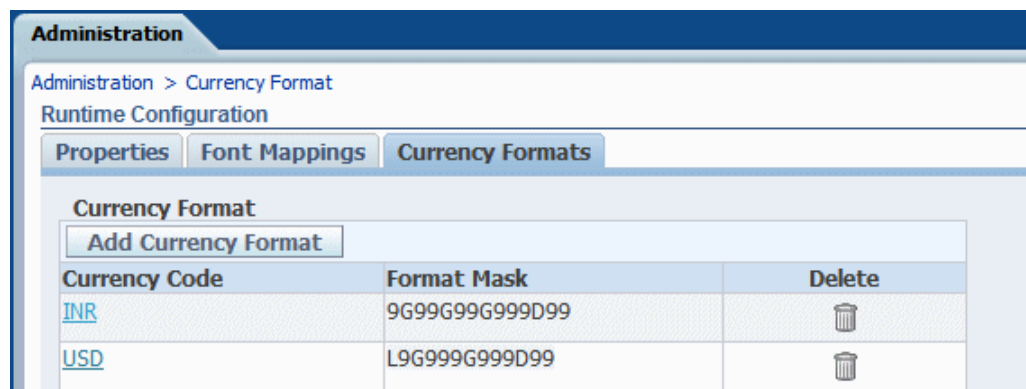
9 represents a displayed number only if present in data

G represents the group separator

D represents the decimal separator

0 represents an explicitly displayed number regardless of incoming data

The figure below shows sample currency formats.



11

Secure Reports

This topic describes how to secure pixel-perfect reporting.

Topics:

- [Use Digital Signature in a PDF Document](#)
- [Use PGP Keys for Encrypted Report Delivery](#)

Use Digital Signature in a PDF Document

You can apply digital signatures to PDF output documents.

Digital signatures enable you to verify the authenticity of the documents you send and receive. You can upload your digital signature file to a secure location, and at runtime sign the PDF output with the digital signature. The digital signature verifies the signer's identity and ensures that the document hasn't been altered after it was signed.

For additional information, refer to the Verisign and Adobe websites.

Prerequisites and Limitations

When you use digital signatures with documents in Publisher , you must be aware of a few limitations.

A digital signature is obtained from a public certificate authority or from a private/internal certificate authority (if for internal use only). You must copy the digital signature file to a secure location accessible by Publisher.

Keep the following limitations in mind:

- You can register only a single digital signature with Publisher.
- Only the reports scheduled in Publisher 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.

Obtain Digital Certificates

You can obtain a digital certificate either by purchasing one or by using the self-sign method.

To obtain a digital certificate:

- Perform one of the following:
 - Purchase a certificate from an authority, verify and trust the authenticity of the certificate, and then use Microsoft Internet Explorer 7 or later to create a PFX file based on the certificate you purchased.
 - Create a self-signed certificate using a software program such as Adobe Acrobat, Adobe Reader, OpenSSL, or OSDT as part of a PFX file, and then use the PFX file to

sign PDF documents by registering it with Publisher. Bear in mind that anyone can create a self-signed certificate, so use care when verifying and trusting such a certificate.

Create 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.

You don't need to create a PFX file if a self-signed certificate PFX file already exists.

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. From the Tools menu, click **Internet 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.

Apply a Digital Signature

You can set up and sign your output PDF documents with a digital signature.

To apply a digital signature:

1. Register the digital signature in the Publisher Administration page and specify the roles that are authorized to sign documents.
2. Specify the display field location.
3. Select the **Enable Digital Signature** property for the report.
4. Log in as a user with an authorized role and submit the report through the Publisher scheduler, choosing the PDF output. When the report completes, it is signed with your digital signature in the specified location of the document.

Register Your Digital Signature and Assign Authorized Roles

Publisher supports the identification of a single digital signature.

To register a digital signature:

1. On the Administration tab, under **Security Center**, click **Digital Signature**.

2. Enter the file path to the digital signature file and enter the password for the digital signature.
3. Enable the Roles that must have the authority to sign documents with this digital signature. Use the shuttle buttons to move Available Roles to the Allowed Roles list.
4. Click **Apply**.

Specify 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:

- Specify a template field in a PDF template for the digital signature.
- Specify the location for the digital signature in the report properties.

If the template is RTF, specify the location for the digital signature in the report properties.

Specify a Template Field in a PDF Template for the Digital Signature

Include a field in the PDF template for digital signatures.

Report authors can add a new field or configure an existing field in the PDF template for the digital signature. See [Add or Designate a Field for a Digital Signature](#).

Specify 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 height and width of the field for the digital signature by using runtime properties. You don't 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** — Doesn't 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 Publisher places the digital signature in the output document sized and positioned appropriately.

If you set this property, then don't 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're 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're setting the X and Y coordinates.

Run and Sign Reports with a Digital Signature

If you've been assigned a role that's been granted the digital signature privilege, you can sign a generated report with a signature, if the report has been configured to include signatures. You can sign only scheduled reports with signatures.

To sign reports with a digital signature:

1. Log in 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 on the Schedule Report Job page, select **PDF output**, and then submit the job.

The completed PDF displays the digital signature.

Use PGP Keys for Encrypted Report Delivery

You can use PGP encryption to secure file delivery through the FTP server and the Content server.

You can configure the FTP server and Content server delivery channels to use the PGP public keys to deliver PGP encrypted files. You can configure an FTP server to use the PGP encrypted files in ASCII armor format.

Use Security Center to upload and download the PGP keys.

View Audit Data of Publisher Catalog Objects

An administrator can enable or disable viewing of the audit data of Publisher catalog objects, configure a connection to the audit data, and create reports to view the audit data.

Topics:

- [About Viewing Audit Data of Publisher Catalog Objects](#)
- [Enable or Disable Viewing of Publisher Audit Data](#)
- [Specify the Data Source Connection for Publisher Audit Data](#)
- [View Publisher Audit Data](#)

About Viewing Audit Data of Publisher Catalog Objects

You can view the audit data of Publisher catalog objects.

You can find out the time of access and who accessed the Publisher catalog objects such as reports, data models, sub-templates, style templates, and folders.

Audit data helps you track:

- Report start, process, end, and download
- Report job pause, resume, and cancellation
- Publisher resource creation, modification, copy, and deletion
- Publisher resource access



Note:

User session data (User Login and User Logout events) isn't included in the audit data. Only the reporting activities performed in the *host:port/ui/xmlpserver* Publisher interface pages are included in the audit data. The reporting activities performed in the *host:port/ui/analytics* interface pages aren't included in the audit data.

Enable or Disable Viewing of Publisher Audit Data

Administrators can enable or disable viewing the audit data of publishing activities.

1. Navigate to the Server Configuration page.
2. To enable viewing of audit data, select **Enable Monitor and Audit** and set **Audit Level** to **Medium**.
3. To disable viewing of audit data, deselect **Enable Monitor and Audit**.

Specify the Data Source Connection for Publisher Audit Data

Configure a data source connection for the audit data.

1. In the Administration page, click **JNDI Connection**.
2. Click **Add Data Source**.
3. In the **Data Source Name** field, enter AuditViewDB.
4. In the **JNDI Name** field, enter `jdbc/AuditViewDataSource`.
5. Click **Test Connection** to confirm the connection to the AuditViewDB data source.
6. Define security for this data source connection. Move the required 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.
7. Click **Apply**.

View Publisher Audit Data

You can download and use the sample reports for viewing the audited information.

Make sure you select **Enable Monitor and Audit** in the Server Configuration page to log audit data, and then configure the JNDI connection to the AuditViewDB data source to view the audit data.

The sample reports use the JNDI connection to fetch data from the data source for auditing. The report layout and data model are pre-designed in the sample reports. You can customize the report layout, but don't change the data model in the sample reports. The sample reports are configured to run as a scheduled job because the size of auditing data can be large. If you want to view an audit report online, select the **Run Report Online** property and make sure you don't select the **Auto Run** property of the report.

1. Download the sample audit reports from the Publisher home page in Oracle Technology Network.
2. Upload the sample audit reports to a shared folder in the catalog.
3. Schedule the sample audit reports you want to view.
 - a. Navigate to the sample audit report in the catalog.
 - b. Click **Schedule**.
 - c. In the General tab, specify the dates for the **Date From** and **Date To** parameters.
 - d. In the Output tab, make sure the output format is PDF.

You can add delivery destinations if required.
4. After the scheduled job completes, view the report in the Report Job History page.

Add Translations for the Catalog and Reports

This topic describes how to export and import translation files both for the catalog and for individual report layouts.

Topics:

- [Introduction](#)
- [Export and Import a Catalog Translation File](#)
- [Template Translation](#)
- [Use a Localized Template](#)

Introduction

Publisher supports two types of translation: Catalog Translation and 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 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 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.

Limitations of Catalog Translation

If you have XLIFF file translations for specific reports and then you import a catalog translation file for the folder in which the existing translations reside, you overwrite the existing XLIFF files.

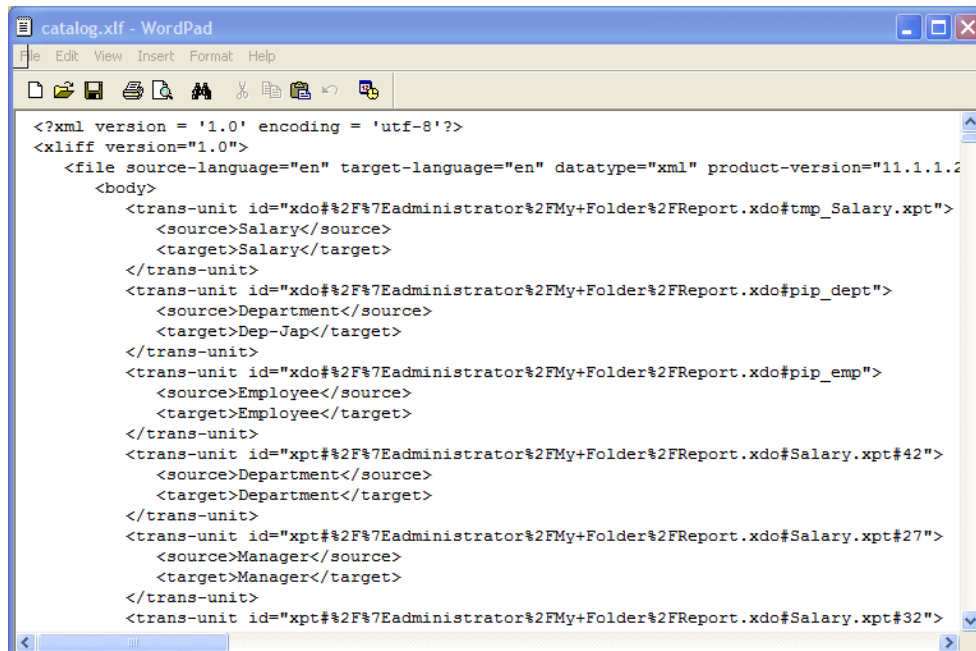
Export and Import a Catalog Translation File

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 the following figure.



4. After the file is translated, upload the XLIFF file to the 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.

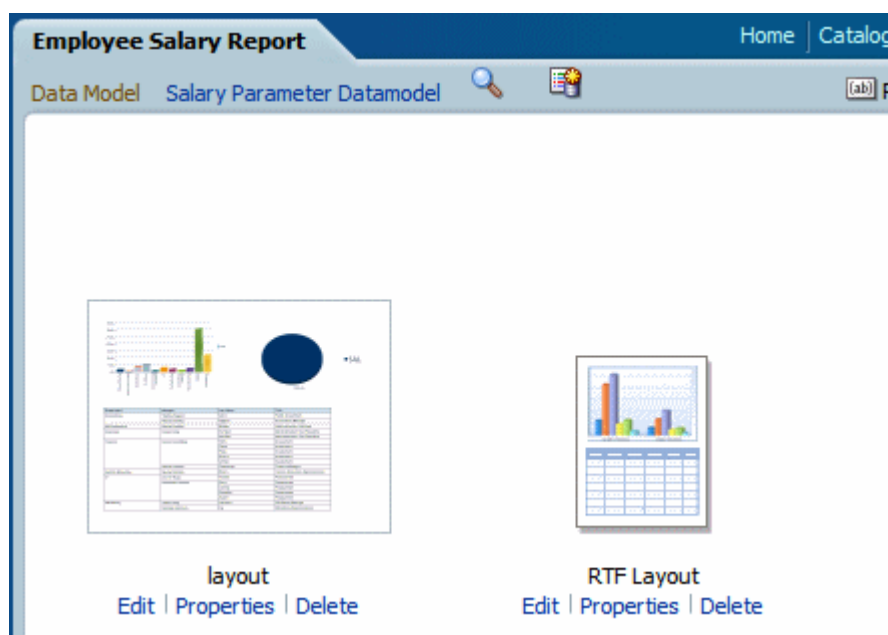
Template Translation

RTF and Publisher (.xpt) templates can be translated from the Properties page.

Template translation includes:

- RTF templates
- RTF sub templates
- Style templates
- Publishertemplates (.xpt)

To access the Properties page, click the **Properties** link for the layout in the Report Editor, as shown below.



From the Properties page you can generate an XLIFF file for a single template. Click **Extract Translation** to generate the XLIFF file.

Generate the XLIFF File from the Layout Properties Page

Generate the XLIFF file for report layout templates, style templates, and sub templates.

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**.

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**.

Publisher extracts the translatable strings from the template and exports them to an XLIFF (.xlf file).

3. Save the XLIFF to a local directory.

Translate the XLIFF File

When you download a XLIFF file, it can be sent to a translation provider, or using a text editor, you can enter the translation for each string.

A "translatable string" is any text in the template 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.

Upload the Translated XLIFF File to Publisher

You can run the Template Manager to upload the translated XLIFF file to Publisher.

To upload a 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.

Use a Localized Template

You can create localized templates for reports.

If you need to design a different layout for the reports that you present for different localizations, then you can create new RTF file designed and translated for the locale and upload this file to the Template Manager.

The localized template option is not supported for XPT templates.

Design 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.

Upload the Localized Template to Publisher

Upload localized template files in rtf format.

To upload a 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.

Move Catalog Objects Between Environments

This topic describes how to move objects between test, production, and development environments using the catalog utility.

Topics:

- [Overview](#)
- [Prepare to Use the Catalog Utility](#)
- [Export the Reporting Objects](#)
- [Import the Reporting Objects](#)
- [Generate Translation Files and Checking for Translatability](#)

Overview

The catalog utility enables administrators and report developers to export the reporting object-related files from the catalog where all the pixel-perfect reports are stored, and to import them to a different catalog.

Use this tool to manage pixel-perfect 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. You must first run the GenerateBIPUtility script to generate the BIPCatalogUtil utility. See [Generate the Utilities](#).

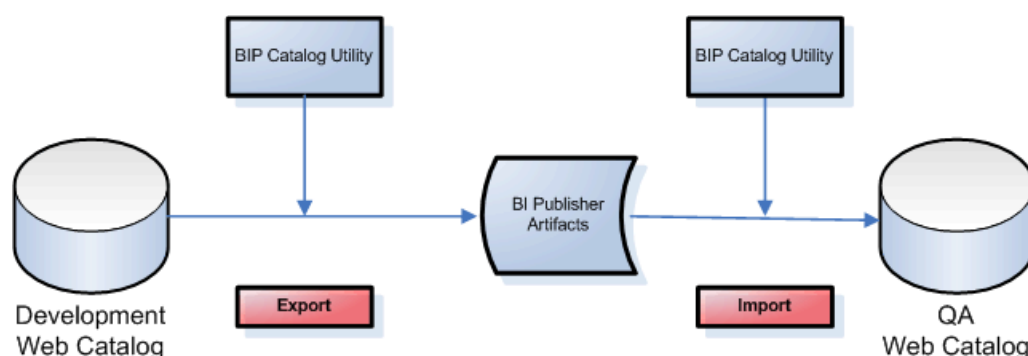
Use the catalog utility to perform the following tasks:

- Export pixel-perfect reports from the catalog
- Import pixel-perfect 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

When to Use the Catalog Utility

Use the catalog utility to move 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 below.



Other Options for Moving Catalog Objects

You can download and upload catalog objects to transfer objects across environments.

To download or upload a small number of objects, the download feature of the 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.

Note:

Do not manually edit the Publisher files in the file system. 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 corrupts, then you can restore it by deleting the corrupt file and restarting Publisher.

What Files Are Moved

Styles and skins are organized into folders that contain Cascading Style Sheets (CSS) and images.

Object	Files
Report Example: Balance+Letter.xdo	• <code>_report.xdo</code> — The report definition file • <code>xdo.cfg</code> — The configuration file that contains the report property settings • <code>~metadata.meta</code> — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. • <code>security.xml</code> file — Specifies the object level permissions defined for the report • <code>template</code> files — All template files loaded to the report definition. The file names include the language suffix, for example: <code>My_RTF_template_en_us.rtf</code>, <code>My_BIP_layout_en_us.xpt</code> • <code>translation</code> files — All translation files (.xlf), for example: <code>My_RTF_template_jp_jp.xlf</code>

Object	Files
Data Model Example: myDataModel.xdm	• <code>_datamodel.xdm</code> — The report definition file • <code>~metadata.meta</code> — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. • <code>security.xml</code> file — Specifies the object level permissions defined for the data model
Subtemplate Example: mysubtemplate.xsb	• <code>_template_en_us.rtf</code> — The subtemplate file with locale designation • <code>~metadata.meta</code> — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. • <code>security.xml</code> file — Specifies the object level permissions defined for the subtemplate • translation files — Any translations, when present; for example: <code>_template_jp_jp.rtf</code>
Style Template Example: myStyleTemplate.xss	• <code>_template_en_us.rtf</code> — The style template file with locale designation • <code>~metadata.meta</code> — The metadata file that contains the catalog path information. This file is used by the utility to import objects back to their original locations. • <code>security.xml</code> file — Specifies the object level permissions defined for the style template • translation files — Any translations, when present; for example: <code>_template_jp_jp.rtf</code>

Maintaining Identical Folder Names and Structure Across Environments

A pixel-perfect 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 doesn't 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 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 doesn't 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.

Prepare to Use the Catalog Utility

The catalog utility is installed in the following location:

```
ORACLE_HOME/clients/bipublisher
```

Configure the Environment

You must configure each environment in which you run 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 Publisher instance URL ("bipurl") and the user name and password of the 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`.

Export the Reporting Objects

Use the export command to export either a single reporting object or a set of reporting objects under a specified folder.

Export commands:

- `-export` — Use this command to export a single report object.
- `-exportfolder` — Use this command to export a folder and its contents.

The table below describes the supported parameters for the `-export` and `-exportfolder` commands.

Parameter	Used With	Description
catalogpath	-export -exportfolder	The path to the object in the catalog. If there're spaces in any of the names, use the '+' sign as a substitute.
target	-export	The destination directory in which to place the extracted reporting objects.
basedir	-exportfolder	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	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', and '.cfg'. Use the default option of exporting the reporting object to a zip format to handle the international characters. If you set the value to 'true', then the utility exports the reporting object-related files under the specified target folder.
subfolders	-exportfolder	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	Specify true to overwrite existing objects in the target area.

Import the Reporting Objects

Use the import command to import either a single reporting object or a set of reporting objects under a specified folder.

The table below describes the supported parameters for the import command.

Parameter	Description
catalogpath	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	The directory where the reporting object is located. Use this parameter when you're importing a single report.
basedir	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	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.

Generate Translation Files and Checking for Translatability

The catalog utility supports the -xliff command to generate a translatable XLIFF file for a specific file.

The table below 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 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.

Parameter	Description
source	The path to the report or template file (RTF or XPT) for which to generate the XLIFF file.
target	The location to save the generated .xlf document.
basedir	The directory to place the generated .xlf files into.

The following examples show how to generate translation files:

- [Generate a Translation File for a Report Definition File \(.xdo\)](#)
- [Generate a Translation File for an RTF Template](#)

Generate 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/
```

Generate 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/
```

Customize the Publisher User Interface

The user interface in Publisher is generated by using scripts and is therefore customizable. The look-and-feel is controlled by skins and styles. Publisher is shipped with the Skyros (default style), and blafplus (browser look-and-feel plus), styles.

Topics:

- [What are Skins and Styles?](#)
- [About Style Customizations](#)
- [Modify the User Interface Styles](#)
- [Customize the Style](#)

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 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.

About Style Customizations

To customize the look-and-feel of 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.

Most of the common Skyros styles and image files, including the style sheet (master.css), are contained in the master directory.

Within the master.css style sheet, each element (or class) that's available for update is documented in the comments.

Other style sheets are also contained within the Skyros style and skin folders. You only update these files if you want to create an advanced custom skin.



Note:

The Skyros style doesn't apply to Administration pages in Publisher.

Modify the User Interface Styles

To change the skin for Publisher, modify the `xmfp-server-config.xml` configuration file located at `CATALOG_DIRECTORY/Admin/Configuration/xmfp-server-config.xml`.

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"/>
```

The `THEME` property must be either `"blafplus"` or `"skyros"`.

Customize 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 shipped with Publisher contains an example folder structure to help you get started.

When creating styles and skins for Publisher, you must create CSS and image files, and make them available to Publisher. Only the CSS defined in `master.css` and images defined in the `master` folder can be customized for Publisher (bundled in the `bicustom.ear` file).

Customize the Style for Publisher Standalone

Update selected configuration files to create a custom style for Publisher when Publisher is not integrated with the Oracle Analytics Server.

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 doesn't overwrite the `bicustom.ear` file.

2. Extract the `bicustom.war` file from the `bicustom.ear` file to the machine where 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 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/aimel/
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 Publisher.

Customize the Style for Publisher Integrated with the Oracle Analytics Server

Update selected configuration files to create a custom style for Publisher integrated with Oracle Analytics Server.

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 doesn't overwrite the bicustom.ear file.

2. Extract the bicustom.war file from the bicustom.ear file to the machine where 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/
aim1/bip/res/s_Custom/master"/>
```

9. Restart 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.

Fallback Mechanism for Custom Styles

When creating custom styles for Publisher (standalone and integrated with Oracle Analytics Server), 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 Publisher, which is the Skyros theme.

Custom Style Sheets

For custom style sheets (css), if THEME_CUSTOM_MASTER_CSS_URL is provided, Publisher references those styles and ignores any others.

For custom style sheets (css), if THEME_CUSTOM_MASTER_CSS_URL is provided, Publisher references those styles and ignores any others. If THEME_MASTER_CSS_URL is provided, Publisher uses those styles. If neither are provided, Publisher uses the styles defined in the base skin.

Images

Publisher constructs image URLs based on certain factors.

For images, if `THEME_CUSTOM_MASTER_IMAGE_PATH` is provided, and the requested image exists in the directory, 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, Publisher uses the value of `THEME_MASTER_IMAGE_URL_PREFIX` to construct the image URL. If neither are provided, Publisher uses the images defined in the base skin.

A

Scheduler Configuration Reference

This appendix describes how to configure the Publisher scheduler for each supported database and how to configure ActiveMQ as the JMS provider.

Topics:

- [Introduction](#)
- [Configure Publisher for ActiveMQ](#)
- [Configure the Quartz Scheduler](#)

Introduction

The 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 [Configure the Scheduler](#).

Configure 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 Publisher if you choose to use ActiveMQ as the JMS provider.

Install ActiveMQ

You can install Apache ActiveMQ version 5.2.0 or later in Windows, UNIX, or Linux.

Follow the installation steps documented in the ActiveMQ documentation.

Register ActiveMQ as a JNDI Service

Update the activemq.xml configuration file to 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`.

Update the Publisher Scheduler Configuration Page

Open the Scheduler Configuration page from the Publisher Administration page.

1. On the 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`.

Configure the Quartz Scheduler

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, 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 Publisher Installer, then you must download, install, and configure the driver manually.

Recommendations for Using DataDirect Connect or Native Database Drivers

DataDirect Connect for JDBC drivers are provided for supported databases

Supported databases include:

- IBM DB2 v8.1, v9.1
- Microsoft SQL Server 2000, 2005
- Sybase Adaptive Server Enterprise
- Oracle 9i, Oracle 10g, Oracle 11g,

The table below displays the driver recommendations for the supported scheduler databases.

Database	Native JDBC Driver	DataDirect JDBC Driver
Oracle 10g, Oracle 11g	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

Set Up a User on Your Scheduler Database

To set up the connection to the scheduler database, make that you create a user on the selected database.

Publisher uses this user to connect to the database. Depending on the database type, this user might require specific privileges.

Connect to Your Scheduler Database and Install the Schema

Following are the general steps for setting up the Scheduler database. Also refer to the subsequent section that's specific to your database.

To set up the Scheduler database:

1. Log in to 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.

The Oracle database drivers and the DataDirect drivers are installed with 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 Publisher.

5. Click **Test Connection** to ensure that BI Publisher can connect to the database.
6. Click **Install Schema** to install the Publisher scheduler schema to your database.

Connect to Oracle Databases

When connecting to an Oracle database, 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.

The table below describes the fields for the Oracle native driver to connect to the Oracle Database.

Field	Description
Database Type:	Select Oracle 11g or Oracle 10g 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

Connect to IBM DB2 Databases

When connecting to an IBM DB2 v8 or IBM DB2 v9 database, 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.

The table below describes the fields for the DataDirect driver to connect to an IBM DB2 v8 or IBM DB2 v9 database.

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

Connect to Microsoft SQL Server Databases

When connecting to a Microsoft SQL Server database, 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.

The table below describes the fields for the DataDirect driver to connect to a Microsoft SQL Server 2000 or 2005 database.

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

Connect to Sybase Adaptive Server Enterprise Databases

When connecting to an Sybase Adaptive Server Enterprise database, ensure that you set the "ddl in tran" mode to true in the database. Consult the Sybase documentation or contact your database administrator for instructions on how to enable this option.

The table below describes the fields for the DataDirect driver to connect to a Sybase Adaptive Server Enterprise 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=bipsc heduler.
Database Driver Class:	hyperion.jdbc.sybase.SybaseDriver

B

Integration Reference

This appendix describes configuration details for integrating Publisher with Oracle BI Presentation Services and BI Server.

Topics:

- [About Integration](#)
- [Integrate with Presentation Services](#)
- [Set Up a JDBC Connection to the Oracle BI Server](#)

About Integration

The information in this chapter is for reference to highlight the integration points between Publisher and the Oracle Analytics Server.

You might need to reference this information in the following scenarios:

- You're upgrading
- You run a separate installation of 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 integration with Oracle BI Presentation Services

Prerequisites

Publisher must be installed on the same server with the other components of Oracle Analytics Server.

The security configuration must be either Oracle Fusion Middleware security or Oracle Analytics Server security.

Integrate with Presentation Services

When you install the Oracle Analytics Server, integration with Publisher is automatically configured and "Server" and "Port" information remains uneditable. Furthermore, the username and password fields are hidden, because both products are configured to use Oracle Fusion Middleware security.

1. From the Administration page, under Integration, click Presentation Services.
2. Enter the following information about your Presentation Services server:
 - **Server Protocol** — Select http or https
 - **Server Version** — Select v10

- **Server** — Enter the server host name. For example: BIEEServer
- **Port** — Enter the port for the server where the Oracle BI Presentation Services plug-in is running. For example: 9502
- **Administrator Username and Password** — These fields are hidden when you use Oracle Fusion Middleware Security.
- **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 enable the Web services between Publisher and Presentation Services.

- Session time out in minutes

Set Up a JDBC Connection to the Oracle BI Server

Make sure all prerequisites have been met before setting up a JDBC Connection to the Oracle BI Server.

 Note:

If you installed Publisher with the Oracle Analytics Server, then this data source is automatically configured.

To add the Oracle Analytics Server as a JDBC data source, follow the guidelines in [Set Up a JDBC Connection to a Data Source](#) with these specific guidelines.

Note that if Oracle BI Server is SSL-enabled, then you must copy the keystore to Publisher 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 Publisher and provide it in the connection string.

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 available on the server. If the catalog is not specified, you see the default. 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 Oracle BI Server.

<Password> — Specifies the password for the Oracle BI Server for the user name.

<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 tries 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;SecondaryCCS=machine02;SecondaryCCSPort=9706;user=example;password=example;ssl=true;sslKeystorefilename=c:\example\OracleBI\ssl\javahost.keystore;sslKeystorepassword=example;trustanyserver=true;
```

Use System User — you must select this box to use the BISystem User. When you select this box, Publisher will use the BISystem Username and password to connect to the Oracle 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 Publisher runtime configuration file.
Topics:

- [Publisher Configuration Files](#)
- [Set Properties in the Runtime Configuration File](#)
- [Structure of the Root Element](#)
- [Properties and Property Elements](#)
- [Font Definitions](#)
- [Predefined Fonts](#)

Publisher Configuration Files

This appendix contains reference information about the following 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. See [Set Runtime Properties](#).

Set 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 don't use the Administration page to set the properties, then Publisher falls back to the properties set in this file.

The Administration interface doesn't update this file. Any settings in the Administration pages take precedence over the settings in the xdo.cfg file.

File Name and Location

The configuration file is named xdo.cfg.

The file is located under the <Publisher Repository>/Admin/Configuration.

Namespace

Namespace for configuration file.

`http://xmlns.oracle.com/oxp/config/`

Configuration File Example

Refer to the sample configuration file below.

```
<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 -->
  <font>
    <!-- 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>
  </font>
</config>
```

Example Element Specification

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.

Structure of the Root Element

The <config> element is the root element.

The element has the following structure:

```
<config version="cdata" xmlns="http://xmlns.oracle.com/oxp/config/">
  <font> ... </font> [0..n]
  <property> ... </property> [0..n]
</config>
```

Attributes of Root Element

The <config> element has the attributes described in the table below.

Attribute	Description
version	The version number of the configuration file format. Specify 1.0.0.
xmlns	The namespace for Publisher's configuration file. Must be http://xmlns.oracle.com/oxp/config/

Description of Root Element

The root element of the configuration file.

The configuration file consists of two parts:

- Properties (<property> elements)
- Font definitions (elements)

The and <property> elements can appear multiple times. If conflicting definitions are set up, the last occurrence prevails.

Properties and Property Elements

This section describes the <properties> element and the <property> element.

<properties> Element

The structure of the <properties> element is shown below.

```
<properties locales="cdata">
  <property>...
  </property> [0..n]
</properties>
```

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>
```

<property> Element

The structure of the <type1> element is shown below.

```
<property name="cdata">
  ...pcdata...
</property>
```

Attribute of <property> Element

The <property> element has a single attribute, name, which specifies the property name.

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 [Define Runtime 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>
```

Font Definitions

Font definitions include the specific elements.

Elements include:

-
-

- `<font-substitute>`
- `<truetype>`
- `<type1>`

For the list of Truetype and Type1 fonts, see [Predefined Fonts](#).

`<font>` Element

The structure of the `<font>` element is shown below.

```
<font locales="cdata">
  <font> ... </font> [0..n]
  <font-substitute> ... </font-substitute> [0..n]
</font>
```

Attribute of `<font>` Element

The `<font>` element has a single optional attribute, `locales`, which specifies the locales for this font definition.

Description of `<font>` Element

The `<font>` element defines a set of fonts. Specify the `locales` attribute to define locale-specific fonts.

```
<!-- Font definitions for all locales -->
<font>
  ..Font definitions here...
</font>

<!-- Korean-specific font definitions -->
<font locales="ko-KR">
  ... Korean Font definitions here...
</font>
```

`<font>` Element

The structure of the `<font>` element is shown below.

```
<font family="cdata" style="normalitalic"
weight="normalbold">
  <truetype>...</truetype>
or <type1> ... <type1>
</font>
```

Attributes of `<font>` Element

The `<font>` element has the attributes described in the table below.

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.

Description of Element

Defines a 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) doesn't refer to this element.

```
<!-- Define "Arial" font -->
<font family="Arial" style="normal" weight="normal">
  <truetype path="/fonts/Arial.ttf"/>
</font>
```

<font-substitute> Element

The structure of the <font-substitute> element is shown below.

```
<font-substitute name="cdata">
  <truetype>...</truetype>
or <type1>...</type1>
</font-substitute>
```

Attributes of <font-substitute> Element

The <font-substitute> element has a single attribute, name, which specifies the name of the font to be substituted.

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>
```

<type1> element

The structure of the <type1> element is shown below.

```
<type1 name="cdata"/>
```

Attribute of <type1> Element

The <type1> element has a single attribute, name, which specifies one of the Adobe standard Latin1 fonts, such as "Courier".

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>
```

Predefined Fonts

The following Type1 fonts are built-in to Adobe Acrobat and 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 the table below.

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
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

Number	Font Family	Style	Weight	Font Name
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 the table below. All TrueType fonts are subsetting and embedded into PDF.

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)
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)

Number	Font Family Name	Style	Weight	Actual Font	Actual Font Type
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)

Included Barcode Fonts

Publisher includes a number of barcode fonts.

Barcode fonts are described in the table below.

Font File	Supported Algorithm
128R00.TTF	code128a, code128b, and code128c
B39R00.TTF	code39, code39mod43
UPCR00.TTF	upca, upce

D

Update the Publisher Context Root

Topics:

- [Update the Publisher URL Context Root](#)
- [Example](#)

Update the Publisher URL Context Root

Change the default context to update the Publisher URL.

When you install Publisher with BI Server, by default the context for the Publisher URL is

```
http://<hostname>:<port>/xmlpserver
```

You change the default context to this:

```
http://<hostname>:<port>/<new context>/xmlpserver
```

1. Unzip the xmlpserver.ear file.
2. Update the following xmlpserver configuration files:
 - META-INF/application.xml
 - WAR/WEB-INF/web.xml
 - WAR/WEB-INF/weblogic.xml
 - \$DOMAIN_HOME/config/fmwconfig/biconfig/bipublisher/Admin/Configuration/xmlpserver-config.xml
3. Repackage the xmlpserver.ear.
4. Unzip the analytics.ear file.
5. Update the following analytics file:
 - META-INF/application.xml
6. Repackage the analytics.ear.
7. Update the instanceconfig.xml.
8. In WebLogic Server, update the bipublisher and analytics applications.

The following [Example](#) details the required updates in each file.

Example

This example details the required updates to change the Publisher context from `xmlpserver` to `/sales/xmlpserver`.

Perform these tasks:

- Update the xmlpserver META-INF/application.xml File
- Update the xmlpserver WAR/WEB-INF/web.xml File
- Update the xmlpserver WAR/WEB-INF/weblogic.xml File
- Update the xmlp-server-config.xml File
- Update the analytics META-INF/application.xml File
- Update the instanceconfig.xml File
- Update the bipublisher and analytics Applications in WebLogic Server

Update the xmlpserver META-INF/application.xml File

Update the context-root of the META-INF/application.xml file.

1. Unzip the xmlpserver.ear file.
2. Navigate to META-INF/application.xml under the xmlpserver WAR.
3. Update the context-root to match you new context. In this example the context root is updated to /sales/xmlpserver:

```
<web>
  <web-uri>xmlpserver.war</web-uri>
  <context-root>/sales/xmlpserver</context-root>
</web>
```

Update the xmlpserver WAR/WEB-INF/web.xml File

Under the xmlpserver WAR/WEB-INF folder, update the web.xml file.

1. Navigate to the WAR/WEB-INF/web.xml file.
2. Update the following parameter values in the file:

```
<init-param>
<!-- This is the root webdir for the xmlpserver application. Modify
this if
xmlpserver.ear is not deployed to its standard location. -->
  <param-name>xmlp-online-web-dir</param-name>
  <param-value>/sales/xmlpserver</param-value>
</init-param>
<init-param>
<!-- Path to the ServiceGateway SOAP end point. Most likely this
will be the
path for services deployed with Axis. -->
  <param-name>service-endpoint</param-name>
  <param-value>/sales/xmlpserver/services/ServiceGateway</param-
value>
</init-param>
<init-param> <!-- Path to report service web directory. -->
  <param-name>web-dir</param-name>
  <param-value>/sales/xmlpserver/report_service</param-value>
</init-param>
```

Update the xmlpserver WAR/WEB-INF/weblogic.xml File

Under the xmlpserver WAR/WEB-INF folder, update the weblogic.xml file.

1. Navigate to the WAR/WEB-INF/weblogic.xml file.
2. Update the cookie-path and context-root in the file:

```
<wls:session-descriptor>
  <wls:cookie-path>/sales/xmlpserver</wls:cookie-path>
</wls:session-descriptor>
<wls:context-root>sales/xmlpserver</wls:context-root>
```

Update the xmlp-server-config.xml File

Update an element in the xmlp-server-config.xml file.

1. Navigate to:

```
Oracle_Home/user_projects/domains/bi/config/fmwconfig/biconfig/bipublisher/
Admin/Configuration/xmlp-server-config.xml
```

2. Update the following element in the file:

```
<property name="SAW_URL_SUFFIX" value="sales/analytics/saw.dll"/>
```

Update the analytics META-INF/application.xml File

Update the elements in the file to match your context-root.

1. Unzip the analytics.ear file.
2. Navigate to the META-INF/application.xml file.
3. Update the following elements to match your context-root:

```
<display-name>analytics</display-name>
<module>
  <web>
    <web-uri>analytics.war</web-uri>
    <context-root>sales/analytics</context-root>
  </web>
</module>
<module>
  <web>
    <web-uri>analytics-ws.war</web-uri>
    <context-root>sales/analytics-ws</context-root>
  </web>
</module>
<module>
  <web>
    <web-uri>analytics.war</web-uri>
    <context-root>sales/analytics-bi-adf</context-root>
  </web>
</module>
```

Update the instanceconfig.xml File

Update the instanceconfig.xml file.

1. Navigate to the instanceconfig.xml located at

```
[ORACLE_INSTANCE]
\config\OracleBIPresentationServicesComponent\coreapplication_obips1\i
nstanceconfig.xml
```

2. Update the <ServerBasedURL> and <WebURL> elements under <AdvancedReporting> in the file:

```
<AdvancedReporting>
<ServerBasedURL>/sales/xmlpserver</ServerBasedURL>
<WebURL>/sales/xmlpserver</WebURL>
</AdvancedReporting>
```

Update the bipublisher and analytics Applications in WebLogic Server

Update the bipublisher and analytics applications in the Oracle WebLogic Server Administration Console.

1. Repackage the xmlpserver.ear file.
2. Repackage the analytics.ear file.
3. Open your Oracle WebLogic Server Administration Console.
4. In the Change Center of the Administration Console, click **Lock & Edit**.
5. In the left pane of the Console, select **Deployments**. A table in the right pane displays all deployed Enterprise Applications and Application Modules.
6. In the table, select the bipublisher application.
7. Click **Update**.
8. Click **Finish** (do not change the source path).
9. Repeat Step 6 through Step 8 for the analytics application.
10. In the **Change Center** of the **Administration Console**, click **Activate Changes** and then click **Release Configuration**.

E

Use Command-Line Utilities

You can run the `GenerateBIPUtility` script to generate the utilities for configuring the memory guard properties and for managing the catalog.

Topics:

- [Generate the Utilities](#)
- [Configure Memory Guard Properties Using Utility](#)

Generate the Utilities

Use the `GenerateBIPUtility` script to generate the `BIPConfigService.zip` and `BIPCatalogUtil.zip` files to configure the memory guard properties and the catalog.

Syntax

```
sh GenerateBIPUtility.sh toolname outputzipdestinationpath
```

where

- *toolname* (Mandatory) specifies either:
 - `configserviceutil` to generate the `BIPConfigService.zip` file.
 - `catalogutil` to generate the `BIPCatalogUtil.zip` file.
 - *Outputzipdestinationpath* (Optional) specifies the path to store the zip file. If you do not provide the *outputzipdestinationpath* path, the zip file will be created in the `BI_HOME/clients/bipublisher` folder.
1. Set `JAVA_HOME=/u01/app/4.0.0/jdk`.
 2. Navigate to `Oracle_Home/bi/clients/bipublisher/utility/bin`.
 3. Run the `GenerateBIPUtility` script from bash.
 - To generate the `BIPConfigService.zip` file, run the following command:

```
sh GenerateBIPUtility.sh configserviceutil
```

You can configure memory guard properties using `BIPConfigService`. See [Configure Memory Guard Properties Using Utility](#).
 - To generate the `BIPCatalogUtil.zip` file in `BI_Home/clients/bipublisher`, run the following command:

```
sh GenerateBIPUtility.sh catalogutil
```

You can manage catalogs using `BIPCatalogUtil`. See [Move Catalog Objects Between Environments](#).

Configure Memory Guard Properties Using Utility

You can use the `runtimepropertiesconfig.sh` command-line utility to configure the memory guard properties to protect against the out-of-memory errors that can occur while processing reports.

Syntax

`runtimepropertiesconfig.sh Operation Options`

where

Operation: update, get, or help

Options for update operation : `KEY1=VALUE1,KEY2=VALUE2`

Options for get operation: `KEY1,KEY2`

Examples

- Command to update the following memory guard properties:
 - `server.ONLINE_REPORT_MAX_DATA_SIZE` property to change the maximum report data size for online reports from the default value of 300MB to 223MB.
 - `server.SQL_QUERY_TIMEOUT` property to change the timeout of SQL query to 550 seconds from the default value of 600 seconds.

```
./runtimepropertiesconfig.sh update
server.ONLINE_REPORT_MAX_DATA_SIZE=223MB,server.SQL_QUERY_TIMEOUT=550
```

- Command to list the values of all memory guard properties:

```
./runtimepropertiesconfig.sh get
```

- Command to list the values of specified memory guard properties:

```
./runtimepropertiesconfig.sh get
server.ONLINE_REPORT_MAX_DATA_SIZE,server.SQL_QUERY_TIMEOUT
```

- Command to list all the memory guard properties along with the default values:

```
./runtimepropertiesconfig.sh help
```

1. Set the environment variable.

For example, `export JAVA_HOME=/home/jdk/jdk1.8.0_40`. By default, `JAVA_HOME=$BI_HOME/jdk`.

2. Navigate to `Oracle_Home/bi/clients/bipublisher/utility/bin`.
3. Run the following command to run the `GenerateBIPUtility` script from bash and generate the `BIPConfigService.zip` file.

```
sh GenerateBIPUtility.sh configserviceutil
```

4. Unzip the `BIPConfigService.zip` file.

```
cd <BI_HOME>/modules
unzip -d BIPConfigService BIPConfigService.zip
```

5. Change directory to the location of the runtimepropertiesconfig.sh command line utility.

```
cd <BI_HOME>/modules/BIPConfigService/bin
```

6. Provide the path for <BI_DOMAIN_HOME> when the utility prompts.

For example: /user_projects/domains/bidomain/

Memory Guard Properties

Configure the memory guard properties to protect against out-of-memory errors.

Use the runtimepropertiesconfig.sh command-line utility to configure the memory guard properties. See [Configure Memory Guard Properties Using Utility](#).

Property	Description
server.BURSTING_REPORT_MAX_DATA_SIZE	Maximum report data size for bursting reports Default value: 500MB
server.DB_FETCH_SIZE	DB fetch size Default value: 20
server.FREE_MEMORY_THRESHOLD	Free memory threshold Default value: 500MB
server.MAX_DATA_SIZE_UNDER_FREE_MEMORY_THRESHOLD	Maximum report data size under the free memory threshold Default value: free_memory_threshold/10
server.MAX_ROWS_FOR_CSV_OUTPUT	Maximum rows for CSV output Default value: 1000000
server.MAX_SAMPLE_XML_DATA_SIZE_LIMIT	Maximum sample data size limit Default value: 1MB
server.MINIMUM_SECONDS_BETWEEN_GARBAGE_COLLECTIONS	Minimum time span between garbage collection runs Default value: 300 (seconds)
server.OFFLINE_REPORT_MAX_DATA_SIZE	Maximum report data size for offline (scheduled) reports Default value: 500MB
server.ONLINE_REPORT_MAX_DATA_SIZE	Maximum report data size for online reports Default value: 300MB
server.ONLINE_REPORT_TIMEOUT	Timeout for online reports. Default value: 600 (seconds)
server.SQL_QUERY_TIMEOUT	SQL Query Timeout Default value: 600 (seconds)
server.WAIT_SECONDS_FOR_FREE_MEMORY	Maximum wait time for free memory to come back above the threshold value Default value: 30 (seconds)
server.XML_DATA_SIZE_LIMIT	Maximum data size limit for data generation Default value: 500MB