

Oracle® Business Intelligence Publisher

New Features Guide

Release 10.1.3.4.1

E14667-01

April 2009

This document describes the following new features added in the Oracle Business Intelligence Publisher release 10.1.3.4.1:

- Section 1, "Support for Oracle WebLogic Server 10.3"
- Section 2, "Extended Support for Single Sign-On Providers"
- Section 3, "Automatic Refresh of LDAP Cache"
- Section 4, "Support for Siebel CRM Security"
- Section 5, "Support for Secure File Transfer Protocol (SFTP) for Burst Reports"
- Section 6, "Data Model Enhancements"
- Section 7, "Support for JNDI for Scheduler Connections"
- Section 8, "Support for Expressions to Calculate Date Parameters"
- Section 9, "RTF Template Enhancements"
- Section 10, "Remove Logos and Links from the BI Publisher Header"
- Section 11, "Disable Access to Guest Page"
- Section 12, "Updates to the BI Publisher Web Service"

1 Support for Oracle WebLogic Server 10.3

The 10.1.3.4.1 release of BI Publisher introduces support for Oracle WebLogic Server 10.3. Follow the manual deployment instructions below to deploy BI Publisher to WebLogic Server 10.3. You will also need the *Oracle Business Intelligence Publisher Installation Guide* to complete the configurations required for the deployment.

1.1 Deploying BI Publisher to Oracle WebLogic Server 10.3

Note: Do not deploy the `xmlpserver.war` or `xmlpserver.ear` file on the WebLogic Server by uploading it from the WebLogic console, because the console deploys the application (or Web module) in an archived file format. This is problematic to a BI Publisher configuration, because you must update `WEB-INF/xmlp-server-config.xml` manually after the deployment. To work around this issue, use an "exploded archive" directory, as described in this procedure.

1. Create a new WebLogic Server domain using a JDK such as Sun JDK 1.5.x or higher.
2. Create an exploded archive directory from the xmlpserver.ear or xmlpserver.war file using the new domain, by following these steps:

- a. Copy the xmlpserver.ear or xmlpserver.war file in the \xmlpserver\web directory to your destination directory, as shown in the following example.

```
-mkdir c:mydestination\xmlpserver\  
-cp c:\xmlpserver\web\xmlpserver.war to  
c:mydestination\xmlpserver\  

```

- b. Manually unpack the xmlpserver.ear or xmlpserver.war file in your destination folder using a jar command, as shown in the following example.

```
cd c:mydestination\xmlpserver\  
jar -xvf xmlpserver.war
```

This is the directory where the application will be deployed.

3. Set up the BI Publisher repository.

To set up the BI Publisher repository, copy the XMLP repository to a location on your server, then edit the configuration file to point to the saved location, as follows:

- a. Copy the XMLP repository directory from \manual on your installation media to your server.

- b. Open the xmlp-server-config.xml file located in the c:mydestination\xmlpserver\WEB-INF\ directory where you unpacked the ear or war file.

- c. Replace \${oracle.home}/xdo/repository with the file path to the location where you copied the XMLP repository directory on your server. For example, if you copied the repository directory to /home/repository/XMLP, update the xmlp-server-config.xml file as follows:

```
<?xml version="1.0" encoding="UTF-8"?>  
<xmlpConfig xmlns="http://xmlns.oracle.com/oxp/xmlp">  
<resource>  
<file path="/home/repository/XMLP"/>  
</resource>  
</xmlpConfig>
```

- d. Save xmlp-server-config.xml after updating.

4. In a browser, open the WebLogic Administration console (for example: <http://hostname:7001/console>).
5. If you have not already done so, in the **Change Center** of the Administration Console, click **Lock & Edit**.
6. In the left pane of the Administration Console, click **Deployments**.
7. In the right pane, click **Install**.
8. Select c:mydestination\xmlpserver\xmlpserver and click **Next**.
9. Select "Install this deployment as an application" and click **Next**.
10. Select "I will make the deployment accessible from the following location" from Source accessibility.
11. Click **Finish**.

12. To activate these changes, in the **Change Center** of the Administration Console click **Activate Changes**.
13. Install the BI Publisher fonts as described in the section "Configuring the BI Publisher Repository and Installing Fonts," in the *Oracle Business Intelligence Publisher Installation Guide*.

1.2 Required Settings for the WebLogic Start Script

WebLogic Server 10.3 includes two implementations of SAAJ (SOAP with Attachments API for Java). In order for BI Publisher to work properly, you must ensure that your WebLogic Server start script points to the SOAP Message Factory in the SAAJ 1.3 implementation. The SAAJ 1.3 implementation is found in package `weblogic.xml.saa`.

To ensure that your deployment is using the correct version, include the following setting for the domain that contains the BI Publisher server in the `/weblogic_home/user_projects/domains/base_domain/bin/startWebLogic.sh` script:

```
Djavax.xml.soap.MessageFactory=weblogic.xml.saa.MessageFactoryImpl
```

2 Extended Support for Single Sign-On Providers

In previous versions of BI Publisher only Oracle Single Sign-On was supported.

Earlier versions of BI Publisher also required that if an administrator configured BI Publisher with Oracle Single Sign-On that BI Publisher must also use Oracle Internet Directory as the security model. This restriction is removed in 10.1.3.4.1 of BI Publisher. Customers may now use any of the supported security models when implementing Single Sign-On. Please note that it is the customer's responsibility to ensure that the user community of the security model matches the user community from the Single Sign-On server.

To configure BI Publisher to use Single Sign-On you must enter your Single Sign-On server information in BI Publisher's Security Configuration page under the Admin tab. You will need to provide details on how the Single Sign-On server passes the user name and the user locale to BI Publisher as well as the parameter names for both user name and user locale. For the certified Single Sign-On servers, the configuration page will provide default values for these configuration settings.

2.1 Configuring Single Sign-On for BI Publisher

1. Create a BI Publisher Local Superuser.

Before performing any security updates, set up a BI Publisher Local Superuser to ensure access to BI Publisher regardless of your selected security configuration. See "Defining a Local Superuser" in the BI Publisher Help or in the *Oracle Business Intelligence Publisher Administrator's and Developer's Guide* for more information.

2. Configure the Single Sign-On server to include BI Publisher as a partner or protected application. See your Single Sign-On server documentation for details.
3. BI Publisher requires that certain locations are unprotected by your Single Sign-On application to enable communication between the BI Publisher server and external components. To enable the following services, you must unprotect the specified location. Follow the guidelines in your Single Sign-On server documentation for unprotecting specific locations.

Location to Unprotect	Needed For:
/xmlpserver/services/	Web service communication between BI Publisher and its client components: Template Builder and Excel Analyzer Web service communication between BI Publisher and BI Presentation Services
/xmlpserver/report_service/	Also needed for Web service communication between BI Publisher and its client components: Template Builder and Excel Analyzer
/xmlpserver/ReportTemplateService.xls/	Web service communication between BI Publisher and Excel Analyzer
/xmlpserver/Guest/	Guest folder access to users not signed on through SSO

4. Log in to BI Publisher with Administrator privileges. Navigate to the **Admin tab**, then select **Security Configuration**.
5. In the Single Sign-On region of the page, select **Use Single Sign-On**.
6. In the details of the Single Sign-On region enter the following:
 - Select from the certified types of certified Single Sign-On servers:
 - Oracle Single Sign On
 - Oracle Access Manager
 - Siteminder
 - Optionally enter the **Single Sign-Off URL**.
 - Based on the type of server you selected, the following fields will be automatically populated with default values:
 - **How to get Username**
 - **User Name Parameter**
 - **How to get User Locale**
 - **User Locale Parameter**
7. If your Single Sign-On server is configured to use different parameters or methods, you can delete the default values and enter the appropriate values for your system.
8. Restart BI Publisher.

3 Automatic Refresh of LDAP Cache

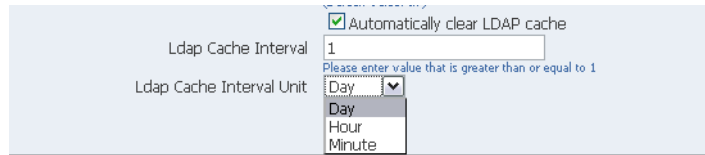
The LDAP Security Model configuration has been enhanced to enable you to schedule the automatic refresh of the LDAP cache. Previously in order to refresh the LDAP cache, the administrator had to navigate to the Refresh Metadata function and click the Synchronize button. With this enhancement, the LDAP cache can be automatically refreshed per a designated interval.

To activate the LDAP cache refresh:

1. Navigate to the Security Configuration page By clicking the **Admin tab**, then choosing **Security Configuration**.

2. Scroll down to your LDAP settings.

The three new fields added to support this new feature are shown in the following figure:



The screenshot shows a configuration interface for LDAP settings. It includes a checkbox labeled "Automatically clear LDAP cache" which is checked. Below it is a text input field for "Ldap Cache Interval" containing the value "1". A blue tooltip message reads "Please enter value that is greater than or equal to 1". Below the input field is a dropdown menu for "Ldap Cache Interval Unit" with "Day" selected. The dropdown menu also shows "Hour" and "Minute" as options.

3. Select **Automatically clear LDAP cache**.
4. Enter an integer for **Ldap Cache Interval**. For example, to clear the LDAP cache once a day, enter 1.
5. Select the appropriate **Ldap Cache Interval Unit**: Day, Hour, or Minute.
6. Restart BI Publisher.

4 Support for Siebel CRM Security

If you are using BI Publisher integrated with Siebel Customer Relationship Management (CRM) you can now use the Siebel security model for BI Publisher.

To configure BI Publisher to integrate with Siebel security, perform the following three general tasks (described in more detail below):

1. Set up BI Publisher roles as Siebel CRM responsibilities.
2. Configure BI Publisher to user Siebel Security.
3. Authorize Siebel responsibilities to access data sources and folders in BI Publisher.

4.1 Setting Up BI Publisher Roles as Siebel CRM Responsibilities

1. Using Siebel Administrator credentials, navigate to Administration - Application, and then Responsibilities.
2. In the Responsibilities list, add a new record for each of the BI Publisher functional roles:
 - XMLP_ADMIN - this is the administrator role for the BI Publisher.
 - XMLP_DEVELOPER - enables users to build reports in the system.
 - XMLP_SCHEDULER - enables users to schedule reports.
 - XMLP_ANALYZER_EXCEL - enables users to use the Excel analysis feature.
 - XMLP_ANALYZER_ONLINE - enables users to use the online analysis feature.
 - XMLP_TEMPLATE_BUILDER - enables users to connect to the BI Publisher server from the Template Builder for Microsoft Word and to upload and download templates directly from Microsoft Word.
3. Assign these responsibilities to the appropriate users. You may also want to create additional reporting responsibilities that you can utilize when setting up your report privileges in the BI Publisher application.

Ensure to assign the XMLP_ADMIN role to a user with administration privileges.

4.2 Configure BI Publisher to Use Siebel Security

1. In the BI Publisher application, log in with Administrator privileges. From the **Admin** tab select **Security Configuration**.
2. Define a Local Superuser in BI Publisher:
 - Under **Local Superuser**, select the box and enter the credentials for the Superuser.

The Local Superuser credentials will enable you to access BI Publisher in case of issues with the configured security model.

3. In the **Security Model** section of the page, select **Siebel Security** from the list.
4. Provide the following connection information:
 - Siebel Web Service Endpoint
 - Administrator Username
 - Administrator Password
5. Restart BI Publisher for the security changes to take effect.

4.3 Authorize Siebel Responsibilities to Access BI Publisher Data Sources and Folders

1. After you restart BI Publisher, log in to BI Publisher using the Siebel user credentials to which you assigned the XMLP_ADMIN role.
2. From the **Admin** tab, under **Security Center**, select **Roles and Permissions**. This will invoke the **Security Center** page. Here you can see the Siebel CRM responsibilities to which you assigned BI Publisher roles.
3. To add report folder access for a role, select the **Add Folders** icon. From the **Add Folders** page, use the shuttle buttons to select the appropriate folders for the role.
4. To add data source access for a role, select the **Add Data Sources** icon. From the **Add Data Sources** page, use the shuttle buttons to select the appropriate data sources for the role.

For more information about BI Publisher's roles and permissions, see the topic "Understanding BI Publisher's Users and Roles" in the *Oracle Business Intelligence Publisher Administrator's and Developer's Guide*.

5 Support for Secure File Transfer Protocol (SFTP) for Burst Reports

BI Publisher now supports Secure FTP (SFTP) as a delivery channel for burst reports. To enable SFTP for your burst report, follow the guidelines for defining the delivery data set for FTP, then specify "true" as the value for parameter 6.

The parameter mappings for the delivery data set are shown in the following table:

Channel	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
FTP	Server Name	User Name	Password	Remote Directory	Remote File Name	Secure Value: true

For the complete description of how to enable a report for bursting, see the topic: "Enabling Bursting" in the *Oracle Business Intelligence Publisher Report Designer's Guide*.

6 Data Model Enhancements

The following enhancements for Web service and SQL Query data set types are included in this release:

- [Section 6.1, "Web Service Data Set Type Enhancement: Specify Path to Data in SOAP Response"](#)
- [Section 6.2, "SQL Query Data Set New Property: Use Default Schema Only"](#)

6.1 Web Service Data Set Type Enhancement: Specify Path to Data in SOAP Response

When using a Web service as a data source, the start of the XML data that you wish to use for your report can often be deeply embedded in the response XML generated by the Web service request. This can make it difficult to work with the data in the BI Publisher template design tools.

In this release, a new field has been added to the Web Service data set type definition: **ResponseData XPath**. Use this field to specify the path to the data that you wish to use in your BI Publisher report.

The new field is shown in the following figure of the Web Service data set definition page:

The screenshot shows the 'Data Set' configuration page. Under 'General Settings', the Name is 'Complex Web Service' and the Type is 'Web Service'. Under 'Details', the Complex Type is 'True', WS-Security is 'Disabled', Username and Password are empty, Time Out is empty (in seconds), WSDL URL is 'http://bip.mycompany.com:9999/xmlpserver/service', Web Service is '-- Select --', Method is 'getScheduledReportInfo', and ResponseData XPath is '/getScheduledReportInfoResponse/getScheduledF'. Parameters are listed as '[name=ns1:userID], [type=string]' with a value of ':userID'.

6.2 SQL Query Data Set New Property: Use Default Schema Only

When building a SQL query data set type using BI Publisher's Query Builder, by default the schema name is prepended to the table name from which you are selecting data. In this release, a new option has been added to the SQL query data set type definition: **Only use default schema**. When you enable this option, the query builder constructs the query without the schema name prepended to the table name.

For example, the following figure shows a sample query with the new property not enabled. Note that the table names are specified with the schema name prepended: OE.DEPARTMENTS and OE.EMPLOYEES

The screenshot shows a query editor window with the following settings:

- General Settings:** Name: Query 1, Type: SQL Query
- Details:** Data Source: demo (selected), Cache Result: unchecked, Only use default schema: unchecked
- SQL Query:**

```
select  EMPLOYEES.FIRST_NAME as FIRST_NAME,
        EMPLOYEES.LAST_NAME as LAST_NAME,
        EMPLOYEES.DEPARTMENT_ID as DEPARTMENT_ID,
        DEPARTMENTS.DEPARTMENT_NAME as DEPARTMENT_NAME
from    OE.DEPARTMENTS DEPARTMENTS,
        OE.EMPLOYEES EMPLOYEES
where   EMPLOYEES.DEPARTMENT_ID=DEPARTMENTS.DEPARTMENT_ID
```

The next figure shows the same query with the new property enabled. Note that the table names are now specified as DEPARTMENTS and EMPLOYEES.

The screenshot shows the same query editor window with the following settings:

- General Settings:** Name: Query 1, Type: SQL Query
- Details:** Data Source: demo (selected), Cache Result: unchecked, Only use default schema: checked
- SQL Query:**

```
select  EMPLOYEES.FIRST_NAME as FIRST_NAME,
        EMPLOYEES.LAST_NAME as LAST_NAME,
        EMPLOYEES.DEPARTMENT_ID as DEPARTMENT_ID,
        DEPARTMENTS.DEPARTMENT_NAME as DEPARTMENT_NAME
from    DEPARTMENTS DEPARTMENTS,
        EMPLOYEES EMPLOYEES
where   EMPLOYEES.DEPARTMENT_ID=DEPARTMENTS.DEPARTMENT_ID
```

Enable this property when your development environment schema may be different from your production environment schema. Enabling this property makes your query portable against different schema.

Important Considerations

Because the query no longer includes the schema identifier, you cannot use this property if your query requires table joins across different schema.

Because the Query Builder builds the query based on the setting of this option, ensure to enable this property before you build the query with the Query Builder.

7 Support for JNDI for Scheduler Connections

In environments with many simultaneous scheduled jobs or requests to display completed reports from job history, the JDBC connection to the scheduler database may get overwhelmed causing performance to be sluggish. To enable better performance the scheduler database now supports using a JNDI connection pool defined on the application server.

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 via Java Naming and Directory Interface (JNDI).

After you set up the connection pool in your application server, enter the required fields in the Scheduler Configuration page. For information on setting up a connection pool in OC4J, see the chapter "Data Sources" in the *Oracle Containers for J2EE Services Guide 10g*.

To connect the Scheduler to your database via a JNDI connection pool

Perform the following:

1. Define a JNDI data source in your application server. Select JDBC connection pool and set an appropriate number of connections for the pool size. The optimum number of connections will depend on the volume of report scheduling required at your site.
2. From the BI Publisher **Admin** tab, navigate to the **Scheduler Configuration** page.
3. For the **Database Connection Type**, select **jndi**.
4. Enter the **JNDI Name** that you defined on your application server.
5. Test the connection. If it is successful, select **Install Schema** to install the Scheduler schema to your database.
6. Restart BI Publisher.

8 Support for Expressions to Calculate Date Parameters

Previously when you created a schedule for a report that included date parameters in the data model, each time the report ran under the schedule, the original entries for the data parameters could not be updated. Now you can create a schedule for a report with date parameters and use a function in the data parameter field to programmatically calculate the dates each time the report runs under that schedule.

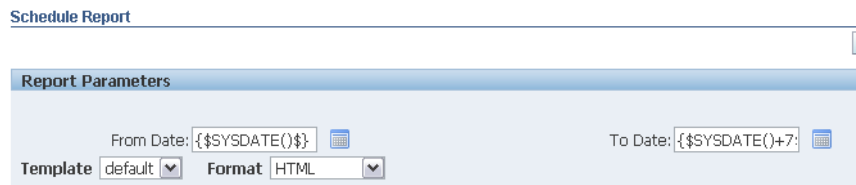
Enter one of the following functions using the syntax shown to calculate the appropriate date at the scheduled runtime for the report:

- `{ $SYSDATE () $ }` - current date (the system date of the server on which BI Publisher is running)
- `{ $FIRST_DAY_OF_MONTH () $ }` - first day of the current month
- `{ $LAST_DAY_OF_MONTH () $ }` - last day of the current month
- `{ $FIRST_DAY_OF_YEAR) $ }` - first day of the current year
- `{ $LAST_DAY_OF_YEAR) $ }` - last day of the current year

Note that you can also enter expressions using the plus sign "+" and minus sign "-" to add or subtract days as follows:

- `{ $SYSDATE () +1 $ }`
- `{ $SYSDATE () -7 $ }`

The following figure shows a sample of the Scheduler page with the date expressions entered for the parameter values:



The date function calls in the parameter values are not evaluated until the report is executed by the Scheduler.

The date functions can also be set up as default parameter values in the data model. In this case, every time a user views the report from the report viewer, the date parameters will be calculated according to the expression supplied for the default value.

9 RTF Template Enhancements

This release includes the following enhancements for RTF template development:

- [Section 9.1, "Get List of BI Publisher Configuration Settings"](#)
- [Section 9.2, "Enable Debug Mode for an RTF Template"](#)
- [Section 9.3, "Number to Word Conversion in Report Output"](#)

9.1 Get List of BI Publisher Configuration Settings

This release introduces a new function that can be inserted into an RTF template to return the list of configuration properties and values.

Insert the following command in an RTF template to return the list of configuration settings:

```
<?xdoxslt:getXDOProperties ($_XDOCTX) ?>
```

Note that if the report has been configured to override the system settings, the report-specific settings will be returned.

9.2 Enable Debug Mode for an RTF Template

This release introduces a new command for RTF templates to enable debug mode for the specific template that contains the command. Setting this property during testing of a template will enable you to receive targeted error information about a specific template without impacting system debug settings.

The new command is `<?xdo-debug-level: debug_level?>` where the valid values are:

```
<?xdo-debug-level: "STATEMENT" ?>
<?xdo-debug-level: "ERROR" ?>
<?xdo-debug-level: "OFF" ?>
```

Note that *debug_level* is not case sensitive, `<?xdo-debug-level: "STATEMENT" ?>` and `<?xdo-debug-level: "statement" ?>` are both valid.

The debug log directory will be generated in one of the following locations:

- `<system-temp-dir>/xdodebug`
- `<java.io.tmpdir>/xdodebug` (if `system-temp-dir` is not defined)

Note: This command can also be used in an XSL template or FO file by adding the following comment:

```
<!-- xdo-debug-level="STATEMENT" -->
```

The comment must reside within the first 512 bytes in the file.

9.3 Number to Word Conversion in Report Output

This release introduces a function that enables the conversion of numbers to words for RTF template output. This is a common requirement for check printing.

The new function is "to_check_number". The syntax of this function is

```
<?xdo:to_check_number(amount, precisionOrCurrency, caseType, decimalStyle)?>
```

The following table describes the function attributes:

Attribute	Description	Available Value
amount	The number to be transformed.	Any number
precisionOrCurrency	For this attribute you can specify either the precision, which is the number of digits after the decimal point; or the currency code, which will govern the number of digits after the decimal point. The currency code does not generate a currency symbol in the output.	An integer, such as 2; or a currency code, such as 'USD'.
caseType	The case type of the output.	Valid values are: 'CASE_UPPER', 'CASE_LOWER', 'CASE_INIT_CAP'

Attribute	Description	Available Value
decimalStyle	Output type of the decimal fraction area.	Valid values are: 'DECIMAL_STYLE_FRACTION1', 'DECIMAL_STYLE_FRACTION2', 'DECIMAL_STYLE_WORD'

The following examples display the function as entered in an RTF template and the returned output:

RTF Template Entry: <?xdofx:to_check_number(12345.67, 2)?>

Returned Output: Twelve thousand three hundred forty-five and 67/100

RTF Template Entry: <?xdofx:to_check_number(12345.67, 'USD')?>

Returned Output: Twelve thousand three hundred forty-five and 67/100

RTF Template Entry: <?xdofx:to_check_number(12345, 'JPY', 'CASE_UPPER')?>

Returned Output: TWELVE THOUSAND THREE HUNDRED FORTY-FIVE

RTF Template Entry: <?xdofx:to_check_number(12345.67, 'EUR', 'CASE_LOWER', 'DECIMAL_STYLE_WORDS')?>

Returned Output: twelve thousand three hundred forty-five and sixty-seven

10 Remove Logos and Links from the BI Publisher Header

If you are integrating BI Publisher into a custom application, you may want to remove the BI Publisher header to integrate more seamlessly into your enterprise applications.

In this release, BI Publisher introduces a configuration property to enable you to remove the header information from the BI Publisher application.

To remove the BI Publisher header region:

1. Navigate to the configuration file located at: <BI Publisher repository>/Admin/Configuration/xmlp-server-config.xml
2. Add the following property and value to the file:

```
<property name="DISPLAY_HEADER_LOGO_MENU" value="false" />
```
3. Restart BI Publisher.

11 Disable Access to Guest Page

Currently, if you have activated Guest user access, then any user can access the BI Publisher Guest report page. If your company does not wish to expose the BI Publisher Guest report page to all users, you can now restrict access to the guest page to only authorized users.

This feature enables you to restrict access by defining a mandatory role that users must have to access the Guest page.

1. Navigate to the <BI Publisher repository>/Admin/Configuration/xmlp-server-config.xml file.
2. Add the following property to the file:

```
<property name="MANDATORY_USER_ROLE" value="ROLE_NAME"/>
```

where ROLE_NAME is the name of the role that is required for a user to access the guest page.

3. Restart BI Publisher server.

Any user without this new role who attempts to access the BI Publisher Guest log in page will receive the HTTP 403 Forbidden Error.

Note: Once an unauthorized user has attempted to log in, the browser must be closed before an authorized user can log in.

12 Updates to the BI Publisher Web Service

In release 10.1.3.4.1, BI Publisher introduces a new version of PublicReportService, called PublicReportService_v11. This Web service includes all the functionality available in PublicReportService, plus introduces additional features.

If you have implemented PublicReportService and do not wish to make any changes to your existing implementation, you can continue to use PublicReportService when you uptake BI Publisher 10.1.3.4.1.

If this is the first time you are using the BI Publisher Web service, use PublicReportService_v11 to get the latest functionality.

This section describes the changes between PublicReportService (as documented in release 10.1.3.4) and PublicReportService_v11. For the full description of PublicReportService, see the *Oracle Business Intelligence Publisher Administrator's and Developer's Guide*.

12.1 Summary of Changes

This topic contains the following sections to describe the changes between PublicReportService and PublicReportService_v11:

- [Section 12.2, "New Complex Types"](#)
- [Section 12.3, "Changed Complex Types"](#)
- [Section 12.4, "New Operations"](#)
- [Section 12.5, "Changed Operations"](#)
- [Section 12.6, "Using the InSession Operations"](#)

12.2 New Complex Types

The following complex types are new in PublicReportService_v11:

- BIPDataSource
- JDBCDataSource
- FileDataSource
- DeliveryServiceDefinition
- TemplateFormatLabelValue
- ArrayOfTemplateFormatLabelValue

- TemplateFormatsLabelValues
- ArrayOfTemplateFormatsLabelValues

12.2.1 BIPDataSource

The new complex type BIPDataSource has been added to enable you to dynamically specify a data source when using the runReport operation. The definition is as follows:

```
<complexType name="BIPDataSource">
  <sequence>
    <element name="JDBCDataSource" nillable="true"
type="impl:JDBCDataSource"/>
    <element name="fileDataSource" nillable="true"
type="impl:FileDataSource"/>
  </sequence>
</complexType>
```

BIPDataSource is used by the [ReportRequest](#) structure.

The following table describes the elements:

Parameter	Type	Description
JDBCDataSource	JDBCDataSource	Contains the elements to describe a JDBC data source. See Section 12.2.2 , "JDBCDataSource."
fileDataSource	FileDataSource	Contains the elements to describe a file data source. See Section 12.2.3 , "FileDataSource."

12.2.2 JDBCDataSource

The new complex type JDBCDataSource has been added to enable you to specify a JDBC data source. The elements that make up the JDBCDataSource structure are as follows:

```
<complexType name="JDBCDataSource">
  <sequence>
    <element name="JDBCDriverClass" nillable="true" type="xsd:string"/>
    <element name="JDBCDriverType" nillable="true" type="xsd:string"/>
    <element name="JDBCPassword" nillable="true" type="xsd:string"/>
    <element name="JDBCURL" nillable="true" type="xsd:string"/>
    <element name="JDBCUserName" nillable="true" type="xsd:string"/>
    <element name="dataSourceName" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
```

This structure is used by the [BIPDataSource](#) structure.

The elements are described in the following table:

Parameter	Type	Description
JDBCDriverClass	string	The JDBC driver class for the data source, for example: "oracle.jdbc.OracleDriver"
JDBCDriverType	string	The driver type as defined in the BI Publisher data source definition page, for example: "Oracle 9i/10g/11g".
JDBCPassword	string	The password for the data source as defined in the BI Publisher data source definition page.

Parameter	Type	Description
JDBCURL	string	The connection string for the data source, for example: "jdbc:oracle:thin:@mydatabase.mycompany.com:1521:orcl".
JDBCUserName	string	The user name for the data source as defined in the BI Publisher data source definition page.
dataSourceName	string	The Data Source Name assigned to the data source in the BI Publisher data source definition page, for example: "Oracle".

12.2.3 FileDataSource

The new complex type FileDataSource has been added to specify a file data source for your report. You can specify a direct path to a location on your server, or indicate that the file is in the temporary directory. The elements that make up FileDataSource are as follows:

```
<complexType name="FileDataSource">
  <sequence>
    <element name="dynamicDataSourcePath" nillable="true"
type="xsd:string"/>
    <element name="temporaryDataSource" type="xsd:boolean"/>
  </sequence>
</complexType>
```

This structure is used by the [BIPDataSource](#) structure.

The elements are described in the following table:

Parameter	Type	Description
dynamicDataSourcePath	string	If you want to specify a path to a data source that resides on an available server, specify the full path to the data source and set temporaryDataSource to "false". For example: "D:\BI\OracleBI\xmlp\XMLP\DemoFiles\Balance.xml")" If the file is located in the system temporary directory, set temporaryDataSource to true, and specify the file name here. For example: "Balance.xml".
temporaryDataSource	boolean	Set to true when the file data source is in the system temporary directory. Set to false when dynamicDataSourcePath specifies the full path.

12.2.4 DeliveryServiceDefinition

The new complex type DeliveryServiceDefinition is returned in the response for the [getDeliveryServiceDefinition](#) operation. This operation enables you to obtain information about the delivery servers set up for BI Publisher. The definition is as follows:

```
<complexType name="DeliveryServiceDefinition">
  <sequence>
    <element name="EmailServerNames" nillable="true"
type="tns1:ArrayOfString"/>
    <element name="FTPServerNames" nillable="true"
type="tns1:ArrayOfString"/>
    <element name="HTTPServerNames" nillable="true"
```

```

type="tns1:ArrayOfString" />
    <element name="SFTPServerNames" nillable="true"
type="tns1:ArrayOfString" />
    <element name="defaultServerNames" nillable="true"
type="tns1:ArrayOfString" />
    <element name="faxServerNames" nillable="true"
type="tns1:ArrayOfString" />
    <element name="printerNames" nillable="true" type="tns1:ArrayOfString" />
    <element name="webDAVServerNames" nillable="true"
type="tns1:ArrayOfString" />
    </sequence>
</complexType>

```

The following table describes the elements:

Parameter	Type	Description
EEmailServerNames	ArrayOfString	The list of e-mail server names returned in the ArrayOfString structure.
FTPServerNames	ArrayOfString	The list of FTP server names returned in the ArrayOfString structure.
HTTPServerNames	ArrayOfString	The list of HTTP server names returned in the ArrayOfString structure.
SFTPServerNames	ArrayOfString	The list of SFTP server names returned in the ArrayOfString structure.
defaultServerNames	ArrayOfString	The list of the default server names for each type defined returned in the ArrayOfString structure.
faxServerNames	ArrayOfString	The list of fax server names returned in the ArrayOfString structure.
printerNames	ArrayOfString	The list of printer names returned in the ArrayOfString structure.
webDAVServerNames	ArrayOfString	The list of WebDAV server names returned in the ArrayOfString structure.

12.2.5 TemplateFormatLabelValue

The new complex type TemplateFormatLabelValue has been added to specify the template format labels and values for a report. TemplateFormatLabelValue is included in the "[ArrayOfTemplateFormatLabelValue](#)" structure. The elements that make up TemplateFormatLabelValue are as follows:

```

<complexType name="TemplateFormatLabelValue">
    <sequence>
        <element name="templateFormatLabel" nillable="true" type="xsd:string" />
        <element name="templateFormatValue" nillable="true" type="xsd:string" />
    </sequence>
</complexType>

```

The elements are described in the following table:

Parameter	Type	Description
templateFormatLabel	string	The label that displays for a template format. Examples include: "HTML" "PDF" "Excel"
templateFormatValue	string	The template format value that corresponds to the label. Examples include: "html" "pdf" "excel"

12.2.6 ArrayOfTemplateFormatLabelValue

The new complex type `ArrayOfTemplateFormatLabelValue` is an array structure to include the `TemplateFormatLabelValue` label-value pairs.

`ArrayOfTemplateFormatLabelValue` is included in the [TemplateFormatsLabelValues](#) structure. The elements that make up `ArrayOfTemplateFormatLabelValue` are as follows:

```
<complexType name="ArrayOfTemplateFormatLabelValue">
  <sequence>
    <element name="item" type="tns1:TemplateFormatLabelValue" minOccurs="0"
maxOccurs="unbounded" />
  </sequence>
</complexType>
```

The elements are described in the following table:

Parameter	Type	Description
item	TemplateFormatLabelValue	Contains the <code>TemplateFormatLabelValue</code> label-value pairs. See Section 12.2.5, "TemplateFormatLabelValue"

12.2.7 TemplateFormatsLabelValues

`TemplateFormatsLabelValues` is a new complex type that provides detailed information about template formats stored in the BI Publisher repository.

`TemplateFormatsLabelValues` is included in the [ArrayOfTemplateFormatsLabelValues](#) structure.

The elements that make up `TemplateFormatsLabelValues` are as follows:

```
<complexType name="TemplateFormatsLabelValues">
  <sequence>
    <element name="listOfTemplateFormatLabelValue" nillable="true"
type="tns1:ArrayOfTemplateFormatLabelValue" />
    <element name="templateAvailableLocales" nillable="true"
type="tns1:ArrayOfString" />
    <element name="templateID" nillable="true" type="xsd:string" />
    <element name="templateType" nillable="true" type="xsd:string" />
    <element name="templateURL" nillable="true" type="xsd:string" />
  </sequence>
</complexType>
```

The elements are described in the following table:

Parameter	Type	Description
listOfTemplateFormatLabelValue	ArrayOfTemplateFormatLabelValue	Contains the TemplateFormatLabelValue label-value pairs. See Section 12.2.6 , "ArrayOfTemplateFormatLabelValue"
templateAvailableLocales	ArrayOfString	The available locale options defined for a template passed in the ArrayOfString object.
templateID	string	The name assigned to the template in BI Publisher, for example: "Employee Listing".
templateType	string	The type of template, for example: "rtf" or "pdf".
templateURL	string	The template file name in the BI Publisher repository, for example: "Employee Listing.rtf".

12.2.8 ArrayOfTemplateFormatsLabelValues

The new complex type ArrayOfTemplateFormatsLabelValues is a structure to include an array of the TemplateFormatsLabelValues structure.

ArrayOfTemplateFormatsLabelValues is included in the ReportDefinition structure to contain the specific fields to describe the available template formats. The elements that make up ArrayOfTemplateFormatsLabelValues are as follows:

```
<complexType name="ArrayOfTemplateFormatsLabelValues">
  <sequence>
    <element name="item" type="impl:TemplateFormatsLabelValues"
minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

The elements are described in the following table:

Parameter	Type	Description
item	TemplateFormatsLabelValues	Contains the fields that describe the available template formats. See Section 12.2.7 , "TemplateFormatsLabelValues"

12.3 Changed Complex Types

The following complex types have changed in PublicReportService_v11:

- DeliveryRequest
- EmailDeliveryOption
- ParamNameValue
- PrintDeliveryOption
- ReportRequest
- ReportResponse

12.3.1 DeliveryRequest

The complex type DeliveryRequest provides the options to deliver a report via the specified delivery methods. Three new elements are added for PublicReportService_v11. The complete complex type definition is as follows:

```
<complexType name="DeliveryRequest">
  <sequence>
    <element name="contentType" nillable="true" type="xsd:string"/>
    <element name="documentData" nillable="true" type="xsd:base64Binary"/>
    <element name="dynamicDataSource" nillable="true" type="impl:BIPDataSource"/>
    <element name="emailOption" nillable="true" type="impl:EMailDeliveryOption"/>
    <element name="faxOption" nillable="true" type="tns1:FaxDeliveryOption"/>
    <element name="ftpOption" nillable="true" type="tns1:FTPDeliveryOption"/>
    <element name="localOption" nillable="true" type="tns1:LocalDeliveryOption"/>
    <element name="printOption" nillable="true" type="impl:PrintDeliveryOption"/>
    <element name="webDAVOption" nillable="true"
type="tns1:WebDAVDeliveryOption"/>
  </sequence>
</complexType>
```

The three new elements added are shown in the following table:

Parameter	Type	Description
contentType	string	The content type of the generated document. Possible values are: "text/html;charset=UTF-8" "text/plain;charset=UTF-8" "application/pdf" "application/vnd.ms-powerpoint" "application/vnd.ms-powerpoint" "application/vnd.ms-excel" "application/msword" "application/x-shockwave-flash" "text/xml" "message/rfc822"
documentData	base64Binary	The output document.
dynamicDataSource	BIPDataSource	See Section 12.2.1 , "BIPDataSource"

12.3.2 EmailDeliveryOption

The complex type EmailDeliveryOption added one new element in PublicReportService_v11. The complete complex type definition is as follows:

```
<complexType name="EMailDeliveryOption">
  <sequence>
    <element name="emailBCC" nillable="true" type="xsd:string"/>
    <element name="emailBody" nillable="true" type="xsd:string"/>
    <element name="emailCC" nillable="true" type="xsd:string"/>
    <element name="emailFrom" nillable="true" type="xsd:string"/>
    <element name="emailReplyTo" nillable="true" type="xsd:string"/>
    <element name="emailServerName" nillable="true" type="xsd:string"/>
    <element name="emailSubject" nillable="true" type="xsd:string"/>
    <element name="emailTo" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
```

```

    </sequence>
</complexType>

```

The new element is described in the following table:

Parameter	Type	Description
emailServerName	string	The e-mail server name, for example: "Oracle Mail".

12.3.3 ParamNameValue

Sixteen new elements were added to the ParamNameValue complex type in PublicReportService_v11. This structure describes parameters defined for a reports and templates in BI Publisher. The complete definition is as follows:

```

<complexType name="ParamNameValue">
  <sequence>
    <element name="UIType" nillable="true" type="xsd:string"/>
    <element name="dataType" nillable="true" type="xsd:string"/>
    <element name="dateFormatString" nillable="true" type="xsd:string"/>
    <element name="dateFrom" nillable="true" type="xsd:string"/>
    <element name="dateTo" nillable="true" type="xsd:string"/>
    <element name="defaultValue" nillable="true" type="xsd:string"/>
    <element name="fieldSize" nillable="true" type="xsd:string"/>
    <element name="label" nillable="true" type="xsd:string"/>
    <element name="lovLabels" nillable="true" type="tns1:ArrayOfString"/>
    <element name="multiValuesAllowed" type="xsd:boolean"/>
    <element name="name" nillable="true" type="xsd:string"/>
    <element name="refreshParamOnChange" type="xsd:boolean"/>
    <element name="selectAll" type="xsd:boolean"/>
    <element name="templateParam" type="xsd:boolean"/>
    <element name="useNullForAll" type="xsd:boolean"/>
    <element name="values" nillable="true" type="tns1:ArrayOfString"/>
  </sequence>
</complexType>

```

The ParamNameValue structure is used in the ArrayOfParamNameValue, which is included in the ReportRequest, ReportDefinition, and JobInfo structures. ParamNameValue is also returned by the [getTemplateParameters](#) and [getReportParameters](#) operations.

The new elements are described in the following table:

Parameter	Type	Description
UIType	string	The type of parameter as defined in the BI Publisher data model user interface. Valid values are: "Menu" "Text" "Date" "Hidden" "Search"

Parameter	Type	Description
dataType	string	The data type of the parameter. Valid values are: "String" "Integer" "Boolean" "Date" "Float"
dateFormatString	string	If the parameter type is "Date", this specifies the Date Format String. The date format string must be a Java date format, for example: "MM-dd-YYYY".
dateFrom	string	If the parameter type is "Date", this specifies the begin value of the date.
dateTo	string	If the parameter type is "Date", this specifies the end value of the date.
defaultValue	string	Specifies the default value of the parameter.
fieldSize	string	For parameter types "Text" and "Date" specifies the text field size for the parameter.
label	string	For all parameter types except "Hidden", specifies the display label for the parameter.
lovLabels	ArrayOfString	If the parameter type is "Menu", specifies the values displayed in the list of values to the user.
multiValuesAllowed	boolean	True indicates that multiple values are allowed for the parameter.
name	string	The parameter Identifier as defined in the BI Publisher parameter definition page.
refreshParamOnChange	boolean	For parameter types "Text" and "Menu", a true value for this parameter indicates that other defined parameters should be refreshed when a selection is made for this parameter.
selectAll	boolean	For parameter type Menu: True indicates that all values can be selected for the LOV.
templateParam	boolean	True indicates the parameter is defined in the RTF template.
useNullForAll	boolean	For parameter type Menu: True indicates that a null will be passed if all values are selected for the parameter.
values	ArrayOfString	The values for the parameter.

12.3.4 PrintDeliveryOption

One new element was added to the PrintDeliveryOption complex type in PublicReportService_v11. The complete definition is as follows:

```
<complexType name="PrintDeliveryOption">
  <sequence>
    <element name="printNumberOfCopy" nillable="true" type="xsd:string"/>
    <element name="printOrientation" nillable="true" type="xsd:string"/>
    <element name="printRange" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
```

```

        <element name="printSide" nillable="true" type="xsd:string"/>
        <element name="printTray" nillable="true" type="xsd:string"/>
        <element name="printerName" nillable="true" type="xsd:string"/>
    </sequence>
</complexType>

```

This structure is used by the [DeliveryRequest](#) structure.

The new element is described in the following table:

Parameter	Type	Description
printOrientation	string	Valid values are: "Portrait" and "Landscape".

12.3.5 ReportRequest

Five new elements are added to the ReportRequest complex type in PublicReportService_v11. The complete definition is as follows:

```

<complexType name="ReportRequest">
    <sequence>
        <element name="attributeCalendar" nillable="true" type="xsd:string"/>
        <element name="attributeFormat" nillable="true" type="xsd:string"/>
        <element name="attributeLocale" nillable="true" type="xsd:string"/>
        <element name="attributeTemplate" nillable="true" type="xsd:string"/>
        <element name="attributeTimezone" nillable="true" type="xsd:string"/>
        <element name="byPassCache" type="xsd:boolean"/>
        <element name="dynamicDataSource" nillable="true"
type="impl:BIPDataSource"/>
        <element name="flattenXML" type="xsd:boolean"/>
        <element name="parameterNameValues" nillable="true"
type="impl:ArrayOfParamNameValue"/>
        <element name="reportAbsolutePath" nillable="true" type="xsd:string"/>
        <element name="reportData" nillable="true" type="xsd:base64Binary"/>
        <element name="reportOutputPath" nillable="true" type="xsd:string"/>
        <element name="reportRawData" nillable="true" type="xsd:string"/>
        <element name="sizeOfDataChunkDownload" type="xsd:int"/>
    </sequence>
</complexType>

```

The new elements are described in the following table:

Parameter	Type	Description
attributeCalendar	string	The formatting calendar to use for the report request. Valid values are: "Gregorian", "Arabic Hijrah", "English Hijrah", "Japanese Imperial", "Thai Buddha", and "ROC Official".
attributeTimeZone	string	Specifies the time zone to use for the request, using a supported Java time zone ID. For example: "America/Los_Angeles".
byPassCache	boolean	True indicates to bypass document cache.
dynamicDataSource	BIPDataSource	See Section 12.2.1, "BIPDataSource"
reportOutputPath	string	Specifies the output path for the generated report.
reportRawData	string	If raw XML data is used for the report, this element contains the XML data.

12.4 New Operations

The following operations are new in PublicReportService_v11:

- getTemplateParameters
- getDeliveryServiceDefinition
- deliveryService

Also new are the corresponding InSession operations for the new operations listed above:

- getTemplateParametersInSession
- getDeliveryServiceDefinitionInSession
- deliveryServiceInSession

For information on using the InSession operations, see [Section 12.6, "Using the InSession Operations."](#)

12.4.1 getTemplateParameters

Use getTemplateParameters to retrieve information about parameters for a template.

12.4.1.1 Input Message: getTemplateParametersRequest The following table displays the parameters required for getTemplateParameters:

Parameter	Type	Nullable	Description
reportAbsolutePath	string	no	The directory path to the report. Example: /HR Manager/Employee Reports/Employee Listing.xdo
templateID	string	no	The layout name of the report template, for example: "Employee Listing".
userID	string	no	The BI Publisher user name.
password	string	no	The password for the entered user name.

12.4.1.2 Return Message: getTemplateParametersResponse The response message returns the parameter information in the ParamNameValue structure:

Parameter	Type	maxOccurs	Description
getTemplateParametersReturn	ParamNameValue	unbounded	See Section 12.3.3, "ParamNameValue."

12.4.2 getDeliveryServiceDefinition

Use getDeliveryServiceDefinition to retrieve information about the BI Publisher delivery servers.

12.4.2.1 Input Message: getDeliveryServiceDefinitionRequest The following table describes the parameters required for getDeliveryServiceDefinition:

Parameter	Type	Nullable	Description
userID	string	no	The BI Publisher user name.
password	string	no	The password for the entered user name.

12.4.2.2 Return Message: getDeliveryServiceDefinitionResponse The response message returns information about the delivery servers in the DeliveryServiceDefinition structure:

Parameter	Type	Description
getDeliveryServiceDefinitionReturn	DeliveryServiceDefinition	See Section 12.2.4, "DeliveryServiceDefinition"

12.4.3 deliveryService

Use deliveryService to execute a report delivery request.

12.4.3.1 Input Message: deliveryServiceRequest The following table describes the parameters required for deliveryService:

Parameter	Type	Nullable	Description
deliveryRequest	DeliveryRequest	no	Includes parameters required to execute a report delivery via a supported delivery channel. See Section 12.3.1, "DeliveryRequest."
userID	string	no	The BI Publisher user name.
password	string	no	The password for the entered user name.

12.4.3.2 Return Message: deliveryServiceResponse The following table describes the parameter for deliveryServiceResponse:

Parameter	Type	Description
deliveryServiceReturn	string	The return response.

12.5 Changed Operations

The following operations are impacted by changes in the noted complex type:

PublicReportService Operation	Includes Complex Type:
getReportParameters	ReportRequest
runReport	ReportRequest
scheduleReport	ScheduleRequest
updateReportDefinition	ReportDefinition

12.6 Using the InSession Operations

For most operations in `PublicReportService` and `PublicReportService_v11`, an "InSession" version of the operation is also available. These operations enable you to pass a BI Publisher token, rather than the username and password and otherwise take the identical parameters. Use the `login` operation to obtain a token to pass to the InSession operations. Use the `logout` operation to terminate the session.

12.6.1 login

Use the `login` operation to establish a BI Publisher session and return the session token for you to pass to the InSession operations.

12.6.1.1 Input Message: loginRequest The following table describes the parameters defined for `login`:

Parameter	Type	Description
<code>userID</code>	string	The BI Publisher user name.
<code>password</code>	string	The password for the entered user name.

12.6.1.2 Return Message: loginResponse The following table describes the parameter for `loginResponse`:

Parameter	Type	Description
<code>loginReturn</code>	string	The BI Publisher session token. Use this token with the InSession operations.

12.6.2 logout

Use the `logout` operation to terminate a BI Publisher session.

12.6.2.1 Input Message: logoutRequest The following table describes the parameters defined for `logout`:

Parameter	Type	Description
<code>bipSessionToken</code>	string	The token established for your BI Publisher session.

12.6.2.2 Return Message: logoutResponse The following table describes the parameter for `logoutResponse`:

Parameter	Type	Description
<code>logoutReturn</code>	boolean	True indicates that the BI Publisher session was successfully terminated.

13 Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to

facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Oracle Business Intelligence Publisher New Features Guide, Release 10.1.3.4.1
E14667-01

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.