Oracle® Business Intelligence

New Features Guide

Release 10.1.3.3.2

Part No. E10416-03

December 2007

ORACLE®

Oracle Business Intelligence New Features Guide, Release 10.1.3.3.2

Part No. E10416-03

Contents

Part II New Features for Oracle BI Publisher

10 Oracle BI Publisher Template Builder Enhancements

11 Flash Templates for Oracle BI Publisher

12 Oracle BI Publisher PowerPoint Output

13 Oracle BI Publisher Integration with Oracle BI Discoverer

14 Performance Improvements for Oracle BI Publisher

15 Converting Reports from Oracle Reports to Oracle BI Publisher

16 Support for Digital Signature in PDF Documents

17 Support for Postscript Printers

18 Additional New Features in Release 10.1.3.3

19 New Features in Release 10.1.3.3.1

20 Oracle Business Intelligence Publisher Web Services

Part III New Features for Oracle Business Intelligence for Microsoft Office

21 Installing Oracle Business Intelligence for Microsoft Office

Preface

This guide describes the new features in Oracle Business Intelligence Enterprise Edition (Oracle BI Enterprise Edition) Releases 10.1.3.3, 10.1.3.3.1, 10.1.3.3.2 and Oracle Business Intelligence Publisher (Oracle BI Publisher) Releases 10.1.3.3, 10.1.3.3.1, 10.1.3.3.2.

This preface contains the following sections:

Audience

This document is intended for:

- Oracle BI Enterprise Edition report and dashboard creators
- Oracle BI Enterprise Edition administrators
- Oracle Business Intelligence Publisher administrators, report authors, and end users

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documents

In addition to this installation guide, Oracle BI Enterprise Edition has the following documentation:

- Oracle BI Enterprise Edition component-level Online Help (for example, Oracle BI Answers Help, Oracle BI Administration Tool Help)

You can also refer to the following:

- The Oracle BI Enterprise Edition Documentation Web site:

  http://www.oracle.com/technology/documentation/bi_ee.html

- The Oracle BI Enterprise Edition Product Web site:

  http://www.oracle.com/technology/products/bi/enterprise-edition.html

- The latest information on supported versions is on the Certify application at:

  http://metalink.oracle.com

- The Oracle BI Enterprise Edition Release Notes are available on SupportWeb at http://supportweb.siebel.com.

- The Oracle Business Intelligence Publisher Documentation pages:

  http://www.oracle.com/technology/products/xml-publisher/xmlpdocs.html

- The Oracle Business Intelligence Publisher Product web site:

  http://www.oracle.com/technology/products/xml-publisher/index.html

Conventions

The following text conventions are used in this document:

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

Part I

General New Features

Part I contains the following chapters:

- Chapter 1, "Oracle BI Answers-Based Metadata Dictionary Feature"
- Chapter 2, "Multi-Select Dashboard Prompts Feature"
- Chapter 3, "Office Integration with BI Dashboard Feature"
- Chapter 4, "Dashboard Link Types Feature"
- Chapter 5, "Import Oracle Catalog Feature"
- Chapter 6, "Embedded Database Functions Feature"
- Chapter 7, "Drag and Drop XMLA Metadata Objects Feature"
- Chapter 8, "General New Features in Release 10.1.3.3.1"
- Chapter 9, "General New Features in Release 10.1.3.3.2"

1

Oracle BI Answers-Based Metadata Dictionary Feature

This new feature enables administrators to analyze metadata repository statistics in Oracle BI Answers.

The Oracle BI Server can generate a Metadata Dictionary that describes the metrics contained within the repository as well as the attributes of repository objects. The Metadata Dictionary output is a static set of XML documents. For more information about the Metadata Dictionary, see *Oracle Business Intelligence Server Administration Guide*.

In this release, this Metadata Dictionary can now be accessed directly from the Oracle BI Answers selection pane where specific metadata information will be shown to guide report construction. Once this Oracle BI Answers-based functionality has been enabled as described in this section, an icon will appear next to each Subject Area, Table name, and Column shown in the selection pane for that Subject Area. The Subject Area icon is always displayed, whereas the Table name and Column name icons are only displayed on mouse-over. Clicking on the icon will open up a specific Metadata Dictionary page to show information for that element and links to related repository information.

1.1 Configuration for the Oracle BI Answers-Based Metadata Dictionary Feature

This section explains how to configure the Oracle BI Answers Based Metadata Dictionary feature.

To configure this functionality, a Metadata Dictionary should first be generated using the Oracle BI Administration Tool, as follows:

1. On the machine where Oracle BI is installed, launch the Oracle BI Administration Tool.
2. Log in to the desired repository in "Offline" mode.
3. Create the dictionary using: **Tools** -> **Utilities** -> **Generate Metadata Dictionary**.

Once the newly-generated Metadata Dictionary has been saved, it should then be packaged and deployed to the application server as appropriate for your installation. If you are using IIS as a web server, placing the Metadata Dictionary in the "OracleBIData" directory is a recommended practice - for example, within OracleBIData\web\res\dictionary. Any http-based location will also work.

The following "SubjectAreaMetadata" node will need to be added to the Presentation Services instanceconfig.xml file to specify this selected location. It should be added under the "ServerInstance" node. Please see the "Making Oracle BI Presentation Services Configuration Changes" section within the *Oracle Business Intelligence Presentation Services Administration Guide* for more information on editing the instanceconfig.xml configuration file.

Here is an example showing a relative reference to the IIS location previously mentioned. For any other application server this path would point to where the WAR file had been deployed.

```
<SubjectAreaMetadata>
<DictionaryURLPrefix>/analyticsres/dictionary/</DictionaryURLPrefix>
</SubjectAreaMetadata>
```

Generate the Metadata Dictionary, make this configuration change, then restart Presentation Services to use this feature. Best practices include keeping repository names to less than 156 characters in order to prevent truncated URLs. Access to the Metadata Dictionary through Oracle BI Answers can also be administered using a new Access to Metadata Dictionary privilege. Finally, some versions of browsers (like IE6 or Firefox 1.6) might have problems showing these XML-based files if they are not configured to access data sources across domains. This setting should be enabled.

2 Multi-Select Dashboard Prompts Feature

This new feature enables users to analyze data more easily with an easier to use and more powerful screen for specifying dashboard prompts. For example, users can now use multi-select to specify prompt values, and a more powerful search facility.

Prior to this release, a multi-select dashboard prompt with a large number of values could become difficult to work with. A new search feature adds the ability to better navigate and select values from a multi-select prompt when there are many (possibly hundreds or thousands) of values. Modifications to the existing multi-select dashboard prompt now allow the user to search through the available values using four matching schemes: begins with, ends with, contains, and is Like (Pattern Match). Wildcards like % can also be used in the query. A more sophisticated paging mechanism is also now present which allows the user to page ahead according to a configured search set size as needed. A plus sign following the "Choices Returned" number and the "More..." button indicate additional values have not yet been shown. Configurations for the search set size are described below.

As with the previous multi-select prompt, to add values to be filtered, choices are selected and moved via the shuttle buttons to the left pane. A single mouse click selects a single value. Multiple values can be selected by clicking while holding down the Ctrl key. Range selection is done by holding down the Shift key while clicking. Buttons then allow the user to move single values, these sets of values, or all visible values to or from the "Selected" pane.

Once in the "Selected" pane, the collection of values may be edited by clicking on the Edit button. This action will show an editable dialog where new values could be typed in or lists of values could be copied and pasted in or out of the prompt. A list of values in Excel, for example, could easily be pasted in and thus "imported" into the prompt. This edit dialog will interpret sequences of values separated by carriage returns/line feeds, tab delimiters, and/or pipe (|) characters. Please also note that while manually typing in values, only carriage returns/line feeds and "|" can be used as value separators. Tab is a valid value separator only when user copy pastes values, which are tab, separated.

2.1 Configuration for the Multi-Select Dashboard Prompts Feature

This section explains how to configure the Multi-Select Dashboard Prompts feature.

For configuration, instanceconfig.xml can be edited to specify the number of values returned per search set. The following "Prompts" node will need to be added under the "ServerInstance" node:

```
<Prompts>
<MaxScrollValues>20</MaxScrollValues>
</Prompts>
```

Please see the "Making Oracle BI Presentation Services Configuration Changes" section within the *Oracle Business Intelligence Presentation Services Administration Guide* for more information on editing the instanceconfig.xml configuration file.

3

Office Integration with BI Dashboard Feature

This new feature enables users to save reports in Powerpoint format, and paste reports directly into MS Office applications.

Two new Interactive Dashboard features have been added to this release that provide additional interactivity with Microsoft Office. These features complement the new Oracle BI Office Add-ins for PowerPoint and Excel that provide functionality natively within those Office applications. For more information on the Oracle BI Office Add-ins for PowerPoint and Excel, see Chapter 21, "Installing Oracle Business Intelligence for Microsoft Office"and Chapter 22, "Microsoft Office Support".

The first new Dashboard feature is "Download to PowerPoint", which adds PowerPoint alongside Excel as a targeted application for downloaded reports. This new link is found within the "Report Links" Download menu and will download a static version of that report to PowerPoint. Like the other Download options, when the link is first clicked, the user will have the choice of directly saving the report to disk or opening it immediately. The Oracle BI Office Add-in does not need to be installed to use this functionality.

The second new Dashboard feature provides a means to copy and paste Dashboard reports directly into Office applications. This functionality is also made available along with the other Dashboard Report Links configured using the Dashboard Editor. A new "Copy" link can be added to a report that will copy the XML definition of the current report to the Windows clipboard. Pasting this link using the Paste function found in the Oracle BI menu will convert the copied XML into an Office version of that report. Naturally this functionality will only be present in versions of either PowerPoint or Excel where the Oracle BI Office Add-in has been installed. Once pasted, this Dashboard report will then be translated into native Office objects that can be subsequently refreshed with live data at any point. Pasting the link directly using Edit/Paste either in Office or any other application (like a text editor) will copy the request XML without translation.

3.1 Configuration for the Office Integration with BI Dashboard Feature

This section explains how to configure Office Integration With BI Dashboard feature.

Access to both of these Dashboard Report Links can be administered using a new Access to Oracle BI Office Add-in privilege.

4 Dashboard Link Types Feature

This new feature enables users to users to share reports and collaborate more easily by providing shortcut links to them, including prompts.

Two new types of Dashboard links can be generated using this release. Intended for collaboration, these are the:

- Bookmark Link
- Prompted Link

4.1 Using Bookmark Links

The Bookmark Link provides a mechanism whereby a user may obtain a shortcut link to a Dashboard page, which contains all aspects of page state. This link can then be e-mailed to another user who (assuming they had identical permissions and access) can enter that link into the URL line of a browser and view the same page content. The recipient could then further analyze the data in the Dashboard. Technically, page state is too complex to capture in a URL so this functionality writes an object to the Presentation Services catalog, which is called by the Bookmark Link URL. To create a Bookmark Link the user selects the "Create Bookmark Link" from the Dashboard "Page Options" menu. Once selected, the Dashboard message bar will show that "A Bookmark Link suitable for saving or sharing has been created for this page. This Link can be found in the browser's Address Bar." Inspection will reveal the URL has indeed changed in the Address Bar to show the Bookmark Link. At this point the user can copy and paste this link from the Address Bar and either save it or share it most likely via e-mail. After a Bookmark Link has been generated, the user may continue to modify the content of the page, e.g., add or delete reports. The Bookmark link will attempt to match up context between the Bookmark Link and the Dashboard Page. The existing Dashboard Saved Selections functionality exploits a similar matching mechanism. For example, if a specific filter on the original Dashboard page were included in the saved Bookmark Link, but all reports containing that filter were later removed from that Dashboard, the Bookmark Link would still work. The Bookmark Link would simply contain a small amount of extraneous information, which would not be applied. All other page context still appropriate would be applied, however.

4.2 Using Prompted Links

A second variant of this functionality generates a Prompted Link, which does not capture all aspects of state, but instead provides a simplified link to the Dashboard Page, which contains both the path to that page and a simplified presentation of Dashboard Prompts. This Prompted Link is intended to support both manual and programmatic manipulation of that link and also aims to provide a simple URL-based

API for generating and customizing Dashboard content. Unlike the Bookmark Link, this link is dynamic, which maps to no saved catalog content.

When configured to show this feature, the Dashboard Page Options menu will show a "Create Prompted Link" option. Selecting this option will generate the Prompted Link and the user will see a message indicating, "A Prompted link capturing the prompts and values of this page has been created."

This is the Prompted Link syntax:

```
<Content Path>, Action=Navigate, Column1 (col1), Operator1 (op1), Values1(val1),
Column2 (col2), Operator2 (op2), Values2 (val2) ...
```

Values are separated by pluses and to further simplify this syntax the operator parameter is not needed for equal (eq). Double quotes are also optional if there are no white spaces inside the values.

These are Prompted Link examples:

```
...Action=Navigate&col1=Products.Color&val1=Armory
...Action=Navigate&col1=Products.Color&val1=Armory+Clear
...Action=Navigate&col1=Products.Color&op1=bet&val1=Armory+Clear
...Action=Navigate&col1=Products.Color&val1="Armory"+"Clear"&col2=Periods."Month"&
val2="04/01/1998%2012:00:00%20AM"
...Action=Navigate&col1=Products.Color&val1="Armory"+"Clear"&col2=Periods."Month"&
op2=gt&val2="05/01/1998%2012:00:00%20AM"
```

Here is an example of the Full URL:

```
@http://localhost/aruba/saw.dll?Dashboard&PortalPath=%2Fshared%2Fnew%2F_
portal%2Fprompt&Page=page%252&Action=Navigate&col1=Products.Color&val1="Armory"+"C
lear"&col2=Periods."Month"&op2=gt&val2="05/01/1998%2012:00:00%20AM"
```

As these links are extensions of the existing Go URL, a full list of operators can be found in the "Using the Oracle BI Presentation Services Go URL to Issue SQL and Pass Filters" section found in the *Oracle Business Intelligence Presentation Services Administration Guide*.

4.3 Configuration for the Dashboard Prompt Types Feature

This section explains how to configure the new Dashboard Prompt Types feature.

For configuration, the Create Bookmark Link is enabled by default. The following nodes must be added to the instanceconfig.xml under the "ServerInstance" node:

```
<Dashboard>
<EnableBookmarkURL>True</EnableBookmarkURL>
<EnablePromptedURL>True</EnablePromptedURL>
<BookmarkExpirationDays>30</BookmarkExpirationDays>
</Dashboard>
```

"EnableBookmarkURL" controls whether or not the "Create Bookmark Link" is shown in the "Page Options" menu while "EnablePromptedURL" controls the appearance of the "Create Prompted Link." The Bookmark Link refers to a stub saved in the Presentation Services catalog and "BookmarkExpirationDays" allows an administrator to set a lifespan for how long these should persist.

Please see the "Making Oracle BI Presentation Services Configuration Changes" section within the *Oracle Business Intelligence Presentation Services Administration Guide* for more information on editing the instanceconfig.xml configuration file.

For advanced applications, and for both the "Create Bookmark Link" and "Create Prompted Link" functionality, two Javascript functions can be called to retrieve these values from custom scripts embedded on the originating Dashboard Page. Note, for Javascript functions to work a HardenXSSconfiguration must be set to false in instanceconfig,xml.

For the Bookmark Link, the function linkToPage(bInlineDrill,bGetBookmarkOnly) can be called. If the second argument is omitted or set to "false," the browser address bar will be updated with the bookmark URL. If the second argument is set to "true," the browser address bar won't be refreshed. In either case, the bookmark URL is retrieved asynchronously and put into variable saw.bookmarkURL. Polling the value of saw.bookmarkURL will give the link. This variable is set to empty string when the function is invoked. As for the first argument, if you see the "Page Options" menu on the page, set it to true, otherwise, set it to false.

For the Prompted Link, the function GetPURL( ) returns the Prompted Link as a string.

5

Import Oracle Catalog Feature

This new feature enables administrators to import Database objects through a native Oracle OCI connection.

The Oracle BI Administration Tool now supports importing of physical Database objects through a native Oracle OCI connection and reuses the same OCI connection for queries. This feature mitigates the need to setup an ODBC connection to an Oracle Database purely for importing metadata objects into the Administration Tool.

6 Embedded Database Functions Feature

This new feature enables users and administrators to create more powerful reports by directly calling Database functions from either Oracle BI Answers or by using a Logical column (in the Logical Table source) within the Metadata (repository).

The Oracle BI server now supports the capability to directly call functions defined within the Database from either the Answers interface or using a Logical column (in the Logical Table source) within the Metadata (repository). This feature is restricted to support SQL sources in this release.

Note: MDX or MOLAP data sources are not supported in this release.

The following new database functions are supported:

- EVALUATE

  For more information, see Section 6.1, "EVALUATE Function".

- EVALUATE_AGGR

  For more information, see Section 6.2, "EVALUATE_AGGR Function".

- EVALUATE_PREDICATE

  For more information, see Section 6.3, "EVALUATE_PREDICATE Function".

6.1 EVALUATE Function

This function is intended for scalar and analytic calculations.

Syntax: `EVALUATE('DB_Function(%1)', <Comma separated Expression>)`

Example: `SELECT e.lastname,sales.revenue,EVALUATE('dense_rank() over(order by %1 )',sales.revenue) FROM sales s, employee e;`

6.2 EVALUATE_AGGR Function

This function is intended for aggregate functions with group by clause.

Syntax: `EVALUATE_AGGR('DB_Aggregate_Function(%1)', <comma separated Expression)`

Example: `SELECT year.year, sales.qtysold, EVALUATE_AGGR('sum(%1)', sales.quantity) From SnowFlakeSales;`

6.3 EVALUATE_PREDICATE Function

This function is intended for functions with a return type of boolean.

Syntax: `EVALUATE_PREDICATE('DB_Function(%1)', <comma separated Expression)`

Example: `SELECT year, Sales as DOUBLE,CAST(EVALUATE('OLAP_EXPRESSION(%1,''LAG(units_cube_sales, 1, time, time LEVELREL time_levelrel)'')', OLAP_CALC) AS DOUBLE) FROM "Global".Time, "Global"."Facts - sales" WHERE EVALUATE_PREDICATE('OLAP_CONDITION(%1, ''LIMIT time KEEP ''''1'''', ''''2'''', ''''3'''', ''''4'''' '') =1', OLAP_CALC) order by year;`

7

Drag and Drop XMLA Metadata Objects Feature

This new feature enables administrators to drag-and-drop Physical metadata objects (cubes) into the Business Model and Mapping layer in Oracle BI Administration Tool as fully configured Logical Model Metadata (Subject area) retaining metrics, attributes and dimensions.

The Oracle BI Administration Tool now supports Drag & Drop feature for XMLA data sourced (SAP/BW and Microsoft Analysis Server). Using this capability its now possible to drag-and-drop Physical metadata objects (cubes) into the Business Model and Mapping layer as fully configured Logical Model Metadata (Subject area) retaining metrics, attributes and dimensions. This mitigates the need to manually redefine Logical dimensional models from data sources, which already contain rich dimensional metadata constructs.

8 General New Features in Release 10.1.3.3.1

The 10.1.3.3.1 release of the Oracle Business Intelligence Enterprise Edition introduces three new BI Server features. This chapter describes the following new features:

- Section 8.1, "Oracle Business Intelligence Installer Now Deploys BI Components into One OC4J Container"
- Section 8.2, "DataDirect Connect ODBC Drivers for Linux and UNIX Operating Systems Bundled with Oracle BI EE 10.1.3.3.1"
- Section 8.3, "Support for Charts in Oracle BI EE for Arabic, Thai, and Hebrew"

8.1 Oracle Business Intelligence Installer Now Deploys BI Components into One OC4J Container

In 10.1.3.x releases prior to version 10.1.3.3.1, during an Advanced installation, the Oracle BI Installer created the following OC4J containers and deployed BI components in them:

- bianalytics - for BI Presentation Services Plug-in component
- bipublisher - for BI Publisher component
- bijmx - for BI Systems Management component
- bioffice - for BI Office server component (in Oracle BI EE version 10.1.3.3)

With Oracle BI EE 10.1.3.3.1, during an Advanced installation, the Oracle BI Installer deploys the BI components into one OC4J container called "home". The memory footprint for the OC4J instance will be between 512 -1024 MB.

8.1.1 Upgrading to Version 10.1.3.3.1 from a BIEE 10.1.3.x Release Prior to 10.1.3.3.1

On upgrade to Oracle BI EE 10.1.3.3.1, if the "Keep User Modified Configurations" upgrade option is selected on the BI Installer screen, the old OC4J containers for BI will be deleted and the BI components will be deployed into the OC4J container named "home". The configuration files for the BI component applications deployed in the individual BI OC4J containers are migrated into the deployed application in the OC4J instance named "home".

The configuration files that are migrated are:

```
$ORACLE_HOME/j2ee/bianalytics/applications/analytics/analytics/WEB-INF/web.xml
```

to

```
$ORACLE_HOME/j2ee/home/applications/analytics/analytics/WEB-INF/web.xml
```

and

```
ORACLE_HOME/j2ee/bibioffice/applications/analytics/analytics/WEB-INF/bioffice.xml
```

to

```
$ORACLE_HOME/j2ee/home/applications/analytics/analytics/WEB-INF/bioffice.xml
```

On upgrade to Oracle BI EE 10.1.3.3.1, the old OC4J containers for BI will be deleted if the "Reset Configurations" upgrade option is selected on the BI Installer screen. The BI components will be deployed in the OC4J instance named "home". The configuration files web.xml for bianalytics and bioffice.xml from the previous release will not be preserved.

Note: If you have deployed any custom applications on any of the BI OC4J containers, you must make a backup of your custom applications and redeploy them into a different OC4J container before performing an upgrade to Oracle BI EE 10.1.3.3.1.

8.2 DataDirect Connect ODBC Drivers for Linux and UNIX Operating Systems Bundled with Oracle BI EE 10.1.3.3.1

Oracle BI EE 10.1.3.3.1 provides DataDirect Connect ODBC drivers and driver managers version 5.3 for Linux and UNIX operating systems for connectivity to Informix, MS SQL Server, and Sybase ASE databases. 32-bit and 64-bit DataDirect 5.3 drivers are available on AIX, Solaris, and HP-UX Intanium. 32-bit drivers are available for Linux and HP-UX PA-RISC.

The Oracle BI EE Installer 10.1.3.3.1 installer installs the DataDirect Connect ODBC drivers in the following location:

`$OracleBI/odbc`

The 32-bit drivers are installed in `$OracleBI/odbc/lib` and 64-bit drivers are installed in `$OracleBI/odbc/lib64`.

Note: Communication between database clients and servers is typically independent of the widths and data paths. In other words, the 32-bit database drivers will communicate with 64-bit database servers, and vice versa.

Refer to the *Systems Requirements and Supported Platforms Guide for Oracle Business Intelligence 10.1.3.3* for further information on supported operating systems and databases for the DataDirect Connect ODBC drivers.

8.2.1 Using the DataDirect Connect ODBC Drivers

The following examples show sample ODBC.ini file entries for Microsoft SQL Server, Sybase ASE, and Informix data sources.

8.2.1.1 For connectivity to Microsoft SQL Server database

The name of the DataDirect ODBC driver file to connect to an MS SQL Server database is SEmsss23.so (SEmsss23.sl on HP-UX PA-RISC). For supported versions of MS SQL Server see the *Systems Requirements and Supported Platforms Guide for Oracle Business Intelligence 10.1.3.3*.

On Linux, AIX, Solaris, HP-UX PA-RISC and HP-UX Intanium operating systems, the 32-bit driver is located in `$OracleBI/odbc/lib`. On AIX, Solaris and HP-UX Itanium operating systems, the 64-bit driver is located in `$OracleBI/odbc/lib64`.

To use the DataDirect ODBC Driver to connect to MS SQL Server:

1. Make the following entries in the `odbc.ini` file. This file is located in `$OracleBI/setup`. Set the Driver parameter to the file name and location of the DataDirect driver for MS SQL Server. In the following example, the Driver parameter is set to the 64-bit DataDirect driver.

Example 8–1 Sample File for Microsoft SQL Server

```
[SQLSERVER_DB
Driver=/usr/OracleBI/odbc/lib64/SEmsss23.so
Description=DataDirect 5.1 SQL Server Wire Protocol
Address=111.111.111.111,1433
AlternateServers=
AnsiNPW=Yes
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=
LoadBalancing=0
LogonID=
Password=
QuoteID=No
ReportCodePageConversionErrors=0
```

2. In the `$OracleBI/server/Config/DBFeatures.INI` file, for the appropriate SQL Server sections, turn the IDENTIFIER_QUOTE_CHAR parameter off by setting the following:

   IDENTIFIER_QUOTE_CHAR = '';

   The default is:

   IDENTIFIER_QUOTE_CHAR = '"';

8.2.1.2 For connectivity to Sybase ASE Database

The name of the DataDirect ODBC driver file to connect to a Sybase ASE database is SEase23.so. For supported versions of Sybase ASE see the *Systems Requirements and Supported Platforms Guide for Oracle Business Intelligence 10.1.3.3*.

On Linux, AIX, Solaris, HP-UX PA-RISC and HP-UX Intanium operating systems, the 32-bit driver is located in `$OracleBI/odbc/lib`. On AIX, Solaris and HP-UX Itanium operating systems, the 64-bit driver is located in `$OracleBI/odbc/lib64`.

To use the DataDirect ODBC Driver to connect to Sybase ASE, make the following entries in the odbc.ini file. This file is located in `$OracleBI/setup`. Set the Driver parameter to the file name and location of the DataDirect driver for Sybase ASE. In the following example, the Driver parameter is set to the 64-bit DataDirect driver.

Example 8–2 Sample File for Sybase ASE Data Source

```
[SybaseASE_DB
Driver=/usr/OracleBI/odbc/lib64/SEase23.so
Description=DataDirect 5.3 Sybase Wire Protocol
AlternateServers=
ApplicationName=
ApplicationUsingThreads=1
ArraySize=50
```

```
AuthenticationMethod=0
Charset=
ConnectionRetryCount=0
ConnectionRetryDelay=3
CursorCacheSize=1
Database=Northwind
DefaultLongDataBuffLen=1024
EnableDescribeParam=0
EnableQuotedIdentifiers=0
EncryptionMethod=0
GSSClient=native
HostNameInCertificate=
InitializationString=
Language=
LoadBalancing=0
LogonID=
NetworkAddress=111.111.111.111,5005
OptimizePrepare=1
PacketSize=0
Password=
RaiseErrorPositionBehavior=0
ReportCodePageConversionErrors=0
SelectMethod=0
ServicePrincipalName=
TruncateTimeTypeFractions=0
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
WorkStationID=
```

8.2.1.3 For connectivity to Informix Database

The name of the DataDirect ODBC driver file to connect to an Informix database is SEifcl23.so. For supported versions of Informix see the *Systems Requirements and Supported Platforms Guide for Oracle Business Intelligence 10.1.3.3*.

On Linux, AIX, Solaris, HP-UX PA-RISC and HP-UX Intanium operating systems, the 32-bit driver is located in `$OracleBI/odbc/lib`. On AIX, Solaris and HP-UX Itanium operating systems, the 64-bit driver is located in `$OracleBI/odbc/lib64`.

To use the DataDirect ODBC Driver to connect to Informix, make the following entries in the `odbc.ini` file. This file is located in `$OracleBI/setup`. Set the **Driver** parameter to the file name and location of the DataDirect driver for Informix. In the following example, the Driver parameter is set to the 64-bit DataDirect driver.

Example 8–3 Sample File for Informix Data Source

```
[Informix_DB]
Driver=/usr/OracleBI/odbc/lib64/SEifcl23.so
Description=DataDirect Informix Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
CancelDetectInterval=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=
HostName=111.111.111.111
LoadBalancing=0
LogonID=informix
Password=inform_10
PortNumber=1526
```

```
ReportCodePageConversionErrors=0
ServerName=
TrimBlankFromIndexName=1
```

8.3 Support for Charts in Oracle BI EE for Arabic, Thai, and Hebrew

Oracle Business Intelligence Enterprise Edition version 10.1.3.3.1 offers support for Arabic , Thai, and Hebrew. Due to a limitation in the charting engine that is used in Oracle BI EE, bidirectional and Thai language characters do not render properly in charts.

To overcome this limitation and enable charting in Oracle BI EE for Arabic, Thai and Hebrew, you can use Oracle BI Publisher charting capabilities. BI Publisher charts are based on the Oracle BI Beans technology. A BI Publisher report containing a chart may be embedded in an Oracle BI Interactive Dashboard in order to meet charting requirements.

To implement this solution, perform the following steps described below:

- Integrate the Data Sources
- Create a Report and Layout in BI Publisher
- Embed the BI Publisher Report in a Dashboard

8.3.1 Prerequisites

This procedure assumes that you have already performed the following:

- Set up users in Oracle BI EE and Oracle BI Publisher with appropriate privileges.

  You must integrate the user security between Oracle Business Intelligence and Oracle BI Publisher using one of the security models available. Within Oracle BI Presentation Services the user must have privileges to update Dashboards. Within Oracle BI Publisher users must be granted the BI Publisher Developer Role.

  For more information on setting up security in Oracle BI, refer to "Security in Oracle BI" in the *Oracle Business Intelligence Server Administration Guide* and "Managing Oracle Presentation Services Security" in the *Oracle Business Intelligence Presentation Services Administration Guide*. For information on setting up security in Oracle BI Publisher, refer to "Defining a Security Model" in the *Oracle Business Intelligence Publisher User's Guide.*

- Created an Answers request that contains the data you wish to display in a chart.
- Installed the BI Publisher Template Builder for Microsoft Word.

  The Oracle BI Publisher Template Builder for Microsoft Word Add-in can be installed from the BI Publisher application for users with the appropriate roles assigned (that is, BI Publisher Administrator, BI Publisher Developer, or BI Publisher Template Designer). The download link appears on the **Developer Tools** region. See the topic "Installing Oracle BI Publisher Desktop" in the *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.

8.3.2 Integrate the Data Sources

You must set up the Oracle BI Server as a data source in BI Publisher. To use an Answers request as a data source, you must also set up integration with Oracle BI Presentation Services.

8.3.2.1 Adding the Oracle BI Server as a Data Source

If you included BI Publisher in your Oracle BI Enterprise Edition installation, the Oracle BI Server will automatically be configured as a data source in BI Publisher. If you did not include BI Publisher during your BI EE install, you must manually add the BI Server as a data source using BI Publisher's Admin interface. For detailed steps, see the topic: "Adding the Oracle BI Server as a JDBC Data Source" in the *Oracle Business Intelligence Publisher User's Guide*.

8.3.2.2 Integrating with Oracle Presentation Services

By setting up integration with Oracle BI Presentation Services you enable connection to Oracle BI Answers requests as data sources for your reports.

The BI Publisher Admin interface provides a specific page for integration with Oracle BI Presentation Services. Navigate to this page from the Admin page by selecting **Oracle BI Presentation Services** under the **Integration** subheading.

If you included BI Publisher in your Oracle BI Enterprise Edition installation, the Oracle BI installer will set the Presentation Services host name, port, and URL values. However, you must manually enter the Presentation Services username and password on the Oracle BI Presentation Services integration page.

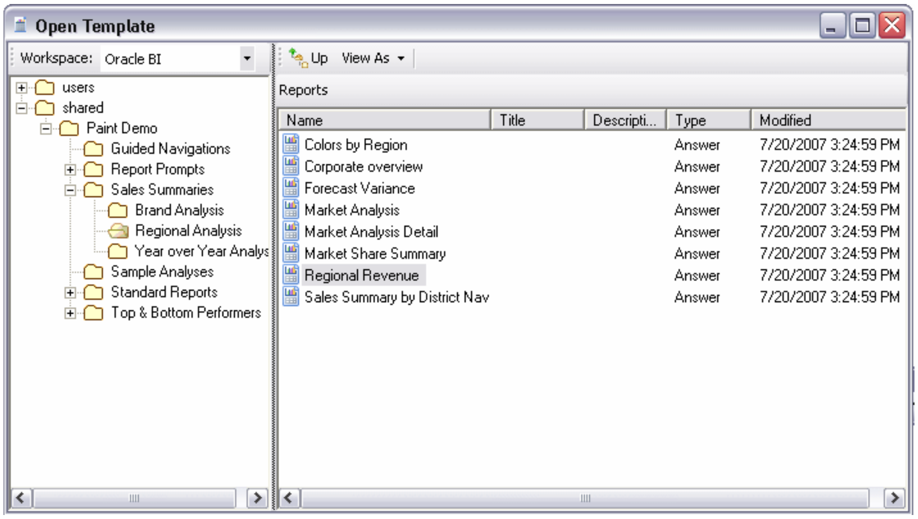
8.3.3 Create a Report and Layout in BI Publisher

Following are summary steps for creating a report and layout in BI Publisher. For detailed description refer to the following sources:

For detailed steps to create a report in BI Publisher, see the chapter "Creating a New Report" in the *Oracle Business Intelligence Pubisher User's Guide*.
For for information on using the Template Builder for Microsoft Word, see the Template Builder online help.

1. Open Microsoft Word (with the Template Builder for Word installed).
2. From the Oracle BI Publisher menu, select **Log On** and enter appropriate credentials (the user must have either the Administrator, Report Designer, or Developer role assigned).
3. From the Open Template dialog, select Oracle BI from the **Workspace** list.
4. Navigate to your Answers request and double-click the request name.

Figure 8–1 Open Template Dialog



Open Template

Workspace: Oracle BI

- users
- shared
  - Paint Demo
    - Guided Navigations
    - Report Prompts
    - Sales Summaries
      - Brand Analysis
      - Regional Analysis
      - Year over Year Analys
    - Sample Analyses
    - Standard Reports
    - Top & Bottom Performers

Reports

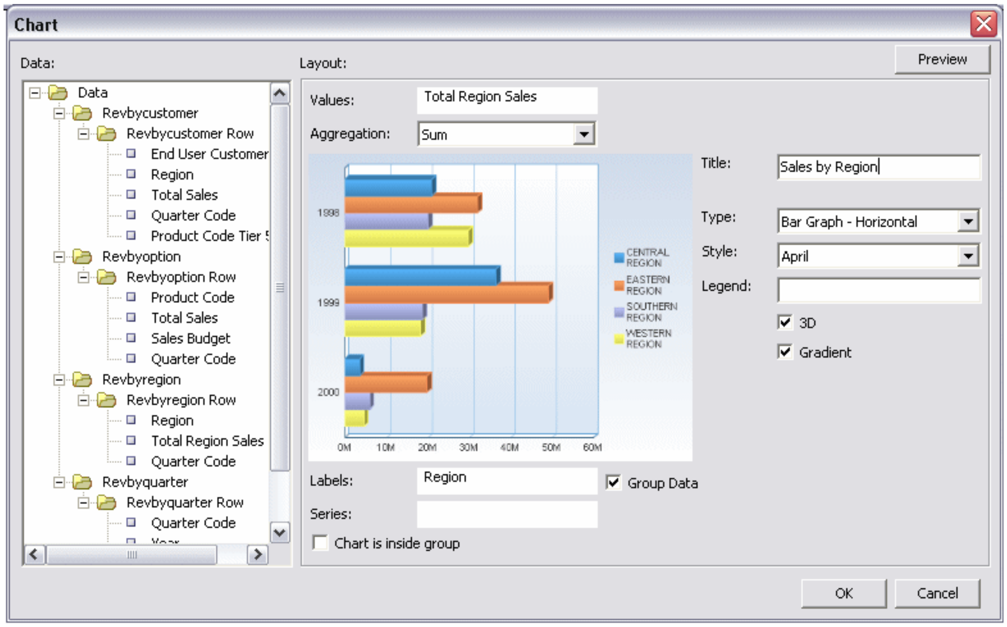
Name	Title	Descripti...	Type	Modified
Colors by Region			Answer	7/20/2007 3:24:59 PM
Corporate overview			Answer	7/20/2007 3:24:59 PM
Forecast Variance			Answer	7/20/2007 3:24:59 PM
Market Analysis			Answer	7/20/2007 3:24:59 PM
Market Analysis Detail			Answer	7/20/2007 3:24:59 PM
Market Share Summary			Answer	7/20/2007 3:24:59 PM
Regional Revenue			Answer	7/20/2007 3:24:59 PM
Sales Summary by District Nav			Answer	7/20/2007 3:24:59 PM

5. You will be prompted to **Save as Oracle BI Publisher Report**. BI Publisher will create a report definition that uses the Answers request as the data source. The template you design will be saved as part of this BI Publisher report definition. Select the folder from the BI Publisher report repository.

   After you save, the Template Builder loads the data and returns you to a blank Microsoft Word page.

6. To insert a chart, launch the Chart dialog from the Oracle BI Publisher menu by selecting **Insert**, then **Chart**. The following figure shows the Oracle BI Publisher Template Builder chart dialog:

Figure 8–2 Template Builder Chart Dialog



Chart

Data:

- Data
  - Revbycustomer
    - Revbycustomer Row
      - End User Customer
      - Region
      - Total Sales
      - Quarter Code
      - Product Code Tier 5
  - Revbyoption
    - Revbyoption Row
      - Product Code
      - Total Sales
      - Sales Budget
      - Quarter Code
  - Revbyregion
    - Revbyregion Row
      - Region
      - Total Region Sales
      - Quarter Code
  - Revbyquarter
    - Revbyquarter Row
      - Quarter Code
      - Year

Layout:

Values: Total Region Sales

Aggregation: Sum

Title: Sales by Region

Type: Bar Graph - Horizontal

Style: April

Legend:

☑ 3D

☑ Gradient

Labels: Region ☑ Group Data

Series:

☐ Chart is inside group

7. After inserting the chart, upload the template to the BI Publisher report repository. From the Oracle BI Publisher menu select **Upload Template As**. Enter a name for the template.

   The report is now available as a BI Publisher object that can be added to a dashboard.

8.3.4 Embed the BI Publisher Report in a Dashboard

The following describes how to embed the BI Publisher report in a Dashboard. For more detailed instructions and options, refer to the topic: "Adding an Oracle BI Publisher Report to an Oracle BI Interactive Dashboard Page" in the *Oracle Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*.

> **Note:** In release 10.1.3.3.1, there is no longer a limitation of one BI Publisher Report per dashboard.

1. In the dashboard page, click the **Page Options** button, and click **Edit Dashboard**.
2. From the selection pane, drag and drop the BI Publisher Report object from the Dashboard Objects area into a section on the dashboard page.
3. Click the newly placed BI Publisher report object's **Properties** button.
4. In the BI Publisher Report Properties dialog, browse for and select the Oracle BI Publisher report.
5. Click **Save** to return to the dashboard and view your added report.

8.3.5 Supported Graph Types

The following graph types are supported in BI Publishe:.

- Area
- Horizontal Bar
- Bubble
- Vertical Bar
- Line
- Line Bar Combo
- Pareto
- Pie
- Radar
- Scatter
- Step

8.3.6 Known Limitations

The BI Publisher chart is static.

The locale information from the Oracle BI Server is not inherited by BI Publisher. BI Publisher charts embedded in the dashboard will take the locale setting from BI Publisher. To ensure consistency, set the same locale for both applications.

9

General New Features in Release 10.1.3.3.2

The 10.1.3.3.2 release of the Oracle Business Intelligence Enterprise Edition introduces the following new BI Server features:

- Section 9.1, "Support for Hyperion Essbase as a Data Source"
- Section 9.2, "Using BI EE OC4J Components with IIS via the IIS Plug-in"

9.1 Support for Hyperion Essbase as a Data Source

Oracle Business Intelligence Enterprise Edition 10.1.3.3.2 now supports Hyperion Essbase as a physical data source.   This topic describes the process to import Essbase structures into the Oracle BI Server metadata repository, as well as usage guidelines and limitations.

9.1.1 Supported Versions of Hyperion Essbase with Oracle BI EE 10.1.3.3.2

At the time of this release, Hyperion Essbase version 7.1.6 and later can be used as a data source with Oracle BI EE 10.1.3.3.2. For the most current information regarding supported versions, please see the *System Requirements and Supported Platforms for Oracle Business Intelligence Suite Enterprise Edition*.

9.1.2 Enabling the Use of Hyperion Essbase as a Data Source

This section describes the steps required to enable Essbase as a data source.

9.1.2.1 Essbase Client Libraries

Oracle BI Server connectivity to Essbase is through the Essbase client libraries. The client libraries must be installed on the Oracle BI Server. Please see the most current *System Requirements and Supported Platforms for Oracle Business Intelligence Suite Enterprise Edition* document for the supported versions of the Essbase Client for connectivity to the Oracle BI Server.

9.1.2.2 BI Server Configuration for UNIX and Linux Platforms

Perform the following configuration steps to access the Essbase client libraries through the BI server on UNIX and Linux Platforms:

9.1.2.2.1 Solaris: Oracle BI 64 bit mode

1. Define ARBORPATH = <*Essbase Client installation folder*>

   For example:

   export `ARBORPATH=/export/home/Hyperion/AnalyticServicesClient`

```
export ARBORPATH
```

2. Add the Essbase Client Libraries folder to LD_LIBRARY_PATH:

   ```
   LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH_64:<Essbase Client
   Libraries folder>
   ```

   For example:

   ```
   LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH_64:$ARBORPATH/bin
   export LD_LIBRARY_PATH_64
   ```

9.1.2.2.2 Solaris: Oracle BI 32 bit mode

1. Define ARBORPATH = <*Essbase Client installation folder*>

   For example:

   ```
    ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to LD_LIBRARY_PATH:

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<Essbase Client Libraries
   folder>
   ```

   For example:

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH$ARBORPATH/bin
   export LD_LIBRARY_PATH
   ```

9.1.2.2.3 HP-UX PARISC: Oracle BI 32 bit mode

1. Define ARBORPATH = <*Essbase Client installation folder*>

   For example:

   ```
   ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to SHLIB_PATH:

   ```
   SHLIB_PATH=$SHLIB_PATH:<Essbase Client Libraries folder>
   ```

   For example:

   ```
   SHLIB_PATH=$SHLIB_PATH$ARBORPATH/bin
   export SHLIB_PATH
   ```

9.1.2.2.4 HP-UX Itanium: Oracle BI 64 bit mode

1. Define ARBORPATH = <*Essbase Client installation folder*>

   For example:

   ```
    ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to SHLIB_PATH:

   ```
   SHLIB_PATH=$SHLIB_PATH:<Essbase Client Libraries folder>
   ```

   For example:

   ```
   SHLIB_PATH=$SHLIB_PATH:$ARBORPATH/bin
   ```

```
export SHLIB_PATH
```

3. Define ESSLANG and LANG

   For example:

   ```
   ESSLANG=English_UnitedStates.UTF-8@Binary
   export ESSLANG
   LANG=en_US.utf8
   export LANG
   ```

4. Comment out the following three lines from the NQSConfig.ini file:

   ```
   [ GENERAL ]
   // Localization/Internationalization parameters.
   LOCALE="English-usa";
   SORT_ORDER_LOCALE="English-usa";
   SORT_TYPE="binary";
   ```

9.1.2.2.5 AIX Oracle BI 32 bit and 64 bit modes

1. Define ARBORPATH = <*Essbase Client installation folder*>

   For example:

   ```
    ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to LIBPATH:

   `LIBPATH=$LIBPATH:<Essbase Client Libraries folder>`

   For example:

   ```
     LIBPATH=$LIBPATH:$ARBORPATH/bin
   export LIBPATH
   ```

9.1.2.2.6 Linux: Oracle BI 32 bit mode

1. Define ARBORPATH = <*Essbase Client installation folder*>

   For example:

   ```
    ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to LD_LIBRARY_PATH:

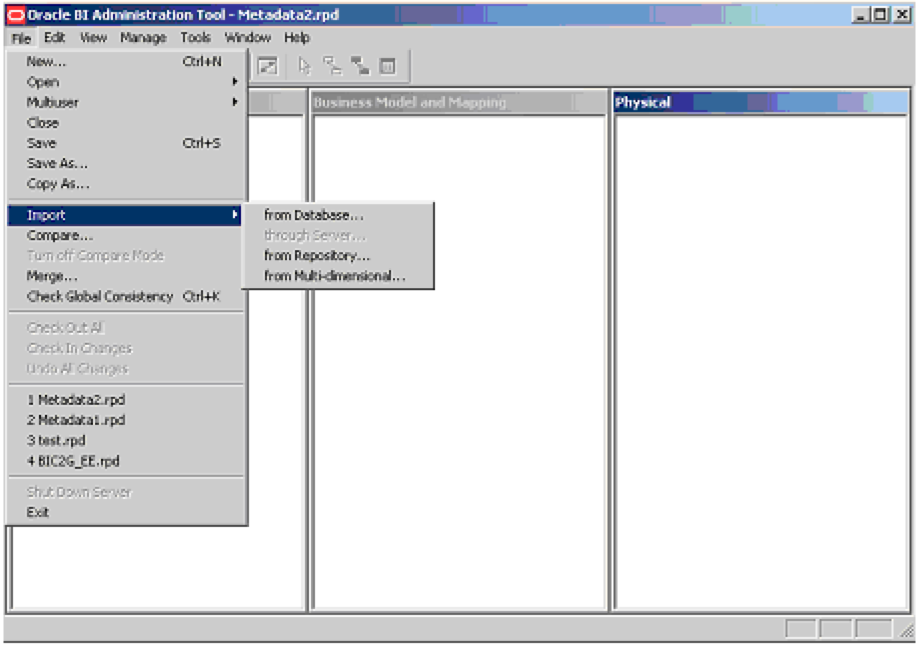
   `LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<Essbase Client Libraries folder>`

   For example:

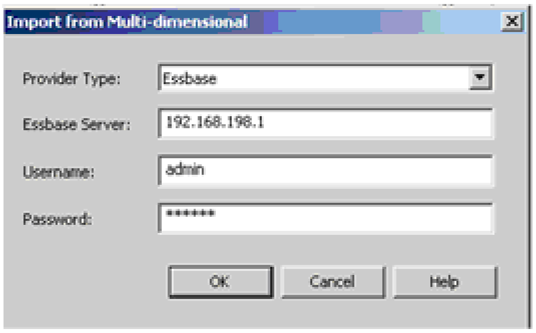
   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ARBORPATH/bin
   export LD_LIBRARY_PATH
   ```

9.1.2.3 Import Process

Essbase cube structures are imported via the "Import from Multi-dimensional" option in the Oracle BI Administration tool, shown in the following figure:



In the "Import from Multi-Dimensional" dialog, select Essbase as the provider type. Input the server and authentication information.



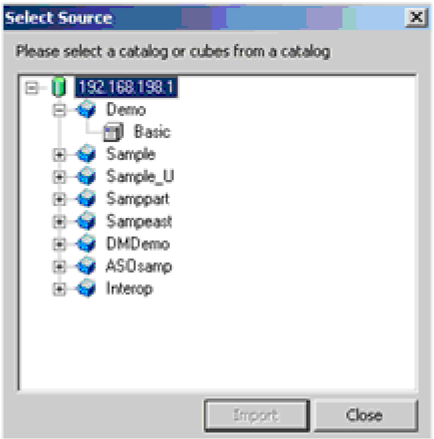
Provider Type: Essbase

Essbase Server: 192.168.198.1

Username: admin

Password: *******

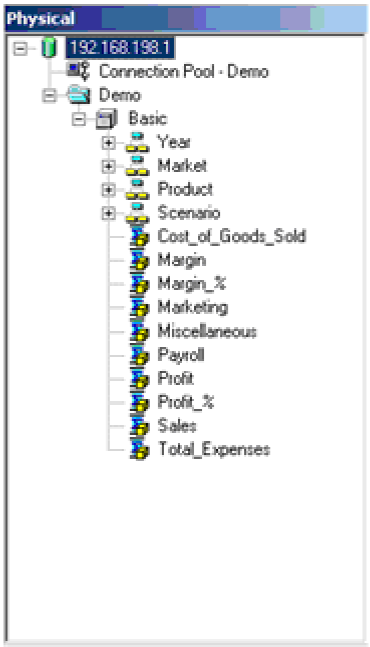
After selecting "OK", the list of Applications and Cubes available on the Essbase Server will be presented as shown in the following figure:



Please select a catalog or cubes from a catalog

```
⊟ 192.168.198.1
  ⊟ Demo
      Basic
  ⊞ Sample
  ⊞ Sample_U
  ⊞ Samppart
  ⊞ Sampeast
  ⊞ DMDemo
  ⊞ ASOsamp
  ⊞ Interop
```

Select the Applications or Cubes to import. Note that Applications will be mapped as Physical Catalogs. The following figure shows an example of the physical layer representation:



Once the physical cubes are imported, they can be dragged to the Business Model and Mapping Layer to automatically create a Business Model that includes associated dimensions. The Business Model can now be customized to take advantage of Oracle BI Server capabilities such as aggregate and fragment navigation.

9.1.3 Guidelines and Limitations

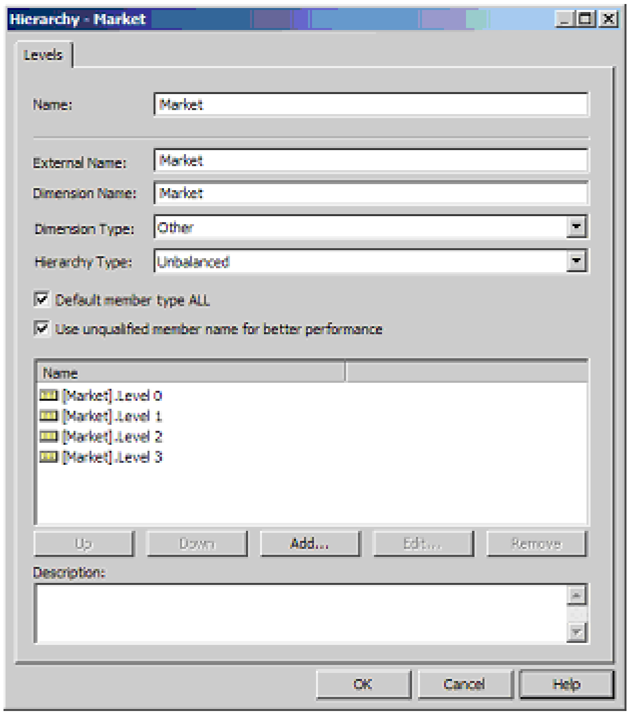
This section describes some unique aspects to Essbase connectivity that the administrator or end user should be aware of when configuring and using Essbase as a data source.

9.1.3.1 Measure Hierarchies

Essbase supports the concept of measure hierarchies. This enables end users to drill from a measure to components that make up the measure, for example, profit drills to revenue and costs. At this time Oracle BI EE does not support measure hierarchies. Oracle BI EE will import measures as a flat list of children to the cube itself independent of the measure hierarchy.

9.1.3.2 Unique Member Names

When member names (including aliases) are unique in a given hierarchy, the Oracle BI Server can take advantage of specific MDX syntax to optimize performance. To enable this capability, select the "Use unqualified member name for better performance" box in the Hierarchy dialog as illustrated in the following figure.



Name: Market

External Name: Market

Dimension Name: Market

Dimension Type: Other

Hierarchy Type: Unbalanced

☑ Default member type ALL

☑ Use unqualified member name for better performance

Name
[Market].Level 0
[Market].Level 1
[Market].Level 2
[Market].Level 3

Description:

The import process will not be able to identify that member names are unique for a given hierarchy, so it is the responsibility of the Administrator to confirm uniqueness. Note that query errors may result if a hierarchy is specified as having unique members when it does not.

9.1.3.3 Time Series Functions

Oracle BI Server Time Series functions, "Ago" and "ToDate", are supported for use with Essbase.

9.1.3.4 Evaluate Function for MDX Sources

In Oracle BI EE 10.1.3.3, Evaluate functions were introduced to provide the ability to leverage unique source-specific functions. This capability was limited to relational sources. In 10.1.3.3.2, you now can leverage Evaluate functions for Essbase. Support for Evaluate does not extend across all multi-dimensional sources at this time.

Sample use cases are described in the following sections. Consider the following examples using the EVALUATE_AGGREGATE and EVALUATE functions. Note that expressions are applied to columns in the Logical Table Source that refers to the physical cube.

EVALUATE_AGGREGATE is used to implement custom aggregations. For example, you may want to compare overall regional profit to profits for the top three products in the region.   A new measure can be defined to represent the profits for top three products resulting in the logical SQL statement:

```
Select Region, Profit, EVALUATE_AGGREGATE('SUM(TopCount(%1.members, 3, %2), %3)',
Products, Profit, Profit) Top_3_prod_Profit From SampleBasic
```

The Oracle BI Server will generate the following expression for the custom aggregation:

```
member [Measures].[MS1] as
'SUM(Topcount([Product].Generations(6).members,3,[Measures].[Profit]),[Measures].[
Profit])'
```

Similarly, use the EVALUATE function to implement scalar functions that are computed post-aggregation. EVALUATE will change the grain of the query, if its definition makes explicit references to dimensions (or attributes) that are not in the query.

For example, if you would like to see the Profits for the top five products ranked by Sales sold in a Region, after creating the applicable measure, the resulting Logical SQL statement is as follows

```
Select Region, EVALUATE('TopCount(%1.members, 5, %2)' as VARCHAR(20), Products,
Sales), Profits
From SampleBasic
```

The Oracle BI Server will generate the following expression to retrieve the top five products:

```
set [Evaluate0]  as
'{Topcount([Product].Generations(6).members,5,[Measures].[Sales]) }'
```

9.1.3.5 User-Defined Attributes

Essbase supports the concept of user-defined attributes (UDAs). A UDA is essentially any arbitrary textual string that can be associated with any member from a dimension. A member can have multiple strings associated to it.   Oracle BI EE will suppress UDAs upon import.

Users can still query using UDAs by leveraging the EVALUATE function. Consider the following example where "Major Market" is a UDA:

After creating the applicable column, the resulting logical SQL statement is as follows:

```
Select EVALUATE('FILTER( %1.Members, isUDA([%2.Dimension, "Major Market"))',
State, State), Sales
From SampleBasic
```

The Oracle BI Server will generate the following expression for the custom column:

```
set [Evaluate0]  as
'{FILTER([Customer].Generations(3).members,isUDA([Customer].Generations(3).Dimensi
on,"Major Market")) }'
```

9.1.3.6 Substitution Variables

Essbase substitution variables are automatically retrieved and populated into corresponding BI Server dynamic system variables.

Depending on the scope of the Essbase variable, the naming convention for the BI Server variable is as described below.

Server instance scope

<server name>:<var name>

Application scope

<server name>:<app name>:<var name>

Cube scope

<server name>:<app name>:<cube name>:<var name>

The refresh interval should be set appropriately to reflect anticipated update cycles for Essbase variables.

9.1.3.7 Attribute Dimensions

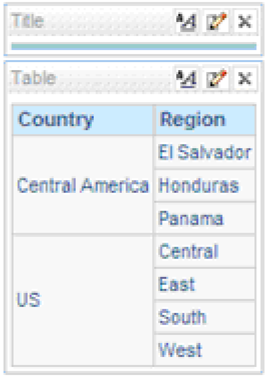
Member attributes are not automatically associated to corresponding dimensions and levels during the import process. To manually create the association, the Oracle BI Administrator may choose one of the following:

- Map the member attribute to the appropriate Logical Table in the Business Model Layer
- Include the attribute in the appropriate Presentation Table in the Presentation Layer

9.1.3.8 Unbalanced Hierarchies

Oracle BI EE 10.1.3.3.2 supports unbalanced hierarchies for Essbase sources. The following is an illustration of how unbalanced hierarchies will be represented to end users in the Answers interface.

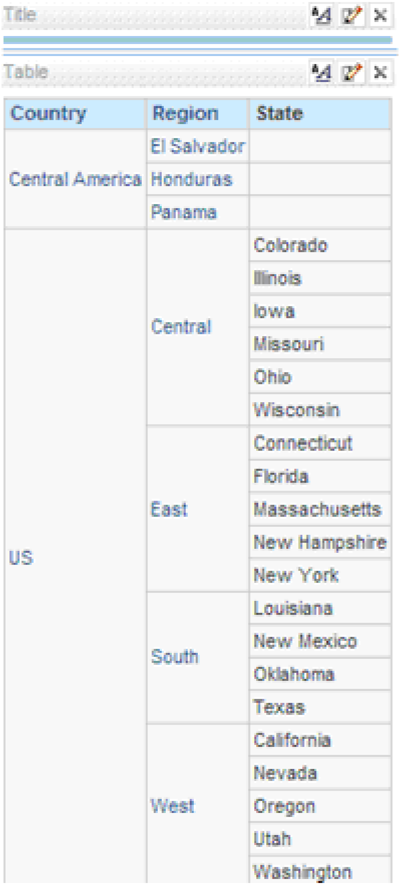
Given a selection of Country, Region as shown in the following image:



Country	Region
Central America	El Salvador
	Honduras
	Panama
US	Central
	East
	South
	West

Now add "State" to the query. We know that for "Central America," Regions do not have States.

The resulting Answers request is shown in the following image:



Country	Region	State
Central America	El Salvador	
	Honduras	
	Panama	
US	Central	Colorado
		Illinois
		Iowa
		Missouri
		Ohio
		Wisconsin
	East	Connecticut
		Florida
		Massachusetts
		New Hampshire
		New York
	South	Louisiana
		New Mexico
		Oklahoma
		Texas
	West	California
		Nevada
		Oregon
		Utah
		Washington

Note that for the unbalanced branch, for columns below the leaf level, nulls will be displayed.

9.2 Using BI EE OC4J Components with IIS via the IIS Plug-in

There are two components of the BI EE installation that require Oracle Containers for Java (OC4J) and will not run natively on Microsoft's Internet Information Server (IIS), those components are:

- Oracle BI Publisher
- Oracle BI Office Server

IIS can be configured as a listener for OC4J. This is accomplished via an IIS proxy plug-in that is provided with the BI EE installation files. When configured, the requests are routed from IIS to OC4J so that it appears to the user that everything is being executed by IIS.

9.2.1 Prerequisites

Following are the prerequisites to use the IIS plug-in:

- Certified Windows Operating System as documented in the *System Requirements and Supported Platforms for Oracle Business Intelligence Suite Enterprise Edition* (SRSP).
- Certified version of IIS as documented in the SRSP.

- Oracle BI EE version 10.1.3.3.1 or later, installed and running with default configuration.

9.2.2 Configuration Steps

The Oracle Application Server Proxy Plug-in is called oracle_proxy.dll and is located in your Oracle BI EE installation files. The following steps describe how to configure your system to use the IIS plug-in:

1. From your BI EE install files, locate oracle_proxy.dll. The navigation path is as follows:

   ```
   <CD-Drive>\Server\Oracle_Business_
   Intelligence\oc4jproxy\oracle_proxy.dll
   ```

2. Create a folder on an accessible drive, for example: `c:\proxy`. Copy oracle_proxy.dll to this folder.

3. In the same folder, create a configuration file called "proxy.conf ". This configuration file is for the OracleAS Proxy Plug-in which will be used to define the proxy server behavior. Following is a sample configuration file:

   ```
   # Server names that the proxy plug-in will recognize.
   oproxy.serverlist=ias1

   # Hostname to use when communicating with a specific server.
   oproxy.ias1.hostname=hostname.domain

   # Port to use when communicating with a specific server.
   oproxy.ias1.port=9704

   # Description of URL(s) that will be redirected to this server.
   oproxy.ias1.urlrule=/xmlpserver/*
   oproxy.ias1.urlrule=/xmlpserver
   oproxy.ias1.urlrule=/bioffice/*
   oproxy.ias1.urlrule=/bioffice
   ```

   When you complete this Step, there will be two files (oracle_proxy.dll and proxy.conf) in the folder that you created in Step 2.

4. Define the OracleAS Proxy Plug-in Registry as follows:

   1. Edit your registry to create a new registry key named HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\IIS Proxy Adapter.

   2. Specify the exact location of your configuration file with the name server_defs, and a value pointing to the location of your configuration file, for example: c:\proxy\proxy.conf.

   3. (Optional) Specify a log_file and log_level: Add a string value with the name log_file, and the desired location of the log file, for example, c:\proxy\plugin.log. Add a string value with the name log_level, and a value for the desired log level. Valid values are "debug", "inform", "error", and "emerg".

5. Create the "oproxy" virtual directory in IIS as follows:

   1. Using the IIS management console, add a new virtual directory to your IIS Web site with the same physical path as that of oracle_proxy.dll. Name the directory "oproxy" and give it execute access.

   2. Using the IIS management console, add oracle_proxy.dll as a filter in your IIS Web site. The name of the filter should be "oproxy" and its executable must

point to the directory that contains oracle_proxy.dll, for example, `c:\proxy\oracle_proxy.dll`.

3. Restart IIS (stop and then start the IIS server), ensuring that the filter is marked with a green arrow pointing up.

6. Check the following configuration files to remove the port 9704 because now IIS is routing all the calls to OC4J.

\oracleBIData\web\Config\instanceconfig.xml
\oracleBI\xmlp\Admin\configutation\xmlp-server-config.xml
\OracleBI\oc4j_bi\j2ee\home\applications\bioffice\bioffice\WEB-INF\bioffice.xml

7. To access BI Servlets from the IIS / OracleAS Proxy Plug-in, you must specify the complete URL for example:

   `http://localhost/xmlpserver/login.jsp`

   or

   `http://localhost/bioffice/about.jsp`

   and

   `http://localhost/analytics/saw.dll?Answers`.

Part II

New Features for Oracle BI Publisher

Part II contains the following chapters:

10

Oracle BI Publisher Template Builder Enhancements

There are many new and exciting enhancements to the Oracle BI Publisher Template Builder. These new features enable users to use Discoverer worksheets as data sources, generate reports in MS PowerPoint format, and to use more powerful formatting facilities and charts.

The Oracle Business Intelligence Publisher version 10.1.3.3 introduces the following new features for the Oracle BI Publisher Template Builder:

- Integration with Discoverer
- Support for Microsoft PowerPoint Output
- Advanced Form Field Support
- Enhanced Chart Builder Features

10.1 Changes to the Oracle BI Publisher Menu

In support of these new features, you will notice the Oracle BI Publisher menu contains the following new items:

Added to the Insert menu:

- **Repeating Group**
- **Conditional Format**
- **Conditional Region**

For information on these menu items, see Section 10.4, "Advanced Form Field Support".

Added to the Preview menu: **PowerPoint**

For information on this menu item, see Section 10.3, "Microsoft PowerPoint Output Support".

In addition, two menu items have been changed:

- **Update Layout Template is now Upload Template**
- **Publish Template As is now Upload Template As**

The functionality of these two menu items has not changed.

10.2 Integration with Oracle BI Discoverer

In this release BI Publisher accepts Discoverer worksheets as report data sources. To facilitate building reports from the Discoverer worksheets, the Template Builder enables you to connect to Discoverer, browse worksheets, and load data to the Template Builder directly from Discoverer.

To open a Discoverer Report in the Template Builder:

Prerequisite: You must have set up integration with Oracle BI Discoverer in the BI Publisher Admin page. For more information see Chapter 13, "Oracle BI Publisher Integration with Oracle BI Discoverer".

1. From the Oracle BI Publisher menu, select **Log On**.
2. Enter your credentials for Oracle BI Publisher. After successfully logging on, the Open Template dialog will launch.
3. From the Workspace list, select **Discoverer**. The Select Data Connection dialog will launch.
4. From the Select Data Connection dialog, select your connection to Discoverer.
5. Browse the Discoverer folders to locate the worksheet you want to open in the Template Builder. Double click the worksheet. A dialog prompts you to save As Oracle BI Publisher Report.
6. From the dialog, select the folder on the BI Publisher server where you want to save your BI Publisher report that will use this Discoverer worksheet as the data source.
7. Click **OK** to create a report in the folder. This also loads the report structure and data to the Template Builder.
8. You can now design the template using the full range of Template Builder features on your Discoverer data source.
9. When finished designing your layout template, save your template to the BI Publisher server by selecting **Upload Template** from the Oracle BI Publisher menu.

10.3 Microsoft PowerPoint Output Support

In this release, BI Publisher now supports Microsoft PowerPoint as an output option. The Template Builder also supports PowerPoint as a preview option.

To preview your report in PowerPoint, select **PowerPoint** from the Oracle BI Publisher Preview menu.

This action will launch Microsoft PowerPoint and display your report as a PowerPoint presentation. Each report page will be displayed on a new slide.

Note that to view BI Publisher PowerPoint output, you must have Microsoft PowerPoint 2003 or later installed on your computer.

For additional information regarding usage and limitations of the PowerPoint output feature, seeChapter 12, "Oracle BI Publisher PowerPoint Output".

10.4 Advanced Form Field Support

In previous releases, BI Publisher inserted abbreviated XSL code into the Form Field Help Text Dialog. Long code strings required inserting code into both the Status Bar and Help Key tabs of the Form Field Help Text Dialog. This made it challenging if you wanted to edit this code. With this release, you can now edit code in the form fields in a single window in the BI Publisher Properties dialog. Further, several additional dialogs have been added to enable you to insert common format constructs in your template and the Template Builder generates the code for you.

The new dialogs simplify the insertion and editing of the following report elements:

- Data field Properties
- Repeating Groups
- Conditional Formatting
- Conditional Regions

To open the Properties dialog for a field, perform one of the following:

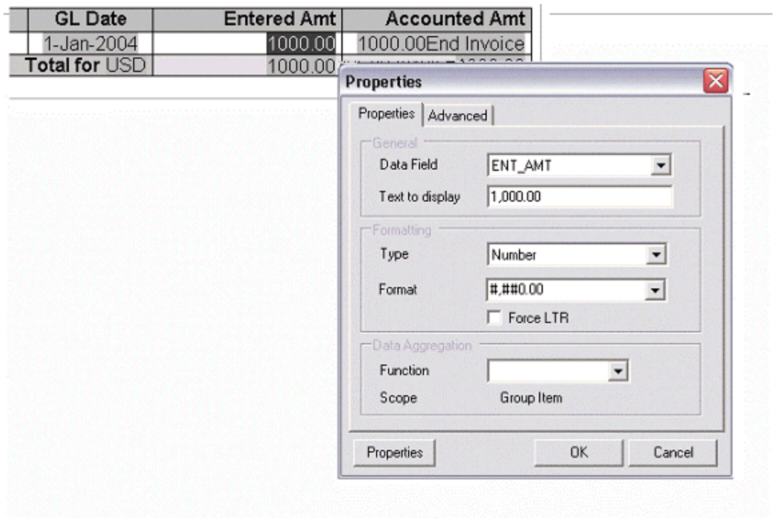
- Double-click the form field in your template.
- Right-click the form field, from the menu select BI Publisher and then select Properties.

For recognized code patterns (for example: data value, for-each, for-each-group, conditional formatting) the dialog displays a Properties tab and an Advanced tab (the Repeating Group dialog also displays a Sorting tab). The Properties tab displays modifiable properties for the field. The Advanced tab displays the underlying code. If the code pattern is not recognized, the Properties dialog displays only the Advanced tab with the code from the form field.

Note that each dialog includes a Properties button that enables you to access the Microsoft Word Form Field Help Text dialog.

10.4.1 Edit Data Fields

The following screenshot displays the Properties dialog invoked for a data field:



GL Date	Entered Amt	Accounted Amt
1-Jan-2004	1000.00	1000.00End Invoice
Total for USD	1000.00	

Data Field: ENT_AMT

Text to display: 1,000.00

Type: Number

Format: #,##0.00

☐ Force LTR

Function:

Scope: Group Item

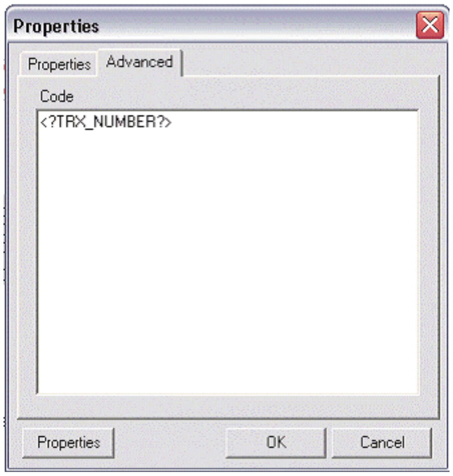
You can set the following properties for a data field:

- **Data Field** — Select the data field from the list of available fields from the loaded data source.
- **Text to Display** — Enter the display text for the form field in the template. This text will be replaced at runtime by the value in the data.

- **Type** — Select the type of data. Options are Regular Text, Number, Date, Current Date, Current Time. The selection in this field will determine the format options.
- **Format** — For any data type except Regular Text, you can select from several number or date display formatting masks or enter your own.
- **Force LTR** — (Force Left-to-Right) This checkbox is only needed if you are using the template in a language that prints the characters from right to left, such as Arabic or Hebrew. In these languages you may want to force left-to-right printing for fields such as phone numbers, addresses, postal/zip codes, or bank account number.
- **Function** — This feature enables you to perform aggregation functions (Sum, Average, Count, Minimum, Maximum) on data fields. For example, if you select sum for a data field, the field will show the sum of all occurring values for this data field depending on the scope (see below).
- **Scope** (informational only) — This field has two possible values:
  - **Group Item** — Indicates that the data field is inside a group. If you choose to perform a function on the field, only the occurrences of the field within the current group will be included in the aggregation.
  - **Normal** — Indicates that the field is not inside a group. Aggregation functions will be performed on all occurrences of the field in the data.

10.4.2 Advanced Tab

The following screenshot shows the Advanced tab of the Properties dialog:



```
<?TRX_NUMBER?>
```

The Advanced tab displays the underlying code.

If the code pattern within the form field is not recognized, the Properties dialog will display this tab only. Use this tab to edit or add code to the form field. Select OK to update the template.

The Properties button opens the Microsoft Word Text Form Field Options dialog.

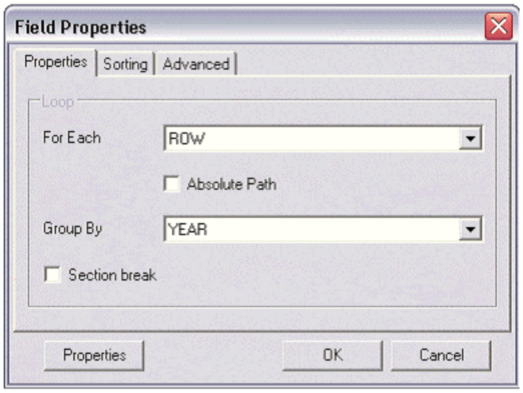
10.4.3 Insert and Edit Repeating Groups

The Template Builder now enables you to insert and edit for-each and for-each group fields using a Field Properties dialog. Use the dialog to choose the data elements to group and the data element to group them by.

10.4.3.1 Creating Repeating Groups

To create a repeating group:

1. Select the section of the template that contains the elements you want repeated.
2. From the Oracle BI Publisher menu, select **Insert**, then **Repeating Group**.
3. Enter the appropriate fields in the Field Properties dialog:



Field Properties

Properties | Sorting | Advanced

Loop

For Each: ROW

□ Absolute Path

Group By: YEAR

□ Section break

   - **For Each** — Select the repeating element to create the loop. When you select the For Each data field you are telling BI Publisher that for each occurrence of the field in the data you want the elements and processing instructions contained within the loop to be repeated.

     Please note the following about creating repeating groups:

     – For loops and groupings not inside another group (that is, outer groups or loops) you must select the repeating XML element to be used. For example if the data set is flat, the only repeatable element is /DATA/ROWSET/ROW. In cases with multiple data sources or hierarchical XML you can choose the data set.

     – If you are creating nested groups (inserting a loop or group inside of another loop in the template), the For Each field is not updateable because it is already defined by the preexisting outer loop. The For Each field will display "Group Item" to inform you that an outer group is already defined.

   - **Absolute Path** — Select this check box to use the Absolute Path to the element in the XML structure. This is important if your data contains the same element name grouped under different parent elements.
   - **Group By** — Select a field from the list by which you want to group the data. If you just want to create a simple loop, do not select a group by element.
   - **Section break** — Select this box if you wish to create a section break between each loop. It is important to note the following characteristics of this feature:

     – The section break can be created only on outer loops that surround the whole document.

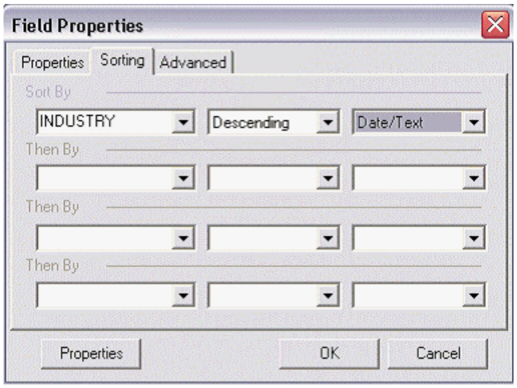
     – As a section break, the page numbering is reset, headers and footers are reset, and any running calculations will be reset.

4. To sort the grouped data, select the Sorting tab. You can select up to four sort-by fields. For each sort by field, select the following:
   - **Sort order** — Select Ascending or Descending.

- **Data type** — Select Number or Date/Text. It is important that you select the correct data type to achieve the expected sort order.

Note that if you are sorting by four criteria and your XML data element names are long, you may exceed the character length limitation (393 characters) of the Microsoft Word form field.

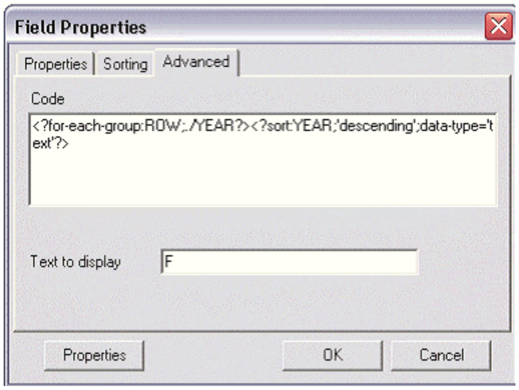
The following figure shows the Sort tab of the Repeating Group Field Properties dialog:



Sort By
INDUSTRY | Descending | Date/Text

5. To change the text display or edit the code directly, select the Advanced tab.

   The following following figure shows the Advanced tab of the Repeating Group Field Properties dialog:



Code

```
<?for-each-group:ROW;./YEAR?><?sort:YEAR;'descending';data-type='t
ext'?>
```

Text to display: F

6. When you have completed the dialog options, click **OK**. This will insert the form fields in your template. By default, the beginning for-each form field will display the text "F" and will be inserted at the beginning of the selected template section. At the end of the selection, an "E" form field will be inserted to denote the end of the repeating group.

10.4.3.2 Create Grouping Fields Around a Block

To create a group around an existing block of text in a template:

1. Select the block of text. For example, a table row.

   Note that if any already inserted BI Publisher tags are included in the block, you must be sure to include the beginning and ending tags. For example, if your block contains any opening for-each, if, or for-each-group tags, you must include the end for-each, end-if, and end for-each-group tags in your selection.

2. From the Oracle BI Publisher menu bar select **Insert**, then **Repeating Group**.

3. In the Field Properties dialog, enter the fields to define the group as described in Section 10.4.3.1, "Creating Repeating Groups".

4. Click **OK** to insert the grouping fields around the block. For example, if the block is a table row, the begin field will be inserted at the beginning of the first cell and the end field will put at the end of the last field.

10.4.4 Insert Conditional Formatting

Using the Advanced Form Field features, you can insert simple conditional formats to apply to table rows or cells. The dialog provides several common options that you can select and the Template Builder inserts the code automatically. The Conditional Format dialog supports two conditions per field.

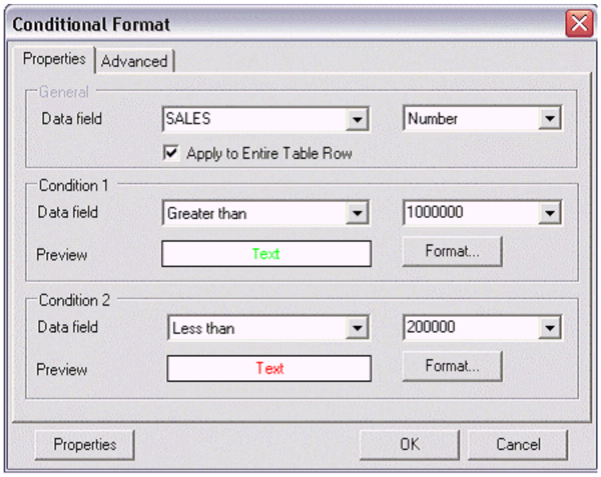
Use this feature to easily construct formats such as displaying values below a certain threshold in one format, and above the threshold in another.

To insert a conditional format:

1. Place the cursor in the table cell of the data element for which you want to define the condition.

2. From the Oracle BI Publisher toolbar, select **Insert**, then **Conditional Format**.

3. Enter the following in the Conditional Format dialog:



Data field: SALES; Number
☑ Apply to Entire Table Row

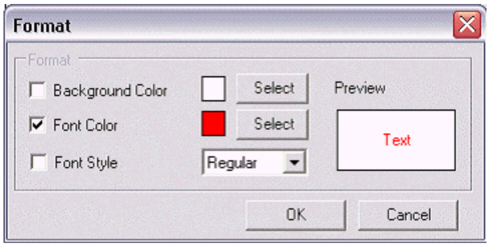
Condition 1
Data field: Greater than; 1000000
Preview: Text

Condition 2
Data field: Less than; 200000
Preview: Text

- **Data Field** — Select the element to test for the condition and the data type of the element (Number or Date/Text).
- **Apply to Entire Table Row** — If you want the format applied to the entire table row, not just the cell of the selected element, select this box.
- **(Condition 1) Data field** — Select the comparison operator. The options are:
  - Equal to
  - Not equal to
  - Greater than
  - Less than
  - Greater than or equal to
  - Less than or equal to

- Select the value to meet the condition. Note that you may enter an integer, enter text, or select another data element to define a comparison based on the incoming values.
- Click **Format** to define the format you want to apply when the condition is met. Options are background color, font color, and font style (regular, bold, italic, bold italic). Select the box and format of each option you want to apply. After you select the format, the Preview region will display the format chosen.

  The following figure shows the Format dialog from the Conditional Format dialog:



Format

Format

□ Background Color □

☑ Font Color 🟥

□ Font Style: Regular

Preview: Text

- Define a second condition if desired.

4. Select **OK**. The conditional format field will be inserted as a form field with the display text "C".

To edit the conditional format, double-click the inserted form field to launch the dialog for editing; or, right-click the form field and select **BI Publisher**, then **Properties**.

10.4.5 Inserting and Editing Conditional Regions

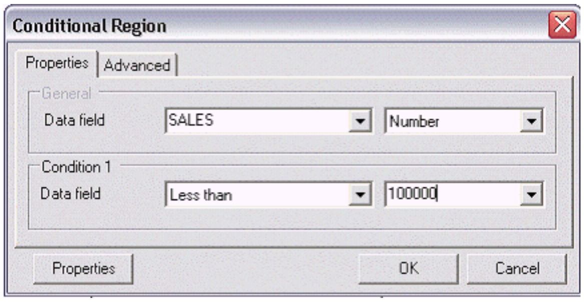
A conditional region is an area that is surrounded by a conditional statement. If the statement tests true, the area is displayed in the report; if the condition tests false, the area is suppressed from the report.

For example, your data contains salary and position information for all employees. Your report contains a table that displays the employee, position, and salary. You want your report to display salary information for managers only. Using the insert conditional region functionality, you can select the region that contains the employee table and insert the condition that the position element must equal manager.

To insert a conditional region:

1. Select the region that you want to apply the condition to. For example, if you want to display a table only for a certain condition, select the region that contains the table. Note that the region must be inside a loop.
2. From the BI Publisher toolbar select **Insert**, then **Conditional Region**.

   The following figure shows the Conditional Region dialog:



Conditional Region

General

Data field: SALES, Number

Condition 1

Data field: Less than, 100000

3. Enter the following fields:
   - **Data field** - Select the field to test for the condition.
   - Select the data type of the field: Number or Date/Text.
   - Select the comparison operator for the data field. Options are:
     - Equal to
     - Not equal to
     - Greater than
     - Less than
     - Greater than or equal to
     - Less than or equal to
     - Enter the value to define the condition, or select from the list of data elements to define a conditional relationship based on the value of another data element.
4. Click **OK**. The form fields containing the conditional logic are inserted around the region. The beginning form field will display the text "C" and the form field closing the region will display the text "EC".

To edit the conditional region, double-click the inserted form field to launch the dialog for editing; or, right-click the form field and select **BI Publisher**, then **Properties**.

10.5 Chart Builder Enhancements

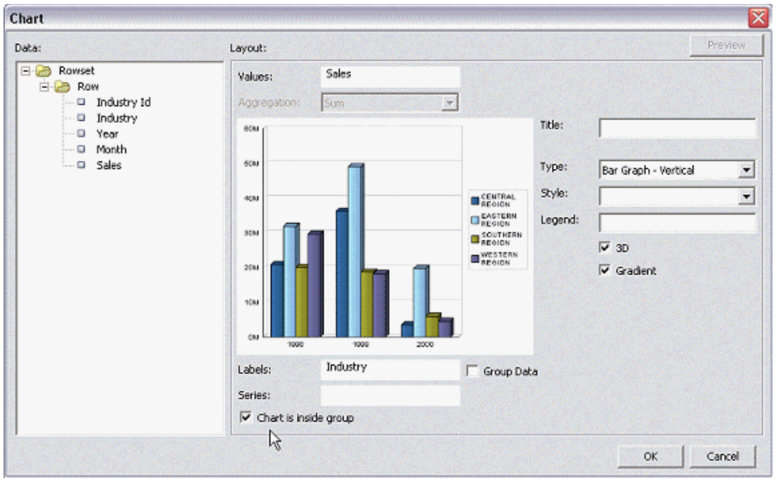
Enhancements to the Template Builder's chart builder include:

- Support for charts inside a grouping
- Support for new chart types
- Support for new chart subtypes

10.5.1 Support for Charts Inside a Grouping

In this release you can now insert a chart inside a grouping of data and indicate through the chart builder to base the chart on the surrounding grouping.

The following figure shows the new check box inside the chart builder dialog:



Values: Sales

Aggregation: Sum

Title:

Type: Bar Graph - Vertical

Style:

Legend:

☑ 3D

☑ Gradient

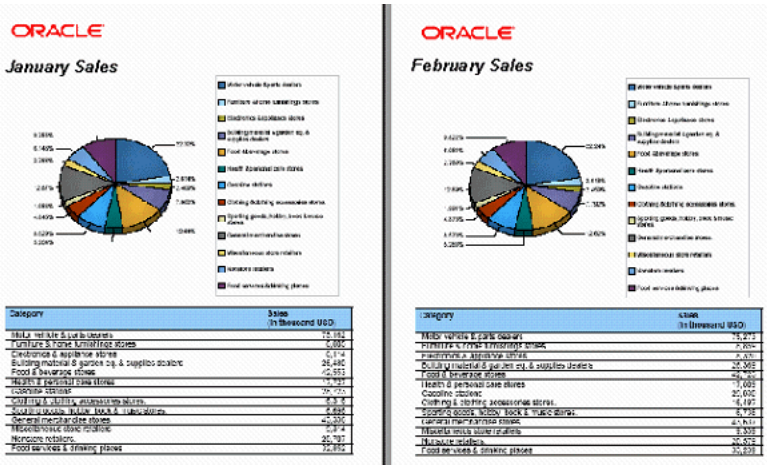
Labels: Industry

☐ Group Data

Series:

☑ Chart is inside group

Using this feature, you can easily insert charts within groups to display the data for each grouping. For example: Your report contains sales data. Your template groups the data by year than by month to report the month's sales by industry. To insert a chart to display each month's data, simply insert the chart within the grouping tags and select the Chart is inside group box. This enables you to create reports as shown in the following figure of a sample report with chart inside grouping:



Category	Sales (In thousand USD)
Motor vehicle & parts dealers	75,162
Furniture & home furnishings stores	[illegible]
Electronics & appliance stores	[illegible]
Building material & garden eq. & supplies dealers	26,480
Food & beverage stores	42,553
Health & personal care stores	15,727
Gasoline stations	[illegible]
Clothing & clothing accessories stores	[illegible]
Sporting goods, hobby, book & music stores	[illegible]
General merchandise stores	42,300
Miscellaneous store retailers	[illegible]
Nonstore retailers	20,787
Food services & drinking places	32,852

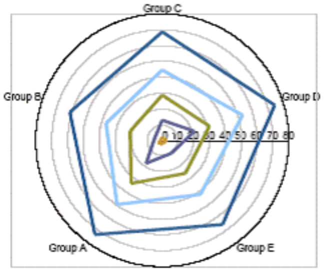
Category	Sales (In thousand USD)
Motor vehicle & parts dealers	75,273
Furniture & home furnishings stores	[illegible]
Electronics & appliance stores	[illegible]
Building material & garden eq. & supplies dealers	26,365
Food & beverage stores	[illegible]
Health & personal care stores	17,085
Gasoline stations	20,600
Clothing & clothing accessories stores	16,497
Sporting goods, hobby, book & music stores	6,736
General merchandise stores	[illegible]
Miscellaneous store retailers	9,335
Nonstore retailers	20,579
Food services & drinking places	30,209

10.5.2 New Chart Types

The following chart types are now available as options from the chart builder:

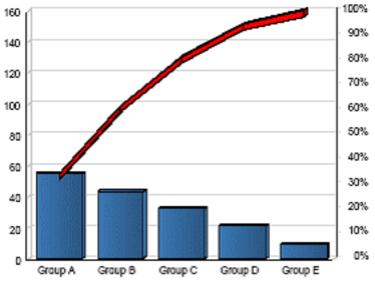
- **Radar**— Plots the same information as a bar graph, but displays the data radiating from the center of the graph.

  The following figure shows a sample radar chart:



- **Pareto** — Displays criteria in descending order to summarize the relative importance of the differences between groups of data. In this graph type, a line shows a cumulative total of the percentages. When using this type of chart, ensure that you sort the data numerically, descending by value (not label).

  The following figure shows a sample pareto chart:



10.5.3 New Chart Subtypes

The chart builder now supports additional chart subtypes in this release. Stacked and percentage subtypes are now supported for bar charts, line charts, and area charts. These are displayed as follows on the chart type list:

- Bar - Vertical - Percent
- Bar - Vertical - Stacked
- Bar - Horizontal - Percent
- Bar - Horizontal - Stacked
- Line - Percent
- Line - Stacked
- Area - Percent
- Area - Stacked

11

Flash Templates for Oracle BI Publisher

This new feature enables report authors to develop Adobe FLEX templates that can be applied to BI Publisher reports to generate highly interactive Flash output documents.

In this release, Oracle BI Publisher offers support for Adobe Corporation's new document format for building interactive forms and reports, called Flex. You can build Flex templates, test them on your desktop, and deploy them to the BI Publisher server to generate Flash output. Users are then able to run the reports from the BI Publisher user interface or schedule them for delivery to report consumers.

This chapter will cover how to set up a Flex template with a BI Publisher "flat" data source (that is, there is no hierarchy in the XML data) and will cover some of the simpler objects such as tables and charts. For more information about interactivity, connectivity between components and more advanced topics, refer to the Adobe's Flex documentation.

11.1 Changes to the BI Publisher User Interface

To support Flash templates, you will notice the following changes in the BI Publisher user interface:

- On the Edit Report page, under Layouts, General Settings, the Template Type list now includes the option **Flash Template**.
- On the View Report page, the output selection list will now include the option **Flash** when a Flash Template is selected as the Template Type. Flash templates only support the output types: Flash, PDF, and data.
- A new set of "Flash" properties have been added to the Runtime Configuration page.

11.2 Prerequisites

The prerequisites are as follows:

- For viewing output:
  - To view the report output from the Flash Template, you must have Adobe Flash Player 9 installed on your computer.
- For building templates:
  - The FlexBuilder IDE from Adobe. Oracle BI Publisher is currently certified with version 2.0.1. The tool can be downloaded and purchased from the Adobe Web site - http://www.adobe.com/products/flex/.

Please note that the charting functionality is available for an additional license fee.

– A report data model set up in BI Publisher that generates flat XML.

11.3 Building Templates

This section includes the following topics:

- "Adding the Data Source"
- "Creating the Layout"
- "Data Binding"

11.3.1 Adding the Data Source

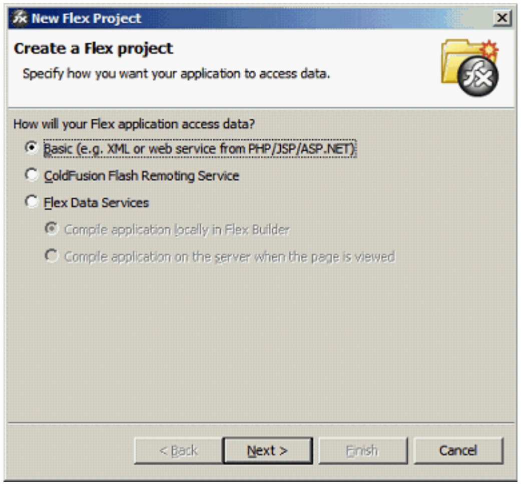
1. Generate a sample data file from your report data model. From the BI Publisher Report Editor or from the Reports page, select View. If no layouts are defined for your report, then the output type will default to xml. Otherwise, choose Data for the output type. Select Export and save the results as an XML file to a local directory. This example is based on the following data:

```
<ROWSET>
<ROW>
<NAME>Neena Kochhar</NAME>
<FIRST_NAME>Neena</FIRST_NAME>
<LAST_NAME>Kochhar</LAST_NAME>
<SALARY>17000</SALARY>
<ANNUAL_SALARY>204000</ANNUAL_SALARY>
<FED_WITHHELD>57120</FED_WITHHELD>
<JOB_TITLE>Administration Vice President</JOB_TITLE>
<DEPARTMENT_NAME>Executive</DEPARTMENT_NAME>
<MANAGER>Steven King</MANAGER>
</ROW>
<ROW>
...
</ROWSET>
```

This data is generated from the following simple query-based report:

```
select
        e.first_name || ' ' || e.last_name name,
        e.first_name,
        e.last_name,
        e.salary,
        e.salary*12 ANNUAL_SALARY,
        e.salary*12*0.28 FED_WITHHELD,
        j.job_title,
        d.department_name,
        m.first_name || ' ' || m.last_name manager
from employees e,
      employees m,
      departments d,
      jobs j
where e.department_id = d.department_id
  and j.job_id = e.job_id
  and e.manager_id = m.employee_id
```

2. Open the Flex IDE and create a new Flex Project, select the "Basic" data access method.


New Flex Project

Create a Flex project

Specify how you want your application to access data.

How will your Flex application access data?

● Basic (e.g. XML or web service from PHP/JSP/ASP.NET)

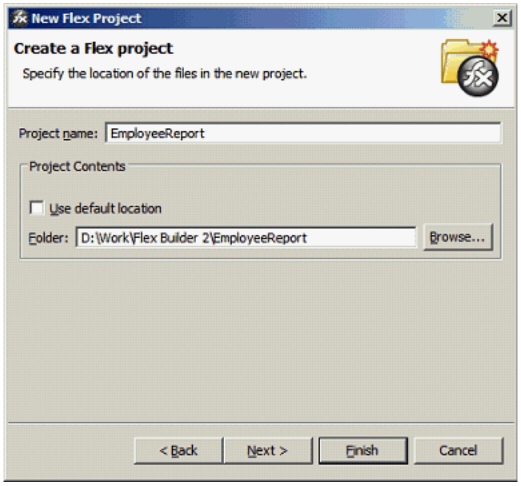
○ ColdFusion Flash Remoting Service

○ Flex Data Services

● Compile application locally in Flex Builder

○ Compile application on the server when the page is viewed

On the next screen give the project a name. The name you use here will be assigned to the template file name you are going to create.


New Flex Project

Create a Flex project

Specify the location of the files in the new project.

Project name: EmployeeReport

Project Contents

□ Use default location

Folder: D:\Work\Flex Builder 2\EmployeeReport

Click **Finish**.

The IDE creates an MXML file; this is the Flex template definition file. It is in an XML format as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
</mx:Application>
```

You can now update it manually or by using the visual builder.

3. Connect the XML you downloaded from your report data model. We will use the XML data services that Flex supports. We will embed the sample data into the MXML file.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
 <mx:Script>
   <![CDATA[
      [Bindable]
      public var dataXML:XML =
<ROWSET>
<ROW>
<NAME>Neena Kochhar</NAME>
<FIRST_NAME>Neena</FIRST_NAME>
<LAST_NAME>Kochhar</LAST_NAME>
<SALARY>17000</SALARY>
<ANNUAL_SALARY>204000</ANNUAL_SALARY>
<FED_WITHHELD>57120</FED_WITHHELD>
<JOB_TITLE>Administration Vice President</JOB_TITLE>
<DEPARTMENT_NAME>Executive</DEPARTMENT_NAME>
<MANAGER>Steven King</MANAGER>
</ROW>
<ROW>
...
</ROWSET>;
        ]]>
 </mx:Script>
</mx:Application>
```

The XML portion should look familiar as the data you downloaded. The important components to note are:

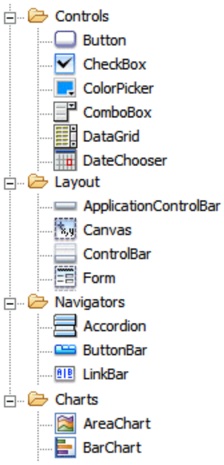
- `<mx:Script>` — This denotes the start of the template scripting code. There is also a closing `</mx:Script>` statement.
- `[Bindable]` — This denotes that the following variable is bindable to a layout component.
- `public var dataXML:XML` — This is the data variable declaration:
  - `public` — The value of the variable is available to the whole template.
  - `var` — Declares there is a variable in the report.
  - `dataXML` — The name of the variable. Note this is a compulsory name. You must use this name to use the template with BI Publisher.
  - `:XML` — Declares that the variable is an XML type.
- `;` — Notice the semicolon after the end of XML data you provided.

At runtime the BI Publisher server will generate the runtime data from the report and inject it into the Flex template replacing the sample data held within the dataXML variable. This feature allows the Flex report to be distributed to users without needing to connect to the server.

11.3.2 Creating the Layout

Now we can start adding report objects to the layout palette. This example uses the Flex Design tab to add the objects to the layout. Click the Design tab to see the available objects in the Component Navigator pane.

The following figure shows an example of the available objects in the Component Navigator pane:



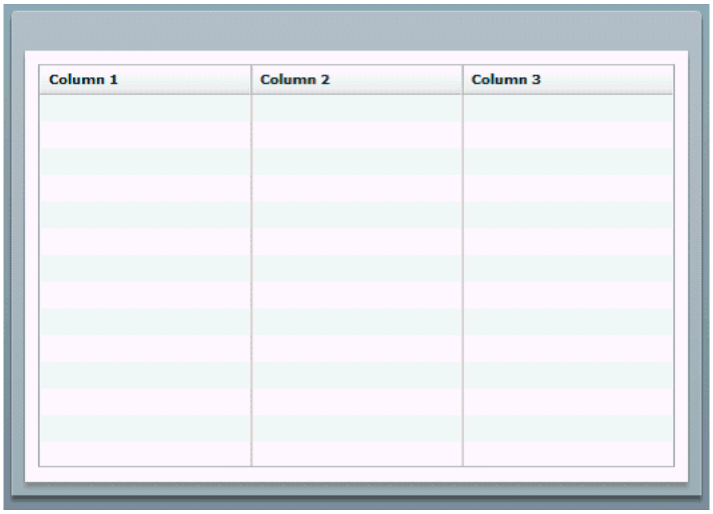
- Controls
  - Button
  - CheckBox
  - ColorPicker
  - ComboBox
  - DataGrid
  - DateChooser
- Layout
  - ApplicationControlBar
  - Canvas
  - ControlBar
  - Form
- Navigators
  - Accordion
  - ButtonBar
  - LinkBar
- Charts
  - AreaChart
  - BarChart

These objects can be dragged and dropped to the design palette.

1. The Flex IDE creates a default canvas for you to drop objects onto. You can modify the canvas as required.

   IMPORTANT: If you intend to embed the Flash output in a PDF document, you must set the Width and Height of the template in the Size region of the Layout properties. Even if you wish to accept the default size, you must explicitly enter values in these fields.

2. Start by dragging a Panel object from under the Layout node to the design palette. Notice as you drag the panel around the edge of the palette, you will see guidelines displayed in blue. Use these guides to aid you in aligning objects.
3. Drop the panel onto the top left hand corner of the palette.
4. Now drag the bottom right edge of the panel across to the right hand side of the palette.
5. Then drag it down to about half the height of the palette. Alternatively, use the property palette on the right hand side to set the size of the panel.
6. Now pick up a Datagrid object. This is the object to render the data in a tabular format. Drop it onto the panel you created in Step 2. The Datagrid is now a child of the panel; you can resize it as needed. The end result is shown in the following figure:



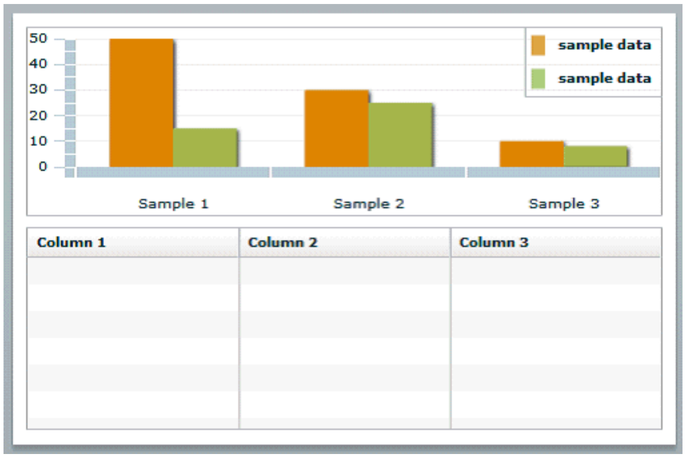
Column 1	Column 2	Column 3

By default three columns are generated. In the next section you will override the default in the MXML code.

11.3.2.1 Adding a Chart

If you have purchased the charting option you can add charts to your layout.

1. First make some room for the chart in your layout. Highlight the Datagrid object and pull the top edge down to about half the height of the hosting panel.
2. For the purposes of this guide we will demonstrate a Column Chart. Select and drag a Column Chart from the design palette and drop it onto the hosting panel. Use the guidelines to get it aligned.
3. Once you drop it you will notice the default size will overlap the Datagrid and the chart legend is in the top left-hand corner. Resize the chart and move the legend to the left to look similar to the following figure:



Column 1	Column 2	Column 3

This is a sample chart. You will bind it to the data in the next section.

11.3.3 Data Binding

Now that our layout is complete, we can bind the layout objects to the data source. Flex does offer some help through the property palette of the objects to define the binding, but not enough to complete the task. Therefore we are going to update the MXML directly using the "Source" editor.

11.3.3.1 Binding the DataGrid

To bind the DataGrid:

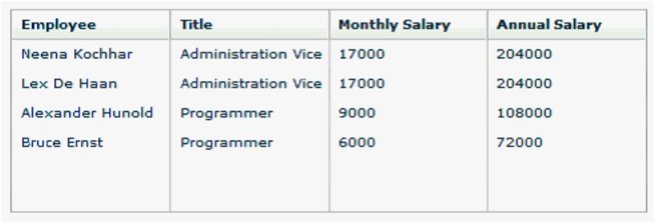
1. Start by highlighting the Datagrid in the design palette, and then click the Source tab to bring up the MXML source. You will see that the first line of the Datagrid code has been highlighted for you. This is a useful feature if you have built complex Flex templates and need to locate the code easily.

   The Datagrid code is as follows:

   ```
   <mx:DataGrid x="10" y="160" width="476" height="152">
      <mx:columns>
         <mx:DataGridColumn headerText="Column 1" dataField="col1"/>
         <mx:DataGridColumn headerText="Column 2" dataField="col2"/>
         <mx:DataGridColumn headerText="Column 3" dataField="col3"/>
      </mx:columns>
   </mx:DataGrid>
   ```

   Notice that it defines the relative x,y position of the grid within its parent container and its width and height. The next element defines the columns with attributes for the header label and the data fields.

   The goal is to achieve a table that looks like the following figure:



Employee	Title	Monthly Salary	Annual Salary
Neena Kochhar	Administration Vice	17000	204000
Lex De Haan	Administration Vice	17000	204000
Alexander Hunold	Programmer	9000	108000
Bruce Ernst	Programmer	6000	72000

2. Make the Datagrid aware of the datasource. To do that, add an attribute to the `<mx:DataGrid>` element as follows:

   ```
   dataProvider="{dataXML.ROW}"
   ```

   This attribute defines the data object to be used at runtime to populate the grid. Remember that we initially defined the XML data variable as "dataXML"; we now use that followed by "ROW". ROW is the repeating group in the data set. Don't forget the curly braces to let the Flex engine know we are defining a datasource.

3. Bind the columns. We already have the basic structure, so it is just a case of replacing values thus:

   ```
   <mx:columns>
       <mx:DataGridColumn headerText="Employee" dataField="NAME" />
       <mx:DataGridColumn headerText="Title" dataField="JOB_TITLE"/>
       <mx:DataGridColumn headerText="Monthly Salary" dataField="SALARY"/>
       <mx:DataGridColumn headerText="Annual Salary" dataField="ANNUAL_SALARY"/>
   </mx:columns>
   ```

4. With this update we now have the following definition for the datagrid:

   ```
   <mx:DataGrid x="10" y="160" width="476" height="152"
   dataProvider="{dataXML.ROW}">
     <mx:columns>
       <mx:DataGridColumn headerText="Employee" dataField="NAME" />
   ```

```
    <mx:DataGridColumn headerText="Title" dataField="JOB_TITLE"/>
    <mx:DataGridColumn headerText="Monthly Salary" dataField="SALARY"/>
    <mx:DataGridColumn headerText="Annual Salary" dataField="ANNUAL_SALARY"/>
  </mx:columns>
</mx:DataGrid>
```

5. You can now try the template with your sample data. Use the Run > Run EmployeeReport. This will open a new browser window and render the table with your sample data as seen in Step 1.

11.3.3.2 Binding the Chart

To bind the chart:

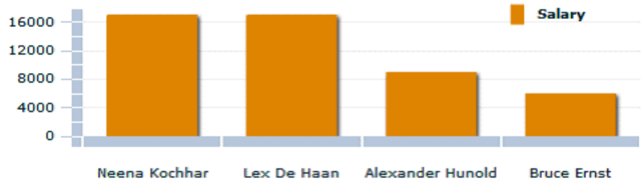
1. Start by bringing up the Design tab and highlight the Chart, then flip back to the Source view to find the chart code:

```
<mx:ColumnChart x="10" y="10" id="columnchart1" width="476" height="142">
  <mx:series>
    <mx:ColumnSeries displayName="Series 1" yField=""/>
  </mx:series>
</mx:ColumnChart>
<mx:Legend dataProvider="{columnchart1}" x="383" y="10"/>
```

2. To bind the data source to the chart object, add the dataProvider attribute to the `<mx:ColumnChart>` as follows:

```
dataProvider="{dataXML.ROW}"
```

3. Next add in the binding for the horizontal axis and the column series. This requires a little more effort. Check the Flex help files for more details. We want to create a chart showing salaries by employees, similar to the following:



To achieve this format, modify the `<series>` group and add a `<horizontalAxis>` element thus:

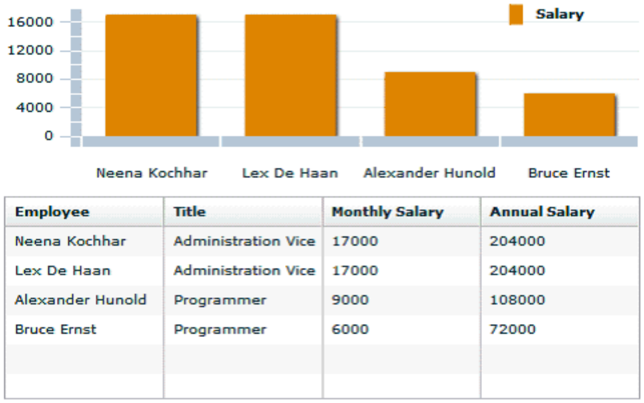
```
<mx:ColumnChart x="10" y="10" id="columnchart1" width="476" height="142"
dataProvider="{dataXML.ROW}">
   <mx:horizontalAxis>
     <mx:CategoryAxis  categoryField="NAME" />
   </mx:horizontalAxis>
   <mx:series >
     <mx:ColumnSeries xField="NAME"  yField="SALARY" displayName="Salary"/>
   </mx:series>
</mx:ColumnChart>
<mx:Legend dataProvider="{columnchart1}" x="383" y="10"/>
```

The `<mx:horizontalAxis>` element has been added and the categoryField attribute has the NAME data element assigned. This element is required to render the chart.

The `<mx:series>` element has been updated binding the SALARY value to each of the employee NAMEs to create a bar for each employee.

The legend code does not need updating. Notice the "id" attribute on the ColumnChart matches the dataProvider attribute value on the `<mx:Legend>` element.

4. You can now run the template using your sample data. You should get an output showing the chart above the tabulated data as shown in the following figure:

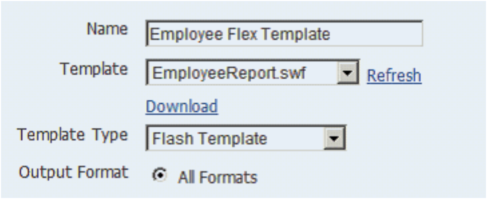


Employee	Title	Monthly Salary	Annual Salary
Neena Kochhar	Administration Vice	17000	204000
Lex De Haan	Administration Vice	17000	204000
Alexander Hunold	Programmer	9000	108000
Bruce Ernst	Programmer	6000	72000

11.4 Loading Templates

Now that the Flex template has been tested on the desktop it can be loaded to the BI Publisher server.

1. Using the Admin or Developer role, navigate to the Edit Report page.
2. Highlight the Layouts node and use the Upload Template field to upload the Flex template to the server. To locate the correct file, use the Browse button and navigate to the Flex project directory. Under this directory open the bin directory and select the EmployeeReport.swf file. Use the Upload button to load the template.
3. With the Layouts node still highlighted click New at the top of the report structure tree. Enter the name, template and template type fields as below.



Name: Employee Flex Template

Template: EmployeeReport.swf

Template Type: Flash Template

Output Format: ● All Formats

4. Save your report. The template will now be available for users to select to either run in real time or to schedule.

11.5 Setting Properties for PDF Output

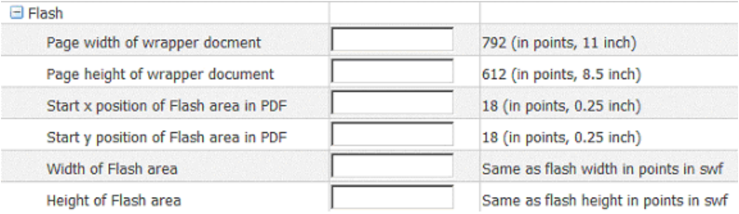
A new set of Flash properties have been added to the Runtime Configuration page to enable you to specify the size and placement of the Flash object when you select PDF as the output type.

IMPORTANT: To produce PDF output you must specify the height and width of the template in the Flex Builder. See Creating the Layout.

To configure the PDF output:

1. Using the Administrator or Developer role, navigate to the report that will use the Flash template and click Configure.

   On the Runtime Configuration page you will see a new set of properties under the Flash heading. These properties control the placement and sizing of the Flash object in your output PDF document. The properties are shown in the following screenshot:



⊟ Flash		
Page width of wrapper docment		792 (in points, 11 inch)
Page height of wrapper document		612 (in points, 8.5 inch)
Start x position of Flash area in PDF		18 (in points, 0.25 inch)
Start y position of Flash area in PDF		18 (in points, 0.25 inch)
Width of Flash area		Same as flash width in points in swf
Height of Flash area		Same as flash height in points in swf

2. Enter values for the properties. Note that no properties are required. If you do not enter any values, the default values assume an 11 inch by 8.5 inch document (standard landscape), with a quarter inch inset from the upper left corner of the page as the insertion point of the Flash object. The default area in the document will be the size of the SWF object.

   - Page width of wrapper document – specify in points the width of the output PDF document. The default is 792, or 11 inches.
   - Page height of wrapper document – specify in points the height of the output PDF document. The default is 612, or 8.5 inches.
   - Start x position of Flash area in PDF – using the left edge of the document as the 0 axis point, specify in points the beginning horizontal position of the Flash object in the PDF document. The default is 18, or .25 inch.
   - Start y position of Flash area in PDF – using the upper left corner of the document as the 0 axis point, specify in points the beginning vertical position of the Flash object in the PDF document. The default is 18, or .25 inch.
   - Width of Flash area – enter in points the width of the area in the document for the Flash object to occupy. The default is the width of the SWF object.
   - Height of Flash area – enter in points the height of the area in the document for the Flash object to occupy. The default is the height of the SWF object.

11.6 Summary

This section has covered the basics of adding and organizing layout objects, binding the objects to BI Publisher data sources to create Flex templates, and then loading the templates to the BI Publisher server to make them available to report users.

This has been a basic introduction to Flex templates but Adobe Flex allows you to build far more complex interactive reports for your users. The animation, "wiring" together and formatting of layout objects can be achieved with Flex. You can also summarize and create calculated fields on the incoming data. Please reference the Flex documentation for these more advanced features.

12

Oracle BI Publisher PowerPoint Output

Oracle BI Publisher 10.1.3.3 adds support for PowerPoint output. This enables you to get report data into your key business presentations.

In this initial version of this functionality, the PowerPoint document generated is a simple export of the formatted data and charts to PowerPoint. In a future release, there will be more sophisticated integration that allows for direct embedding and refreshing from within PowerPoint.

12.1 Supported Versions of PowerPoint

The supported versions of PowerPoint are:

- Microsoft Office PowerPoint 2003
- Microsoft Office PowerPoint 2007

12.2 Configuring Fonts for BI Publisher Server

Support for PowerPoint output does not include the font fallback mechanism that is used for other types of output in BI Publisher. On the BI Publisher server, you must configure each font used in the RTF template for generating PowerPoint output. You will need to copy these fonts to your BI Publisher Server and define the Font Mappings for RTF templates. This can be done for the entire system or for individual reports.

Please see the *Oracle Business Intelligence Publisher User's Guide* section "Defining Font Mappings" for more details:

http://download.oracle.com/docs/cd/B40078_02/doc/bi.1013/b40017/T421739T421745.htm#453835

12.3 Configuring Fonts for BI Publisher Template Builder

In order to correctly preview PPT output that uses non-English or non-standard fonts, you will need to define the fonts in the BI Publisher configuration file. This configuration file is called xdo.cfg and is typically found in:

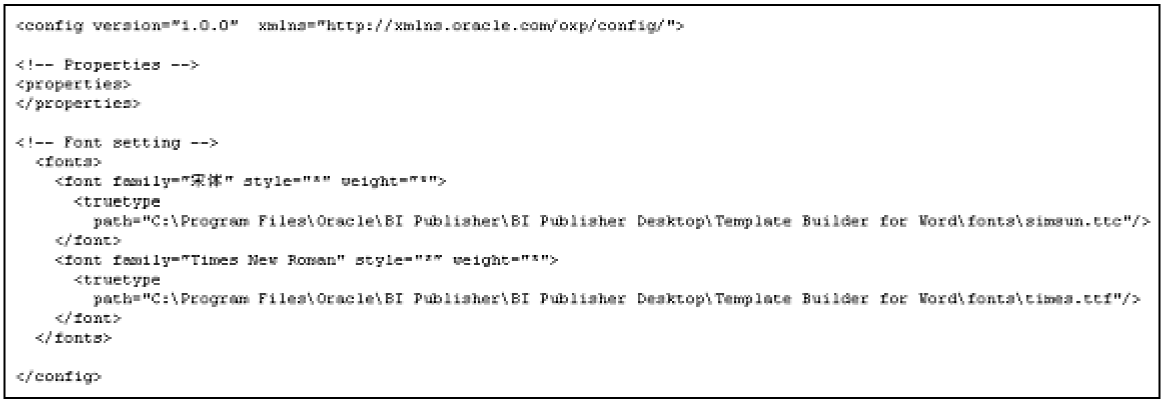
C:\Oracle\BI Publisher\BI Publisher Desktop\Template Builder for Word\config\

Note that if you have not used this file yet you may find the file "xdo example.cfg" instead. This file must be saved with an encoding of UTF-8 and provide a full and absolute path for each font defined. Otherwise, you will encounter issues such as characters overlays and wrapping that does not work.

To summarize if you are using a font that is not defined in xdo.cfg it might not display correctly in the generated Microsoft Office PowerPoint file. To correct this situation:

1. Add these font items into xdo.cfg, and specify the absolute path of the font files locations.
2. Save the xdo.cfg in UTF-8 format.
3. Make sure all the fonts used in the template file are defined correctly in xdo.cfg.

The following figure shows an example of a xdo.cfg file:



```
<config version="1.0.0"  xmlns="http://xmlns.oracle.com/oxp/config/">

<!-- Properties -->
<properties>
</properties>

<!-- Font setting -->
  <fonts>
    <font family="宋体" style="*" weight="*">
      <truetype
        path="C:\Program Files\Oracle\BI Publisher\BI Publisher Desktop\Template Builder for Word\fonts\simsun.ttc"/>
    </font>
    <font family="Times New Roman" style="*" weight="*">
      <truetype
        path="C:\Program Files\Oracle\BI Publisher\BI Publisher Desktop\Template Builder for Word\fonts\times.ttf"/>
    </font>
  </fonts>

</config>
```

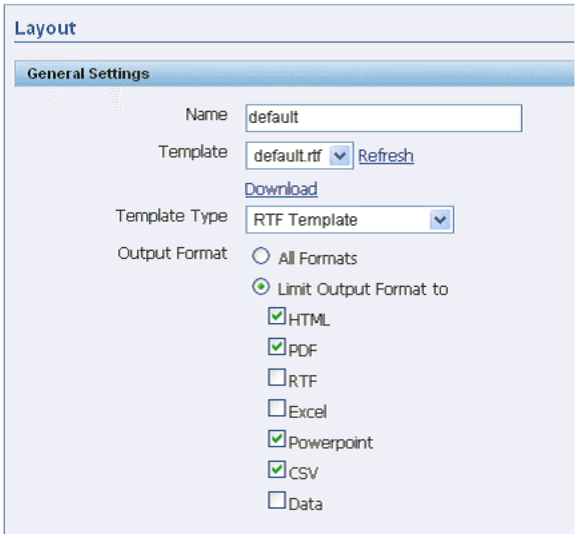
12.4 Usage and Limitations

PowerPoint Output is supported for RTF templates. Please see the *Oracle Business Intelligence Publisher User's Guide* section "Creating an RTF Template" for details on how to create an RTF template.

Here are some usage guidelines that will help you get the most out of your templates for PowerPoint:

- Page breaks will be treated as new slides.
- Background color of generated PowerPoint file is always white.
- Table border type must be single line (double border, dash, etc. are not supported).
- Hyperlinks are not supported.
- Charts and Graphs will be generated as PNG images (SVG is not supported).
- Shapes are not supported.
- Text position may be slightly incorrect if you use right-align.
- PowerPoint output will preserve the page orientation (portrait or landscape) defined in the RTF template. Most presentations are oriented in landscape so this is the recommended orientation of your RTF template.
- Paper size must be the same on all pages of the RTF template. You cannot have mixed paper sizes in the same document.
- All Unicode languages, except bi-directional languages, are supported.
- Text position may be slightly incorrect for Chinese, Japanese, and Korean fonts when using bold or italic effects on characters. This is because Microsoft has bold/italic emulation when there is no bold/italic font.

Once you have uploaded an RTF template to a report, you will have the option to determine if PowerPoint is a suitable output format for the template by selecting the **Limit Output Format to** radio button choice and checking or un-checking the box next to **Powerpoint**. See figure below.



Layout

General Settings

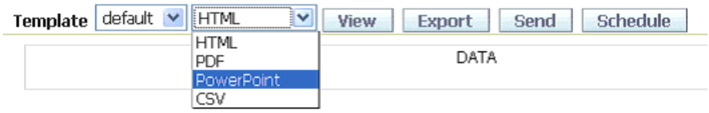
Name: default

Template: default.rtf

Template Type: RTF Template

Output Format:
○ All Formats
● Limit Output Format to
☑ HTML
☑ PDF
□ RTF
□ Excel
☑ Powerpoint
☑ CSV
□ Data

Users will then have the option to then set the output to **PowerPoint** when viewing the report.



Template: default

HTML

DATA

13

Oracle BI Publisher Integration with Oracle BI Discoverer

Oracle BI Publisher uses a modern report architecture that separates the report data model from the report layout. By doing so it allows users to create multiple layouts for a single data model. Oracle BI Publisher also supports many types of data sources such as SQL query against a database using JDBC, file data, HTTP feed, and web service.

Using the new Discoverer web service you will be able to create reports based on the data returned from Discoverer worksheets. Once you install the Discoverer web service and configure Oracle BI Publisher to use it, the web service will return data from selected worksheets that provide the data for Oracle BI Publisher reports.

Oracle BI Publisher 10.1.3.3 includes the ability to use Oracle BI Discoverer worksheets as the data set for a report. This provides 3 key benefits to Discoverer users. Discoverer users will now be able to:

- Create highly formatted reports
- Schedule execution and Delivery of reports to wide range of destinations
- Generate reports in a greater range of formats

This chapter describes how to:

- Configure BI Discoverer and Oracle BI Publisher to work with each other
- Create reports and layouts based on BI Discoverer worksheets

13.1 Configuring Oracle BI Publisher and BI Discoverer Integration

The process to configure Oracle BI Publisher and BI Discoverer Integration includes the following steps:

Step 1: Install BI Discoverer Web Service

Step 2: Install Oracle BI Publisher 10.1.3

Step 3: Configure BI Discoverer to Use Oracle Application Server Identity Management

Step 4: Set Up a Dedicated User to Access the BI Discoverer Web Service from Oracle BI Publisher

Step 5: Configure Oracle BI Publisher to Use the Discoverer Web Service

13.1.1 Step 1: Install BI Discoverer Web Service

Install Application Server Patch Set 10.1.2.3 patch or the Discoverer one off patch that includes the Discoverer Web Service. This will install and deploy the Discoverer Web Service (discWS.jar) into the same Oracle Containers for J2EE (OC4J) that hosts Discoverer services. Once installed, the Discoverer web service will be accessible from:

http://<1012AppServer>:<port>/discoverer/wsi

13.1.2 Step 2: Install Oracle BI Publisher 10.1.3

The 10.1.3.3 version of Oracle BI Publisher supports access to Discoverer worksheets as data sets in Oracle BI Publisher reports. Oracle BI Publisher and Discoverer can, and in most cases will, be on separate machines.

13.1.3 Step 3: Configure BI Discoverer to Use Oracle Application Server Identity Management

To configure BI Discoverer to use Oracle Application Server Identity Management see the *Oracle Business Intelligence Discoverer Configuration Guide*, Section 14.7 "Using Discoverer with Oracle Identity Management Infrastructure" available from the Documentation pages on the Oracle Technology Network (OTN) or directly at:

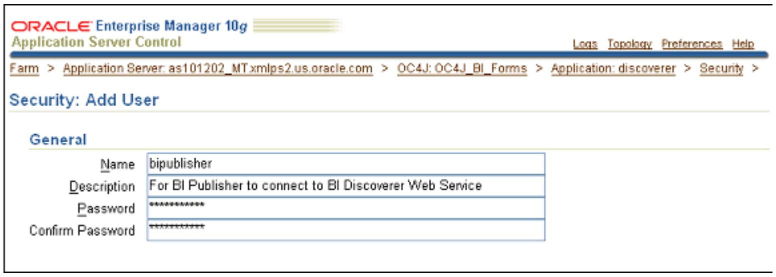
`http://download.oracle.com/docs/html/B13918_03/security2.htm#i1021521`

13.1.4 Step 4: Set Up a Dedicated User to Access the BI Discoverer Web Service from Oracle BI Publisher

The Discoverer web service is accessible only to trusted clients. The Administrator needs to setup the dedicated user that Oracle BI Publisher will use to access the web service.

To setup the user:

1. Login to an Application Server Control instance that has access to the BI Discoverer middle tier machine.
2. Select the BI Discoverer middle tier machine.
3. Select **OC4J_BI_FORMS**.
4. Select the **Applications**.
5. Select the deployed application named 'discoverer'.
6. Under the Administration section, choose Security.
7. Click on the **Add User** button.

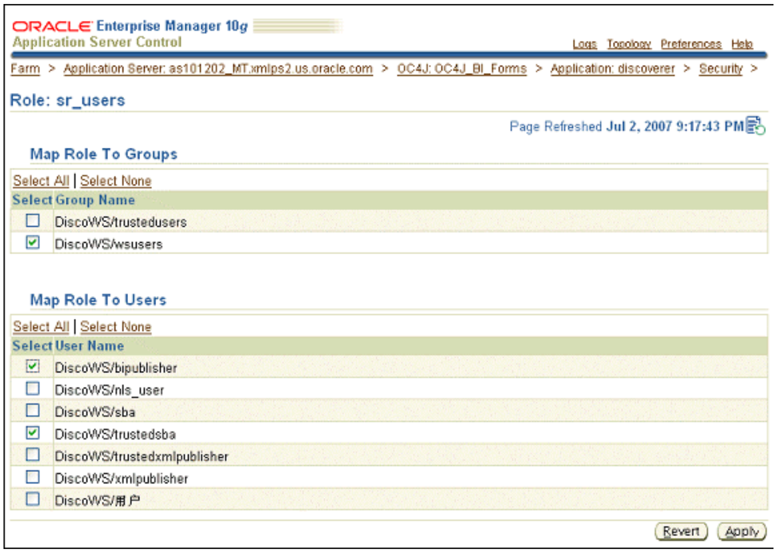

ORACLE Enterprise Manager 10g
Application Server Control
Logs
Topology
Preferences
Help
Farm > Application Server: as101202_MT.xmlps2.us.oracle.com > OC4J: OC4J_BI_Forms > Application: discoverer > Security >

Security: Add User

General

Name	bipublisher
Description	For BI Publisher to connect to BI Discoverer Web Service
Password	***********
Confirm Password	***********

8. Map this user to security role sr_users by selecting the sr_users Security Role and clicking on the "Map Role to Principals" button. On the Role: sr_users page, check the box next to the newly created user in the "Map Role To Users" section and click **Apply**.



Farm > Application Server: as101202_MT.xmlps2.us.oracle.com > OC4J: OC4J_BI_Forms > Application: discoverer > Security >

Role: sr_users

Page Refreshed Jul 2, 2007 9:17:43 PM

Map Role To Groups

Select All | Select None

Select	Group Name
□	DiscoWS/trustedusers
☑	DiscoWS/wsusers

Map Role To Users

Select All | Select None

Select	User Name
☑	DiscoWS/bipublisher
□	DiscoWS/nls_user
□	DiscoWS/sba
☑	DiscoWS/trustedsba
□	DiscoWS/trustedxmlpublisher
□	DiscoWS/xmlpublisher
□	DiscoWS/用户

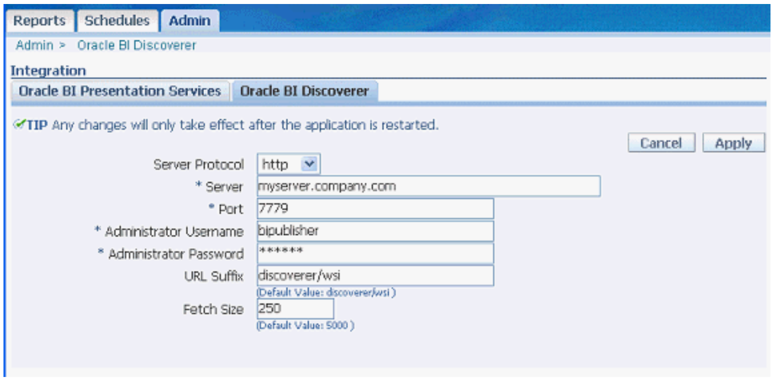
The role sr_users is pre-defined in the BI Discoverer web.xml.

For more detailed information, refer to "Security Role Mapping" in the *Oracle Application Server Containers for J2EE Security Guide* 10g Release 2 (10.1.2) B14013-02 available from the Documentation pages on the Oracle Technology Network (OTN) or directly at:

http://download.oracle.com/docs/cd/B14099_19/web.1012/b14013/jaas_integrate.htm#sthref270

13.1.5 Step 5: Configure Oracle BI Publisher to Use the Discoverer Web Service

Configure the Discoverer web service in Oracle BI Publisher Admin pages using the Discoverer web service URL from Step 1: Install BI Discoverer Web Service (http://<1012AppServer>:<port>/discoverer/wsi) and the dedicated user defined in Step 4: Set Up a Dedicated User to Access the BI Discoverer Web Service from Oracle BI Publisher.



TIP Any changes will only take effect after the application is restarted.

Server Protocol	http
* Server	myserver.company.com
* Port	7779
* Administrator Username	bipublisher
* Administrator Password	******
URL Suffix	discoverer/wsi (Default Value: discoverer/wsi)
Fetch Size	250 (Default Value: 5000)

Note that in this release, any Oracle BI Publisher server or cluster of servers can be configured with only one Discoverer instance.

Restart the application server that is running Oracle BI Publisher for these changes to take effect.

13.1.6 BI Discover Web Service Configuration Settings

The following configuration parameters that control the behavior of the Discoverer web service can be set in the Discoverer configuration.xml file:

- **Loglevel** — Can be set to any one of following: WARNING, DEBUG, FATAL, ERROR, TRACE, NOTIFICATION.
- **MaxSessionPoolSize** — Determines the maximum number of Discoverer Sessions that can be active and available at the same time.

These should only be changed under the direction of Oracle Support.

13.2 Creating Reports Based on Discoverer Worksheets

BI Publisher's modern report architecture provides benefits in report development and maintenance.

With this architecture it is possible to have IT staff focus on extracting exactly the needed data with minimal impact on transactional systems. Then end users or IT staff can easily create as many layouts as needed for each query or data model. Layout development can even be done offline to further insulate transactional systems. Likewise it is a very straightforward process to create translations as needed for each layout.

13.2.1 About Creating a Report Based on a Discoverer Worksheet

In Oracle BI Publisher each report has a data model that can be made up of one or more data sets. Each data set is based on a data source defined in the Admin page.

For integration with BI Discoverer, the data source is configured through the Oracle BI Discoverer tab on the Integration page from the Oracle BI Publisher Admin page (see Section 13.1.5, "Step 5: Configure Oracle BI Publisher to Use the Discoverer Web Service"). When you want to create a report that uses data from BI Discoverer you will identify a BI Discoverer connection and select a worksheet that will define the data set. **Note**: A data model can only include only one data set based on a Discoverer Worksheet.

There are two ways to create an Oracle BI Publisher report that uses a Discoverer Worksheet as a data set. One way is to create the report directly through the Oracle BI Publisher web interface. The other way is to indirectly create a report and data set while creating a layout template using the Oracle BI Publisher Template Builder for Word.

Creating a report directly through the Oracle BI Publisher web interface gives you control over the naming of the report and affords you the option of creating sample data for layout development. This is helpful when the Discoverer Worksheet returns a lot of data or has parameters with default values that return no data. In general, creating a report directly through the web interface is the preferred method of creating the report and data model.

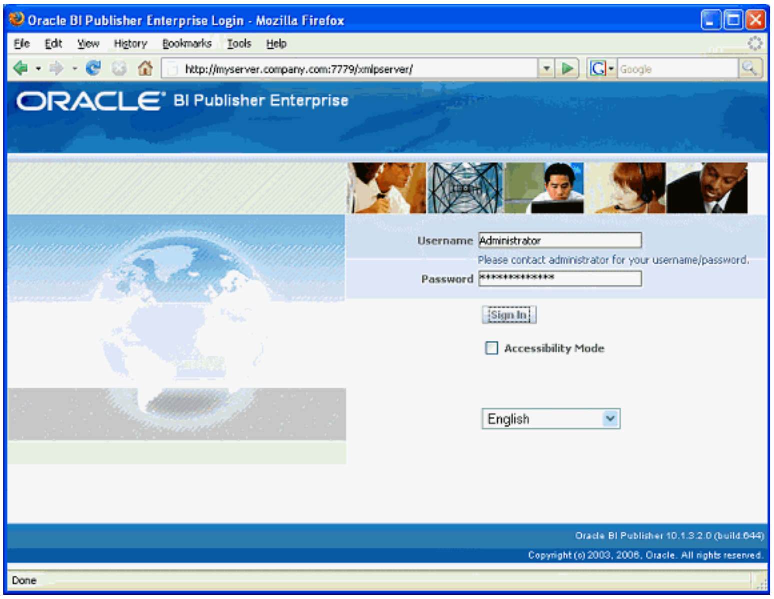
You may also create a report indirectly through the Oracle BI Publisher Template Builder add-in to Word. This method will create a report with a name that is of the form workbook.worksheet in the directory of your choosing in the Oracle BI Publisher report repository. This method does not allow you to change or set default parameter values or to limit the amount of data that is returned from the Discoverer worksheet.

This method is recommended only if the Discoverer worksheet returns a relatively small amount of data and does not have default parameter values that return no data. If your worksheet returns too much data, it will overwhelm the Template Builder -- see the known issues section below. Conversely, if your worksheet has parameters with default values such that no data is returned, no sample data will be available to build the layout template.

13.2.2 Creating a Report Directly Through the Oracle BI Publisher Web Interface

To create a report and data model based on a Discoverer worksheet through the web interface to the Oracle BI Publisher server:

1. Connect to Oracle BI Publisher through your browser (for example, myserver.company.com:7779/xmlpserver)


Oracle BI Publisher Enterprise Login - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
Google
ORACLE BI Publisher Enterprise
Sign In
Done

http://myserver.company.com:7779/xmlpserver/

Username Administrator

Please contact administrator for your username/password.

Password **************

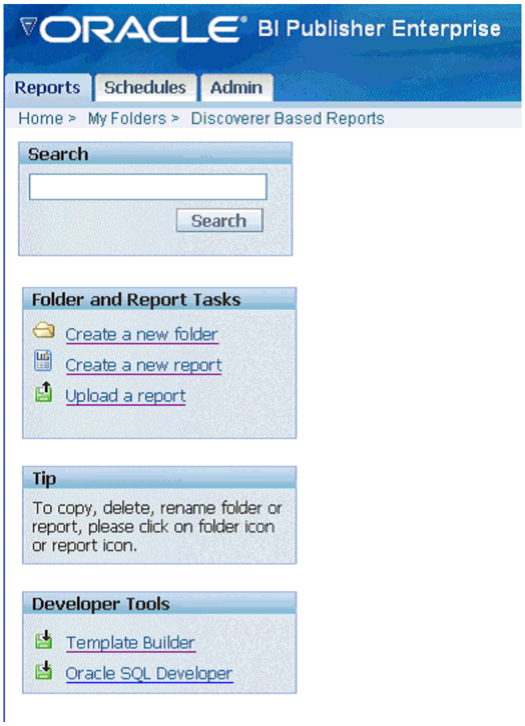
☐ Accessibility Mode

English

Oracle BI Publisher 10.1.3.2.0 (build.644)

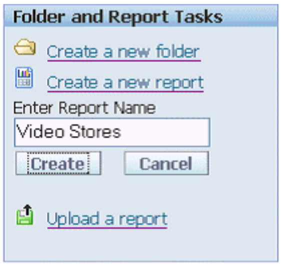
Copyright (c) 2003, 2006, Oracle. All rights reserved.

2. Navigate to or make a directory to hold the report you want to create.



To copy, delete, rename folder or report, please click on folder icon or report icon.

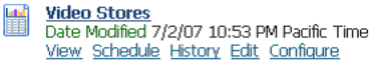
3. Click **Create a new report** in the Folder and Report Tasks frame on the left side of your browser page. Give it a name and click **Create**.



Enter Report Name: Video Stores

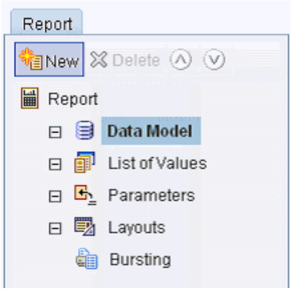
The report will appear on the right hand side of the screen.

4. Click the **Edit** link the under the report name.

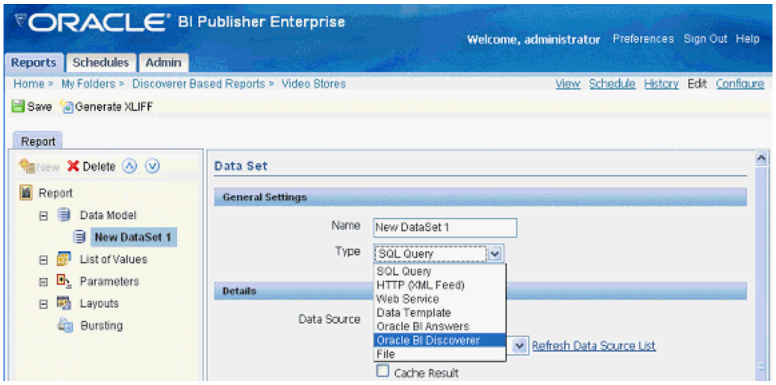


Date Modified 7/2/07 10:53 PM Pacific Time

5. Select **Data Model** from the Report tree and click the **New** button at the top of the tree to create a new Data Set.



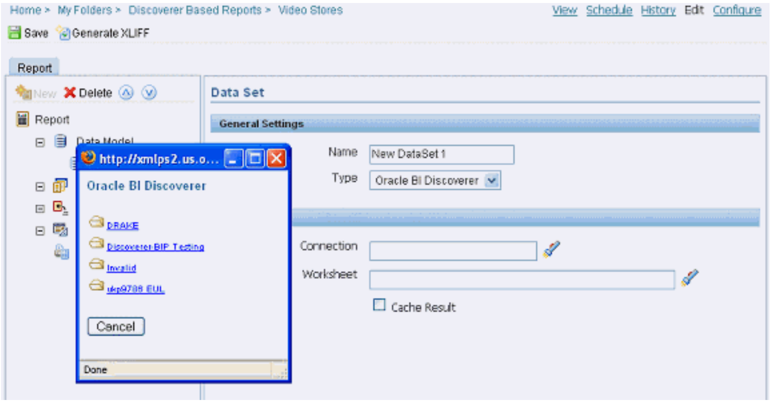
6. Under the General Settings section of the Data Set, optionally name the data set and select **Oracle BI Discoverer** in the **Type** field.



Name: New DataSet 1
Type: SQL Query

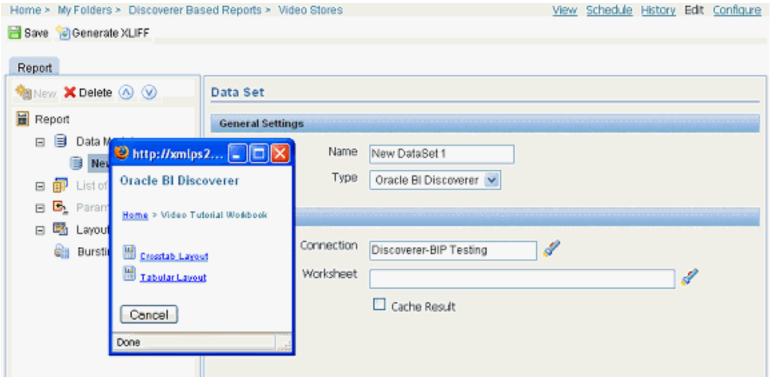
- SQL Query
- HTTP (XML Feed)
- Web Service
- Data Template
- Oracle BI Answers
- Oracle BI Discoverer
- File

7. Under the Details section of the Data Set, click on the flashlight to the right of the **Connection** field. Navigate to and select the Discoverer connection that owns the worksheet you want to use in your report.



Name: New DataSet 1
Type: Oracle BI Discoverer

8. Once you have selected a connection, click on the flashlight to the right of the Worksheet field. Navigate to and select the workbook and then the worksheet you want to base your report on.

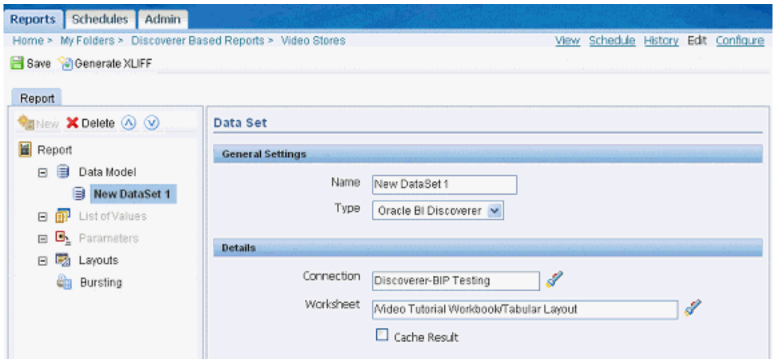


Name: New DataSet 1
Type: Oracle BI Discoverer
Connection: Discoverer-BIP Testing

Note that the fully qualified path to the worksheet is entered in the **Worksheet** field.

Note that regardless of whether the Discoverer worksheet is a tabular or crosstab layout, the Discoverer Web Service will return the data to Oracle BI Publisher as flat tabular data. This is by design so that you can take full advantage of the layout capabilities in BI Publisher. You can layout this data as a crosstab in BI Publisher.

9. Save your report by clicking the **Save** icon on the upper left side of the page below the navigation links to the report.



Name: New DataSet 1

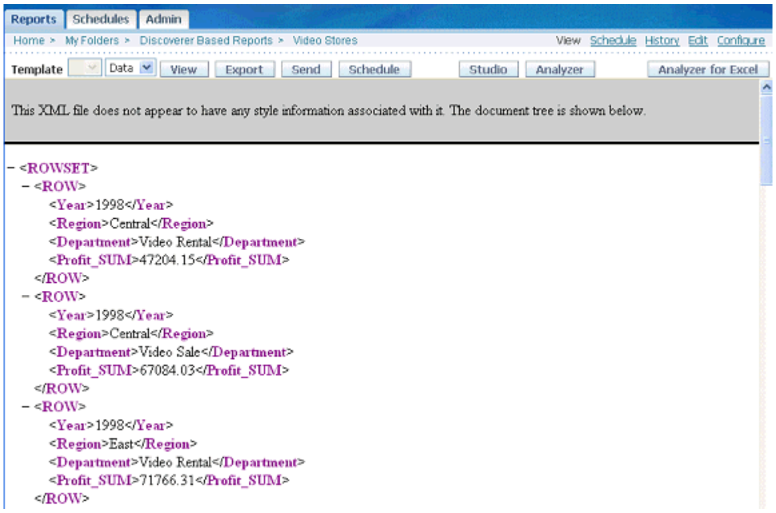
Type: Oracle BI Discoverer

Connection: Discoverer-BIP Testing

Worksheet: /Video Tutorial Workbook/Tabular Layout

☐ Cache Result

10. Once the report is saved, click the **View** link in the upper right corner of the screen to the right of the navigation links to see that the Discoverer Worksheet you selected returns data to the report. If your worksheet contains parameters, you may need to select values other than the default and click **View** to get the worksheet to return data.


Reports
Schedules
Admin
Home > My Folders > Discoverer Based Reports > Video Stores
View Schedule History Edit Configure
Template
Data
View
Export
Send
Schedule
Studio
Analyzer
Analyzer for Excel

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <ROWSET>
  - <ROW>
      <Year>1998</Year>
      <Region>Central</Region>
      <Department>Video Rental</Department>
      <Profit_SUM>47204.15</Profit_SUM>
    </ROW>
  - <ROW>
      <Year>1998</Year>
      <Region>Central</Region>
      <Department>Video Sale</Department>
      <Profit_SUM>67084.03</Profit_SUM>
    </ROW>
  - <ROW>
      <Year>1998</Year>
      <Region>East</Region>
      <Department>Video Rental</Department>
      <Profit_SUM>71766.31</Profit_SUM>
    </ROW>
```

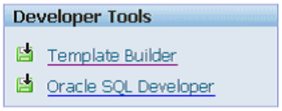
You have successfully created a report with a data model based on a Discoverer Worksheet.

At this point you can continue by creating a layout template by using either the online method or offline method. Continue with the section entitled Section 13.2.3.1, "About Creating Layout Templates".

13.2.3 Creating a Report Indirectly Through Oracle BI Publisher Template Builder

You can also create a report using Oracle BI Publisher Template Builder, an Add-in to Microsoft Office Word.

If you have not installed Oracle BI Publisher Template Builder, download it by clicking the **Template Builder** link in the Developer Tools section on the left side of any page, visible in the Oracle BI Publisher web interface when viewing any folder below Home in the report repository.



Developer Tools

Template Builder

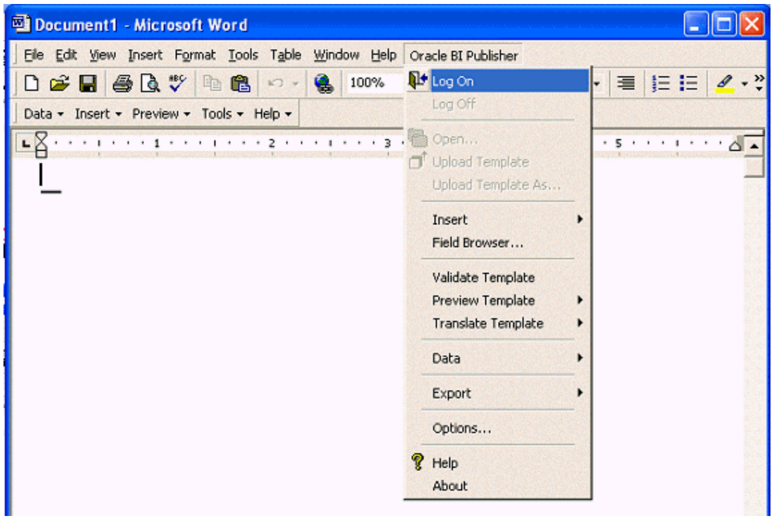
Oracle SQL Developer

Then run the BIPublisherDesktop.exe that is downloaded. This will install Oracle BI Publisher Template Builder.

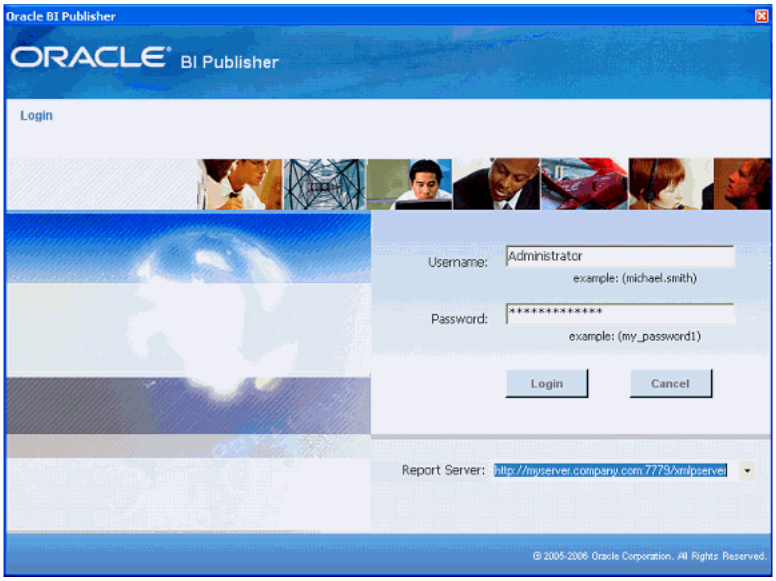
Note that you should exit all Microsoft Office products before installing and that you may also be prompted to download and install Microsoft .Net 2.0 if it is not already installed.

After the installation is finished, continue with the following steps.

1. Start Microsoft Office Word. Connect to the Oracle BI Publisher server by selecting **Log On** from the Oracle BI Publisher menu.



2. In the Oracle BI Publisher Login dialog, supply your Oracle BI Publisher username and password. Note that, if this is the first time you are logging in to the Oracle BI Publisher server, you will need to supply the report server URL (for example, myserver.company.com:7779/xmlpserver).



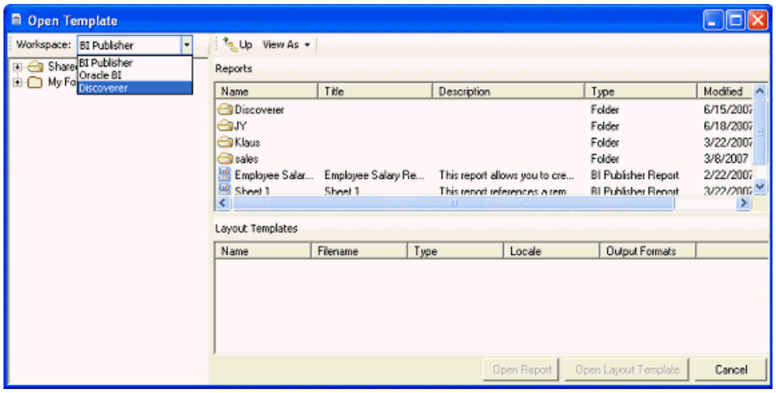
Username: Administrator
example: (michael.smith)

Password: *************
example: (my_password1)

Report Server: http://myserver.company.com:7779/xmlpserver

After the first log on, you may select the Report Server from the drop down list.

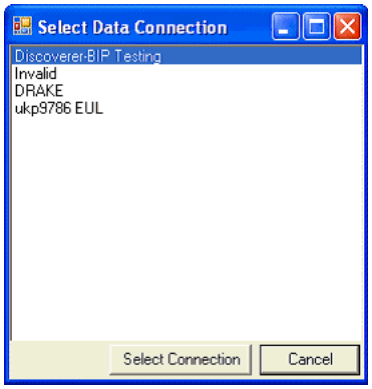
3. Once connected you will see the Open Template dialog. Select **BI Discoverer** from the Workspace pull down.



Name	Title	Description	Type	Modified
Discoverer			Folder	6/15/2007
JY			Folder	6/18/2007
Klaus			Folder	3/22/2007
sales			Folder	3/8/2007
Employee Salar...	Employee Salary Re...	This report allows you to cre...	BI Publisher Report	2/22/2007
Sheet 1	Sheet 1	This report references a rem	BI Publisher Report	3/22/2007

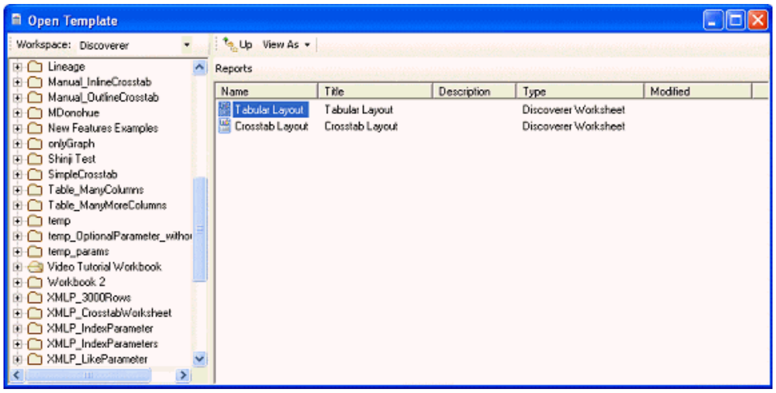
Name	Filename	Type	Locale	Output Formats

4. You will be prompted to choose a Discoverer connection



Discoverer-BIP Testing
Invalid
DRAKE
ukp9786 EUL

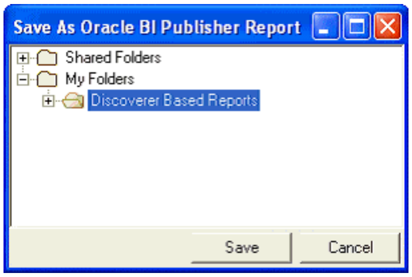
5. Once you have selected a Discoverer connection, the Open Template dialog will show a list of workbooks on the left. As you select workbooks, the worksheets will show in the right pane. Navigate the workbooks and select a worksheet by double clicking on the worksheet name.



Lineage
Manual_InlineCrosstab
Manual_OutlineCrosstab
MDonohue
New Features Examples
onlyGraph
Shinji Test
SimpleCrosstab
Table_ManyColumns
Table_ManyMoreColumns
temp
temp_OptionalParameter_withou
temp_params
Video Tutorial Workbook
Workbook 2
XMLP_3000Rows
XMLP_CrosstabWorksheet
XMLP_IndexParameter
XMLP_IndexParameters
XMLP_LikeParameter

Name	Title	Description	Type	Modified
Tabular Layout	Tabular Layout		Discoverer Worksheet	
Crosstab Layout	Crosstab Layout		Discoverer Worksheet	

6. You will be prompted to save the worksheet as an Oracle BI Publisher report. You will be able to navigate the Oracle BI Publisher report repository and choose the folder where the report will be created.



A report will be created that has a name of the form workbook.worksheet and it will use that worksheet as the data set. You cannot name or rename the report from this dialog or from the Template Builder. If a report already exists with that name, a new report will not be created and the existing report will remain as it is. If you want to rename the report, use the web interface to the Oracle BI Publisher report server.

At this point you can proceed with creating a layout template by following the online method described below. Continue with the section entitled Section 13.2.3.1, "About Creating Layout Templates".

13.2.3.1 About Creating Layout Templates

Oracle BI Publisher provides several types of layout templates. The most common are RTF layout templates created using the Template Builder. You can also create PDF templates in Adobe Acrobat, eText templates, and Adobe Flex templates. This document briefly describes only the process of creating an RTF layout template. See the *Oracle Business Intelligence Publisher User's Guide* for more details on creating other types of templates and for more depth on RTF templates.

Note that even though a layout template created with the Template Builder is in RTF format, at runtime the layout template is converted to Extensible Stylesheet Language Formatting Objects (XSL-FO) and the report output format can be PDF, HTML, Excel, PowerPoint, as well as RTF.

There are two methods you can use to create an RTF layout template for a report. One way is Offline - meaning not connected to the Oracle BI Publisher report server. The other is Online - meaning connected to the Oracle BI Publisher report server.

Using the offline method, you will generate sample XML data for your report. This method has the advantage that you will be placing no load on the report server or underlying data source while creating your layout. It is also the preferred method if your report returns a large amount of data or has parameters that need to be set to limit the amount of data returned or to ensure that some data is returned.

The online method allows you to connect directly to the report server to download report data. This method is suitable if your report has been set up to provide a reasonable amount of data. This method also can be seamlessly integrated with the indirect method of creating a Discoverer report as described above.

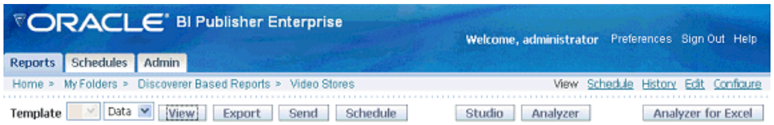
13.2.3.2 Creating a Layout Template Offline

1. Generate sample XML data.

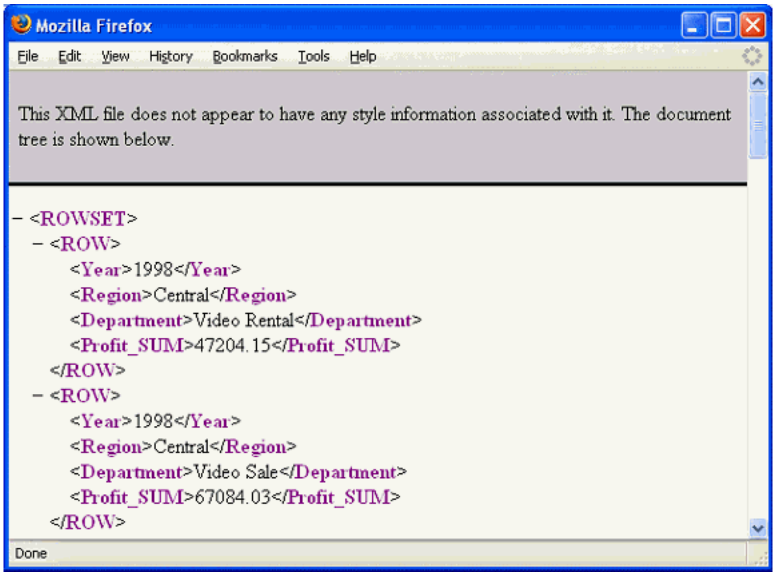
   From the Oracle BI Publisher web interface navigate to the report you want to create a layout against and select View to see the report output.

   If there are parameters that can be set to either reduce the amount of data or to ensure that some data is returned, set the parameter values and click the **View** button.

Then select Data from the pull down menu to the left of the **View** button and click the **Export** button.



A new browser or tab will open containing the exported XML data.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

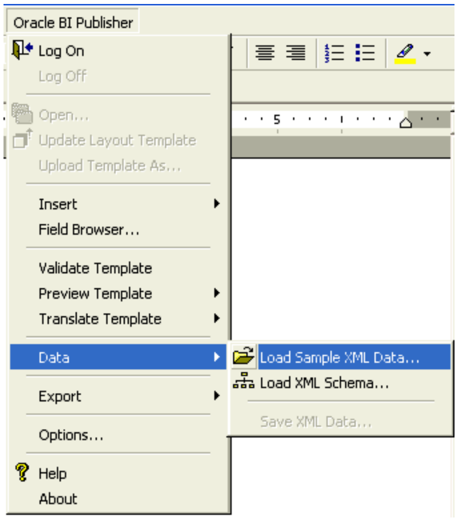
```
- <ROWSET>
  - <ROW>
      <Year>1998</Year>
      <Region>Central</Region>
      <Department>Video Rental</Department>
      <Profit_SUM>47204.15</Profit_SUM>
    </ROW>
  - <ROW>
      <Year>1998</Year>
      <Region>Central</Region>
      <Department>Video Sale</Department>
      <Profit_SUM>67084.03</Profit_SUM>
    </ROW>
```

Using the web browser File menu, save the file with an XML extension.

If the size of the XML file is still excessive, you will want open it in a text editor and remove data from the file to get it to a smaller size. Be sure to preserve the tag structure in the file. You will also want to make sure you have enough data so that previewing your report will show how the actual report will be generated. For example, if you plan to group data by Year, be sure to include some data from at least 2 years so that you can confirm that the grouping displays as you want.

2. Upload the sample XML data file to the Template Builder.

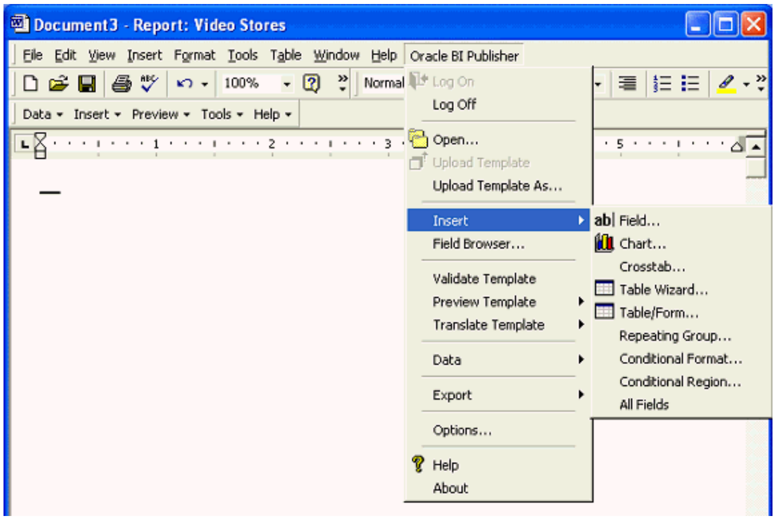
   After you generate the sample XML data file move it to the machine where you are running the Template Builder. From the Oracle BI Publisher menu or the Template Builder toolbar, load the sample XML data.



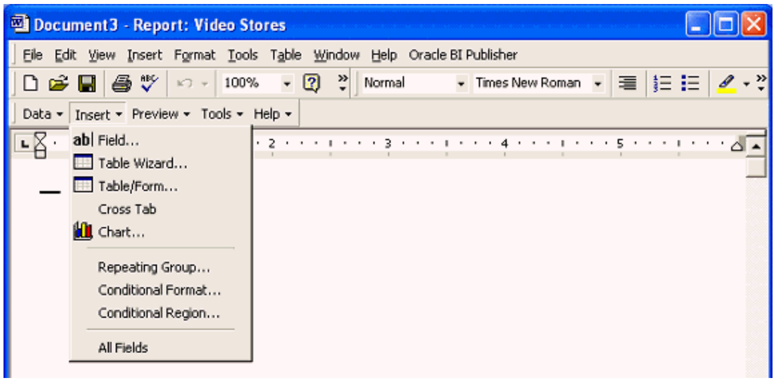
3. Create a layout template.

   Now that you have uploaded the sample XML data to the Template Builder, you can use the Template Builder Insert menu and toolbar to launch wizards and dialogs that help you create your layout template. You can also insert whatever boilerplate text you want as well as format fields and tables using standard Microsoft Office Word layout and formatting functionality.

   The following figure shows the Template Builder Insert Menu:



   The following figure shows the Template Builder Insert Toolbar:



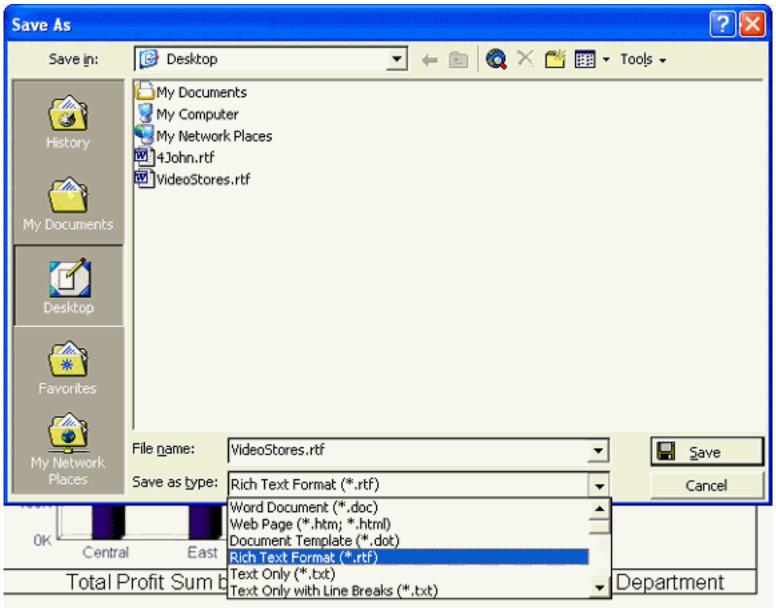
You can also manually include data field tags and other markup to your template using BI Publisher's simplified tags for XSL expressions. These tags associate the XML report data to your report layout. If you are familiar with XSL and prefer not to use the simplified tags, Oracle BI Publisher also supports the use of pure XSL elements in the template.

Please see the section "Creating an RTF Template" in the *Oracle Business Intelligence Publisher User's Guide* for more details, available from the Documentation pages on the Oracle Technology Network (OTN) or directly at:

`http://download.oracle.com/docs/cd/E10091_01/doc/bip.1013/b40017/T421739T481157.htm`

4. Save the layout template as an RTF file.

   From the File menu, select **Save As**. Be sure to set the type to Rich Text Format (*.rtf).



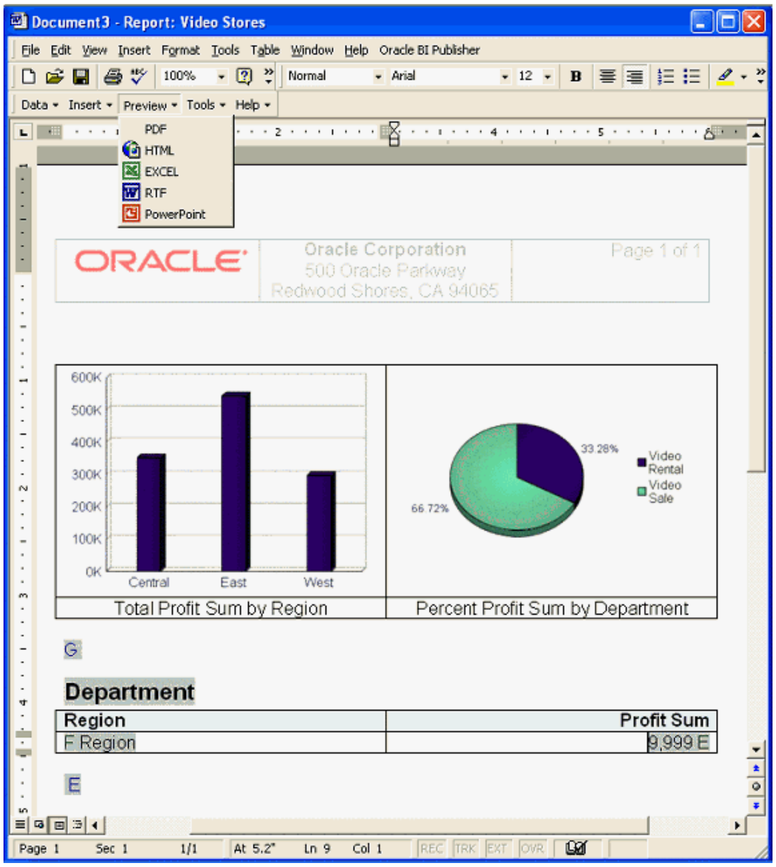
File name: VideoStores.rtf

Save as type: Rich Text Format (*.rtf)

5. Preview the report output.

   After you have created and formatted your layout, you can preview it in the desired output format from the Template Builder to confirm that you have created the layout exactly as you want. Previewing will generate the output document, merging the layout template you just created with the data that is loaded in the Template Builder.

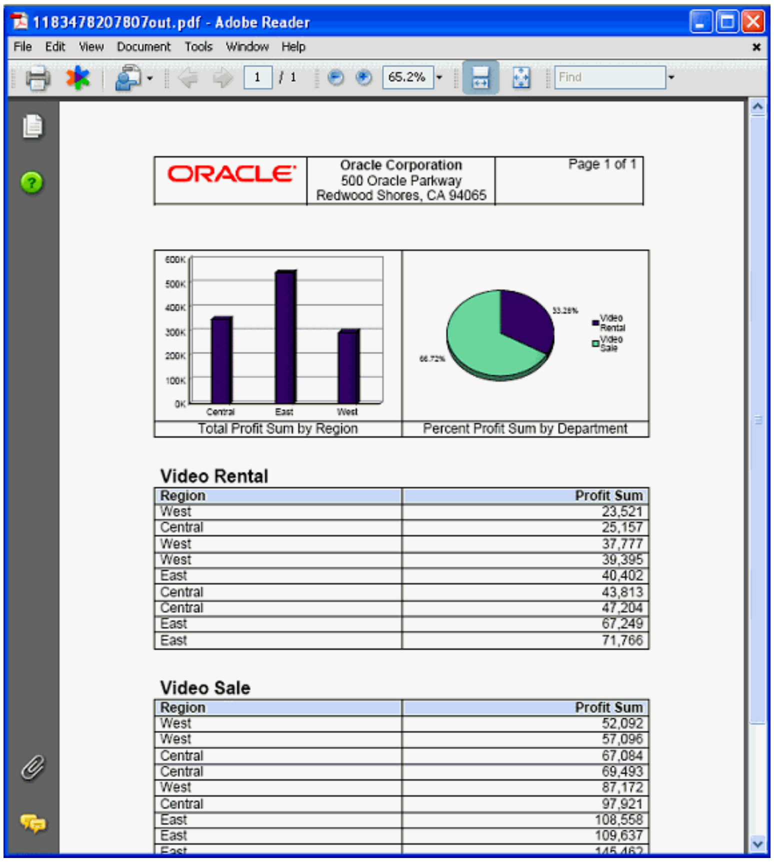
Select the desired output format from the Preview menu item on the Template Builder toolbar or from the Oracle BI Publisher menu.


Document3 - Report: Video Stores
File Edit View Insert Format Tools Table Window Help Oracle BI Publisher
100%
Normal
Arial
12
B
Data Insert Preview Tools Help
PDF
HTML
EXCEL
RTF
PowerPoint
ORACLE
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
Page 1 of 1
600K
500K
400K
300K
200K
100K
0K
Central
East
West
Total Profit Sum by Region
33.28%
66.72%
Video Rental
Video Sale
Percent Profit Sum by Department
G
E
Page 1 Sec 1 1/1 At 5.2" Ln 9 Col 1 REC TRK EXT OVR

Department

Region	Profit Sum
F Region	9,999 E

You will see the generated document in the application suitable for displaying the requested output format. Here is an example previewing PDF output in Adobe Reader:



	Oracle Corporation 500 Oracle Parkway Redwood Shores, CA 94065	Page 1 of 1

Total Profit Sum by Region

Percent Profit Sum by Department

Video Rental

Region	Profit Sum
West	23,521
Central	25,157
West	37,777
West	39,395
East	40,402
Central	43,813
Central	47,204
East	67,249
East	71,766

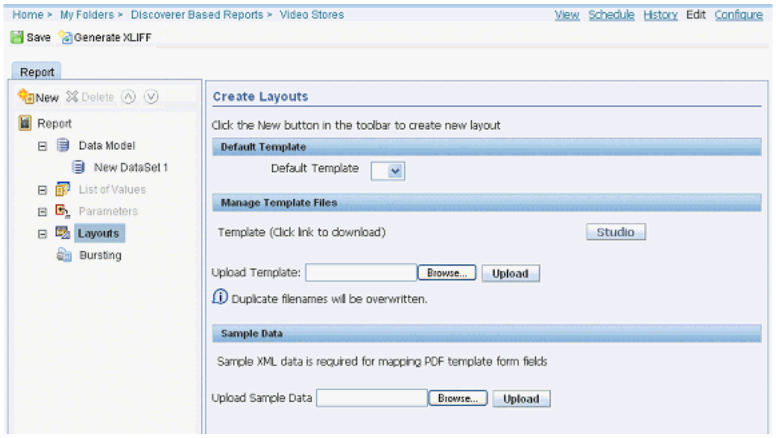
Video Sale

Region	Profit Sum
West	52,092
West	57,096
Central	67,084
Central	69,493
West	87,172
Central	97,921
East	108,558
East	109,637
East	145,462

6. Upload the RTF template to the report.

   Once you have made the changes you want to your RTF template, save it to the file system. Return to the Oracle BI Publisher web interface and navigate to the report associated with this layout template.

   From within the report edit page of the Oracle BI Publisher web interface, select **Layouts** from the Report tree. In the Manage Template Files section of the Create Layouts frame, click the **Browse** button to select the RTF template. Once the RTF template is selected and the **Upload Template** field is populated with the full path to the RTF template, click the **Upload** button.



Click the New button in the toolbar to create new layout

Default Template

Template (Click link to download)

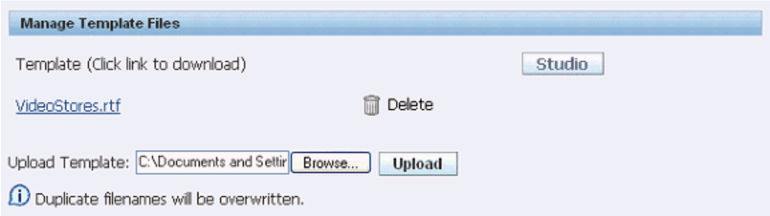
Upload Template:

Duplicate filenames will be overwritten.

Sample XML data is required for mapping PDF template form fields

Upload Sample Data

You will then see the uploaded template in the Manage Template Files section of the page.

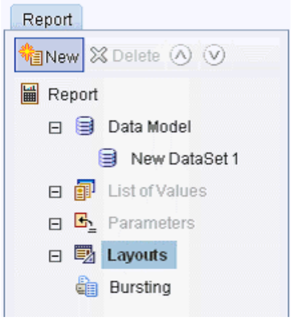


Upload Template: C:\Documents and Settir

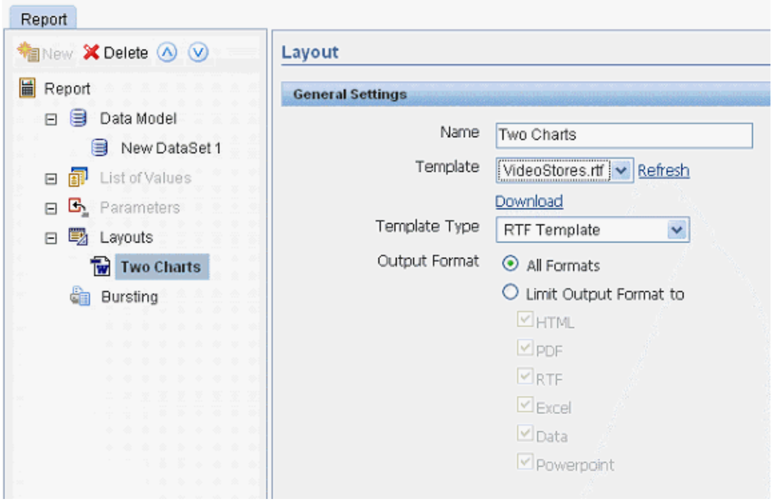
Duplicate filenames will be overwritten.

7. Create a named layout and associate it with the template file.

   From the report tree, with Layouts still selected, click the **New** button at the top of the report tree. This will create a new layout.



   Provide a name for the layout and associate it with the template file you uploaded. You can also optionally determine what output formats are available for this layout.



Name: Two Charts

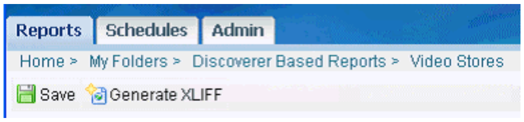
Template: VideoStores.rtf

Template Type: RTF Template

Output Format: ● All Formats ○ Limit Output Format to ☑ HTML ☑ PDF ☑ RTF ☑ Excel ☑ Data ☑ Powerpoint

8. Save the report.

   Be sure to save the changes you have made to the report by clicking the **Save** button below the path to the report.



9. Continue with Section 13.2.3.4, "Viewing and Scheduling Reports".

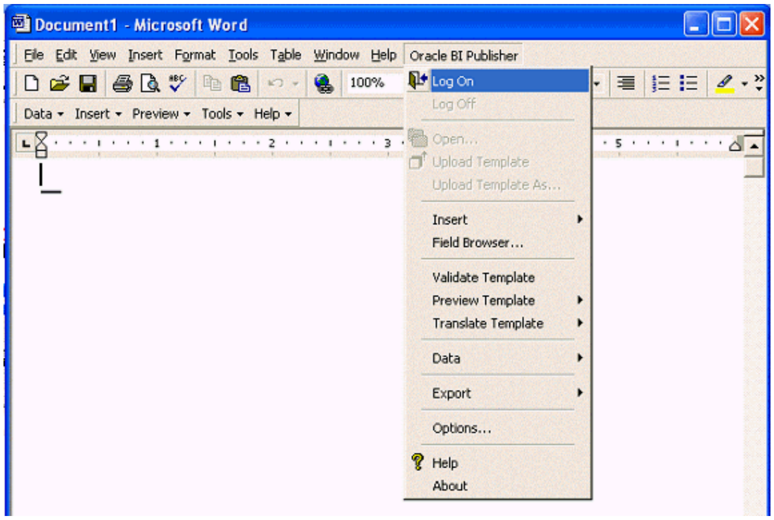
13.2.3.3 Creating a Layout Template Online

1. Connect to the Oracle BI Publisher report server through the Template Builder.

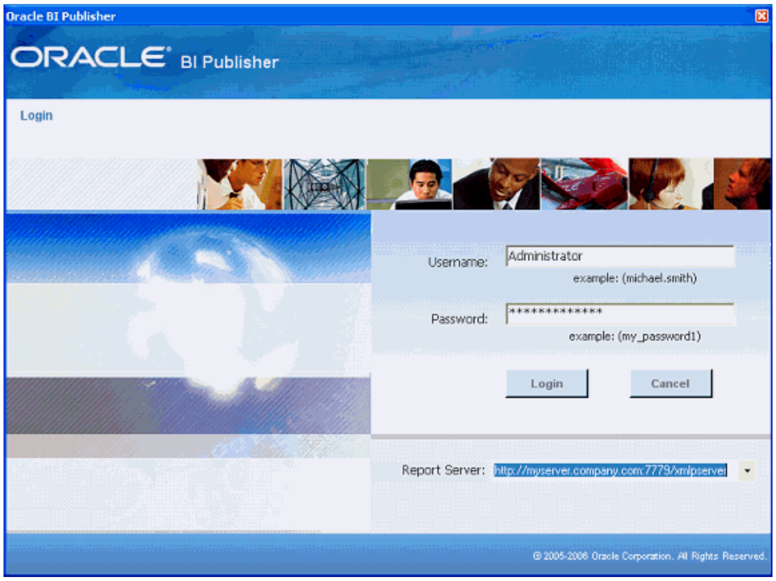
> **Note:** If you have followed the method above to create a report indirectly through the Oracle BI Publisher Template Builder, you will already be logged on to the Oracle BI Publisher Server and the report will already be open. Skip ahead to Step 3 Create a layout template.

If you have not already logged on to the Oracle BI Publisher server, do so following these steps:

a. Start Microsoft Office Word. Connect to the Oracle BI Publisher server by selecting **Log On** from the Oracle BI Publisher menu.



b. In the Oracle BI Publisher Login dialog supply your Oracle BI Publisher username and password. Note that if this is the first time you are logging in to the Oracle BI Publisher server, you will need to supply the report server URL (for example, myserver.company.com:7779/xmlpserver).



Username: Administrator
example: (michael.smith)

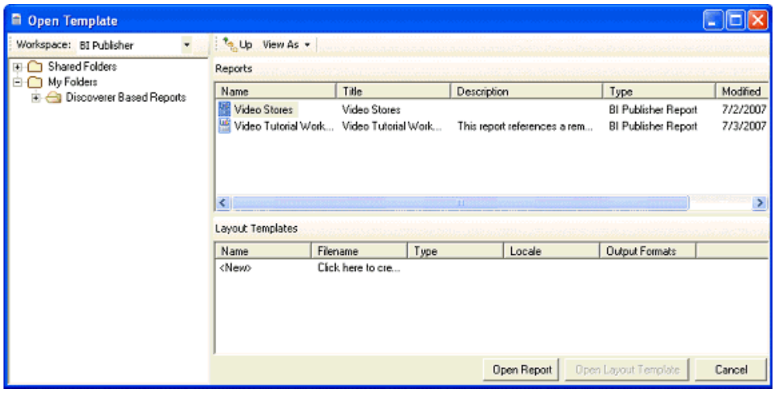
Password: *************
example: (my_password1)

Report Server: http://myserver.company.com:7779/xmlpserver

2. Create or open the desired report.

   The method for creating a report indirectly through the Template Builder is described above in the section entitled Section 13.2.3, "Creating a Report Indirectly Through Oracle BI Publisher Template Builder".

   To open a report that is already created, in the Open Template dialog, select BI Publisher as the Workspace. Navigate the folder tree until the report you want appears in the list of Reports.



Reports

Name	Title	Description	Type	Modified
Video Stores	Video Stores		BI Publisher Report	7/2/2007
Video Tutorial Work...	Video Tutorial Work...	This report references a rem...	BI Publisher Report	7/3/2007

Layout Templates

Name	Filename	Type	Locale	Output Formats
<New>	Click here to cre...			

   Select the report name and click **Open Report**. Alternatively, select the report and then select <New> or any one of the existing layout templates listed in the Layout Templates section and click **Open Layout Template**.

   Once you have opened an existing report or layout template or created a new report or new layout template, the data model definition and fields and data will be loaded into the Template Builder and you can start to create your layout template.

3. Create a layout template.

   Now that you have opened or created a report based on a Discoverer worksheet, you can use the Template Builder Insert menu and toolbar to build your layout template.

   See Step 3 in Section 13.2.3.2, "Creating a Layout Template Offline" above for guidance on creating a layout template.

4. Save the layout template as an RTF file.

   From the File menu, select **Save As**. Be sure to set the type to Rich Text Format (*.rtf).

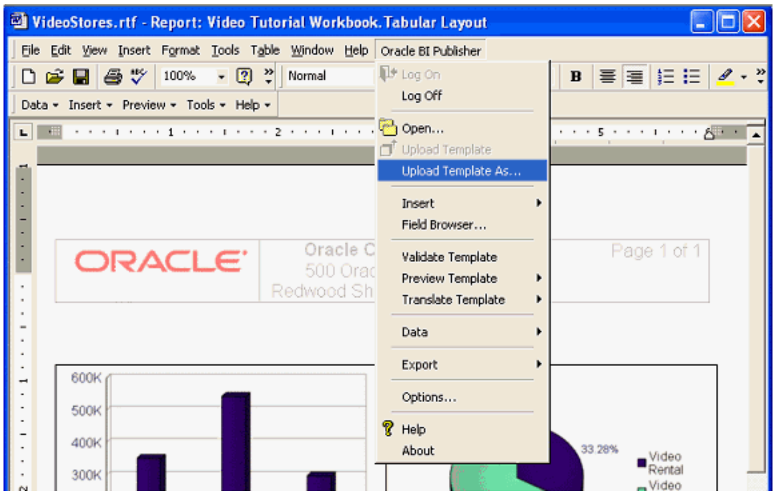
   See Step 4 in Section 13.2.3.2, "Creating a Layout Template Offline".

5. Preview the report output.

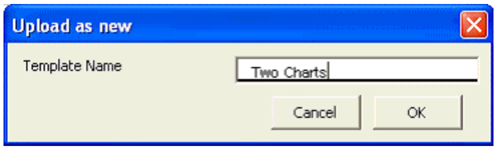
   See Step 5 in Section 13.2.3.2, "Creating a Layout Template Offline".

6. Upload the RTF layout template to the Oracle BI Publisher server.

   Once you have completed the layout template, upload it to the BI server. Select **Upload Template As** from the Oracle BI Publisher menu.

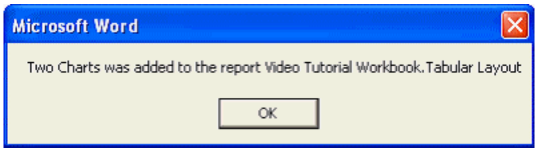


If this is a new template you will be prompted for a template name. Enter the template name and click **OK**.


Upload as new

Template Name: Two Charts

Once the template is uploaded you will see a notification dialog. Click **OK**.


Microsoft Word

Two Charts was added to the report Video Tutorial Workbook.Tabular Layout

The template has been added to the report.

7. Continue with Section 13.2.3.4, "Viewing and Scheduling Reports".

13.2.3.4 Viewing and Scheduling Reports

Please see "Viewing and Scheduling Reports" in the *Oracle BI Publisher User's Guide* available from the Documentation pages on the Oracle Technology Network (OTN) or directly at:

http://download.oracle.com/docs/cd/E10091_01/doc/bip.1013/b40017/T421739T421742.htm

13.2.4 Prerequisites, Limitations, and Known Issues

This section describes the prerequisites, limitation, and known issues in this release.

13.2.4.1 Prerequisites

The following are prerequisites for this release:

- Oracle BI Publisher 10.1.3.3
- Oracle BI Discoverer 10.1.2.3 or BI Discoverer 10.1.2.2 with a BI Discoverer one off patch that includes the BI Discoverer Web Service

- BI Discoverer must be configured to use Oracle Application Server Identity Management

13.2.4.2 Limitations

The following are limitations with this release:

- Only one BI Discoverer Web Service may be configured per Oracle BI Publisher server or cluster.
- An Oracle BI Publisher report may only contain one data set based on a BI Discoverer worksheet.
- Discoverer Web Service does not support Oracle Applications Single Sign On (SSO), Oracle Applications ICX and non-SSO modes.

13.2.4.3 Known Issues

The following are known issues in this release:

Issue: Unable to enter parameter values directly for parameters that have associated list of values.

Description: For Discoverer worksheets that have parameter based on a List of Values (LOV) or an underlying item has an associated LOV, Publisher will only allow parameter value selection from the LOV. Users will not be able to directly enter parameter values as they can in Discoverer.

Workaround: If users wish to allow users to enter parameter values directly, they need to remove the associated LOV in the Discoverer worksheet or item.

Issue: Oracle BI Publisher not able to communicate with Discoverer Web Service when Discoverer is configured with Secure Sockets Layer (SSL) (i.e. uses https://)

Description: Not able to make any calls to Discoverer Web Service. May get errors like "SSL received a record that exceeded the maximum permissible length."

Workaround: When using a web service that is protected via SSL you may need to export the certificate from the web server hosting the web service and import it into the java keystore on the machine running BI Publisher. This step should not be required if the server certificate is linked to some certificate authority (like Verisign) -- the default java keystore can handle that. But if the web service server is using some self-generated certificate (e.g. in a testing environment), then it needs to be done.

Issue: BI Discoverer server CPU hits 100%.

Description: If there is a report based on a BI Discoverer worksheet that takes several minutes to return data and a user become impatient and repeatedly clicks the View or Export buttons it is possible this may overwhelm the Discoverer Server with requests and it may stop responding.

Workaround: There is no workaround for this issue except to not repeatedly click on the **View** or **Export** buttons while waiting for a report to return data.

Issue: Oracle BI Publisher does not return values for a Discoverer worksheet parameter's list of values past the 1000th value.

Description: Oracle BI Publisher limits the number of values in a list of values to 1000.

Workaround: Administrators can change this limit by setting the property DISCO_LOV_LIMIT in xmlp-server-config.xml. Example:

```
<property name=" DISCO_LOV_LIMIT " value="2000"/>
```

Note that the larger the list of values, the slower it will render.

Issue: Get the following error trying to open a report in the Template Builder: "'Error occurred. Please check the settings and try again'"

Description: It is possible that you are loading a report that returns too much data for the Template Builder to handle.

Workaround: Use the method described above for creating a layout template offline. Be sure to reduce the amount of data in the generated XML file.

14

Performance Improvements for Oracle BI Publisher

Oracle BI Publisher 10.1.3.3 has added performance improvements to its already fast Formatting Objects Processor (FOP) and Bursting engine. If you are running BI Publisher on a multiprocessor machine or on multicore processors, you can take better advantage of your hardware.

To enable these performance improvements, go to the Runtime Configuration page on either the Admin page or by following the Configure link for a specific report.

14.1 Multithreading for Bursting

If you have a multiprocessor machine or even a machine with a dual core single processor, you may be able to achieve even higher bursting throughput using the new multithreading functionality for Bursting.

Under the Bursting section on the Runtime Configuration page, you will see two properties that need to be set. Set "Enable multithreading" to True and set "Thread count" to some number greater than 1 and up to the number of processors or cores present on the machine.

If the report delivery channel is File System there will not be any considerable performance gain using multithreading. For delivery destinations other then File delivery, you should notice the performance gain.

Due to other processes that might be running on your system you may need to empirically determine what is the optimal setting for "Thread count." Try a series of tests by varying the setting "Thread count" to see what is optimal for your environment.

Note: Leave these settings at the default if your system does not have multicore processors or more than one processor. Setting "Enable multithreading" to True and "Thread count" to some number greater than the number of cores on the machine will lead to higher CPU usage without any gain in performance.

14.2 Multithreading for Document Generation

If you have a multiprocessor machine or even a machine with a dual core single processor, you may be able to achieve even faster document generation using the new multithreading functionality for FO Processing.

Under the FO Processor section of the properties, set "Enable multithreading" to True.

The amount of performance gain seen by enabling this setting will depend on how much the current system resources are utilized.

On a system that has numerous users running and relatively high CPU utilizations, you will likely only see minor improvements after setting "Enable multithreading" to True.

If the system is used by only a few users, or reports are scheduled sequentially one at a time, or the number of CPUs is more than the number of concurrent reports, then turning on multiple threads will speed up report generation.

Note: Memory utilization is likely to increase once "Enable multithreading" is set to True.

These properties can be set at the server and report level. See "Setting Runtime Properties" in the *Oracle Business Intelligence Publisher User's Guide* for more information on setting properties.

Note: If you are running BI Publisher on a single-core, one processor machine, leave these multithreading configuration settings at the default value of False.

15

Converting Reports from Oracle Reports to Oracle BI Publisher

This chapter describes how to convert reports from Oracle Reports to Oracle BI Publisher.

15.1 Overview

Oracle BI Publisher provides a utility for converting reports from Oracle Reports to Oracle BI Publisher.

In Oracle Reports, the data model (SQL query or extraction logic) and layout are contained in a single file. In Oracle BI Publisher the data model and the layout are separate objects. The conversion utility therefore generates several files that will make up your report in Oracle BI Publisher. In most cases this will include a PL/SQL specification and body that you will need to create in the database that contains the data for the report. The utility will also generate a report definition file and a layout template file that you will upload to the Oracle BI Publisher repository.

Once uploaded, test the report to ensure that the output is as expected and make any changes to the report as needed. Some reports will contain structures that the utility cannot convert. These must be manually implemented in the converted reports.

The overall flow for the conversion process is as follows:

1. Run the conversion utility.
2. Load the PL/SQL package into the database.
3. Upload the report to the Oracle BI Publisher repository.
4. Test the report and check the conversion log files to identify any manual modifications needed to complete the conversion.

15.2 Prerequisites

This section describes the steps required before running the conversion utility.

15.2.1 Reports and Oracle Reports XML Format

During the conversion process, the source reports must eventually be in Oracle Reports XML format. Oracle Reports XML format is supported in Oracle Reports 9i and above.

If your source reports are not in Oracle Reports XML format, the conversion utility will do this conversion automatically. However, this requires Oracle Reports Designer

9i or later on the same machine where you will do the conversion so that the Oracle BI Publisher conversion utility can call the rwconverter executable to get the reports into Oracle Reports XML format.

To get Oracle Reports Designer, download Oracle Developer Suite 10g (10.1.2.0.2) from:

http://www.oracle.com/technology/software/products/ids/index.html

and install Reports Designer.

If your reports are already in Oracle Reports XML format, there is no need to install Oracle Reports.

15.2.2 Update CLASSPATH with Needed JAR Files

To run the conversion utilities you need the following jar files in your classpath:

- Collections.zip -
- xmlparserv2-904.jar -
- xdocore.jar
- aolj.jar

For customers using Oracle Enterprise Business Suite (EBS), these libraries are available under JAVA_TOP. If these libraries are not there, the corresponding classes will be there under the JAVA_TOP.

For customer using Oracle BI Publisher Enterprise (sometimes referred to as "standalone"), all these libraries are available under WEB-INF\lib. Example:

C:\Oracle\bi\oc4j_bi\j2ee\home\applications\xmlpserver\xmlpserver\WEB-INF\lib

15.2.3 JDK Version 1.1.8

The conversion utility requires JDK version 1.1.8 or later.

15.3 Conversion Utility

BIPBatchConversion is the utility to migrate reports from Oracle Reports to BI Publisher.

It takes the following parameters:

- **-source** — (required) Source directory for Oracle Reports files. All reports must be in the same format - either RDF or XML.
- **-target** — (required) Target directory to create Oracle BI Publisher report objects. This includes the Oracle BI Publisher Report file (.xdo), the layout template file (.rtf), the PL/SQL package, and log file.
- **-oraclehome** — (optional) If your reports are in Oracle Reports XML format do not specify this parameter. If your reports are not in Oracle Reports XML format, specify the Oracle home path where Oracle Report Designer (9i or later version) is installed. BIPBatchMigration assumes that rwconverter is contained in the bin directory beneath the Oracle Home path.

  BIPBatchConversion requires rwconverter from Oracle Reports to convert the report from RDF format to XML format.

- **-debug —** (optional) To run the utility in debug mode and write debug statements to the log file.

Command Line Usage:

```
java ... BIPBatchConversion [-debug] -source SourceDirectory -target
TargetDirectory [-oraclehome OracleHomePath]
```

Sample Commands to run BIPBatchConversion:

Example 15–1 Source Report in Oracle Reports RDF File

This example requires you to specify the -oraclehome path.

```
java.exe  -classpath
D:\Jdev\project\xdocore.jar;d:\Jdev\project\collections.zip;d:\Jdev\project\aolj.j
ar;d:\Jdev\project\xmlparserv2-904.jar
oracle.apps.xdo.rdfparser.BIPBatchConversion -source d:\reports\pay -target
d:\reports\pay\output -oraclehome D:\oracle\BIToolsHome_1 -debug
```

Example 15–2 Source Report in Oracle Reports XML Format

Do not specify the -oraclehome parameter in this example.

```
java.exe  -classpath
D:\Jdev\project\xdocore.jar;d:\Jdev\project\collections.zip;d:\Jdev\project\aolj.j
ar;d:\Jdev\project\xmlparserv2-904.jar
oracle.apps.xdo.rdfparser.BIPBatchConversion -source d:\reports\pay -target
d:\reports\pay\output  -debug
```

See the Oracle BI Publisher 10.1.3.3 Java API reference (Javadoc) for more details on this utility and other related conversion APIs in the rdfparser package.

15.3.1 Output Files

The conversion utility will generate the following output files for each report:

- Report definition file that includes the Data Model (for example: REPORT.xdo)

  (Note: This file is not needed for E-Business Suite users; see the following note under DataTemplate.)
- DataTemplate (for example: REPORT_template.xml)

  (Note: this is not required for Oracle BI Publisher Enterprise users because the data template is embedded in REPORT.xdo.)
- Default PL/SQL package specification (for example: REPORTS.pls)
- Default PL/SQL package body (for example: REPORTB.pls)
- RTF Layout Template (for example: REPORT.rtf)
- Log file (for example: REPORT.log)

For example, assume you have a report called raxinv.rdf located in D:\reports\pay. Running this command:

```
java.exe  -classpath
D:\Jdev\project\xdocore.jar;d:\Jdev\project\collections.zip;d:\Jdev\project\aolj.j
ar;d:\Jdev\project\xmlparserv2-904.jar
oracle.apps.xdo.rdfparser.BIPBatchConversion -source d:\reports\pay -target
d:\reports\pay\output -oraclehome D:\oracle\BIToolsHome_1 -debug
```

Will generate the following output files:

- PL/SQL Package Specification: C:\BIPublisher_reports\ raxinv\raxinvS.pls
- PL/SQL Package Body: C:\BIPublisher_reports\ raxinv\raxinvB.pls
- Report definition file: C:\BIPublisher_reports\ raxinv\raxinv.xdo
- DataTemplate: C:\BIPublisher_reports\ raxinv\raxinv_template.xml
- RTF Layout Template: C:\BIPublisher_reports\raxinv\raxinv.rtf
- Log file: C:\BIPublisher_reports\raxinv\raxinv.log

15.4 Upload PL/SQL Package to the Database

Many converted Oracle Reports will generate a PL/SQL package specification file and a PL/SQL package body file as follows:

- <report_name>S.pls
- <report_name>B.pls

Run the PL/SQL package files against your Oracle Database as follows. This will create the PL/SQL package specification and body.

```
SQL> @C:\BIPublisher_reports\ raxinv\raxinvS.pls
SQL> @C:\BIPublisher_reports\ raxinv\raxinvB.pls
```

15.5 Putting Converted Reports into the Oracle BI Publisher Repository

Making your reports visible in the Oracle BI Publisher repository is a two-step process:

1. Copy the report folders into the repository.
2. Refresh the repository metadata.

If you have a file-based repository, copy the report folder structure to the file system. For example, suppose your repository path was set to:

C:\oracle\bi\xmlp\XMLP

For the report in the previous example, simply copy the report directory and all its files to the desired folder structure in the existing report repository.

For example:

```
> copy C:\BIPublisher_reports\raxinv C:\oracle\bi\xmlp\XMLP\Shared
Folders\Converted reports
```

If you are using the XML DB functionality of the Oracle Database as your report repository, there are several ways you can upload content. One method is through WebDAV.

For options and details on how to load reports into your XML DB based repository, please refer to the following topics in the *Oracle XML DB Developer's Guide* (http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14259/toc.htm):

- "Loading XML Content into Oracle XML DB"
- "FTP, HTTP(S), and WebDAV Access to Repository Data"

Once you have loaded report directories into your repository, log on to Oracle BI Publisher with administrator privileges and select "Refresh Metadata" from the System

Maintenance Section of the Oracle BI Publisher Admin page. Now all reports should be available for testing and execution.

15.6 Testing and Tweaking Your Reports

Now that you have successfully converted your reports, created the needed PL/SQL package, and moved the report definition and RTF template files into the Oracle BI Publisher repository, you should test your reports to make sure they are providing the data and formatting that you want.

Most converted reports will run exactly as expected without further modification. More complex reports may require some additional modification to work as desired. Following are some common issues with converted reports.

15.6.1 Data Template and PL/SQL Files

Occasionally when converting a more complex Oracle Reports report, the Data Template or PL/SQL may contain minor errors and require manual correction.

The conversion utility will move all formula columns to the select clause of the SQL query in the data model. In most cases this will not cause a problem. However, if any of the arguments to the formula is a summary column, this will not work because the summary column will not be calculated at the time the query is executed.

To correct this problem you will need to remove this formula from the select clause and implement the formula as XSL in your layout template. Most of these formulas are used either for simple addition or summation or currency conversion, formatting, and rounding.

15.6.2 RTF Layout Template File

The majority of Oracle Reports reports use simple 'if' formatting logic that is automatically converted to equivalent XSL-FO and inserted into form fields in the RTF layout template. However, there is no support for PL/SQL in RTF layout templates. The conversion utility does not convert any PL/SQL format trigger logic present in the report. Instead the conversion utility writes all the format trigger code to a log file. You will need to implement any corresponding PL/SQL logic as XSL code.

To aid in this process, the resulting RTF template will contain form fields that hold the format trigger names that are called; these fields will be highlighted in red. You can then refer to the log to find the actual PL/SQL code used in the original Oracle Report. You will need to rewrite these PL/SQL triggers as XSL-FO.

For more information, see the "Supported XSL-FO Elements" section of the *Oracle Business Intelligence Publishers User's Guide* (http://download.oracle.com/docs/cd/E10091_01/doc/bip.1013/b40017/toc.htm)

16

Support for Digital Signature in PDF Documents

This chapter describes Oracle BI Publisher support for Digital Signature in PDF Documents.

16.1 Introduction

Oracle BI Publisher now enables you to add a digital signature to PDF output documents. Digital signatures enable you to verify the authenticity of the documents you send and receive.

Oracle BI Publisher can access your digital ID file from a central, secure location and at runtime sign the PDF output with the digital ID. The digital signature verifies the signer's identity and ensures that the document has not been altered after it was signed.

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

- Digital ID Introduction by Verisign
  - http://www.verisign.com/support/tlc/per/whitepaper.htm
  - http://www.verisign.com/stellent/groups/public/documents/guides/005326.pdf
- Digital Signature by Adobe
  - http://www.adobe.com/security/digsig.html
- Digital Signatures in PDF and Acrobat
  - http://www.acrobatusers.com/articles/2006/07/digital_signatures/index.php

16.2 Prerequisites

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

- A digital ID obtained from a public certificate authority or from a private/internal certificate authority (if for internal use only). You must copy the digital ID file to a secure location of the file system on the server that is accessible by the BI Publisher server.

16.3 Limitations

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

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

16.4 Integration With Oracle BI Publisher

The following steps summarize the tasks required to set up and use digital signature with Oracle BI Publisher:

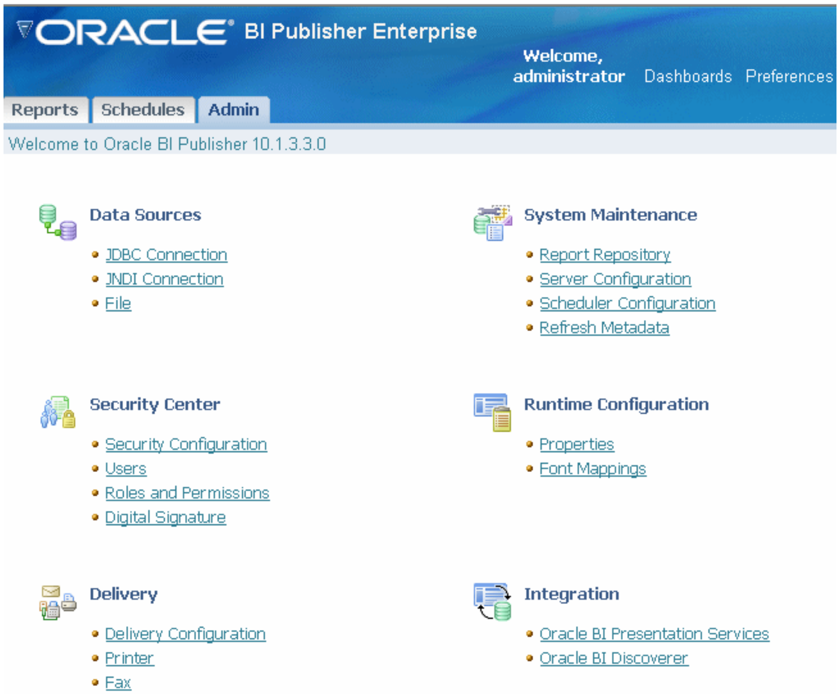
1. Register the digital ID in the BI Publisher Admin interface.
2. Specify the Roles that are authorized to sign documents.
3. (Optional for PDF templates) Add a signature field to the PDF template in which to place the digital signature at runtime.
4. Enable Digital Signature for the report in the Runtime Configuration properties and specify the position to place the digital signature on the completed document. This can be a signature field (for PDF templates), general location (top left, top center, or top right), or you can specify x and y coordinates.
5. Log in to BI Publisher as a user with an authorized role and submit the report through the BI Publisher scheduler, choosing PDF output. When the report completes it will be signed with your digital ID in the specified location of the document.

16.4.1 Digital Signature ID Registration And Allowed Role Assignment

In this release, BI Publisher supports the identification of a single digital ID file only. Register the digital ID in the BI Publisher Admin page as follows:

1. Log in to BI Publisher with Administrator credentials.
2. Select the Admin tab, and then from the Security Center menu select Digital Signature.

   The following screenshot shows the new Digital Signature entry available from the Admin tab.



ORACLE® BI Publisher Enterprise

Welcome, administrator   Dashboards   Preferences

Reports   Schedules   Admin

Welcome to Oracle BI Publisher 10.1.3.3.0

Data Sources

- JDBC Connection
- JNDI Connection
- File

System Maintenance

- Report Repository
- Server Configuration
- Scheduler Configuration
- Refresh Metadata

Security Center

- Security Configuration
- Users
- Roles and Permissions
- Digital Signature

Runtime Configuration

- Properties
- Font Mappings

Delivery

- Delivery Configuration
- Printer
- Fax

Integration

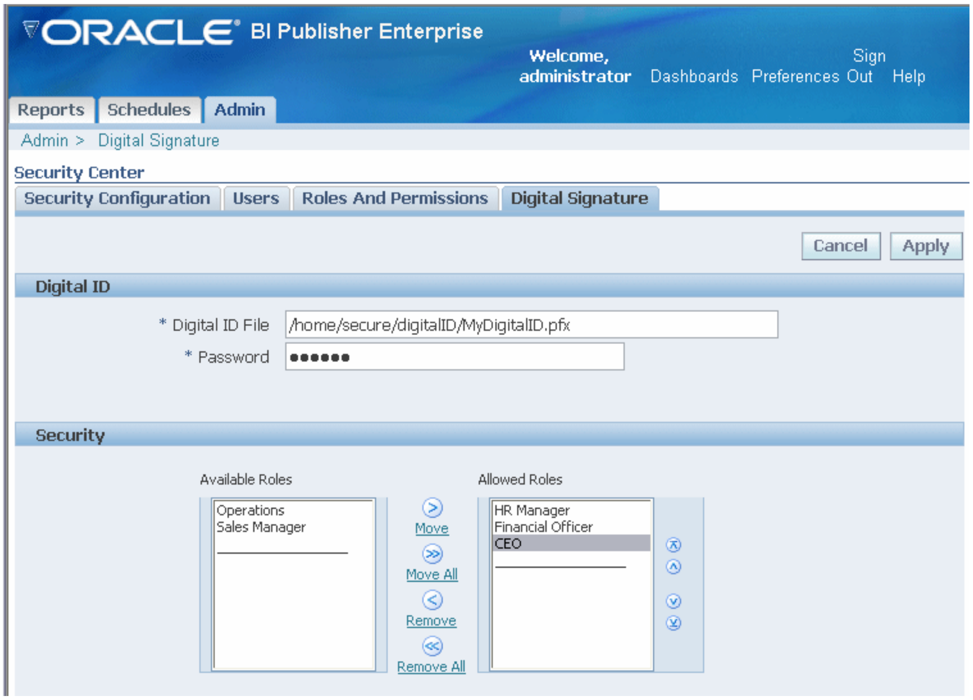
- Oracle BI Presentation Services
- Oracle BI Discoverer

3. On the Digital Signature sub-tab, enter the file path to the digital ID file and enter the password for the digital ID.

4. Enable the Roles that will have the authority to sign documents with this digital ID. Use the shuttle buttons to move Available Roles to the Allowed Roles list.

5. Click Apply.

   The following screenshot shows the Digital Signature sub-tab:



ORACLE® BI Publisher Enterprise

Welcome, administrator   Dashboards   Preferences   Sign Out   Help

Reports   Schedules   Admin

Admin > Digital Signature

Security Center

Security Configuration | Users | Roles And Permissions | Digital Signature

Cancel   Apply

Digital ID

* Digital ID File: /home/secure/digitalID/MyDigitalID.pfx

* Password: ••••••

Security

Available Roles: Operations, Sales Manager

Allowed Roles: HR Manager, Financial Officer, CEO

16.4.2 Specify the Signature Display Field or Location

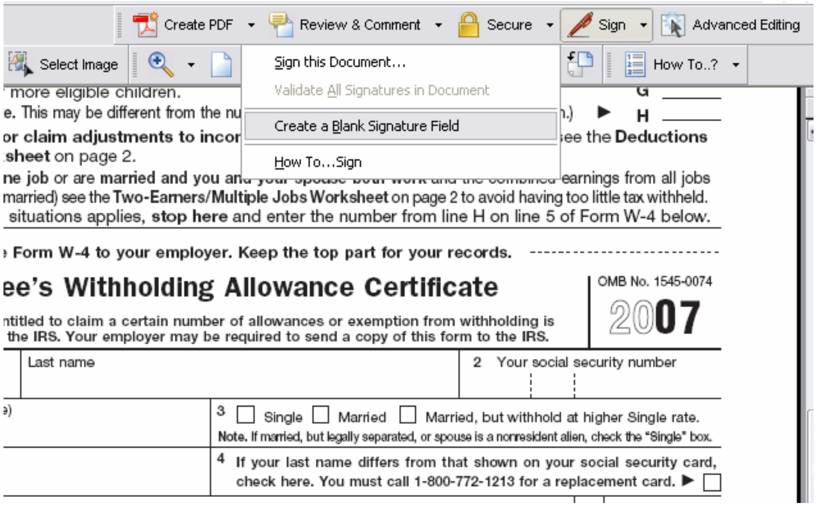
You must specify the location for the digital signature to appear in the completed document. If your template is PDF, you can either specify a template field in which to put the digital signature or you can just specify a location on the document using Runtime Configuration properties. If your template is RTF, you can only specify a location; you cannot specify a field.

16.4.2.1 Designating a Field in a PDF Template for the Digital Signature

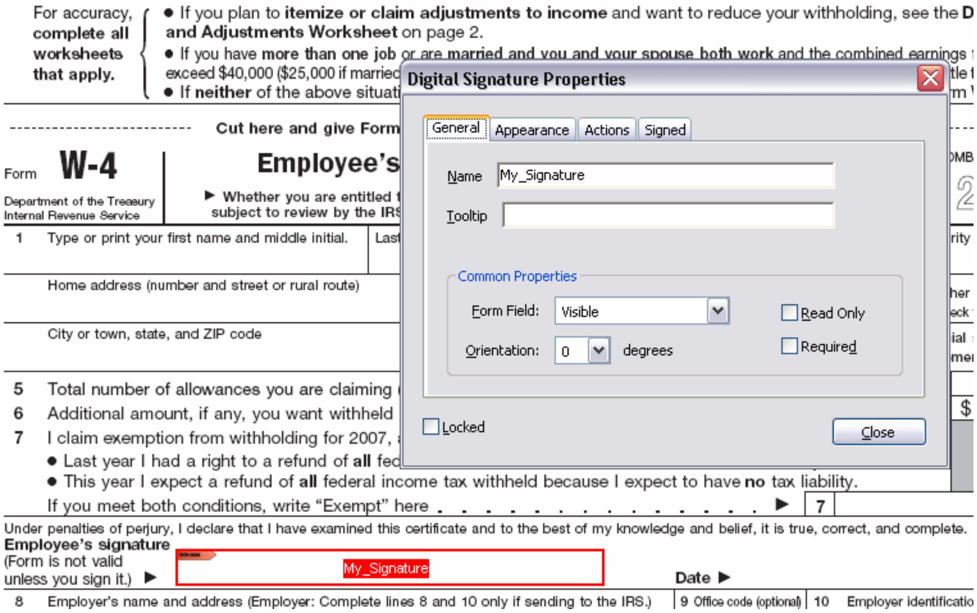
If your template is a PDF, you can designate an existing field in which to place the digital signature or insert a new signature field on the form.

Add a signature field:

1. Open the template in Adobe Acrobat Professional.
2. Select Sign from the toolbar and then select Create a Blank Signature Field. The following figure displays this selection from the Adobe Acrobat Professional toolbar.


Create PDF
Review & Comment
Secure
Sign
Advanced Editing
Select Image
How To..?
Sign this Document...
Validate All Signatures in Document
Create a Blank Signature Field
How To...Sign
more eligible children.
e. This may be different from the nu
or claim adjustments to incor
sheet on page 2.
ne job or are married and you an
married) see the Two-Earners/Multiple Jobs Worksheet on page 2 to avoid having too little tax withheld.
situations applies, stop here and enter the number from line H on line 5 of Form W-4 below.
see the Deductions
G
H
Form W-4 to your employer. Keep the top part for your records.
ee's Withholding Allowance Certificate
ntitled to claim a certain number of allowances or exemption from withholding is
the IRS. Your employer may be required to send a copy of this form to the IRS.
OMB No. 1545-0074
2007
Last name
2 Your social security number
3 □ Single □ Married □ Married, but withhold at higher Single rate.
Note. If married, but legally separated, or spouse is a nonresident alien, check the "Single" box.
4 If your last name differs from that shown on your social security card, check here. You must call 1-800-772-1213 for a replacement card. ▶ □

3. Draw the signature field in the desired location on the template. When you are finished drawing (that is, when you release the mouse button), the Digital Signature Properties dialog will display.
4. Assign a name to your signature field. The following figure shows an inserted digital signature field called "My_Signature."



For accuracy, complete all worksheets that apply.

- If you plan to **itemize or claim adjustments to income** and want to reduce your withholding, see the **D** **and Adjustments Worksheet** on page 2.
- If you have **more than one job** or are married and you and your spouse both work and the combined earnings exceed $40,000 ($25,000 if marrie
- If **neither** of the above situati

Cut here and give Form

Form **W-4**

Department of the Treasury
Internal Revenue Service

Employee's

▶ Whether you are entitled subject to review by the IRS

1 Type or print your first name and middle initial. | Last

Home address (number and street or rural route)

City or town, state, and ZIP code

5 Total number of allowances you are claiming
6 Additional amount, if any, you want withheld
7 I claim exemption from withholding for 2007,
- Last year I had a right to a refund of **all** fed
- This year I expect a refund of **all** federal income tax withheld because I expect to have **no** tax liability.

If you meet both conditions, write "Exempt" here . . . . ▶ 7

Under penalties of perjury, I declare that I have examined this certificate and to the best of my knowledge and belief, it is true, correct, and complete.

Employee's signature
(Form is not valid unless you sign it.) ▶ My_Signature **Date** ▶

8 Employer's name and address (Employer: Complete lines 8 and 10 only if sending to the IRS.) | 9 Office code (optional) | 10 Employer identificatio

Digital Signature Properties

Name: My_Signature

Tooltip:

Common Properties

Form Field: Visible

Orientation: 0 degrees

□ Read Only

□ Required

□ Locked

5. Save your template.

Now perform the following task to configure BI Publisher to insert the digital signature to this field.

Upload the template and configure the report:

1. Log in to BI Publisher with Developer or Administrator credentials.

2. Upload the template to the appropriate report definition.

   a. Navigate to the report.

   b. Select the Edit link.

   c. From the Report definition pane select Layouts.

   d. Select Browse to locate your template, then select Upload.

   e. After the template uploads successfully, with Layouts still selected on the Report definition pane, select New to add a layout.

   f. In the General Settings of the Layout definition, enter a Name for your layout, then select the uploaded template from the list. Select the appropriate Template Type and Output Formats.

   g. Click Save.

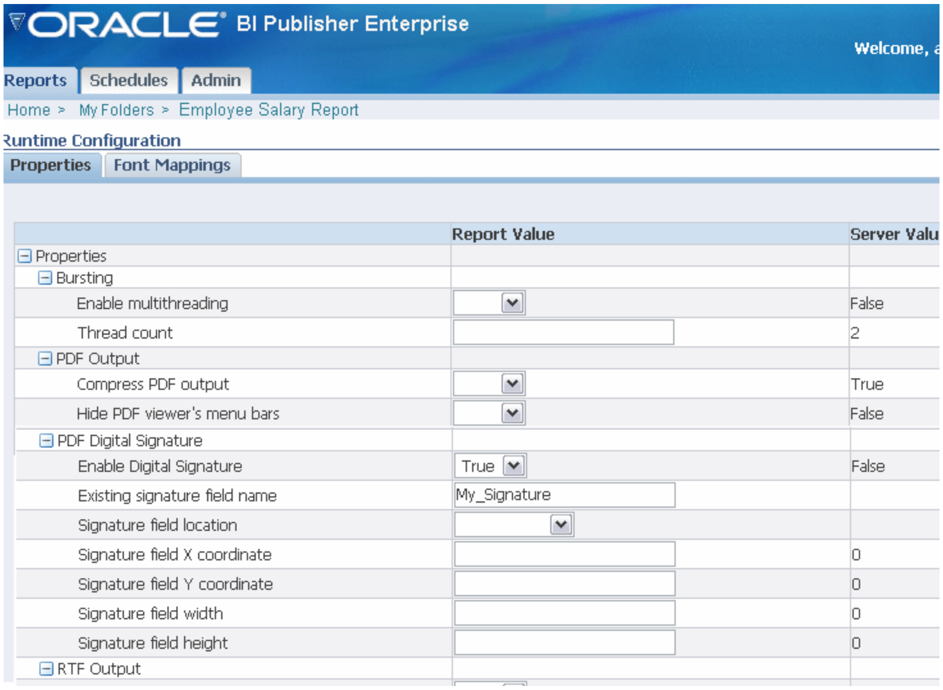
3. Select the Configure link for the report.

4. Scroll down to the PDF Digital Signature group of properties.

5. Set Enable Digital Signature to True.

6. Enter the field name from the PDF template for the property Existing signature field name.

The following screenshot shows the "My_Signature" field name entered into the properties field.



	Report Value	Server Valu
Properties		
Bursting		
Enable multithreading		False
Thread count		2
PDF Output		
Compress PDF output		True
Hide PDF viewer's menu bars		False
PDF Digital Signature		
Enable Digital Signature	True	False
Existing signature field name	My_Signature	
Signature field location		
Signature field X coordinate		0
Signature field Y coordinate		0
Signature field width		0
Signature field height		0
RTF Output		

7. Select Apply.

   Note that the runtime properties that you have just set are at the report level and not the template level. Therefore any templates associated with the report will now include the digital signature as specified in the Runtime Configuration properties for the report. When an Existing signature field name is specified, the template must contain the field for the signature to be applied.

16.4.2.2 Specifying a Location in the Document to Insert the Digital Signature (for RTF and PDF Templates)

When you specify a location in the document to place the digital signature, you can either specify a general location (Top Left, Top Center, or Top Right) or you can specify x and y coordinates in the document. You can also specify the field height and width. This is done through properties on the Runtime Configuration page. Therefore you do not need to alter the template to include a digital signature.

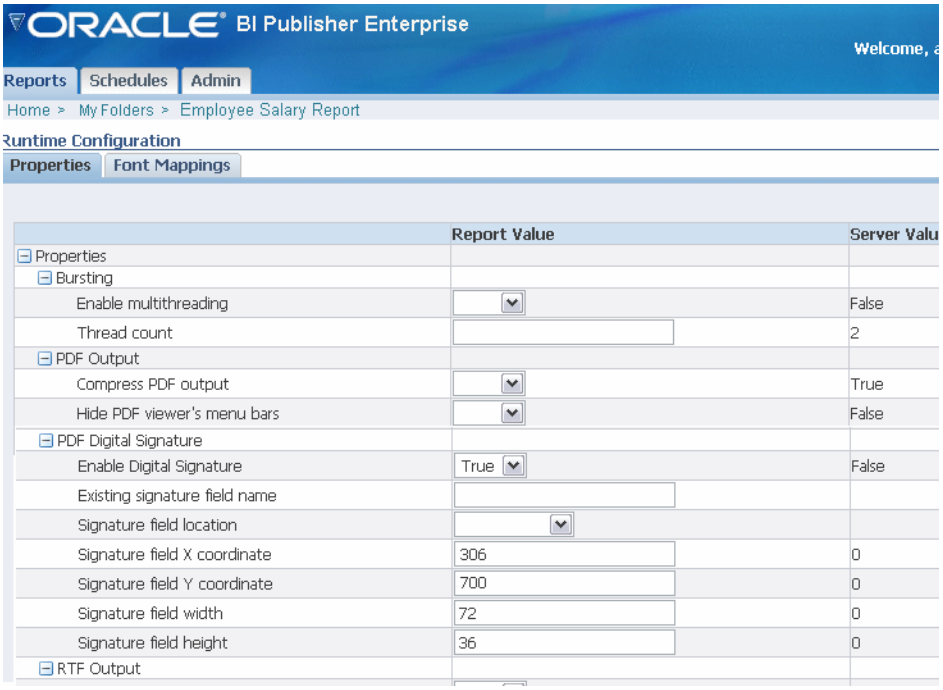
1. Log in to BI Publisher with Developer or Administrator credentials.
2. Navigate to the report you wish to allow authorized users to sign with the digital ID.
3. Select the Configure link for the report.
4. Scroll to the PDF Digital Signature group of properties.
5. Set Enable Digital Signature to True.
6. Specify the location in the document where you want the digital signature to appear by setting the appropriate properties as follows (note that the signature is inserted on the first page of the document only):

   - Existing signature field name – does not apply to this method.
   - Signature field location – provides a list containing the following values: Top Left, Top Center, Top Right. Choose one of these general locations and BI Publisher will insert the digital signature to the output document sized and

positioned appropriately. If you set this property, do not enter X and Y coordinates or width and height properties.

- Signature field X coordinate – using the left edge of the document as the zero point of the X axis, enter the position in points 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.
- Signature field Y coordinate – 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.
- Signature field width – enter in points the desired width of the inserted digital signature field. This applies only if you are setting the X and Y coordinates.
- Signature field height – enter in points the desired height of the inserted digital signature field. This applies only if you are setting the X and Y coordinates.

The following screenshot shows a report configured to place the digital signature at specific x and y coordinates in the document.



	Report Value	Server Valu
⊟ Properties		
⊟ Bursting		
Enable multithreading		False
Thread count		2
⊟ PDF Output		
Compress PDF output		True
Hide PDF viewer's menu bars		False
⊟ PDF Digital Signature		
Enable Digital Signature	True	False
Existing signature field name		
Signature field location		
Signature field X coordinate	306	0
Signature field Y coordinate	700	0
Signature field width	72	0
Signature field height	36	0
⊟ RTF Output		

16.4.3 Running and Signing Reports with the Digital Signature

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

1. Log in to BI Publisher as a user with a role granted digital signature privileges.
2. Navigate to the report that has been enabled for digital signature and select Schedule.
3. Complete the scheduler fields, selecting PDF output, and then submit the job.

4. The completed PDF will display the digital signature.

16.4.4 Additional PDF Security Features

BI Publisher also supports secured PDF to prevent unauthorized users from opening, printing, updating or cutting and pasting the PDF document contents. Digital signature is also supported for PDF reports configured with these security features enabled.

Users with the Developer or Administrator role can change PDF security settings for a report. To set PDF security properties, select the Configure link for the report. For more information about the PDF security properties, see the topic: Setting Runtime Properties in the Oracle Business Intelligence Publisher User's Guide.

17

Support for Postscript Printers

This chapter describes Oracle BI Publisher support for postscript printers.

17.1 Introduction

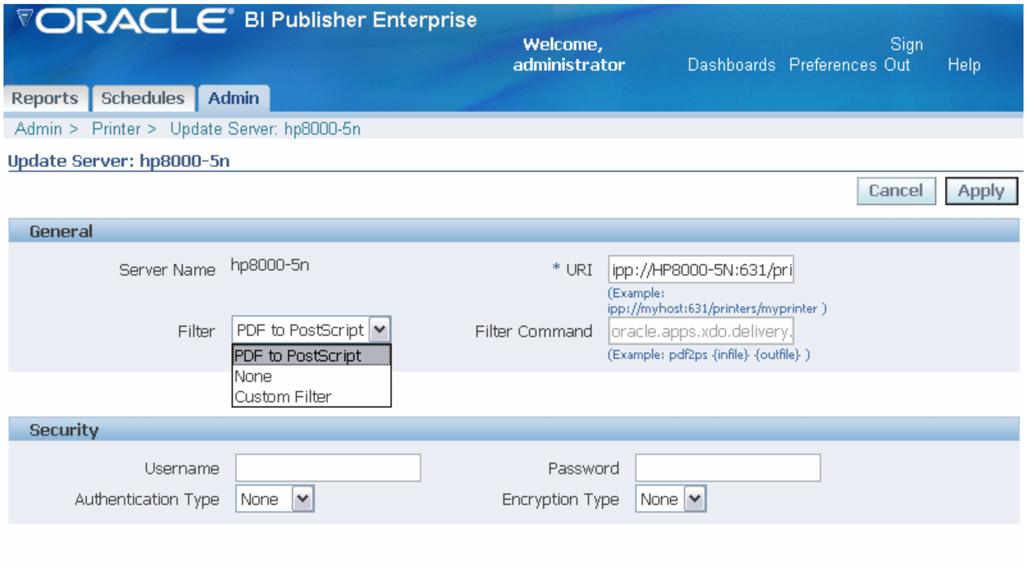
PDF is a popular output format for business reports and is printable from viewer software such as Adobe Reader. However, some reports require printing directly from the report server. For example, paychecks and invoices are usually printed as scheduled batch jobs. Some newer printers with Postscript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

This release introduces native support for PostScript Level printers. Previously, to enable printing from the BI Publisher server to a PostScript printer a filter had to be configured to invoke a third-party utility to perform the conversion. Now BI Publisher provides the utility to convert PDF to PostScript 2.0. If your printer supports PostScript 2.0, you will no longer be required to use a third-party utility to print directly from the server.

17.2 Setting Up Your Print Server to Use the Converter

The conversion utility is provided as an option from the Filter list of the Print Server definition page:

1. Log in to Oracle BI Publisher as an Administrator.
2. From the Admin tab, under the Delivery heading, select Printer.
3. Select an existing print server, or to add a new one, click Add Server.
4. Under the General properties, select PDF to PostScript from the Filter field list. The following screenshot shows the Filter field list of the print server definition.



Update Server: hp8000-5n

General

Server Name hp8000-5n

* URI ipp://HP8000-5N:631/pri
(Example: ipp://myhost:631/printers/myprinter)

Filter PDF to PostScript

Filter Command oracle.apps.xdo.delivery.
(Example: pdf2ps {infile} {outfile})

Security

Username

Password

Authentication Type None

Encryption Type None

For information on setting up the other fields in the Print Server definition page see the Oracle Business Intelligence Publisher User's Guide.

After completing all other required fields for the print server, you will be able to schedule reports to print directly from the BI Publisher server to any printer in your system that supports PostScript Level 2.

17.3 Limitations

This release of the PDF to PostScript converter has the following limitations:

- Only PDF version 1.4 or earlier is supported. This is also the currently supported version of the Oracle BI Publisher PDF output.
- CID fonts, which are used mainly to support languages with large sets of glyphs like Chinese, Japanese, and Korean, are only supported when embedded in the PDF.
- Transparent and half-tone renderings are not supported.
- DeviceN color space is not supported.
- Shading patterns are not fully supported.
- Vertical writing is not supported.

18

Additional New Features in Release 10.1.3.3

The 10.1.3.3 release of the Oracle Business Intelligence Enterprise Edition also includes miscellaneous new features for BI Publishr. These are described in the following sections:

- Section 18.1, "Support for CSV Output"
- Section 18.2, "Report Definition Features"
- Section 18.3, "Report Viewing Features"
- Section 18.4, "New Property for LDAP Security Model"

18.1 Support for CSV Output

Comma Separated Value (CSV) is now supported as an output type for any report with a simple <rowset>/<row> data structure. Note that support for this output type is based on the structure of the data. No template is required.

18.2 Report Definition Features

The following new features have been added to enhance options when defining reports:

- BI Publisher's data template data source now provides native support for BLOB, CLOB, long, and raw data types.
- When defining parameters for your report, if you select Menu as your parameter type, you now have the option under the "Can select all" check box to choose between "Null value passed" and "All values passed". Previously, selecting "all" passed a null value by default.
- Properties added to the Runtime Configuration page:
  - **RTF Output**: Default Font - enter the default font name and size (for example: Arial:12) to use when no font is defined. This is particularly useful to control the sizing of empty table cells in generated reports.
  - **HTML Output**: Use percentage width for table columns - set this property to True to render table columns according to a percentage value of the total width of the table rather than as a value in points.
  - **FO Processing**:
    - Disable external references - a "True" setting (default) disallows the importing of secondary files such as subtemplates or other XML documents during XSL processing and XML parsing. This increases the

security of your system. Set this to "False" if your report or template calls external files.

- * FO Parsing Buffer Size - (default 1,000,000) sets the size of the buffer for the FO Processor. When the buffer is full, the elements from the buffer will be rendered in the report. Reports with large tables or crosstabs 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 will affect the memory consumption of your system.

18.3 Report Viewing Features

The following new features are available from the View report page:

- Views created in the online Analyzer can be saved as templates.
- When in View mode, you will now see "Link to this Report" displayed on the page. This enables you to capture the URL of the currently viewed report. The Link to this Report selection has four options:
  - Current Page
  - No Header
  - No Parameters
  - Document Only

18.4 New Property for LDAP Security Model

For the LDAP security model, a new property, Group Retrieval Size, has been added. This enables support of the LDAPv3 control extension for simple paging of search results. By default, pagination is not used. If this property is provided then the 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 OID 10.1.4. See your LDAP server documentation for information on support of this control type. For more information about LDAP pagination and the required control type, see the article: RFC 2696 - LDAP Control Extension for Simple Paged Results Manipulation (http://www.faqs.org/rfcs/rfc2696.html) .

19

New Features in Release 10.1.3.3.1

The 10.1.3.3.1 release of the Oracle Business Intelligence Enterprise Edition introduces the following new BI Publisher features:

- Section 19.1, "Time Out Value for Web Service Data Sources"
- Section 19.2, "Bidirectional Function for eText Data Fields"
- Section 19.3, "Default Repository"

19.1 Time Out Value for Web Service Data Sources

You can now set a time out value when creating a data set based on a Web service. The Data Set Details definition page contains the "Time Out" field. The default is 60 seconds. Enter a numeric value to equal the number of seconds that BI Publisher will wait for a response from the Web service before giving up.

To set the time out value for a Web service data set:

1. Navigate to the report, and select Edit.
2. Under the report Data Model region, select the Web service data set.
3. In the Details region, enter the desired time out period in seconds (default is 60 seconds).

19.2 Bidirectional Function for eText Data Fields

Starting with 10.1.3.3.1 there is a new BIDI function to preserve the correct order of characters for bidirectional languages in eText templates. If you have data fields that contain bidirectional data (for example, Arabic or Hebrew fields) you can use the BIDI function to preserve the correct right to left ordering of the characters.

To use the BIDI function, enter the following in your template for each field that will contain data to be treated as bidirectional:

```
BIDI(ELEMENT_NAME)
```

For example:

```
<POSITION>  <LENGTH>  <FORMAT>  <PAD>    <DATA>                 <COMMENTS>
     9         21        Alpha     L,' '  BIDI(VENDOR_ADDRESS)  Vendor Address
```

In this example, the data for the element VENDOR_ADDRESS will be treated as bidirectional.

19.3 Default Repository

BI Publisher now sets a default repository location.

The repository path is defined in 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="${oracle.home}/xdo/repository"/>
 </resource>
</xmlpConfig>
```

If the "oracle.home" Java system property is set to a valid path, ${oracle.home} is replaced with that value.

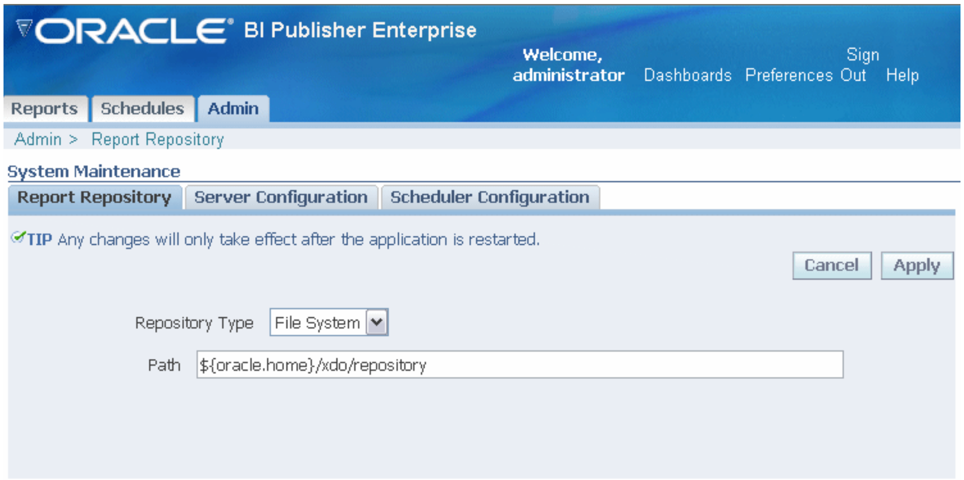
When BI Publisher is deployed on OC4J, `${oracle.home}` is automatically set to a directory path where OC4J is installed.

For example, in an OC4J deployment in which the OC4J home is `C:\oc4j`, the BI Publisher repository is automatically set to:

```
C:\oc4j\xdo\repository
```

Note that the repository can also be set through BI Publisher Admin screens as shown in the following figure:

Figure 19–1 Setting the BI Publisher Report Repository Through the Admin Screen



TIP Any changes will only take effect after the application is restarted.

Repository Type: File System

Path: ${oracle.home}/xdo/repository

20

Oracle Business Intelligence Publisher Web Services

This chapter explains the Oracle Business Intelligence Publisher (Oracle BI Publisher) Web services APIs and related information. This chapter assumes a basic understanding of Web services and application programming interfaces. It contains the following topics:

20.1 Overview of the Oracle BI Publisher Web Services

20.1.1 Functionality Overview

PublicReportService is the Web service that wraps the Oracle BI Publisher public APIs. Starting with release 10.1.3.3.1, the following operations are supported. These operations can be grouped into three categories:

- Operations for validation of privileges
- Operations to get information about reports and the repository
- Operations to run and schedule reports

20.1.2 WSDL Definition for PublicReportService

After you have installed or deployed Oracle BI Publisher, there is a unique URL associated with PublicReportService:

```
http://<host>:<port>/xmlpserver/services/PublicReportService?wsdl
```

Enter this URL in your browser, substituting in the correct host and port number, to display the full Web Service Description Language definition for the publicly supported BI Publisher Web service.

On some J2EE servers, this URL may not correctly display the WSDL for this Web service. Please refer to the section Section 20.5, "Oracle BI Publisher Web Services WSDL" to see the full definition.

20.2 Oracle BI Publisher Web Services Data Types

This section covers the following topics:

- Base Data Types
- XML-to-Java Data Type Mappings
- Complex Types

20.2.1 Base Data Types

Oracle BI Publisher Web services use the following base data types:

Base Type	Description	Example(s)
xsd:boolean	Boolean	true, false
xsd:dateTime	Date and Time	2007-10-26T21:32:52
xsd:int	Integer	23
xsd:string	String	/Home/Shared/HR Reports/Salary Report
xsd:base64Binary	64 bit binary	A document in PDF, HTML, etc. format

20.2.2 XML-to-Java Data Type Mappings

BI Publisher Web Services use RPC-encoded formats. The mapping between Web service XML schema data types and Java data types depends on the SOAP development environment. The following table shows mappings for the Oracle JDeveloper environment:

XML Schema	Oracle JDeveloper
xsd:boolean	java.lang.Boolean
xsd:dateTime	java.util.Date
xsd:int	java.lang.Integer
xsd:string	java.lang.String
xsd:base64Binary	java.lang.Byte

20.2.3 Complex Types

Oracle BI Publisher Web services define and use the following complex types:

ArrayOf_xsd_string

```
<complexType name="ArrayOf_xsd_string">
 <complexContent>
  <restriction base="soapenc:Array">
   <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
  </restriction>
 </complexContent>
</complexType>
```

Description: an array of strings used in other complex types.

ParamNameValue

```
<complexType name="ParamNameValue">
 <sequence>
```

```
  <element name="isMultiValuesAllowed" type="xsd:boolean" />
  <element name="multiValuesAllowed" type="xsd:boolean" />
  <element name="name" nillable="true" type="xsd:string" />
  <element name="values" nillable="true" type="impl:ArrayOf_xsd_string" />
 </sequence>
</complexType>
```

Description: a parameter name and array of parameter values.

ArrayOfParamNameValue

```
<complexType name="ArrayOfParamNameValue">
 <complexContent>
  <restriction base="soapenc:Array">
   <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:ParamNameValue[]" />
  </restriction>
 </complexContent>
</complexType>
```

Description: an array of ParamNameValues.

ReportDefinition

```
<complexType name="ReportDefinition">
 <sequence>
  <element name="defaultOutputFormat" nillable="true" type="xsd:string" />
  <element name="defaultTemplateId" nillable="true" type="xsd:string" />
  <element name="parameterNames" nillable="true"
    type="impl:ArrayOf_xsd_string" />
  <element name="reportParameterNameValues" nillable="true"
    type="impl:ArrayOfParamNameValue" />
  <element name="templateIds" nillable="true" type="impl:ArrayOf_xsd_string" />
 </sequence>
</complexType>
```

Description: object returned from `getReportDefinition`.

ItemData

```
<complexType name="ItemData">
 <sequence>
  <element name="absolutePath" nillable="true" type="xsd:string" />
  <element name="creationDate" nillable="true" type="xsd:dateTime" />
  <element name="displayName" nillable="true" type="xsd:string" />
  <element name="fileName" nillable=
"true" type="xsd:string" />
  <element name="lastModified" nillable="true" type="xsd:dateTime" />
  <element name="lastModifier" nillable="true" type="xsd:string" />
  <element name="owner" nillable="true" type="xsd:string" />
  <element name="parentAbsolutePath" nillable="true" type="xsd:string" />
  <element name="type" nillable="true" type="xsd:string" />
 </sequence>
</complexType>
```

Description: description of an object contained in the report repository.

ArrayOfItemData

```
<complexType name="ArrayOfItemData">
 <complexContent>
  <restriction base="soapenc:Array">
   <attribute ref="soapenc:arrayType"
 wsdl:arrayType="impl:ItemData[]" />
  </restriction>
 </complexContent>
</complexType>
```

Description: an array of ItemData - objects contained in the report repository.

ReportRequest

```
<complexType name="ReportRequest">
 <sequence>
  <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="parameterNameValues" nillable="true"
   type="impl:ArrayOfParamNameValue"/>
  <element name="reportAbsolutePath" nillable="false" type="xsd:string"/>
 </sequence>
</complexType>
```

Description: the collection of settings needed to run a report. Note that allowable values for `attributeFormat` will vary according to the type of template used (for example, PDF templates can only generate PDF output.)

ReportResponse

```
<complexType name="ReportResponse">
 <sequence>
  <element name="reportBytes" nillable="true" type="xsd:base64Binary" />
  <element name="reportContentType" nillable="true" type="xsd:string" />
  <element name="reportLocale" nillable="true" type="xsd:string" />
 </sequence>
</complexType>
```

Description: the document (in whatever file format `attributeFormat` is set to in `ReportRequest`), document type, and document locale that is returned from the runReport operation.

EMailDeliveryOption

```
<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="false" type="xsd:string"/>
  <element name="emailReplyTo" nillable="true" type="xsd:string"/>
  <element name="emailSubject" nillable="false" type="xsd:string"/>
  <element name="emailTo" nillable="false" type="xsd:string"/>
 </sequence>
</complexType>
```

Description: the options to set for e-mail delivery of a report.

FaxDeliveryOption

```
<complexType name="FaxDeliveryOption">
 <sequence>
  <element name="faxNumber" nillable="false" type="xsd:string"/>
  <element name="faxServer" nillable="false" type="xsd:string"/>
 </sequence>
```

```
</complexType>
```

Description: the options to set for facsimile (fax) delivery of a report.

FTPDeliveryOption

```
<complexType name="FTPDeliveryOption">
 <sequence>
  <element name="ftpServerName" nillable="false" type="xsd:string"/>
  <element name="ftpUserName" nillable="false" type="xsd:string"/>
  <element name="ftpUserPassword" nillable="false" type="xsd:string"/>
  <element name="remoteFile" nillable="false" type="xsd:string"/>
  <element name="sftpOption" type="xsd:boolean"/>
 </sequence>
</complexType>
```

Description: the options to set for FTP delivery of a report.

LocalDeliveryOption

```
<complexType name="LocalDeliveryOption">
 <sequence>
  <element name="destination" nillable="false" type="xsd:string"/>
 </sequence>
</complexType>
```

Description: the options to set for delivery of a report to the BI Publisher repository.

PrintDeliveryOption

```
<complexType name="PrintDeliveryOption">
 <sequence>
  <element name="printNumberOfCopy" nillable="true" type="xsd:string"/>
  <element name="printRange" nillable="true" type="xsd:string"/>
  <element name="printSide" nillable="true" type="xsd:string"/>
  <element name="printTray" nillable="true" type="xsd:string"/>
  <element name="printerName" nillable="false" type="xsd:string"/>
 </sequence>
</complexType>
```

Description: the options to set for printer delivery of a report.

WebDAVDeliveryOption

```
<complexType name="WebDAVDeliveryOption">
 <sequence>
  <element name="deliveryAuthType" nillable="true" type="xsd:string"/>
  <element name="deliveryAuthTypeBasic" nillable="true" type="xsd:string"/>
  <element name="deliveryAuthTypeDigest" nillable="true" type="xsd:string"/>
  <element name="password" nillable="true" type="xsd:string"/>
  <element name="remoteFilePath" nillable="false" type="xsd:string"/>
  <element name="server" nillable="false" type="xsd:string"/>
  <element name="userName" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
```

Description: the options to set for Web-based Distributed Authoring and Versioning (WebDAV) delivery of a report.

DeliveryRequest

```
<complexType name="DeliveryRequest">
```

```
 <sequence>
  <element name="emailOption" nillable="true" type="impl:EMailDeliveryOption"/>
  <element name="faxOption" nillable="true" type="impl:FaxDeliveryOption"/>
  <element name="ftpOption" nillable="true" type="impl:FTPDeliveryOption"/>
  <element name="localOption" nillable="true" type="impl:LocalDeliveryOption"/>
  <element name="printOption" nillable="true" type="impl:PrintDeliveryOption"/>
  <element name="webDAVOption" nillable="true"
    type="impl:WebDAVDeliveryOption"/>
 </sequence>
</complexType>
```

Description: an object that provides the options to deliver a report to multiple destinations.

ScheduleRequest

```
<complexType name="ScheduleRequest">
 <sequence>
  <element name="deliveryRequest" nillable="false"
    type="impl:DeliveryRequest"/>
  <element name="endDate" nillable="true" type="xsd:dateTime"/>
  <element name="jobCalendar" nillable="true" type="xsd:string"/>
  <element name="jobLocale" nillable="true" type="xsd:string"/>
  <element name="jobTZ" nillable="true" type="xsd:string"/>
  <element name="notificationTo" nillable="true" type="xsd:string"/>
  <element name="notifyWhenFailed" nillable="true" type="xsd:string"/>
  <element name="notifyWhenSuccess" nillable="true" type="xsd:string"/>
  <element name="notifyWhenWarning" nillable="true" type="xsd:string"/>
  <element name="repeatCount" type="xsd:int"/>
  <element name="repeatInterval" type="xsd:int"/>
  <element name="reportRequest" nillable="false" type="impl:ReportRequest"/>
  <element name="saveDataOption" nillable="true" type="xsd:string"/>
  <element name="saveOutputOption" nillable="true" type="xsd:string"/>
  <element name="scheduleBurstringOption" nillable="true" type="xsd:string"/>
  <element name="schedulePublicOption" nillable="true" type="xsd:string"/>
  <element name="startDate" nillable="true" type="xsd:dateTime"/>
  <element name="useUTF8Option" nillable="true" type="xsd:string"/>
  <element name="userJobName" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
```

Description: the options to schedule a report.

AccessDeniedException

```
<complexType name="AccessDeniedException">
 <sequence />
</complexType>
```

Description: message returned when the user credentials do not have adequate privileges to complete an operation.

OperationFailedException

```
<complexType name="OperationFailedException">
 <sequence />
</complexType>
```

Description: the message returned when an operation fails.

InvalidParametersException

```
<complexType name="InvalidParametersException">
 <sequence />
</complexType>
```

Description: message returned when invalid parameters are supplied for a Web service operation.

20.3 Oracle BI Publisher Web Services Operations

The operations available in the Oracle BI Publisher Web services can be described by the following categories:

- Operations for validation of privileges
- Operations to get information about reports and the repository
- Operations to run and schedule reports

This section describes the input message, output message, possible exception messages and also provides sample code for each operation.

20.3.1 About Using the Code Samples

Each operation listed in the following section includes sample code. The sample code is based on Apache Axis technology. Before working with the examples you must ensure you have included the prerequisite files in your classpath and generated the required BI Publisher stub classes to import into your project.

Include the Required jar Files

- Include the following jar files in your classpath (these files can be obtained from the Web):
  - `activation.jar`
  - `mail-1.4.jar`
- Include `xmlpserver.jar` file (installed with Oracle BI Publisher) in your classpath. This is to ensure that you have all the required request and response objects. This way you will not need to generate response and request stub objects.

Generate the Required BI Publisher Stub Classes

To generate the required BI Publisher stub classes to import to your project for this sample, perform the following:

1. Download the BI Publisher Web services WSDL.
2. Run the following in a command file:

   ```
   export CLASSPATH="D:/bin/axis-1_4/lib/axis.jar;D:/bin/axis-1_
   4/lib/jaxrpc.jar;D:/bin/axis-1_4/lib/saaj.jar;D:/bin/axis-1_
   4/lib/commons-logging.jar;D:/bin/axis-1_
   4/lib/commons-discovery.jar;D:/bin/axis-1_4/lib/wsdl4j.jar"
   echo $CLASSPATH
   java org.apache.axis.wsdl.WSDL2Java PublicReportService.wsdl
   ```

This will generate the following BI Publisher stub classes for you to import into your project:

```
import com.oracle.xmlns.oxp.service.PublicReportService.ItemData;
```

```
import com.oracle.xmlns.oxp.service.PublicReportService.ReportRequest;
import com.oracle.xmlns.oxp.service.PublicReportService.ReportResponse;
import com.oracle.xmlns.oxp.service.PublicReportService.ParamNameValue;
import  com.oracle.xmlns.oxp.service.PublicReportService.ReportDefinition;
import com.oracle.xmlns.oxp.service.PublicReportService.ScheduleRequest;
import com.oracle.xmlns.oxp.service.PublicReportService.DeliveryRequest;
import com.oracle.xmlns.oxp.service.PublicReportService.EMailDeliveryOption;
```

Import the Required Apache Axis and Java Stub Classes

In addition to importing the BI Publisher Web service stub classes that you generated, you must also import the following Apache Axis and Java classes:

```
import  java.io.FileOutputStream;
import  java.io.OutputStream;
import  java.util.Calendar;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import org.apache.axis.encoding.ser.BeanDeserializerFactory;
import org.apache.axis.encoding.ser.BeanSerializerFactory;
import  javax.xml.namespace.QName;
import  javax.xml.rpc.ParameterMode;
import  java.net.URL;
```

20.3.2 Operations for Validation of Privileges

validateLogin

Use validateLogin to validate that a UserID and Password have the privilege to access the Oracle BI Publisher report server.

input message **validateLoginRequest**

```
<wsdl:message name="validateLoginRequest">
  <wsdl:part name="userID" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
```

output message **validateLoginResponse**

```
<wsdl:message name="validateLoginResponse">
  <wsdl:part name="validateLoginReturn"
type="xsd:boolean"/>
</wsdl:message>
```

Example 20–1 Sample Code for validateLogin

```
    public static void validateLogin() throws Exception {

        final String bipEndpoint =
"http://mycompany.com:9999/xmlpserver/services/PublicReportService?wsdl";
        final String bipNamespace =
"http://xmlns.oracle.com/oxp/service/PublicReportService";
        final String reportPath = "/HR Manager/Employee Salary Report/Employee Salary
Report.xdo";

        service = new Service();
        call = (Call) service.createCall();
        call.setTargetEndpointAddress(new URL(bipEndpoint));

        System.out.println("TESTING login Service BEGIN");
        call.setOperationName(new QName(bipNamespace, "validateLogin"));
        call.addParameter("userID", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("password", XMLType.XSD_STRING, ParameterMode.IN);
        call.setReturnType(XMLType.XSD_BOOLEAN);

        // issue the request
        Boolean valid = (Boolean) call.invoke(
            new Object[] {"administrator", "Administrator"});

        if (valid) { System.out.println("user valid"); }
        else { System.out.println("user invalid");  }
        System.out.println("Success for validateLogin");
    }
```

hasReportAccess

Use hasReportAccess to validate that a UserID and Password have the privilege to access a specific report.

input message **hasReportAccessRequest**

```
<wsdl:message name="hasReportAccessRequest">
  <wsdl:part name="reportAbsolutePath" type="xsd:string"/>
  <wsdl:part name="userID" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
```

output message **hasReportAccessResponse**

```
<wsdl:message name="hasReportAccessResponse">
  <wsdl:part name="hasReportAccessReturn"
type="xsd:boolean"/>
</wsdl:message>
```

Possible exceptions:

`InvalidParametersException`

See Section 20.3.5, "Messages for Errors" for descriptions of the possible exceptions.

Example 20–2 Sample Code for hasReportAccess

```
    public static void hasReportAccess(String reportPath) throws Exception {

        final String bipEndpoint =
"http://mycompany.com:9999/xmlpserver/services/PublicReportService?wsdl";
        //final String bipEndpoint =
"http://12.123.8232.107/xmlpserver/services/PublicReportService?wsdl";
        final String bipNamespace =
"http://xmlns.oracle.com/oxp/service/PublicReportService";

        service = new Service();
        call = (Call) service.createCall();
        call.setTargetEndpointAddress(new URL(bipEndpoint));

        System.out.println("TESTING hasReportAccess BEGIN");

        call.setOperationName(new QName(bipNamespace, "hasReportAccess"));
        call.addParameter("reportAbsolutePath", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("userID", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("password", XMLType.XSD_STRING, ParameterMode.IN);
        call.setReturnType(XMLType.XSD_BOOLEAN);

        // issue the request
        Boolean valid = (Boolean) call.invoke(
            new Object[] {reportPath, "administrator", "Administrator"});

        if (valid==true) { System.out.println("User has access."); }
        else { System.out.println("User does not have access");  }
        System.out.println("Success for AccessLogin");
    }
```

20.3.3 Operations to Get Information About Reports and the Repository

getFolderContents

Use getFolderContents to get all of the items in a folder. This will return all the reports and folders contained in the specified folder. You can then use these items to determine what reports you might want to execute or what folders you may want to further search to identify a report.

input message **getFolderContentsRequest**

```
<wsdl:message name="getFolderContentsRequest">
  <wsdl:part name="folderAbsolutePath" type="xsd:string"/>
  <wsdl:part name="userID" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
```

output message **getFolderContentsResponse**

```
<wsdl:message name="getFolderContentsResponse">
  <wsdl:part name="getFolderContentsReturn"
type="impl:ArrayOfItemData"/>
</wsdl:message>
```

Possible exceptions:

```
InvalidParametersException
AccessDeniedException
OperationFailedException
```

See Section 20.3.5, "Messages for Errors" for descriptions of the possible exceptions.

Example 20–3 Sample Code for getFolderContents

```
    public static void getFolderContent(String reportPath) throws Exception {

        final String bipEndpoint =
"http://mycompany.com:9999/xmlpserver/services/PublicReportService?wsdl";
        final String bipNamespace =
"http://xmlns.oracle.com/oxp/service/PublicReportService";

        service = new Service();
        call = (Call) service.createCall();
        call.setTargetEndpointAddress(new URL(bipEndpoint));

        System.out.println("TESTING getFolderContent Service BEGIN");
        // register the ReportDefinition class
        QName reportDef = new QName(bipNamespace, "ReportDefinition");
        call.registerTypeMapping(ReportDefinition.class, reportDef,
                    BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // register the ParamNameValue class
        QName nmvals = new QName(bipNamespace, "ParamNameValue");
        call.registerTypeMapping(ParamNameValue.class, nmvals, BeanSerializerFactory.class,
BeanDeserializerFactory.class);

        call.setOperationName(new QName(bipNamespace, "getFolderContents"));
        call.addParameter("reportAbsolutePath", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("userID", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("password", XMLType.XSD_STRING, ParameterMode.IN);
        call.setReturnClass(ItemData [].class);

        // issue the request
        ItemData items[] =  (ItemData []) call.invoke(
            new Object[] { reportPath, "administrator", "Administrator"});

        for (int i=0; i<items.length; i++) {
            System.out.println(items[i].getDisplayName());
        }
        System.out.println("Success for GetFolderContent");
    }
```

getReportDefinition

Use getReportDefinition to get information about a report such as default template, output type and a listing of template IDs. Once you have a list of template IDs you may wish to generate a report with a template other than the default.

input message **getReportDefinitionRequest**

```
<wsdl:message name="getReportDefinitionRequest">
  <wsdl:part name="reportAbsolutePath" type="xsd:string"/>
  <wsdl:part name="userID" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
```

output message **getReportDefinitionResponse**

```
<wsdl:message name="getReportDefinitionResponse">
  <wsdl:part name="getReportDefinitionReturn"
type="impl:ReportDefinition"/>
</wsdl:message>
```

Possible exceptions:

```
InvalidParametersException
AccessDeniedException
OperationFailedException
```

See Section 20.3.5, "Messages for Errors" for descriptions of the possible exceptions.

Example 20–4 Sample Code for getReportDefinition

```
    public static void getReportDef(String xdofile) throws Exception {

        final String bipEndpoint =
"http://mycompany.com:9999/xmlpserver/services/PublicReportService?wsdl";
        final String bipNamespace =
"http://xmlns.oracle.com/oxp/service/PublicReportService";

        service = new Service();
        call = (Call) service.createCall();
        call.setTargetEndpointAddress(new URL(bipEndpoint));

        System.out.println("BEGIN TESTING getReportDefinition");

        // register the ReportDefinition class
        QName reportDef = new QName(bipNamespace, "ReportDefinition");
        call.registerTypeMapping(ReportDefinition.class, reportDef,
                        BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // register the ParamNameValue class
        QName nmvals = new QName(bipNamespace, "ParamNameValue");
        call.registerTypeMapping(ParamNameValue.class, nmvals, BeanSerializerFactory.class,
BeanDeserializerFactory.class);

        call.setOperationName(new QName(bipNamespace, "getReportDefinition"));
        call.addParameter("reportAbsolutePath", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("userID", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("password", XMLType.XSD_STRING, ParameterMode.IN);
        call.setReturnClass(ReportDefinition.class);

        // issue the request
        ReportDefinition reportDefn = (ReportDefinition) call.invoke(
            new Object[] { xdofile, "administrator", "Administrator"});

        System.out.println("Report Definition Returns with \n Default Output Format = " +
reportDefn.getDefaultOutputFormat());
        ParamNameValue params [] = reportDefn.getReportParameterNameValues();

        if (params != null) {
            for (int i = 0; i < params.length; i++) {
                System.out.print("Parameter " + params[i].getName() + ":");
                if (params[i].getValues() != null) {
                    for (int j = 0;  j < params[i].getValues().length; j++)
                        System.out.print(" " + params[i].getValues()[j]);
                } else
                    System.out.print(" null");
                System.out.println(" - multiple values? " +
params[i].isMultiValuesAllowed());
            }
        }
        System.out.println("END TESTING getReportDefinition");
    }
```

getReportParameters

Use getReportParameters to get an array of report parameters and their default values. Once you have the list of parameters you can set parameter values before running or scheduling a report.

input message **getReportParametersRequest**

```
<wsdl:message name="getReportParametersRequest">
  <wsdl:part name="reportRequest"
type="impl:ReportRequest"/>
  <wsdl:part name="userID" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
```

output message **getReportParametersResponse**

```
<wsdl:message name="getReportParametersResponse">
  <wsdl:part name="getReportParametersReturn"
    type="impl:ArrayOfParamNameValue"/>
</wsdl:message>
```

Possible exceptions:

```
InvalidParametersException
AccessDeniedException
OperationFailedException
```

See Section 20.3.5, "Messages for Errors" for descriptions of the possible exceptions.

Example 20–5 Sample Code for getReportParameters

```
    public static void getParameters(String xdofile) throws Exception {
        //final String bipEndpoint =
"http://10.138.232.107/xmlpserver/services/PublicReportService?wsdl";
        final String bipEndpoint =
"http://mycompany.com:9999/xmlpserver/services/PublicReportService?wsdl";
        final String bipNamespace =
"http://xmlns.oracle.com/oxp/service/PublicReportService";

        service = new Service();
        call = (Call) service.createCall();
        call.setTargetEndpointAddress(new URL(bipEndpoint));
        System.out.println("BEGIN TESTING: getParameters Service");

        // register the ReportRequest class
        QName reportReq = new QName(bipNamespace, "ReportRequest");
        call.registerTypeMapping(ReportRequest.class, reportReq,
                        BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // register the ParamNameValue class
        QName nmvals = new QName(bipNamespace, "ParamNameValue");
        call.registerTypeMapping(ParamNameValue.class, nmvals, BeanSerializerFactory.class,
BeanDeserializerFactory.class);

        // Define parameters
        ParamNameValue[] paramNmVals = new ParamNameValue[2];
        paramNmVals[0] = new ParamNameValue(false, false, "dept", null) ;
        paramNmVals[1] = new ParamNameValue(false, false, "emp", new String[] {"*"}) ;
        ReportRequest req = new ReportRequest("pdf", "en-US", "Simple", paramNmVals,
xdofile);


        call.setOperationName(new QName(bipNamespace, "getReportParameters"));
        call.addParameter( "ReportRequest", reportReq, ParameterMode.IN );
        call.addParameter("userID", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("password", XMLType.XSD_STRING, ParameterMode.IN);
        call.setReturnClass(ParamNameValue [].class);

        // issue the request
        ParamNameValue params [] = (ParamNameValue []) call.invoke(
            new Object[] { req, "administrator", "Administrator"});

        if (params != null) {
            for (int i = 0; i < params.length; i++) {
                System.out.print("Parameter " + params[i].getName() + ":");
                if (params[i].getValues() != null) {
                    for (int j = 0;  j < params[i].getValues().length; j++)
                        System.out.print(" " + params[i].getValues()[j]);
                } else
                    System.out.print(" null");
                System.out.println(" - multiple values? " +
params[i].isMultiValuesAllowed());
            }
        }
        System.out.println("END TESTING: getParameters");
    }
```

20.3.4 Operations to Run and Schedule Reports

runReport

Use runReport to execute a report and return a generated document. Note that the document is returned in the specified file format.

input message **runReportRequest**

```
<wsdl:message name="runReportRequest">
  <wsdl:part name="reportRequest"
type="impl:ReportRequest"/>
  <wsdl:part name="userID" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
```

output message **runReportResponse**

```
<wsdl:message name="runReportResponse">
  <wsdl:part name="runReportReturn"
type="impl:ReportResponse"/>
</wsdl:message>
```

Possible exceptions:

```
InvalidParametersException
AccessDeniedException
OperationFailedException
```

See Section 20.3.5, "Messages for Errors" for descriptions of the possible exceptions.

Example 20–6 Sample Code for runReport

```
    public static void runReport() throws Exception {

        final String bipEndpoint =
          "http://mycompany.com:9999/xmlpserver/services/PublicReportService?wsdl";
        final String bipNamespace =
"http://xmlns.oracle.com/oxp/service/PublicReportService";
        final String xdofile ="/HR Manager/Employee Salary Report/Employee Salary
Report.xdo";

        // set up the call object
        service = new Service();
        call = (Call) service.createCall();
        call.setTargetEndpointAddress(new URL(bipEndpoint));

        //  TEST Run Report
        System.out.println("TESTING runReport Service BEGIN");

        // register the ReportRequest class
        QName reportReq = new QName(bipNamespace, "ReportRequest");
        call.registerTypeMapping(ReportRequest.class, reportReq,
                        BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // register the ParamNameValue class
        QName nmvals = new QName(bipNamespace, "ParamNameValue");
        call.registerTypeMapping(ParamNameValue.class, nmvals, BeanSerializerFactory.class,
BeanDeserializerFactory.class);

        // register the BIPReportResponse class
        QName reportRespqn = new QName(bipNamespace, "ReportResponse");
        call.registerTypeMapping(ReportResponse.class, reportRespqn,
          BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // Default return type based on what we expect
        call.setOperationName( new QName(bipNamespace, "runReport" ));
        call.addParameter( "ReportRequest", reportReq, ParameterMode.IN );
        call.addParameter("userID", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("password", XMLType.XSD_STRING, ParameterMode.IN);
        call.setReturnClass(ReportResponse.class);
        ParamNameValue[] paramNmVals = new ParamNameValue[2];
        paramNmVals[0] = new ParamNameValue(false, false, "dept", null) ;
        paramNmVals[1] = new ParamNameValue(false, false, "emp", new String[] {"*"}) ;

        ReportRequest req = new ReportRequest("pdf", "en-US", "Simple", paramNmVals,
xdofile);

        // issue the reques
        ReportResponse reportResp = (ReportResponse) call.invoke(
            new Object[] {req, "Administrator", "Administrator"} );
        System.out.println("Return ReportResponse: ContentType = " +
reportResp.getReportContentType());

        // Save the report
        byte[] binaryBytes = reportResp.getReportBytes();
        OutputStream out = new FileOutputStream("D:\\temp\\out.pdf");
        out.write(binaryBytes);
        out.close();
        System.out.println("Success for Run Report");
    }
```

scheduleReport

Use scheduleReport to schedule a report for execution and delivery to either printer, fax, email, WebDAV, ftp or simply save in the report repository. Jobs can be scheduled to run immediately, once, or on a recurring pattern and can have an end date to stop the recurrence. This operation returns JobID upon successfully scheduling the report job.

input message **scheduleReportRequest**

```
<wsdl:message name="scheduleReportRequest">
  <wsdl:part name="scheduleRequest"
type="impl:ScheduleRequest"/>
  <wsdl:part name="userID" type="xsd:string"/>
  <wsdl:part name="password" type="xsd:string"/>
</wsdl:message>
```

output message **scheduleReportResponse**

```
<wsdl:message name="scheduleReportResponse">
  <wsdl:part name="scheduleReportReturn"
type="xsd:string"/>
</wsdl:message>
```

Possible exceptions:

```
InvalidParametersException
AccessDeniedException
OperationFailedException
```

See Section 20.3.5, "Messages for Errors" for descriptions of the possible exceptions.

Example 20–7 Sample Code for scheduleReport

```
    public static void scheduleReport() throws Exception {

        final String bipEndpoint =
"http://mycompany.com:9999/xmlpserver/services/PublicReportService?wsdl";
        final String bipNamespace =
"http://xmlns.oracle.com/oxp/service/PublicReportService";
        final String xdofile ="/HR Manager/Employee Salary Report/Employee Salary
Report.xdo";
```

```
        // set up the call object
        service = new Service();
        call = (Call) service.createCall();
        call.setTargetEndpointAddress(new URL(bipEndpoint));

        //  TEST Run Report
        System.out.println("TESTING scheduleReport Service BEGIN");

        // register the ReportRequest class
        QName reportReq = new QName(bipNamespace, "ReportRequest");
        call.registerTypeMapping(ReportRequest.class, reportReq,
                        BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // register the ParamNameValue class
        QName nmvals = new QName(bipNamespace, "ParamNameValue");
        call.registerTypeMapping(ParamNameValue.class, nmvals, BeanSerializerFactory.class,
BeanDeserializerFactory.class);

        // register the BIPReportResponse class (part of Schedule Request)
        QName reportRespqn = new QName(bipNamespace, "ReportResponse");
        call.registerTypeMapping(ReportResponse.class, reportRespqn,
BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // register the Schedule Request class
        QName scheduleReq = new QName(bipNamespace, "ScheduleRequest");
        call.registerTypeMapping(ReportRequest.class, reportReq,
                        BeanSerializerFactory.class, BeanDeserializerFactory.class);

        // Default return type based on what we expect
        call.setOperationName( new QName(bipNamespace, "scheduleReport" ));
        call.addParameter( "ScheduleRequest", scheduleReq, ParameterMode.IN );
        call.addParameter("userID", XMLType.XSD_STRING, ParameterMode.IN);
        call.addParameter("password", XMLType.XSD_STRING, ParameterMode.IN);
        call.setReturnType(XMLType.XSD_STRING);

        // Define Report Request Object
        ParamNameValue[] paramNmVals = new ParamNameValue[2];
        paramNmVals[0] = new ParamNameValue(false, false, "dept", null) ;
        paramNmVals[1] = new ParamNameValue(false, false, "emp", new String[] {"*"}) ;
        ReportRequest report = new ReportRequest("pdf", "en-US", "Simple", paramNmVals,
xdofile);

        // Define Email Delivery Options
        EMailDeliveryOption email =
            new EMailDeliveryOption("manager@mycompany.com", "Body", // bcc, boddy
            "manager@mycompany.com", // cc
            "bipublisher@oracle.com", "bipublisher@oracle.com", // from, reply to
            "Test Email Subject", "manager@mycompany.com" // subject, to
            );

        // Define only email delivery
        DeliveryRequest delivery = new DeliveryRequest(email, null, null, null, null, null);

        ScheduleRequest req = new ScheduleRequest(
            delivery,               // delivery options
            null,                   // enddate (Calendar.getInstance())
            null,                   // job calendar
            "en-US", null,          // job locale, timezone
            null, null, null, null, // notifications
            1, 1,                   // repeat count, repeat interval
            report,                 // Report and Parameters
            "true", "true",         // Save Data, Save Output
            "false", "true",        // Bursting, Public
            null,                   // start date
            "true",                 // Use UTF-8
            "my Job4"               // job name
            );

        // issue the request
        String result = (String) call.invoke(
            new Object[] {req, "Administrator", "Administrator"} );
        System.out.println("Success for Schedule Report");
    }
```

20.3.5 Messages for Errors

One of the following messages may be returned to any of the operations if there is an error that occurs in the execution of the operation.

OperationFailedException

```
<wsdl:message name="OperationFailedException">
  <wsdl:part name="fault" type="impl:OperationFailedException"/>
</wsdl:message>
```

AccessDeniedException

```
<wsdl:message name="AccessDeniedException">
  <wsdl:part name="fault" type="impl:AccessDeniedException"/>
</wsdl:message>
```

InvalidParametersException

```
<wsdl:message name="InvalidParametersException">
  <wsdl:part name="fault" type="impl:InvalidParametersException"/>
</wsdl:message>
```

20.4 Debugging Web Service Applications

As a Web service developer you may need to see the SOAP request messages being used to invoke Web services along with the SOAP response to those request messages. Oracle BI Publisher bundles the Apache Axis SOAP Monitor utility to monitor the SOAP message flow without requiring any special configuration or restarting of the server.

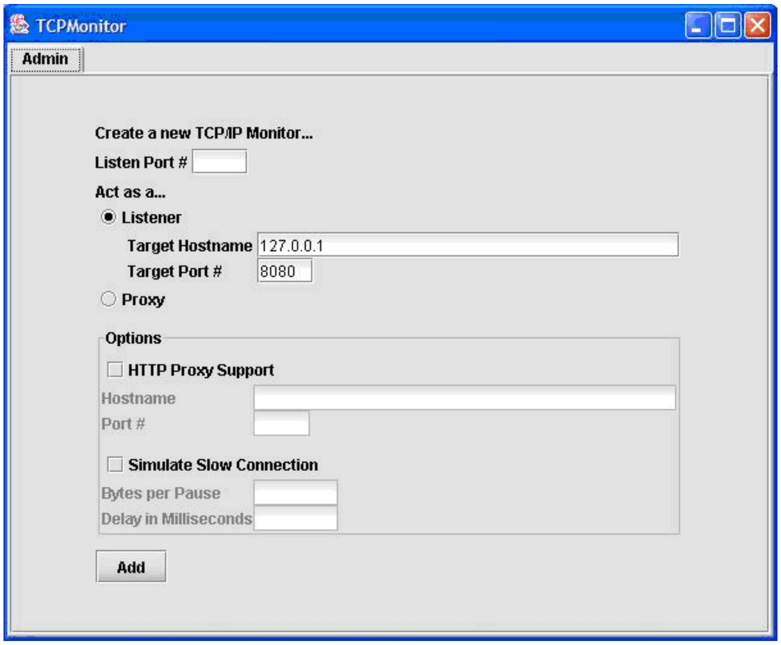
The client sample code sends a SOAP request to listening port 8888, and then takes advantage of Axis TCP monitor to route SOAP request to port 8080, where the PublicReportService is running. This will be a useful debugging tool for the Web service developer.

To start Axis TCP Monitor (tcpmon), from the command line enter the following:

```
% java org.apache.axis.utils.tcpmon
```

This launches the TCP monitor window. A sample of the window without any of the optional arguments is shown in the following figure:

Figure 20–1 TCPMonitor Window



Create a new TCP/IP Monitor...

Listen Port #

Act as a...

● Listener

Target Hostname 127.0.0.1

Target Port # 8080

○ Proxy

Options

□ HTTP Proxy Support

Hostname

Port #

□ Simulate Slow Connection

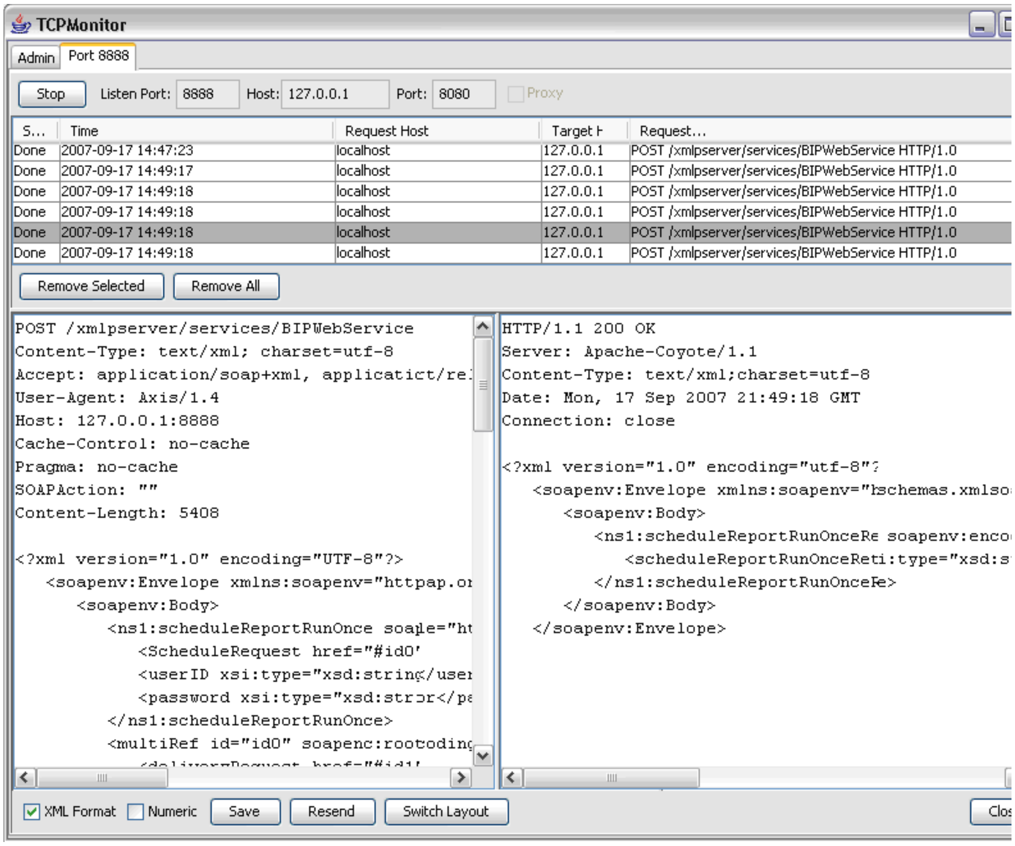
Bytes per Pause

Delay in Milliseconds

To use the program, select a local port that tcpmon will monitor for incoming connections, a target host where it will forward such connections, and the port number on the target machine which should be "tunneled" to. Then click Add. Another tab will then display for your new tunneled connection. From there, you will see the SOAP Request and Response message, which facilitates debugging.

The following figure shows the connection tab of the TCPMonitor window:

Figure 20–2 TCP Monitor Window Connection Tab



Listen Port: 8888 Host: 127.0.0.1 Port: 8080

S...	Time	Request Host	Target H	Request...
Done	2007-09-17 14:47:23	localhost	127.0.0.1	POST /xmlpserver/services/BIPWebService HTTP/1.0
Done	2007-09-17 14:49:17	localhost	127.0.0.1	POST /xmlpserver/services/BIPWebService HTTP/1.0
Done	2007-09-17 14:49:18	localhost	127.0.0.1	POST /xmlpserver/services/BIPWebService HTTP/1.0
Done	2007-09-17 14:49:18	localhost	127.0.0.1	POST /xmlpserver/services/BIPWebService HTTP/1.0
Done	2007-09-17 14:49:18	localhost	127.0.0.1	POST /xmlpserver/services/BIPWebService HTTP/1.0
Done	2007-09-17 14:49:18	localhost	127.0.0.1	POST /xmlpserver/services/BIPWebService HTTP/1.0

```
POST /xmlpserver/services/BIPWebService
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, applicatict/re
User-Agent: Axis/1.4
Host: 127.0.0.1:8888
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 5408

<?xml version="1.0" encoding="UTF-8"?>
   <soapenv:Envelope xmlns:soapenv="httpap.or
      <soapenv:Body>
         <ns1:scheduleReportRunOnce soaple="ht
            <ScheduleRequest href="#id0'
            <userID xsi:type="xsd:string/user
            <password xsi:type="xsd:stror</pa
         </ns1:scheduleReportRunOnce>
         <multiRef id="id0" soapenc:rootoding
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Date: Mon, 17 Sep 2007 21:49:18 GMT
Connection: close

<?xml version="1.0" encoding="utf-8"?
   <soapenv:Envelope xmlns:soapenv="Eschemas.xmlso
      <soapenv:Body>
         <ns1:scheduleReportRunOnceRe soapenv:enco
            <scheduleReportRunOnceReti:type="xsd:s
         </ns1:scheduleReportRunOnceRe>
      </soapenv:Body>
   </soapenv:Envelope>
```

To run tcpmonitor, enter the following:

```
set CLASSPATH=.
set CLASSPATH=%CLASSPATH%;%WEB_HOME%\PORTAL\WEB-INF\lib\axis.jar
set TCPMON_PORT=8081
if not %1.==. TCPMON_PORT=%1
java org.apache.axis.utils.tcpmon %TCPMON_PORT% %ADMINSERVER_HOSTNAME%
%ADMINSERVER_PORT%
```

20.5 Oracle BI Publisher Web Services WSDL

Following is the full BI Publisher PublicReportService Web Services WSDL. It is provided for those instances where BI Publisher has been deployed to a J2EE server that is not able to display this WSDL.

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
targetNamespace="http://xmlns.oracle.com/oxp/service/PublicReportService"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://xmlns.oracle.com/oxp/service/PublicReportService"
xmlns:intf="http://xmlns.oracle.com/oxp/service/PublicReportService"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <!-- WSDL created by Apache Axis version: 1.3
Built on Oct 05, 2005 (05:23:37 EDT)
  -->
```

```
- <wsdl:types>
- <schema
targetNamespace="http://xmlns.oracle.com/oxp/service/PublicReportService"
xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
- <complexType name="ArrayOf_xsd_string">
- <complexContent>
- <restriction base="soapenc:Array">
  <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]" />
  </restriction>
  </complexContent>
  </complexType>
- <complexType name="ParamNameValue">
- <sequence>
  <element name="isMultiValuesAllowed" type="xsd:boolean" />
  <element name="name" nillable="true" type="xsd:string" />
  <element name="values" nillable="true" type="impl:ArrayOf_xsd_string" />
  </sequence>
  </complexType>
- <complexType name="ArrayOfParamNameValue">
- <complexContent>
- <restriction base="soapenc:Array">
  <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:ParamNameValue[]" />
  </restriction>
  </complexContent>
  </complexType>
- <complexType name="ReportRequest">
- <sequence>
  <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="parameterNameValues" nillable="true"
type="impl:ArrayOfParamNameValue" />
  <element name="reportAbsolutePath" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
- <complexType name="AccessDeniedException">
  <sequence />
  </complexType>
- <complexType name="OperationFailedException">
  <sequence />
  </complexType>
- <complexType name="InvalidParametersException">
  <sequence />
  </complexType>
- <complexType name="ReportResponse">
- <sequence>
  <element name="reportBytes" nillable="true" type="xsd:base64Binary" />
  <element name="reportContentType" nillable="true" type="xsd:string" />
  <element name="reportLocale" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
- <complexType name="ReportDefinition">
- <sequence>
  <element name="defaultOutputFormat" nillable="true" type="xsd:string" />
  <element name="defaultTemplateId" nillable="true" type="xsd:string" />
  <element name="parameterNames" nillable="true" type="impl:ArrayOf_xsd_string" />
  <element name="reportParameterNameValues" nillable="true"
type="impl:ArrayOfParamNameValue" />
  <element name="templateIds" nillable="true" type="impl:ArrayOf_xsd_string" />
  </sequence>
  </complexType>
- <complexType name="ItemData">
- <sequence>
  <element name="absolutePath" nillable="true" type="xsd:string" />
  <element name="creationDate" nillable="true" type="xsd:dateTime" />
  <element name="displayName" nillable="true" type="xsd:string" />
  <element name="fileName" nillable="true" type="xsd:string" />
  <element name="lastModified" nillable="true" type="xsd:dateTime" />
  <element name="lastModifier" nillable="true" type="xsd:string" />
  <element name="owner" nillable="true" type="xsd:string" />
  <element name="parentAbsolutePath" nillable="true" type="xsd:string" />
  <element name="type" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
- <complexType name="ArrayOfItemData">
- <complexContent>
- <restriction base="soapenc:Array">
  <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:ItemData[]" />
  </restriction>
  </complexContent>
  </complexType>
- <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="emailSubject" nillable="true" type="xsd:string" />
  <element name="emailTo" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
- <complexType name="FaxDeliveryOption">
- <sequence>
  <element name="faxNumber" nillable="true" type="xsd:string" />
  <element name="faxServer" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
- <complexType name="FTPDeliveryOption">
- <sequence>
  <element name="ftpServerName" nillable="true" type="xsd:string" />
  <element name="ftpUserName" nillable="true" type="xsd:string" />
  <element name="ftpUserPassword" nillable="true" type="xsd:string" />
  <element name="remoteFile" nillable="true" type="xsd:string" />
  <element name="sftpOption" type="xsd:boolean" />
  </sequence>
  </complexType>
- <complexType name="LocalDeliveryOption">
- <sequence>
  <element name="destination" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
- <complexType name="PrintDeliveryOption">
- <sequence>
  <element name="printNumberOfCopy" nillable="true" type="xsd:string" />
  <element name="printRange" nillable="true" type="xsd:string" />
  <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>
- <complexType name="WebDAVDeliveryOption">
- <sequence>
  <element name="deliveryAuthType" nillable="true" type="xsd:string" />
  <element name="deliveryAuthTypeBasic" nillable="true" type="xsd:string" />
  <element name="deliveryAuthTypeDigest" nillable="true" type="xsd:string" />
  <element name="password" nillable="true" type="xsd:string" />
  <element name="remoteFilePath" nillable="true" type="xsd:string" />
  <element name="server" nillable="true" type="xsd:string" />
  <element name="userName" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
- <complexType name="DeliveryRequest">
- <sequence>
  <element name="emailOption" nillable="true" type="impl:EMailDeliveryOption" />
  <element name="faxOption" nillable="true" type="impl:FaxDeliveryOption" />
  <element name="ftpOption" nillable="true" type="impl:FTPDeliveryOption" />
  <element name="localOption" nillable="true" type="impl:LocalDeliveryOption" />
  <element name="printOption" nillable="true" type="impl:PrintDeliveryOption" />
  <element name="webDAVOption" nillable="true" type="impl:WebDAVDeliveryOption" />
  </sequence>
  </complexType>
- <complexType name="ScheduleRequest">
- <sequence>
  <element name="deliveryRequest" nillable="true" type="impl:DeliveryRequest" />
  <element name="endDate" nillable="true" type="xsd:dateTime" />
  <element name="jobCalendar" nillable="true" type="xsd:string" />
  <element name="jobLocale" nillable="true" type="xsd:string" />
  <element name="jobTZ" nillable="true" type="xsd:string" />
  <element name="notificationTo" nillable="true" type="xsd:string" />
  <element name="notifyWhenFailed" type="xsd:boolean" />
  <element name="notifyWhenSuccess" type="xsd:boolean" />
  <element name="notifyWhenWarning" type="xsd:boolean" />
  <element name="repeatCount" type="xsd:int" />
  <element name="repeatInterval" type="xsd:int" />
  <element name="reportRequest" nillable="true" type="impl:ReportRequest" />
  <element name="saveDataOption" type="xsd:boolean" />
  <element name="saveOutputOption" type="xsd:boolean" />
  <element name="scheduleBurstringOption" type="xsd:boolean" />
  <element name="schedulePublicOption" type="xsd:boolean" />
  <element name="startDate" nillable="true" type="xsd:dateTime" />
  <element name="useUTF8Option" type="xsd:boolean" />
  <element name="userJobName" nillable="true" type="xsd:string" />
  </sequence>
  </complexType>
  </schema>
  </wsdl:types>
- <wsdl:message name="getReportParametersResponse">
  <wsdl:part name="getReportParametersReturn" type="impl:ArrayOfParamNameValue" />
  </wsdl:message>
- <wsdl:message name="getReportDefinitionResponse">
  <wsdl:part name="getReportDefinitionReturn" type="impl:ReportDefinition" />
  </wsdl:message>
- <wsdl:message name="getReportParametersRequest">
  <wsdl:part name="reportRequest" type="impl:ReportRequest" />
  <wsdl:part name="userID" type="xsd:string" />
  <wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
- <wsdl:message name="hasReportAccessResponse">
```

```
  <wsdl:part name="hasReportAccessReturn" type="xsd:boolean" />
  </wsdl:message>
- <wsdl:message name="AccessDeniedException">
  <wsdl:part name="fault" type="impl:AccessDeniedException" />
  </wsdl:message>
- <wsdl:message name="validateLoginResponse">
  <wsdl:part name="validateLoginReturn" type="xsd:boolean" />
  </wsdl:message>
- <wsdl:message name="scheduleReportResponse">
  <wsdl:part name="scheduleReportReturn" type="xsd:string" />
  </wsdl:message>
- <wsdl:message name="runReportResponse">
  <wsdl:part name="runReportReturn" type="impl:ReportResponse" />
  </wsdl:message>
- <wsdl:message name="OperationFailedException">
  <wsdl:part name="fault" type="impl:OperationFailedException" />
  </wsdl:message>
- <wsdl:message name="getFolderContentsRequest">
  <wsdl:part name="folderAbsolutePath" type="xsd:string" />
  <wsdl:part name="userID" type="xsd:string" />
  <wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
- <wsdl:message name="hasReportAccessRequest">
  <wsdl:part name="reportAbsolutePath" type="xsd:string" />
  <wsdl:part name="userID" type="xsd:string" />
  <wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
- <wsdl:message name="getReportDefinitionRequest">
  <wsdl:part name="reportAbsolutePath" type="xsd:string" />
  <wsdl:part name="userID" type="xsd:string" />
  <wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
- <wsdl:message name="getFolderContentsResponse">
  <wsdl:part name="getFolderContentsReturn" type="impl:ArrayOfItemData" />
  </wsdl:message>
- <wsdl:message name="InvalidParametersException">
  <wsdl:part name="fault" type="impl:InvalidParametersException" />
  </wsdl:message>
- <wsdl:message name="scheduleReportRequest">
  <wsdl:part name="scheduleRequest" type="impl:ScheduleRequest" />
  <wsdl:part name="userID" type="xsd:string" />
  <wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
- <wsdl:message name="runReportRequest">
  <wsdl:part name="reportRequest" type="impl:ReportRequest" />
  <wsdl:part name="userID" type="xsd:string" />
  <wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
- <wsdl:message name="validateLoginRequest">
  <wsdl:part name="userID" type="xsd:string" />
  <wsdl:part name="password" type="xsd:string" />
  </wsdl:message>
- <wsdl:portType name="PublicReportService">
- <wsdl:operation name="getReportParameters" parameterOrder="reportRequest userID
password">
  <wsdl:input message="impl:getReportParametersRequest"
name="getReportParametersRequest" />
  <wsdl:output message="impl:getReportParametersResponse"
name="getReportParametersResponse" />
  <wsdl:fault message="impl:AccessDeniedException" name="AccessDeniedException" />
  <wsdl:fault message="impl:OperationFailedException"
name="OperationFailedException" />
  <wsdl:fault message="impl:InvalidParametersException"
name="InvalidParametersException" />
  </wsdl:operation>
- <wsdl:operation name="validateLogin" parameterOrder="userID password">
  <wsdl:input message="impl:validateLoginRequest" name="validateLoginRequest" />
  <wsdl:output message="impl:validateLoginResponse" name="validateLoginResponse"
/>
  </wsdl:operation>
- <wsdl:operation name="runReport" parameterOrder="reportRequest userID password">
  <wsdl:input message="impl:runReportRequest" name="runReportRequest" />
  <wsdl:output message="impl:runReportResponse" name="runReportResponse" />
  <wsdl:fault message="impl:AccessDeniedException" name="AccessDeniedException" />
  <wsdl:fault message="impl:OperationFailedException"
name="OperationFailedException" />
  <wsdl:fault message="impl:InvalidParametersException"
name="InvalidParametersException" />
  </wsdl:operation>
- <wsdl:operation name="getReportDefinition" parameterOrder="reportAbsolutePath
userID password">
  <wsdl:input message="impl:getReportDefinitionRequest"
name="getReportDefinitionRequest" />
  <wsdl:output message="impl:getReportDefinitionResponse"
name="getReportDefinitionResponse" />
  <wsdl:fault message="impl:AccessDeniedException" name="AccessDeniedException" />
  <wsdl:fault message="impl:OperationFailedException"
name="OperationFailedException" />
  <wsdl:fault message="impl:InvalidParametersException"
name="InvalidParametersException" />
  </wsdl:operation>
- <wsdl:operation name="getFolderContents" parameterOrder="folderAbsolutePath
userID password">
  <wsdl:input message="impl:getFolderContentsRequest"
name="getFolderContentsRequest" />
  <wsdl:output message="impl:getFolderContentsResponse"
name="getFolderContentsResponse" />
  <wsdl:fault message="impl:AccessDeniedException" name="AccessDeniedException" />
  <wsdl:fault message="impl:OperationFailedException"
name="OperationFailedException" />
  <wsdl:fault message="impl:InvalidParametersException"
name="InvalidParametersException" />
  </wsdl:operation>
- <wsdl:operation name="scheduleReport" parameterOrder="scheduleRequest userID
password">
  <wsdl:input message="impl:scheduleReportRequest" name="scheduleReportRequest" />
  <wsdl:output message="impl:scheduleReportResponse" name="scheduleReportResponse"
/>
  <wsdl:fault message="impl:AccessDeniedException" name="AccessDeniedException" />
  <wsdl:fault message="impl:OperationFailedException"
name="OperationFailedException" />
  <wsdl:fault message="impl:InvalidParametersException"
name="InvalidParametersException" />
  </wsdl:operation>
- <wsdl:operation name="hasReportAccess" parameterOrder="reportAbsolutePath userID
password">
  <wsdl:input message="impl:hasReportAccessRequest" name="hasReportAccessRequest"
/>
  <wsdl:output message="impl:hasReportAccessResponse"
name="hasReportAccessResponse" />
```

```
  <wsdl:fault message="impl:InvalidParametersException"
name="InvalidParametersException" />
  </wsdl:operation>
  </wsdl:portType>
- <wsdl:binding name="PublicReportServiceSoapBinding"
type="impl:PublicReportService">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"
/>
- <wsdl:operation name="getReportParameters">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="getReportParametersRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:input>
- <wsdl:output name="getReportParametersResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:output>
- <wsdl:fault name="AccessDeniedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="AccessDeniedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="OperationFailedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="OperationFailedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="InvalidParametersException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="InvalidParametersException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
  </wsdl:operation>
- <wsdl:operation name="validateLogin">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="validateLoginRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:input>
- <wsdl:output name="validateLoginResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:output>
  </wsdl:operation>
- <wsdl:operation name="runReport">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="runReportRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:input>
- <wsdl:output name="runReportResponse">
```

```
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:output>
- <wsdl:fault name="AccessDeniedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="AccessDeniedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="OperationFailedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="OperationFailedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="InvalidParametersException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="InvalidParametersException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
  </wsdl:operation>
- <wsdl:operation name="getReportDefinition">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="getReportDefinitionRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:input>
- <wsdl:output name="getReportDefinitionResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:output>
- <wsdl:fault name="AccessDeniedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="AccessDeniedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="OperationFailedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="OperationFailedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="InvalidParametersException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="InvalidParametersException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
  </wsdl:operation>
- <wsdl:operation name="getFolderContents">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="getFolderContentsRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
```

```
  </wsdl:input>
- <wsdl:output name="getFolderContentsResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:output>
- <wsdl:fault name="AccessDeniedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="AccessDeniedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="OperationFailedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="OperationFailedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="InvalidParametersException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="InvalidParametersException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
  </wsdl:operation>
- <wsdl:operation name="scheduleReport">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="scheduleReportRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:input>
- <wsdl:output name="scheduleReportResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:output>
- <wsdl:fault name="AccessDeniedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="AccessDeniedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="OperationFailedException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="OperationFailedException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
- <wsdl:fault name="InvalidParametersException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="InvalidParametersException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
  </wsdl:operation>
- <wsdl:operation name="hasReportAccess">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="hasReportAccessRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
```

```
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:input>
- <wsdl:output name="hasReportAccessResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:output>
- <wsdl:fault name="InvalidParametersException">
  <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="InvalidParametersException"
namespace="http://xmlns.oracle.com/oxp/service/PublicReportService" use="encoded"
/>
  </wsdl:fault>
  </wsdl:operation>
  </wsdl:binding>
- <wsdl:service name="PublicReportServiceService">
- <wsdl:port binding="impl:PublicReportServiceSoapBinding"
name="PublicReportService">
  <wsdlsoap:address
location="http://mycompany.com:9999/xmlpserver/services/PublicReportService" />
  </wsdl:port>
  </wsdl:service>
  </wsdl:definitions>
```

Part III

New Features for Oracle Business Intelligence for Microsoft Office

Part III contains the following chapters:

21

Installing Oracle Business Intelligence for Microsoft Office

Oracle Business Intelligence for Microsoft Office includes the Oracle Business Intelligence Office server and the Oracle Business Intelligence Add-in for Microsoft Office client. The client includes both the Oracle BI Office Add-In for Microsoft Excel and the Oracle BI Office Add-in for Microsoft PowerPoint. This chapter describes how to install and configure the BI Office server and client.

This chapter includes the following topics:

21.1 System Requirements

Following are the required components for the BI Office client and server. For the detailed list of system requirements for Oracle Business Intelligence Enterprise Edition and for updates to this list, see *System Requirements and Supported Platforms for Oracle Business Intelligence Suite Enterprise Edition, Version 10.1.3.3*.

21.1.1 Supported Client Operating Systems

- Windows XP
- Windows Vista

21.1.2 Supported Versions of Microsoft Office

- Microsoft Office 2003
- Microsoft Office 2007

21.1.3 Supported Application Servers

- Oracle Application Server (OC4J/HTTP) 10.1.3.1 on Linux

- Oracle Application Server 10.1.3.3
- Apache Tomcat 5.5x on Linux
- BEA Weblogic 9.x
- IBM Websphere Application Server 5.1 on Windows
- IBM Websphere Application Server 6.0, 6.1
- Sun Java System Web Server 6.1, 7.0

21.2 Installing the BI Office Server

The Oracle BI EE version 10.1.3.3 installer installs the BI Office server. The installer performs the following tasks:

- Deploys the bioffice.ear file.
- Initializes values in the BI Office configuration file.
- Copies the client OracleBIOffice.exe file to the location where Presentation Services is running.
- Updates the instanceconfig.xml file with the location of the client OracleBIOffice.exe.
- Backs up the BI Office server configuration file, bioffice.xml, to a subdirectory under <OracleBI_Home>/Install_Backup/<upgrade_timestamp>.

> **Note:** During an upgrade installation this file is used to retain the existing configuration settings.

21.2.1 Deployment of the bioffice.ear

The Oracle BI Enterprise Edition installer for 10.1.3.3 installs the Oracle BI Office server into the J2EE container with the Presentation Services Plug-in.

- The basic BIEE install option deploys the bioffice.ear to the standalone OC4J home.
- The advanced BIEE install option creates a new OC4J instance called "bioffice" in Oracle Application Server and deploys the bioffice.ear to this new OC4J instance.

21.2.2 Initialization of the BI Office Configuration File

The BI Office configuration file is bioffice.xml. The location of the file depends on your installation type. For example:

- In a BIEE basic install, it is installed under <OracleBI_Home>/oc4j_bi/j2ee/home/applications/bioffice/bioffice/WEB-INF/bioffice.xml
- In a BIEE advanced install, it is installed under ORACLE_HOME/j2ee/bibioffice/applications/bioffice/bioffice/WEB-INF/bioffice.xml
- In a BIEE install with IBM WebSphere 5.1 (Windows), it is installed under C:\Program Files\WebSphere\AppServer\installedApps\leil-server\bioffice.ear\bioffice.war
- In a BI EE install with Apache Tomcat 5.5 (Windows), it is installed under C:\Program Files\<Apache_Home>\Tomcat 5.5\webapps\bioffice
- In other non-Oracle Application Servers, it is installed under the bioffice application directory, parallel to the Oracle BI analytics application.

The Oracle BI EE installer configures the BI Office server to point to the Oracle BI Presentation Services server by setting the property "SawBaseURL" in the bioffice.xml file. To configure the Oracle BI Office server to point to a different Presentation Services server, modify this property with the new Presentation Services URL. For more details about other properties in the bioffice.xml file, see Section 21.5, "Setting Properties in the bioffice.xml Configuration File".

21.2.3 Placement of the Client OracleBIOffice.exe File

The installer copies the client/OracleBIOffice.exe file to the location where Presentation Services is running and updates the instanceconfig.xml file with this location. This makes the client install executable available for download from the More Products menu in Oracle BI Interactive Dashboards and Oracle BI Answers.

For example, in an OC4J install, the OracleBIOffice.exe is installed to <OracleBI_Home>/oc4j_bi/j2ee/home/applications/analytics/analytics/client/OracleBIOffice.exe.

In an Oracle Application Server install, the OracleBIOffice.exe is installed to ORACLE_HOME/j2ee/bianalytics/applications/analytics/analytics/client/OracleBIOffice.exe.

The installer modifies the instanceconfig.xml (OracleBIData/web/config/instanceconfig.xml) by adding the new element `<BIforOfficeURL>` and initializing the property to point to the location of the client download. For example:

```
<BIforOfficeURL>client/OracleBIOffice.exe</BIforOfficeURL>
```

21.3 Deploying the BI Office Server on J2EE Application Servers

If you deployed Oracle BI Enterprise Edition on a non-Oracle J2EE Application Server, you must manually deploy the bioffice.ear or bioffice.war file. It is not included in the analytics.ear file. The bioffice application must be deployed to the same container as the analytics application.

Deploy the BI Office server:

1. From a browser, open the Web application deployment page for your J2EE application server.
2. Locate the bioffice.ear or bioffice.war under OracleBI/office/server.
3. Using the guidelines for your application server, deploy the bioffice.ear or bioffice.war file as an application named "bioffice" to your J2EE application server environment.

Make the BI Office Client Executable Available to Presentation Services (Required for 10.1.3.3 only):

> **Note:** Do not perform this step if you are installing Oracle BI EE version 10.1.3.3.1. In the 10.1.3.3.1 release, the installer automatically places the BI Office Client install executable in the deployed analytics/analytics/client directory.

Copy the <OracleBI_Home>/office/client directory to the deployed analytics/analytics/client directory.

Update Configuration Files:

1. Update the instanceconfig.xml to point to the location of the OracleBIOffice.exe file.

   a. Locate the instanceconfig.xml file in the directory <OracleBI_Home>/web/config.

   b. Add the following entry to the file:

   ```
   <BIforOfficeURL>client/OracleBIOffice.exe</BIforOfficeURL>
   ```

   > **Note:** For more information on modifying instanceconfig.xml, refer to the *Oracle Business Intelligence Presentation Services Administration Guide*.

2. Update the bioffice.xml file to point to the Presentation Services server.

   Update the property `<SawBaseURL>` with the hostname and port of the Presentation Services server. See Section 21.5, "Setting Properties in the bioffice.xml Configuration File" for the file location and more information.

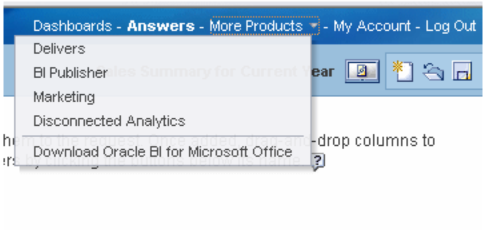
21.4 Installing the BI Office Client

After installing Oracle BI EE 10.1.3.3, the BI Office client is available for download from Oracle BI Answers and Oracle BI Interactive Dashboards.

To install the BI Office Client:

1. Log in to Oracle BI Answers or Oracle BI Interactive Dashboards with your Oracle BI EE user credentials.

2. Select the **More Products** link and then select **Download Oracle BI for Microsoft Office**.


Dashboards - Answers - More Products - My Account - Log Out
Delivers
BI Publisher
Marketing
Disconnected Analytics
Download Oracle BI for Microsoft Office
-drop columns to

3. A dialog will prompt you to run or save OracleBIOffice.exe. Save the file to your local directory.

4. Navigate to the saved location. Close all Microsoft Office applications. Double-click OracleBIOffice.exe to launch the installer.

   Microsoft .Net Framework 2.0 is required for the BI Office client. If it is not installed on your computer you will be prompted to exit and install Microsoft .Net Framework 2.0 from the Microsoft Web site.

   > **Note:** In the Oracle BI EE version 10.1.3.3, Microsoft .Net Framework 2.0 was included with the BI Office client installer. If you are installing this version, you will be prompted to install Microsoft .Net Framework 2.0 without being required to exit.

In addition, if the following prerequisites are not on your computer, you will be prompted to install them:

- Office2003-kb907417sfxcab-ENU_Patch
- Shared Add-In Extensibility Microsoft .Net 2.0
- Shared Add-In Support Update Microsoft .Net 2.0

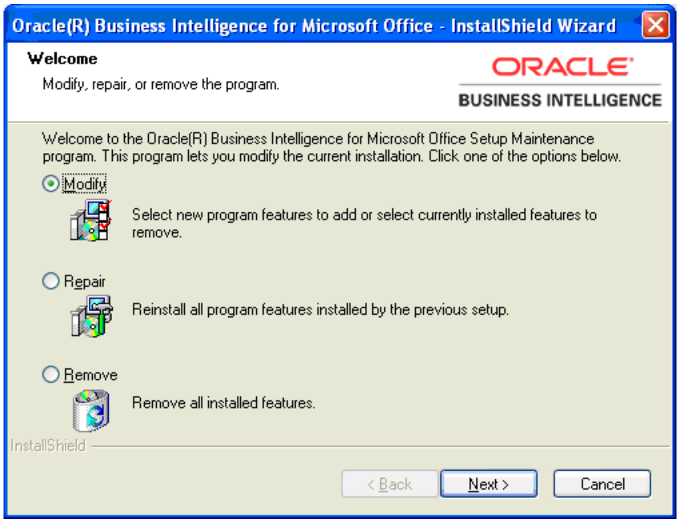
After installing the prerequisites, you may be prompted to restart your computer.

After restarting, navigate back to the location of the OracleBIOffice.exe file and double-click to reinitiate the install.

5. From the Welcome screen, select **Next**.
6. In the Customer Information screen, enter your User Name and Company Name, and select the desired sharing option.
7. On the Setup Type screen choose one of the following:
   - **Typical** — Performs installation of both add-ins to C:\Program Files\Oracle\BIOffice
   - **Custom** — Enables you to select the install directory, and choose which components to install.
8. After all components are installed, the InstallShield Wizard Complete dialog box will display. Click **Finish**.

21.4.1 Performing a Maintenance or Upgrade Install

If you have an existing installation on your machine, you will be prompted to Modify, Repair or Remove the existing installation.



Oracle(R) Business Intelligence for Microsoft Office - InstallShield Wizard

Welcome

Modify, repair, or remove the program.

Welcome to the Oracle(R) Business Intelligence for Microsoft Office Setup Maintenance program. This program lets you modify the current installation. Click one of the options below.

● Modify

Select new program features to add or select currently installed features to remove.

○ Repair

Reinstall all program features installed by the previous setup.

○ Remove

Remove all installed features.

Modify: Prompts you to select the specific Add-in you wish to install.

Repair: Uses a cached copy of the original install.

Remove: Removes the BI Office Add-Ins.

21.4.2 Configuring the Client

Once downloaded and installed, each user must enter the connection information to the Oracle BI for Microsoft Office Add-in Preferences dialog on the client computer to enable connection to Presentation Services.

To add connection information to the client computer:

1. Open Microsoft Excel or Microsoft PowerPoint. (The connection information is shared, therefore you can enter it through either application and it will be available to both.)

2. From the Oracle BI menu, select **Preferences**.

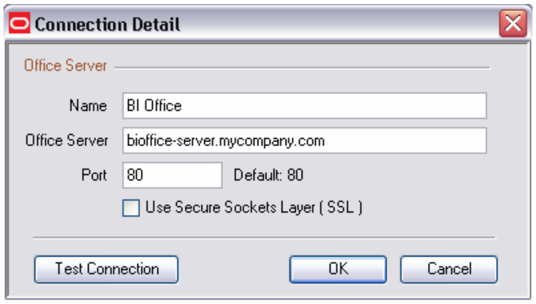
3. On the Connections tab, select **New**.

4. Enter the following fields:

   a. **Name** for the connection

   b. **Office Server** — The server URL for the Oracle BI for Microsoft Office server (for example: bioffice-server.mycompany.com)

   c. **Port** for the Oracle for BI for Microsoft Office server



Connection Detail

Office Server

Name: BI Office

Office Server: bioffice-server.mycompany.com

Port: 80 Default: 80

□ Use Secure Sockets Layer (SSL)

> **Note:** If your environment is SSL enabled, see Section 21.6, "Setting Up SSL-Enabled OC4J for BI Office" for the appropriate set up steps.

5. Click **Test Connection** to test the connection between the add-in and the BI Office server.

21.5 Setting Properties in the bioffice.xml Configuration File

This section describes the properties that you can configure in the bioffice.xml file. For file location see Section 21.2.2, "Initialization of the BI Office Configuration File".

Following is a sample bioffice.xml file:

```
<bioffice>
   <!-- log -->
   <!-- LogDir. Default is [O4CJ
dir]\j2ee\home\applications\bioffice\bioffice\WEB-INF\log -->
    <!-- <property name="LogDir">D:\BIOffice\Server\log\</property> -->
    <!-- LogLevel Never = 1; Error = 2; Warning = 3; Msg = 4; Debug = 5; -->
    <property name="LogLevel" type="int">3</property>
    <!-- saw -->
   <property name="SawBaseURL">http://localhost/analytics/saw.dll</property>
    <!-- Does SAW use SSO (Single Sign-On): yes = 1; no = 0; -->
   <property name="SawUseSSO" type="int">0</property>
    <!-- Specifies the maximum number of rows to be returned by SAW
```

```
executeXMLQuery or fetchNext method. -->
    <property name="SawMaxRowsPerPage" type="int">5000</property>
   <!-- Parse hyperlink column: turn on = 1; turn off = 0; -->
   <property name="SawParseHyperLink" type="int">1</property>
   <!-- Parse percentage column: turn on = 1; turn off = 0; -->
   <property name="SawParsePercentageColumn" type="int">1</property>
   <!-- Fetch fresh data: turn on = 1; turn off = 0; -->
    <!-- Turn on will make SAW re-submits the query to refresh data, in stead of
reading data from cache. -->
   <property name="SawFetchFreshData" type="int">0</property>
</bioffice>
```

The following table describes the bioffice.xml properties:

Property Name	Valid Values	Description
LogDir	N/A	Enter the path to the directory for the BI Office server to write the log file. Default is: <O4CJ dir>\j2ee\home\applications\bioffice\bioffice\WEB-INF\log
LogLevel	1 = Never 2 = Error 3 = Warning (Default) 4 = Message 5 = Debug	Set the level of information you wish to be written to the log file. Note that as the log level value increases, performance is impacted. Therefore do not set this property to debug (5) except when troubleshooting an issue.
SawBaseURL	http://<HOSTNAME>:<PORT>/analytics/saw.dll or https://<HOSTNAME>:<PORT>/analytics/saw.dll	Enter the URL for the Oracle BI Presentation Services. This property is set initially by the BIEE installer.
SawUseSSO	0 = No (Default) 1 = Yes	Default is 0. Set this property to 1 if your BIEE implementation is SSO-enabled. In an SSO-enabled implementation, the BI Office add-in features "Insert as Flash," "Insert as Image," and "Edit View" are not supported. Setting this option to "1" suppresses these options from the office add-in menu.
SawMaxRowsPerPage	N/A	Default setting is 5000. Specifies the maximum number of rows to be returned by SAW executeXMLQuery or fetchNext method. Increase this setting in a multiple-user environment.
SawParseHyperLink	0 = Off 1 = On (Default)	When set to 1, the BI Office server parses the HTML content in the results set, and converts the content to HTML display format in Excel. When set to 0, the BI Office server inserts the hyperlink text as is., that is, as text with markup tags. If your data does not include any hyperlink text, set this property to 0 for better performance.

Property Name	Valid Values	Description
SawParsePercentageColumn	0 = Off 1 = On (Default)	In Oracle BI Answers, if the Column Properties dialog box for a given column is set up as follows: On the Data Format tab, "Override Default Data Format" is checked and "Treat Numbers as Percentage" is selected; then setting this property to 1 instructs the BI Office server to treat the number as a percentage (divide by 100). If the user then formats the data as a percentage in Excel, the data will display correctly. If the property is set to 0, no percentage processing is done by the BI Office server. For more information on setting Column Properties in Oracle BI Answers, see the *Oracle Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*.
SawFetchFreshData	0 = Off (Default) 1 = On	Default is 0. Set this property to 1 to resubmit the query to refresh data. When set to 0, the data is read from the cache.

21.6 Setting Up SSL-Enabled OC4J for BI Office

Oracle Containers for J2EE supports Secure Sockets Layer (SSL) communication. For information on setting up SSL in your OC4J environment, see *Oracle Containers for J2EE Security Guide 10g (10.1.3.1.0)*.

Assuming your Oracle BI EE is installed under C:\OracleBI, the following instructions describe how to set up SSL in a standalone OC4J environment on Windows.

> **Note:** In the BI EE install, you will find the keytool utility in this location: C:\OracleBI\uninstall_jvm\bin\ keytool.exe.

1. Create a keystore with an RSA private/public key pair using the keytool utility.

   Example:

   This example generates a keystore to reside in a file named mykeystore.jks, which has a password of 123456, using the RSA key pair generation algorithm.

   ```
   cd C:\OracleBI\oc4j_bi\j2ee\home\config
   keytool -genkey -keyalg RSA -keystore mykeystore.jks -storepass 123456
   ```

   The keytool prompts you for additional information, as follows:

   What is your first and last name?

   [Unknown]: Test User

   What is the name of your organizational unit?

   [Unknown]: Support

   What is the name of your organization?

   [Unknown]: Oracle

What is the name of your City or Locality?

[Unknown]: Redwood Shores

What is the name of your State or Province?

[Unknown]: CA

What is the two-letter country code for this unit?

[Unknown]: US

Is <CN=Test User, OU=Support, O=Oracle, L=Redwood Shores, ST=CA, C=US> correct?

[no]: yes

Enter key password for <mykey>

(RETURN if same as keystore password):

Always press RETURN for the key password. In OC4J 10.1.3.x implementations, the keystore password must be the same as the key entry password.

The mykeystore file is created in the current directory. The default alias of the key is mykey.

2. If you do not have a secure-web-site.xml file, create one in the following location: C:\OracleBI\j2ee\home\config\secure-web-site.xml. You can start by copying content from default-web-site.xml.
3. Update secure-web-site.xml with the following elements:
   a. Update the `<web-site>` element to add secure="true" and to set the port value to some available port. (For example, port="4443". To use the default of 4443, you have to be a super user.) For standalone OC4J, use HTTP protocol, which is the default setting. (The setting protocol="http" in combination with secure="true" results in HTTPS being used.)

      ```
      <web-site port="4443" secure="true" protocol="http" display-name="Default
      OracleAS Containers for J2EE Web Site" >
      ...
      </web-site>
      ```

   b. Add the following under the <web-site> element to define the keystore and password.

      ```
      <ssl-config keystore="./mykeystore.jks" keystore-password="123456" />
      ```

      Note that the relative path and filename extensions are used for example: ./mykeystore.jks.

   c. Make sure that the server.xml points to the secure-web-site.xml file.

      As necessary, uncomment or add the following line in server.xml:

      ```
      <web-site path="./secure-web-site.xml" />
      ```

4. Restart OC4J to initialize the secure-web-site.xml file additions.
5. Test the SSL port by accessing the site in a browser on the SSL port. For example: https://bieeserver.mycompany.com:4443/em
6. Import the certificate into the OracleBI JDK and export the self-signed certificate.
   a. cd JAVA_HOME\jre\lib\security

b. Use the keytool utility to export a certificate from your keystore to mykeystore.cer.

```
keytool -export -keystore C:\OracleBI\oc4j_
bi\j2ee\home\config\mykeystore.jks -storepass 123456 -alias mykey -rfc
-file mykeystore.cer
```

c. Use the keytool utility to import the mykeystore.cer to replace the OracleBI JDK system-wide keystore cacerts.

```
keytool -import -keystore cacerts -storepass changeit -alias mykeycert
-file mykeystore.cer
```

7. Update SawBaseURL in the BI Office configuration file bioffice.xml as follows:

```
<property name="SawBaseURL">https://
/bioffice-server.mycompany.com:4443/analytics/saw.dll</property>
```

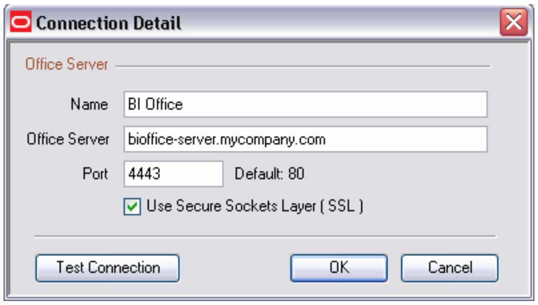
8. Now you can access this SSL-enabled OC4J and Oracle BI EE from your Oracle BI Add-In for Microsoft Office. In the Add-In, create a connection as follows:

   Name: OBI

   Office Server: bioffice-server.mycompany.com

   Port: 4443

   Use Secure Sockets Layer (SSL): selected



Connection Detail

Office Server

Name: BI Office

Office Server: bioffice-server.mycompany.com

Port: 4443 Default: 80

☑ Use Secure Sockets Layer (SSL)

Tip: If you can connect to SSL successfully in the browser, yet the connection fails in the client, confirm that the certificate was exported and imported properly (see Step 6).

21.7 Setting Up an SSO-Enabled Oracle BI Presentation Server for BI Office

Oracle BI Office currently cannot be fully integrated with SSO. If your Oracle BI Enterprise Edition implementation is SSO-enabled, users can use their SSO credentials from the Microsoft Excel or PowerPoint add-in to log in to the Oracle BI Office system. However, the following options from the client add-in insert menu are not supported:

- Edit View
- Insert as Image
- Insert as Flash

In SSO-enabled implementations, it is recommended that you set the SAWUseSSO parameter of the bioffice.xml file to "1". This setting suppresses the unsupported options from the client add-in insert menu.

For more information on the bioffice.xml file, see Section 21.5, "Setting Properties in the bioffice.xml Configuration File".

22

Microsoft Office Support

This new feature enables users to analyze and deploy business intelligence reports more easily in MS Office.

This release of Oracle BI Enterprise Edition features an improved version of Oracle Business Intelligence Add-in for Microsoft Office, with a new Excel and PowerPoint Add-in. These add-ins allow you to insert BI EE request views as native Office tables and BI chart views as native Office charts, or as static but refreshable images, or even as high quality Flash objects, leveraging Office functionality while referencing the Oracle BI semantic layer, thus ensuring that a single version of the truth is available to users across an enterprise.

Oracle BI for Microsoft Office provides the following functionality:

- Oracle BI Add-in for Microsoft Excel.
- Oracle BI Add-in for Microsoft PowerPoint.
- When installed on Excel 2007 the Oracle BI Add-in's functionality is made available via a native Office ribbon interface. When installed on Excel 2003 the Oracle BI Add-in's functionality is made available via a menu and a toolbar. In both cases an Office pane is available for browsing the presentation catalog and for selecting views to insert inside Office documents.
- Oracle BI Office Server. This is a server-side J2EE application that communicates between the BI Office client and the BI Presentation Server using web services.
- Ability to secure BI data in BI views inserted inside Excel spreadsheets or PowerPoint presentations. Metadata describing the BI view is preserved, but all data is wiped by the BI Add-in. Users can obtain the latest data available by refreshing their BI views using the BI Add-in.
- Users can copy views from Answers and Dashboards pages and then paste them into Excel and PowerPoint documents using the 'Paste' feature of the Oracle BI Add-in. Copied views are pasted as Office documents - Excel or PowerPoint tables and charts. This functionality also extends to copying compound views from Answers and Dashboards. Compound views copied from Answers or Dashboards are pasted into Office documents as native Office tables and charts. Only BI views supported by the Oracle BI Add-in are pasted inside Office documents.
- Users can use this 'copy' feature to copy the state of BI views in Dashboards and paste this state into Office documents. A user may choose to drill on a chart from year to quarter, then drill from region to the district level, using the 'copy' link, if available for that view, to copy this state and then paste it into an Office document using the BI Add-in. When a user refreshes the BI views inside the Office document, the data is refreshed and displayed base on this state of the request.

For more information on how this 'Copy' feature works from Answers and Dashboards, please refer to the section describing how to copy requests from Answers and Dashboards.

- A 'Flat Data View' is provided for every request. This view provides the entire data set for the original request in a simple tabular format. This view is more suited when the user wants to obtain only the data and then use Excel's functionality to perform further analysis on the data.
- The Table view displays results in a tabular format. Users can navigate and drill down in the results, add totals, customize headings, and change the formula or aggregation rule for a column. Drilling on a Table view is available only in Excel. If users have the BI Office Add-in installed, they can make use of this drilling functionality even in offline mode, without being connected to a BI Presentation Server.
- A 'Pivot Table View', available in the Excel Add-in, presents data from BI requests with page items as well as a section based layout if these had been defined in the BI Pivot Table view for the request in Answers.
- Support for the Oracle BI Security model, including support for both encrypted (SSL) and Single-Sign On modes. The same login that you use to sign-in to Answers or Dashboards is used to login to the BI Office Add-in.
- A native Excel 2007 and PowerPoint 2007 ribbon interface.
- BI Presentation Catalog browser as a pane in Microsoft Excel and PowerPoint to browse requests, both user-created and shared. The user can also view the individual views available for each request.
- Ability to insert table, pivot table, and chart views into Excel documents as Excel tables and Excel charts (subject to the limitations of Excel).
- Support for prompts defined in the BI request.
- If a view selected for insertion has multiple levels from one or more dimensions, the user can choose to limit the data for each dimension by unchecking these levels from a 'level selection' dialog that is displayed.
- Ability to insert table and chart views into PowerPoint presentations as PowerPoint tables and charts (subject to the limitations of PowerPoint).
- Ability to insert chart views as images inside Excel spreadsheets and PowerPoint presentations that can be refreshed using the Oracle BI Add-in.
- Ability to insert chart views as high quality Flash objects inside PowerPoint presentations that can then be refreshed using the Oracle BI Add-in.
- Users can define Excel specific conditional formats to data from BI views. These conditional formats are preserved during data refreshes.
- Users can change the formats of data items displayed in BI views in Excel and PowerPoint documents. To change the format of an item in an Excel sheet, the user would need to modify the style for that cell. These formatting changes are then preserved even during refreshes. Furthermore, these formatting changes are applied to all cells with the same style.
- For BI chart views inserted as native Excel or PowerPoint charts, users can change the chart type, apply other formatting changes using Excel and PowerPoint's charting capabilities. These changes are preserved during refreshes.

- Ability to customize the default appearance and placement of BI tables and pivot table views inserted in Excel sheets using an Excel template installed during the client add-in installation.

The following figure shows Oracle Business Intelligence Add-in for Microsoft Excel:

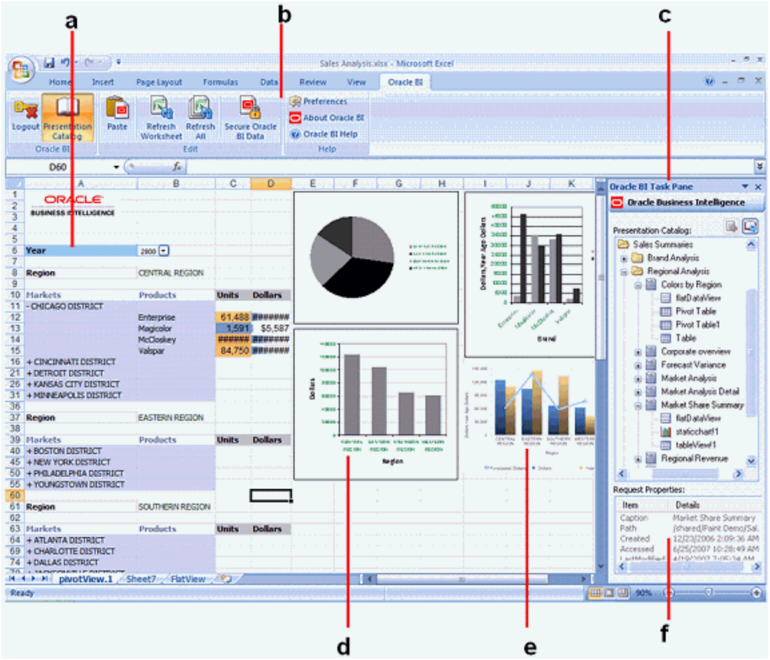


Figure Key:

a - Oracle BI toolbar as native Excel 2007 ribbon
b - BI Table View inserted as PowerPoint table
c - BI Presentation Catalog with list of requests and views available for insertion
d - BI Chart view inserted as refreshable PNG image
e - BI Chart view inserted as refreshable Flash format
f - Request properties

23

New Features in Release 10.1.3.3.1

The 10.1.3.3.1 release of the Oracle Business Intelligence Enterprise Edition introduces the following new BI Office features:

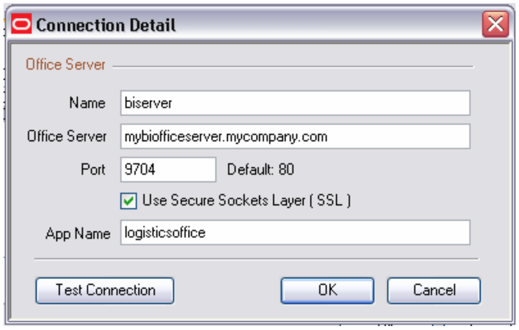
- Section 23.1, "Configure the BI Office Connection URL"
- Section 23.2, "Language Support"

23.1 Configure the BI Office Connection URL

When you deploy the BI Office Server application file to OC4J or to your supported J2EE container, you can provide any name for the application. The default is "bioffice", but you can specify any other name such as "myoffice", "financeoffice", "logisticsoffice". In 10.1.3.3.0, the URL that the BI Office client connects to always appended the default application name "bioffice" to the server (for example: `http://mybiserver.mycompany.com:9704/bioffice/....`), and there was no way to change the application name appended to the end of the URL.

In 10.1.3.3.1 you can now specify an "App name" for the BI Office server to configure the URL that the BI Office client connects to. Specify the App Name in the Connection Details dialog available from the BI Office > Preferences menu selection. The value for the App Name field will be used to construct the BI Office client connection URL. For example: `http://mybiserver.mycompany.com:9704/financeoffice/...` or `http://mybiserver.mycompany.com:9704/logisticsoffice/...`

The following image shows an example of the Connections Details dialog:



Connection Detail

Office Server

Name: biserver

Office Server: mybiofficeserver.mycompany.com

Port: 9704 Default: 80

☑ Use Secure Sockets Layer (SSL)

App Name: logisticsoffice

23.2 Language Support

The 10.1.3.3.1 release of BI Office adds the capability to support all the languages that BI Enterprise Edition supports. This is reflected in the login dialog also where you can now select from any of the approximately 25 languages. If BI EE metadata is available in multiple languages then BI Office shall also display this metadata in the language selected. For example, the language you choose to display BI EE metadata such as dimensions, ('country', 'brand'), items, and measures ('Sales', 'Profits') will also be used when you connect to this BI EE instance from BI Office.