Chapter No. 3
"Multiple Data Sources"
In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.3 "Multiple Data Sources"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Daniela Bozdoc is an IT professional with experience of almost a decade working as a software developer, an analyst, and a data and software architect on different technologies from MS .NET Framework to Sybase Power Builder, Oracle and MS SQL Server. She is continuing as an ERP consultant, and a report developer for MS dynamics NAV, Oracle EBS, and BI Publisher respectively in the present.

She is a graduate of Babes-Bolyai University of Cluj-Napoca, Romania, with a B.D. in Computer Science.

Daniela lives in Romania, where she has her family's support in her career and enjoys taking pictures of nature's beautiful landscapes.

I dedicate this book to my family for their unconditional love and support in every way possible throughout the process of writing this book and beyond. Their confidence in me especially has made this book a real thing.

I would like to express my gratitude to all those who gave me the opportunity to meet, and then to master, all the technology needed to write this book.

In addition, a special thanks to the people from Packt Publishing, editors, and project coordinators I worked with for their professionalism in offering me guidance and support.

For More Information:

In the last 15 years, some of us have witnessed mail becoming e-mail, grocery stores becoming www.store, and step-by-step technology becoming part of our daily life.

Technology development has radically changed the enterprise's way of doing business. First, daily tasks such as sales, client, and vendor management became computerized, and now all the resources are managed by enterprise applications. But this type of daily task support proved not to be enough for the business process, especially in the case of big organizations, which ended up with all kinds of applications, according to their vast field of activities. At some point they realized the need for some unified point of view. The choice was between replacing some applications and providing an integration process tool. Thus, Business Intelligence (BI) appeared in the scene. BI uses many types of data input, it doesn't take any note (if not necessary) of department, organization, or specific activities, and provides a unique answer for the corporate level.

Business Intelligence, in this way has become essential in most organizations. The goal in the near future is to support more effective business processes. Initially, it was used only for analysis and predictions based on the historical data; however, the new tendency of BI is to be actively involved in the business process. Integration seems to be the key to exploring the business possibilities, and providing the right tools seems to be the necessary step for an advanced business management.

In response to fast-growing demands, software development companies have come up with complex solutions, which can be used to plan, manage, and analyze a company's resources.

For More Information:
Somewhere in between, or being an active part for both the business process and software development, is the IT consultant. He has to stay in touch with the latest technologies, business solutions, and tools. He is an important factor when it comes to advising a company to adopt a new technology, a new tool, or a new vision concerning the IT field. There are two main categories of IT professionals involved in the process of providing these tools—the software developers and the report developers.

Having in mind the report developer's point of view, I will go through explaining the Business Intelligence concept—definition and process, Oracle BI, and finally an important component of Oracle BI—the Oracle BI Publisher—the document factory from Oracle.

**What is Business Intelligence?**

**Business Intelligence (BI)** is the process of transforming data gathered from all the business data sources into decision support business information. Most companies gather data from their business activity, even using ERP and legacy applications with different databases. The amount of data input depends on the software tools used. Data characteristics depend on the company's structure, such as departments or activity profile. At this point, the complex process of data processing and formatting, which is necessary to generate even a simple report becomes visible. The need for a tool to process the amount of data gathered becomes visible, as well.

**Premises**

The first thing that makes you think about a BI solution is the lack of information needed for the good course of the business process. Answers to questions such as these are very important:

- Which are my best suppliers?
- How much will it cost to start a new product's production?
- Where does all the money go?

The very first technical request to be accomplished is the data input quality. You won't have a good result if you have nothing to start with.

Another factor to be considered is the cost of a BI software implementation. This could be an expensive investment both from the perspective of time and financial resources.

*For More Information:*

**BI software application functions**

For a software application to work as a BI solution, it has to provide tools for the following:

- Data mining
- OLAP (Online Analytical Processing)
- ETL (extract, transform, load)
- Predictive analytics
- Business performance management
- Reporting

The reporting tool has to be flexible enough to allow the creation of reports, charts, or dashboards along with running and scheduling them at different access levels. It has to allow reports viewing, printing, and saving in many formats. Integration with other products like Microsoft Office is also important.

**How it works**

For a better understanding of how the described tools work together, take a look at the following figure:

As you can see, the process begins with the corresponding data of the business. There are a lot of data sources types, such as databases and operational applications, and local data sources such as spreadsheets and XML files.

---

**For More Information:**

An ETL (or ELT, if you focus on Oracle Data Integrator) tool will perform the BI data integration process. In this way, the Data becomes accessible to the BI solution's end user.

The data warehouse is the central point of the BI solution. This contains structured data—detailed and consistent—for query analysis, and provides support for all BI operations from data mining to reporting. Unlike the Online Transaction Processing (OLTP) databases, data warehouses have a very different design to support a large amount of data (which does not need validation in this case), only a small number of users, and at the same time a particular access to the data depending on the particular queries that have to be processed. The historical data coming from business transaction processes is stored using a structure based on business entities, such as customer, product, and time.

The result of the business intelligence process is also visible through dashboards, analysis, reports, alerts, scorecards, and all these being available in a variety of designs and formats according to the end user requests.

**Business Intelligence software**

As an introduction to the BI software world, here is a list of suggestions of actual software offers available in the market:

- Oracle BI
- SAP Business Objects
- Microsoft BI
- IBM
- SAS
- Microstrategy
- Actuate 7
- JasperSoft
- Olik View

**Oracle Business Intelligence**

Among the leading industry of BI software, Oracle's solution was designed to address the entire spectrum of analytical requirements facing businesses including information access, analysis and reporting, and data integration and management.

**For More Information:**

Oracle's offer could be divided into two main categories of software:


- **Oracle BI Applications**: These are built on OBIEE platform. Oracle BI Applications also include CRM Analytics and ERP Analytics applications. These solutions, being smart, agile, and aligned, will provide support for companies to achieve management excellence.

Oracle has also acquired Hyperion to expand their BI solution. The company claimed to be a leader in Enterprise Performance Management, by unifying Performance Management and BI solutions. It will support a broad range of strategic, financial, and operational management processes.

In the following sections, we will see how the Oracle BI solution maps to the Business Intelligence concept, including a short introduction to some Oracle BI components.

**Oracle Data sources**
This refers to all data coming from sources interacting with the Oracle Business Intelligence server. Oracle BI supports Oracle Database, Oracle E-Business Suite and other Oracle based sources, IMB DB2 Database, Microsoft SQL Server, SAP NetWeaver BI, Microsoft Excel, flat files, ODBC sources, and XML data sources.

**Oracle Data Integrator**
Data integration products are used to improve the speed of handling data, to reduce business process execution times, and to reduce development costs. Oracle Data Integrator combines all the elements of data integration to provide timely, accurate, and consistent information, which are as follows:

- Real-time and bulk data movement
- Transformation
- Synchronization
- Data quality
- Data management
- Data service

For More Information:
Oracle BI Interactive Dashboards

Oracle Dashboards are in fact customized points of access for analytics information. According to the user's requests, the personalized information displayed is used in the decision making process. The resulting Web interface is provided to users according to their group membership and permissions.

Oracle BI Server

This is the OLAP server. The Oracle BI server collects and aggregates information from all, even disparate data sources. It provides services to the other components, and processes the request, forming queries, and sending these queries to the underlying data source for processing. A very important factor in this case is the time of response, for the user to access immediately the answer to complex business questions, to be able to simulate various complex business scenarios.

Oracle BI server is the heart that drives all the other components.

Oracle BI Answers

Providing queries for the BI server, BI Answers is an ad hoc query and analysis tool. The web environment used is the gate to interactive charts, pivot tables, reports, and dashboards for the user. The user can save, modify, or format his view of information through the BI Answers tool.

Oracle BI Delivers

Based on analytics results, BI Delivers creates alerts. Specified results can be detected within reports and the triggered alerts can be sent via multiple channels including e-mail, dashboards, and mobile devices. The notified dashboards can again trigger other alerts, resulting in a very close monitoring device for the business process.

Alerts are being sent to users based on a subscribing service.

Oracle Scorecard and Strategy Management

This tool provides the ability to set a goal at the management level, to trace, and to apply all related activities involved in reaching the objective. The performance is monitored through Key Performance Indicators (KPI), and many visualization types like KPI watch lists, maps, strategy trees, and diagrams are provided.

For More Information:

**Oracle BI Publisher**

**Oracle BI Publisher** (formerly known as XML Publisher) is a reporting engine based on a very versatile open source language: XML. It can access relational, OLAP, and other data sources; in fact any data sources mapped to Oracle BI server.

It enables the creation, management, and delivery of all kinds of operational reports, financial reports, and any other customer-facing documents.

The result, consisting of high fidelity and highly formatted documents is delivered in a wide diversity of formats, such as: PDF, Excel, RTF, HTML, and electronic transfer documents. The results can be viewed online, saved for further processing, can be e-mailed, can be sent over FTP or scheduled for a delivery by, and for, a wide range of users and destinations.

However, the most important feature of Oracle BI Publisher is the fact that the report developer (not necessarily the software developer) is able to choose data sources and design the necessary reports.

The result types a user can get from Oracle BI are:

- Interactive dashboard: Provides with security, driven navigation
- Ad hoc analysis and interactive reporting: Provides with metrics, hierarchies, and calculations
- Enterprise reporting: It is provided by BI Publisher
- Proactive detection and alert: The alert engine can trigger workflows based on business events and notify stakeholders via their preferred medium or channel, such as: on the cell phone, via e-mail, a PDF file, or an Excel file
- Actionable Intelligence: The business process can be invoked from within the BI platform
- Microsoft Office integration: Information can be passed from Oracle BI to Microsoft Office documents such as Excel, Word, or Power Point
- Spatial Intelligence: It is provided via map-based visualization
- Scorecard and strategy management: Communicates strategic goals across the organization and monitors the process over time
- Server based query, reporting, and analysis: Provided by Oracle BI Server, which generates a query optimized from each data source, aggregates them, and presents the result.

For More Information:

What This Book Covers

This book introduces Oracle Business Intelligence Publisher 11g, providing a suite of examples to help illustrate its main functionalities. Here is a synopsis of what you will find in the book:

Chapter 1, What's New in Oracle BI Publisher 11g, is a short presentation of Oracle BI Publisher 11g, with focus on the differences between the 10g release and the 11g release.

Chapter 2, Creating a Data Model for a Report, explains how Data Model Editor – the new feature provided by BI Publisher 11 – looks and works.

Chapter 3, Multiple Data Sources, describes how BI Publisher retrieves and structures the data used for a report.

Chapter 4, Report Layout Template, is about data presentation, which include layout types, visual components of the report, and template types.

Chapter 5, The New XPT Format introduces the new report format used by Oracle BI Publisher 11g. It generates almost pixel perfect output and could be a good substitute for PDF forms.

Chapter 6, Oracle BIP Template Builder for Microsoft Word, describes layout designing in MS Word.

Chapter 7, The Report Configuration, demonstrates how reports are managed, and how to view, run, or set properties for reports.

Chapter 8, Exploring BI Publisher 11g: A Simple Report Example, is a simple report example. It presents an example, covering all the steps described in the previous chapters.

Chapter 9, BI Publisher 11g and E-Business Suite, discusses integration with Oracle e-Business Suite.

Appendix A, Report Translations, is a walk through all the translation techniques that BI Publisher offers.

Appendix B, Migrating Oracle Reports to BI Publisher, describes the steps required to migrate Oracle Reports to BI Publisher Reports.

Appendix C, Debugging Oracle Reports to BIP Migration, deals with an error that frequently occurs in Oracle Reports to BIP migration process. The necessary steps required to correct this type of error are described here.

Appendix D, Glossary, a short list of BI Publisher specific terms, for a better understanding of the concepts explained.

For More Information:
Multiple Data Sources

We met the Data Model Editor in the previous chapter. The editor’s interface deals with all the components and functionalities needed for the data model to achieve the structure you need. However, the main component is **Data Set**. In BIP, to create your data model structure you can choose from a variety of data set types:

- SQL Query
- MDX Query
- Oracle BI Analysis
- View Object
- Web Service
- LDAP Query
- XML file
- Microsoft Excel file
- Oracle BI Discoverer
- HTTP

Taking advantage of this variety requires multiple **Data Sources** of different types to be defined in the BIP. In this chapter, we will see:

- How data sources are configured
- How the data is retrieved from different data sets
- How data set type characteristics and the links between elements influence the data model structure

For More Information:

Administration

Let’s first see, how you can verify or configure your data sources. You must choose the Administration link found in the upper-right corner of any of the BIP interface pages, as shown in the following screenshot:

The connection to your database can be chosen from the following connection types:

- Java Database Connectivity (JDBC)
- Java Naming and Directory Interface (JNDI)
- Lightweight Directory Access Protocol (LDAP)
- Online Analytical Processing (OLAP)

Available Data Sources

To get to your data source, BIP offers two possibilities:

1. You can use a connection. In order to use a connection, these are the available connection types:
   - JDBC
   - JNDI
   - LDAP
   - OLAP

2. You can also use a file.

In the following sections, the Data Source types — JDBC, JNDI, OLAP Connections, and File — will be explained in detail.

For More Information:
JDBC Connection

Let's take the first example. To configure a Data Source to use JDBC, from the Administration page, choose JDBC Connection from the Data Sources types list, as shown in the following screenshot:

You can see the requested parameters for configuring a JDBC connection in the following screenshot:

- **Data Source Name**: Enter a name of your choice.
- **Driver Type**: Choose a type from the list. The relating parameters are:
  - **Database Driver Class**: A driver, matching your database type.
  - **Connection String**: Information containing the computer name on which your database server is running, for example, port, database name, and so on.
- **Username**: Enter a database username.
- **Password**: Provide the database user's password.

For More Information:

Multiple Data Sources

The **Use System User** option allows you to use the operating system’s credentials as your credentials. For example, in this case, your MS SQL Database Server uses Windows authentication as the only authentication method.

When you have a system administrator in-charge of these configurations, all you have to do is to find which are the available Data Sources and eventually you can check if the connection works. Click on the **Test Connection** button at the bottom of the page to test the connection:

![ORACLE BI Publisher Enterprise](image)

**JNDI Connection**

JNDI Connection pool is in fact another way to access your JDBC Data Sources. Using a connection pool increases efficiency by maintaining a cache of physical connections that can be reused, allowing multiple clients to share a small number of physical connections.

For More Information:

In order to configure a Data Source to use JNDI, from the Administration page, choose JNDI Connection from the Data Sources types list. The following screen will appear:

As you can see in the preceding screenshot, on the Add Data Source page you must enter the following parameters:

- **Data Source Name**: Enter a name of your choice
- **JNDI Name**: This is the JNDI location for the pool set up in your application server, for example, `jdbc/BIP10gSource`

The users having roles included in the Allowed Roles list only will be able to create reports using this Data Source.

### OLAP Connection

Use the OLAP Connection to connect to OLAP databases. BI Publisher supports the following OLAP types:

- Oracle Hyperion Essbase
- Microsoft SQL Server 2000 Analysis Services

For More Information:

Multiple Data Sources

- Microsoft SQL Server 2005 Analysis Services
- SAP BW

In order to configure a connection to an OLAP database, from the Administration page, choose **OLAP Connection** from the **Data Sources** types list. The following screen will appear:

On the **Add Data Source** page, the following parameters must be entered:

- **Data Source Name**: Enter a name of your choice
- **OLAP Type**: Choose a type from the list
- **Connection String**: Depending on the supported OLAP databases, the connection string format is as follows:
  - **Oracle Hyperion Essbase**: Format: [server name]
  - **Microsoft SQL Server 2000 Analysis Services**: Format: Data Source=[server];Provider=msolap;Initial Catalog=[catalog]
  - **Microsoft SQL Server 2005 Analysis Services**: Format: Data Source=[server];Provider=msolap.3;Initial Catalog=[catalog]
- **Username** and **Password**: Used for OLAP database authentication

For More Information:
File

Another example of a data source type is File. In order to gain access to XML or Excel files, you need a File Data Source. In order to set up this kind of Data Source, only one step is required—enter the path to the Directory in which your files reside. You can see in the following screenshot that demo files Data Source points to the default BIP files directory. The file needs to be accessible from the BI Server (not on your local machine):

![Screenshot of File Data Source]

Multiple Data Sources

Having a large amount of data at your disposal for creating a Data Source necessitates a great effort in creating a good structure of data with the necessary relationships between its elements.

Here are some of the tools, provided by BIP 11g Data Model Editor to design links between elements in your Data Model structure.

Add different types of Data Sets to a Data Model

You will see how the structure of the Data Model changes, after different data connections (links) are created.

As shown in the following screenshot, the Data Sets added are:

- **Independent**: An HTTP type Data Set
- **trend**: An Oracle BI Analysis type Data Set

For More Information:

Multiple Data Sources

- **Sales**: A XML file type Data Set
- **countries**: A Microsoft Excel file type Data Set
- **reader, library, and book**: SQL Query type Data Sets

No links between the inserted Data Sets are created at this moment.

From the XML file and HTTP Data Sets, no metadata is available, hence you cannot create links using these types of Data Set.

For More Information:
In the following screenshot, you can see from the **Structure** view that there is no hierarchy in the **Data Model**:

For More Information:
Joins between Data Set objects

Using the Query Builder, you can create links between table columns as shown in the following screenshot. This kind of relationship is called a join and specifies how the rows of one table are combined with the rows of the second table. Columns of a table, view, or materialized view can be used to create links. Query Builder supports inner, outer, left, and right joins. You can find a detailed description of these operations in the Structured Query Language (SQL) documentation:

In the SQL tab of the Query Designer, you can see the SQL syntax generated for the created links. Three conditions were generated in a where clause. This is shown in the following screenshot:

For More Information:
Links between Data Sets

There are two options when creating Data Sets links. You can create element-level or group-level links.

Element-level links

An **Element-level link** refers to a link created between an element of a Data Set and an element of another Data Set. In this way, a parent-child relationship is created between different Data Sets.

As the following screenshot shows, by binding the **READER_ID** column from the **G_Library** group and the **READER_ID_1** column from the **G_Reader** group, a link was created between the two mentioned groups:

You can link different types of Data Sets using element-level links. The procedure for creating a link has been covered in the previous chapter. For example, as you can see in the following screenshot, a link between a SQL Query Data Set and a Microsoft Excel file Data Set was created by dragging the parent field **Country** from the **G_Reader** group over the Code field in the **G_Country** child group:

For More Information:

**Group-level links**

A **group-level link** also defines a link between two Data Sets to obtain a hierarchically-structured XML. But in this case, the child group must have the same Data Set type as the parent group. For example, you cannot create a group-level link between `G_Reader` and `G_Country` (from the preceding screenshot), as one is a SQL Query Data Source type and the other is an MS Excel type.

To create a group-level link, go to the group menu (upper-right corner of the group) and choose the **Create Group Link** option. The following screenshot shows, how available **Child Groups** are displayed for selection when a **Group link** is created:

There is another request for a group-level link to work. You have to define a bind variable in the child query, as shown in the following screenshot. This variable is linked to a column from the parent Data Set.

Considering `G_Book` (the child group), you can see that only the `Book_ID` field could be used as a link between `G_Library` and `G_Book`, as `Book_ID` is the only field which is common to both the groups:

---

For More Information:

Group-level links are provided for backward compatibility with Data Templates from earlier versions of BIP. However, element-level links are preferred.

**Group-level aggregate elements**

Once a parent-child link is created between two Data Sets, you can insert values obtained by applying aggregate functions on the child’s elements into the parent data set. Depending on the element type, different aggregation functions are available. As you can see in the following screenshot, for a string data type element, you can choose between the following functions:

- COUNT
- FIRST
- LAST

For More Information:

Global-level functions

There is a special group in the Data Model designer: **Global Level Functions**, and you can add values to this group, obtained by the aggregation of elements belonging to any of the available Data Sets. For example, let's add a value called **TOTAL_BOOKS** to the **Global Level Functions** group. To obtain this value, as the following screenshot reveals, the aggregation function COUNT is used on the BOOK_ID field from the G_Book group. The same procedure is used to obtain the values: **TOTAL_LOANS**, **TOTAL_READERS**, and **TOTAL_COUNTRIES**:
You can further refine the result and create more complicated expressions, such as the example shown in the following screenshot:

**Parameters**

In order to add parameters to a data set, parameters must be first declared.

For example, let’s add a List of Values (LOV) named **countries**, as shown in the following screenshot:

For More Information:

Multiple Data Sources

A parameter named **country** is defined and it uses the LOV **countries**, as shown in the following screenshot:

After a parameter is defined, there are many ways to use it. The advantage is more accurate and precise data for the reports using the parameterized data model.

**SQL Query statements**

You can use parameters in **SQL Query** statements. For example, by including the parameter **country**, as shown in the following screenshot, the readers list is narrowed to the readers from a specific country:

---

For More Information:

Group filter expressions

Another option is to use parameters in group filter expressions. The result is similar to the previously described case, with the difference being that a group filter can also be used for another Data Set type than SQL type; for example, an Oracle BI Analysis Data Set (G_Trend) can be filtered using a parameter-based filter expression as you can see in the following screenshot:

Data Set parameters

You can also specify parameters when you create a Data Set. In the following screenshot, you can see the interface to add parameters to an HTTP Data Set:
Multiple Data Sources

The following screenshot shows the interface for a Microsoft Excel file data set type:

Microsoft Excel Data Set types support only one value per parameter.

As opposed to HTTP and Microsoft Excel Data Set types, for an Oracle BI Analysis Data Set type, you cannot add parameters, as you can see in the following screenshot. Parameters and lists of values will be inherited from the BI analysis and they will show up at runtime:

However, there are a few actions supported by this Data Set type:

- Global level functions
- Group filters
- Setting the value for elements in case of null values

For More Information:
Finally, the following screenshot shows, how the Data Model structure changed to reflect the links between Data Sets and group-level filters:

<table>
<thead>
<tr>
<th>Diagram</th>
<th>Structure</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table View</td>
<td>Output</td>
<td>Independent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sales</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATA_DS</td>
</tr>
</tbody>
</table>
| | | S_1
| | | BOOK_ID |
| | | S_6
| | | AUTHOR_ID |
| | | TITLE |
| | | YEAR |
| | | PUBLISHING |
| | | LOCATION |
| | | PRICE |
| | | S_2
| | | BOOK_ID_1 |
| | | READER_ID |
| | | LOAN_DATE |
| | | LOAN_PERIOD |
| | | READER_FIRST_NAME |
| | | S_3
| | | FIRST_NAME |
| | | LAST_NAME |
| | | READER_ID_1 |
| | | ADDRESS |
| | | COUNTRY |
| | | TELEPHONE_NO |

There are only a few cases in which Data Models with multiple Data Sources are recommended:

- When you need to perform functions not supported by the query type
- When the Data Model has to support complex views
- When you want to simulate a view, in case you don't have a view or you don't want to use one

Otherwise, it is recommended that the number of Data Sets used be reduced. The reason is that single Data Set models execute faster than multiple Data Set models—in a parent-child hierarchy the child is executed for each element of the parent. However, the main advantage of a Data Model based on multiple data sets remains, and thus, a model offers a more intuitive and simple view of the data structure.

For More Information:
Multiple Data Sources

Summary
In this chapter, we saw how to use different tools offered by the Data Model Editor to shape a data structure composed of multiple data sets, into a compact and organized form.

For reports, the advantage of using a good structured Data Model is easy and also provides intuitive data access.

Keeping in mind the various aspects related to the Data Model structure we will proceed with the report layout design in the next chapter.

For More Information:
Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our [shipping policy](http://www.packtpub.com/oracle-bi-publisher-11g-practical-guide-for-enterprise-reporting/book).

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.