Oracle Business Intelligence
Enterprise Edition 11g:
A Hands-On Tutorial

Haroun Khan
Christian Screen
Adrian Ward

Chapter No. 17
"Oracle Essbase and OLAP Integration"
In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.17 "Oracle Essbase and OLAP Integration"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Haroun Khan is one of Europe's leading OBIEE consultants. Being a Computer Science graduate from Imperial College, London, he has been involved with OBIEE from its early days as an acquisition from nQuire by Siebel, and subsequently as part of the Oracle family. Haroun has worked as a consultant on projects worldwide for Siebel and as a Principal Consultant for Oracle over a period of 10 years. He has specialized in BI and data warehousing over a longer period including time working at MicroStrategy. Haroun now freelances in leading and designing projects in the BI and data warehousing space, combining this with entrepreneurial activities, such as his own e-commerce business JRPass.com. In his downtime, Haroun likes to spend as much time as possible climbing in the mountains, away from a computer screen and avoiding numerous requests to write a blog or get on another plane!

Haroun works through his own company Awaan and can be contacted at his company address haroun@awaan.eu.

For More Information:
It goes without saying, but I would like to thank my parents, Zainab and Ayub for their support, encouragement and for everything, really. Also I have to include my sisters Sophia and Soraya so that they can have their name in print and I can avoid admonishments! Last, but not least, thanks to my toddler niece Sharifa for providing me with boundless distractions and opportunities to procrastinate.

Thanks goes to all those people (too many to mention) who I’ve worked with on great projects during my time at Oracle and MicroStrategy. Thanks also go out to Packt and their editors for their work. Finally, I would also like to extend my appreciation to my co-authors for their commitment and energy. It’s been tiring but fun!

Christian Screen is a Business Intelligence evangelist with over 15 years of experience in technology ranging from low-level programming, E-Commerce, Data Warehousing, Enterprise Performance Management and, of course, Business Intelligence. In his spare time, he enjoys writing technical articles, learning new technologies, developing products, writing software, spending time with his family, trying to change the world, and running his blog and podcast at ArtOfBi.com. He is an Oracle ACE, an Oracle Deputy CTO, and holds several technology and project management certifications.

For More Information:
I would like to thank my family (wife Kirsten and three super children—Riley, Jaxon, and Dylan) seems so cliché but indeed their patience and support has been essential to the process of writing my first book. I'd also like to thank all of my colleagues who have provided continual intellectual ferment for our work with Business Intelligence and Enterprise Performance Management solutions. Thanks to Greg, Jason, and Tom at Analytic Vision for giving me a chance with my first consulting gig. Thanks to Amy Mayer of BI Consulting Group and Capgemini for my second consulting gig and having an unbelievable culture that I'll take with me wherever I may go. There are a few people at Oracle to thank as well, such as Mike Hallett in the UK who handles the BI/EPM partner community for EMEA and does an amazing job at building the community and keeping even us in the USA informed, the Oracle ACE Program leaders Justin Kestelyn and Lillian Buziak (Brian Stover, of course, many thanks!) for the Oracle ACE moniker, and the Oracle BI development team Matt Bedin and Phillipe Lions for their SampleApp and other assistance. I'd like to generally thank other authors of technical books that I may have referred to over the years and the many bloggers that find it their duty to share the tidbits, prose, or other knowledge that make finding an answer to daily issues merely a Google search away—thanks for sharing. A big thanks goes to the very keen reviewers of this book—Kevin McGinley, Daan Bakboord, and Ramke Ramakrishnan. We truly appreciate the time, effort, and suggestions you've provided to make this book better. Lastly, thanks to you for reading our first book—hopefully you purchased it legally—I trust you will find it useful.

Adrian Ward started working in Siebel Analytics back in 2001 and quickly realized the potential in the technology. He formed the UK's first independent consultancy focusing purely on OBIEE (née Siebel Analytics) and Oracle BI Applications. He has led many large successful OBIEE implementations in a wide range of business sectors, from Investment Banking to Military operations. His deep technical OBIEE and BI Applications knowledge has been applied on dozens of projects throughout the globe including HR, Sales, Service, Pharma, and Custom Analytics.

He was also one of the first bloggers on Oracle BIEE and today runs the Addidici OBIEE consultancy which has operations in the UK, Europe, and South Africa.

For More Information:
Adrian runs one of the largest Oracle BI networking groups on LinkedIn—"Oracle Business Intelligence", and helps others to network and learn about the product and its application, including organizing networking social events in London.

In his spare time he loves sailing, skiing, enjoying life with his family, and learning new technologies.

Firstly, I would like to thank my wife Sarah for her enormous love and support over the years, and in particular whilst I was writing my part of this book. Thanks too, must go to my cool children, Hugh and Hatty, for their help in keeping the house quiet and delivering endless cups of tea, and also to my Mum and Dad for being the best parents you could ask for—always there when you need.

I will be eternally grateful to Narmada for her selfless support and to Chet Justice for his great sense of humor. Special thanks go to James Robinson for my first decent job, to my great clients for employing me, and the great people I have worked with over the last 20 years (including Steve Lomax, Robert Patterson, Trev Harvey, Eric Gravil, Adrian Ball, Haider Tirmizi, Luis, Pierre, Andi Schloegl, Neil Ashton, Daniel and many more).

I am also indebted to Graeme Hampshire for inspiring me to get writing, and for helping to keep sailing fun, John Dunnet and the crew of The Beefeater for putting up with my captaincy, and Jon Spencer for being a great PRO.

Finally thanks to my special friends Daniel, Andy & Jenny, and Jamie & Jackie for putting up with me over the years!

For More Information:
Oracle Business Intelligence Enterprise Edition 11g: A Hands-On Tutorial

Oracle Business Intelligence Enterprise Edition (OBIEE) 11g is packed full of features and has a fresh approach to information presentation, system management and security. This book will introduce the reader to those features, providing a step-by-step guide to building a complete system from scratch. The aim of the book is to equip a developer or analyst with a good basic understanding of what the product contains, how to install and configure it, and how to create effective business intelligence.

What This Book Covers

Chapter 1, Understanding the Oracle BI 11g Architecture, helps you in understanding the 11g architecture. As with any good software suite, a solid architectural foundation is required. In today's marketplace the ability for software to scale well, meet the growing needs of an enterprise, and integrate with an organization's existing Information Technology (IT) investments is expected. Having the software be transparent enough for the average IT professional to implement it is definitely a plus. Being simple enough for an end-user to use or consume the product is a must. Oracle Business Intelligence (Oracle BI) 11g fits within all of these paradigms.

Chapter 2, Installing the Metadata Repository, covers how to install the required database components for your Oracle BI system to use them. Before installing an OBIEE 11g system, you will need to prepare a database, not for end user reporting but for the OBIEE system itself. You will also learn how to customize the installation to change the options available, and how to use the silent installation. You cannot afford to skip this chapter if you're installing your own development system! It's crucial to understand what is possible so that you can advise the database administrators, who will be managing your production environments.

Chapter 3, Installing on Windows Server 2008, provides step-by-step instructions for installing Oracle BI 11g on Windows Server 2008. This installation will walk you through the Enterprise installation of Oracle BI 11g, which is one of the three possible installation options, and will give you the best of all worlds for an Oracle BI 11g platform implementation. It will allow you to work/play with all of the features seen in a production Oracle BI 11g environment.

For More Information:
Chapter 4, Installation Options, covers additional installation options. The installation conducted in Chapter 3, Installing on Windows Server 2008, is perfect for a sandbox or a development environment. There is one more installation option, Software Only Install, which is an advanced way to conduct the Enterprise installation option. Several advanced configuration options and many production environment considerations may be made using the Software Only Install option. Additional installation and environment configuration options are what we will cover in this chapter.

Chapter 5, Understanding the Systems Management Tools, goes into greater detail on the administration interfaces, explaining what these components are, what they do, and how they work together. We will dive into the navigation of these tools so that you will become more familiar with the interfaces. Finally, this chapter is crucial to the remainder of the book as it contains the security exercises for creating the users and groups that will be used to access the Tennis Repository's dashboards and reports, which you will develop in subsequent chapters.

Chapter 6, Upgrading the RPD and Web Catalog to 11g, looks at the upgrade process for the most fundamental parts of the system from a version 10g implementation to 11g. The upgrade process for this is extremely straightforward, as Oracle has provided an easy-to-use upgrade tool that we will step through in this chapter. If you are upgrading a current live implementation, then we must consider the wider implications of the upgrade, especially the possible effects on current functionality. Therefore, in this chapter we will also touch upon the thinking and planning that is needed prior to a full upgrade for a current live implementation.

Chapter 7, Reporting Databases, introduces the main concepts of a reporting database. The process of creating an efficient database is the subject of dozens of books and blogs, and therefore the details in this chapter should provide enough information to get you started in creating a database that is fit for using in an Oracle Business Intelligence system.

Chapter 8, Developing a BI Repository, covers the development of a simple RPD from tables in a database through to how those objects are presented to us when we move on to create an actual request. This will be carried out via the Oracle BI Administration tool, which is the primary method of accessing and modifying an RPD file. This tool provides an inviting graphical interface for developing and administering an RPD file. By the end of this chapter, you will be able to complete the major tasks associated with RPD development. We will also describe the more advanced options that are available.

Chapter 9, Features of the Presentation Catalog, introduces the new interface of the web catalog and the tools that are integrated into the Presentation Services. It also explores aspects of search, catalog administration, privileges, object security, and more.

For More Information:  
Chapter 10, Creating Dashboards and Analysis, teaches you how to create reports and group them in dashboards. In this chapter, we will also look at the various ways of representing and formatting data that are available, along with advice on best practices gained from implementation experience. This is one of the more robust chapters in the book and provides an insightful look at dashboard and report development.

Chapter 11, Agents and the Action Framework, covers the new functionality that Oracle has introduced in 11g for more integration with business processes, and the actions that result from producing the analytical reports. In previous versions, we had the Delivers portion of OBIEE where you could invoke basic actions, such as the delivery of reports or dashboard alerts. 11g has drastically enhanced this capability through the Action Framework, through which we are now able to initiate a multitude of additional noncore actions. During this chapter, we will look at examples of the new actions that Oracle has provided in attempting to succeed in this goal.

Chapter 12, Developing Reports Using BI Publisher, covers some of the new features of BI Publisher 11g as well as the general functionality of BI Publisher in order to get you up-to-speed on using the tool. It is aimed at providing a crash-course that should give any reader enough hands-on exercises to get their feet wet and enough food for thought for further research.

Chapter 13, Customizing the Style of Dashboards, provides a step-by-step how-to guide for branding OBIEE dashboards to match your corporate look and feel. It also provides several other insights for continued development and research.

Chapter 14, Improving the Performance, explores some common techniques to reduce the bottlenecks that can exist in the process of delivering dashboards and reports to the users. We will look across the whole system, defining poor performance and where required, take steps to improve the performance.

Chapter 15, Using the BI Admin Change Management Utilities, builds on the fundamental techniques of OBIEE development learned in previous chapters explores other capabilities in the Administration tool. In larger projects we may have a group of developers accessing and modifying the same RPD. The OBIEE Administration tool provides the ability to merge multiple versions of an RPD as well as functionality for groups to manage development on a sole repository (multiuser development). In this chapter, we will go over various methods of dealing with multiuser development.

Chapter 16, Usage Tracking, will demonstrate how to activate the "usage tracking" feature, and create useful reports from it. One of the great features of Oracle BI is that you can use the system—Dashboards and Analysis—to monitor the system itself, which means to say that you can use an OBIEE Analysis that tells you how OBIEE is performing for your users!

For More Information:
Chapter 17, Oracle Essbase and OLAP Integration, shows how OLAP technologies integrate into Oracle BI. After exploring the options, we will then focus on Oracle Essbase as the preferred OLAP technology. This chapter will show you how to integrate Oracle Essbase as a data source in Oracle BI and define several best practices for the integration. At the end of the chapter, you should have a well-balanced sense of how Essbase integrates with Oracle BI and the added value that it can bring to an organization.

Appendix A, Programs and Definitions, describes the main OBIEE command utilities and provides some reusable examples. One of the powerful features of OBIEE 11g administration is its ability to be controlled by User Interfaces (web browsers and Admin tools) as well as by command-line utilities. Many of the manual tasks that you undertake each day can be scripted and therefore automated. In this chapter you will learn about those automations.

Appendix B, Useful Resources: Join the Oracle BI Movement, lists some of the best books, events, groups, blogs for further reading and further practice on OBIEE 11g. Over the last decade, the number of resources focusing on Oracle Business Intelligence has skyrocketed. Oracle’s documentation of the software has become increasingly more useful and user-friendly. Blogs all over the globe have popped-up in large numbers (though some better maintained and better written than others). Use this chapter to find where to learn more about OBIEE.

For More Information:
Since Oracle acquired Hyperion in 2007, they have steadily increased customer awareness that bringing Business Intelligence and Enterprise Performance Management (EPM) together provides a much more complete view of how an organization can analyze its business. Hyperion was best known for its core Online Analytics Processing (OLAP) tool, Essbase. Oracle has successfully rolled Hyperion Essbase into its Application stack. However, Essbase is not the only OLAP tool that Oracle owns. Nor is it the only OLAP engine that Oracle BI can leverage as a data source.

In this chapter, we will look at how OLAP technologies integrate into Oracle BI. After exploring the options, we will focus on Oracle Essbase as the preferred OLAP technology. This chapter will show you how to integrate Oracle Essbase as a data source in Oracle BI and define several best practices for the integration. At the end of the chapter, you should have a well-balanced sense of how Essbase integrates with Oracle BI and the added value that it can bring to an organization.

A bit about OLAP

OLAP technology has been around for many years. It is a technology system in and of itself. Over the years, it has developed its own standards and earned itself a stay in almost every organization's analytical decision support reporting toolbox. The preferred language for querying OLAP data is a technology called Multidimensional Expressions (MDX). Just as a relational database management system (RDBMS) uses SQL (Structured Query Language) to read and write its data, an OLAP structure uses MDX but solely in a read-only capacity.
**Oracle Essbase and OLAP Integration**

**Competition**
There are four major OLAP tools that Oracle BI supports: Oracle Essbase, Oracle OLAP, SAP BW, and Microsoft SQL Server Analytic Services (SSAS). There are pros and cons to each. An organization's bias, if any, is usually due to its corporate technology standard. An organization declaring itself as a Microsoft "shop" will usually integrate SSAS if requiring an OLAP technology. However, that does not preclude it from using another OLAP technology. Essbase is usually the exception that finds its way into an organization's OLAP mix, but in many cases it was already the incumbent under its Hyperion moniker.

**MOLAP, ROLAP, HOLAP, XOLAP**
What would a technology book be without more technology acronyms? Well, here are some more for you to add to your already lengthy list.

We've talked about OLAP technology and in many cases the terminology is misused. When this disruptive technology emerged, there were very few software vendors that understood how valuable it would become to the evolution of structuring data. There were many variations on the same concept. OLAP is a general encompassing term that has an overarching definition for aggregate data storage. **Aggregate Data Storage** means that the data is taken from a lower level or transactional level of granularity such as the date, customer, and store level in a data warehouse and summed to a higher level at some delayed frequency and stored for later retrieval. This process and the OLAP engine allow for complicated data calculations and pre-aggregation of data against varied dimensionality to be achieved. The result provides the ability to quickly query an otherwise large and complex amount of data which is most often required by executive levels of reporting. Compare that to a SQL query against an RDBMS every time a report is needed. The amount of time and processor resources required for an RDBMS to sum, count, or otherwise aggregate, millions of records from the data warehouse to multiple levels of granularity for every combination of data relationship is immense.

Although OLAP is the general term, the OLAP solutions mentioned earlier that are offered by the major software vendors are actually **Multidimensional Online Analytical Processing (MOLAP)** solutions. This was the original terminology of this standard, which is commonly referred to as OLAP. However, due to different business analytics reporting requirements affecting this type of multidimensional system, several variations of MOLAP technology were forged. Requirements such as the need to blend real-time data, combining MOLAP data with its relational data source counterparts, and applying security at different levels helped in bringing on these variations.

---

For More Information:
Here are short definitions of those variations of OLAP:

- **Relational OLAP (ROLAP):** This is a system using a multidimensional architecture sourced primarily against one or more relational databases; typically a Data Warehouse. The appearance of accessing a multidimensional cube is a derivative created by an Extract Transform Load (ETL) process which creates aggregated information. Slicing and dicing is ultimately boiled down to interpreted SQL query syntax, such as using a simple `WHERE` clause.

- **Hybrid OLAP (HOLAP):** This is a technology that allows for data stored within separate repositories, such as a multidimensional OLAP system and an RDBMS to be combined at some level of granularity. The goal is to allow for more efficient querying of associated data slices allowing for the query to be optimized by the system that performs best at each respective level. This is usually OLAP for higher summarized levels and the RDBMS for the more detailed levels.

- **eXtended OLAP (XOLAP):** This is a technology spawned for Essbase that attempts to use the structure of an Essbase multidimensional database metadata outline to control the retrieval of data from a relational data source at runtime. Basically, it is an integration of a relational data source with Essbase, which is primarily used as its own multidimensional OLAP system.

What all of these variations have in common is the concept of dimensions and the ability to aggregate. Dimensions are a way of slicing or gaining perspective against the underlying "fact" data or metrics. Ultimately, the storage repository associated with an OLAP engine is commonly referred to as a **cube** and sometimes a **hypercube**.

**Essbase's entrenched past**

Hyperion has been a long-time incumbent in most organizations but not under the Information Technology umbrella as some may think. Essbase was originally the champion of the Finance department due to its ability to aggregate financial data, handle calculations, and process complicated allocations with relative ease. It also allowed for ad-hoc analysis of said data via the familiar tool, Microsoft Excel, via a proprietary Hyperion Excel add-in. As a matter of fact, Essbase was born out of the inherent challenges of using spreadsheets for analytics in a standard and scalable way. Essbase is actually an acronym that stands for **Extended Spreadsheet dataBASE**. Prior to being acquired by Oracle, Essbase had been developed by software company, Arbor, before it merged with Hyperion Solutions Corporation in 1998. It has been the subject of conversation in many organizations, specifically when it comes to which department would maintain the software – Finance or IT. After its acquisition by Oracle, this conversation arises less because of the repositioning of Essbase into Oracle Fusion Middleware, which is owned by IT.
Unfortunately, with this type of history inside of an organization, building an integration path between Oracle BI and Oracle Essbase can have complications like any other integration. This chapter should give you enough insight and know-how in order to roll out a quick **Proof-of-Concept (POC)** in order to dazzle the business team that usually owns Essbase. Just don't forget that the business users often need a little extra **Tender Loving Care (TLC)** during this process; Essbase is their baby you're playing with.

**Oracle Essbase Studio**

Oracle Essbase is a server engine. The ability to create the metadata associated with the database cube is usually conducted through the **Essbase Administration Service (EAS)** console. Introduced shortly before Hyperion was purchased by Oracle, a different GUI development interface called **Essbase Studio** was introduced. This separate application allows a user to create an Essbase cube via a GUI IDE. It also is the first of the tools to assist with XOLAP technology development. It can even use Oracle BI as a data source. Using **Oracle Essbase Studio** to build an Essbase cube with Oracle BI as a data source allows an existing enterprise Oracle BI logical model to be translated into an aggregate storage repository. This could potentially create something of a feedback loop, in that the resulting Essbase cube could then be imported into the Oracle BI repository and leveraged as a data source.


**Oracle BI SampleApp v107+ – VM image**

The Oracle BI development team has worked hard at getting its best in class solution to customers. They've produced a composite set of materials and artifacts called the **Sample Application (SampleApp)**, which is comprised of an Oracle BI RPD, Presentation Catalog, supporting sample data, and more. These SampleApp artifacts, downloadable from the OTN in compressed file format, are ready for installation into an existing Oracle BI 11g environment for demonstration, education, and testing purposes. More importantly, since the Oracle BI development team has produced the dashboards, models, and so on in the SampleApp, they are sure to highlight the deepest capabilities of Oracle BI.
This team has also developed a ready to deploy sandbox server environment leveraging **Virtual Machine** (VM) technology. This VM image contains an Oracle BI 11g server, Essbase Server, Times Ten Server, Oracle DB 11g Server, and ancillary software all in one demo environment. It is available for download via the OTN with a registered user account. The download is approximately 25 GB, so prepare in advance for the download time. The VM image is based on **VirtualBox**, which is an Oracle sponsored open source operating system agnostic VM technology. The VM image itself is based on the Linux operating system, **Oracle Enterprise Linux** (OEL). Download the VM Image from the OTN, [http://www.obi11gbook.com/u/18](http://www.obi11gbook.com/u/18)

The SampleApp artifacts (not the Oracle software technologies) are open source. This means that you can leverage the data and models where you see fit, as long as it complies with the open source license or third-party licenses under which it is released.

### Getting started – let's get set up

As a pre-requisite to the exercise in this chapter, you must have an Essbase Server from which to operate. If you already have an Essbase Server at your disposal, you are ready to start the exercises. If not, as mentioned above, the Oracle BI **SampleApp** has an Essbase Server ready for you to use and it is perfect for what you will do in the exercise. We are actually using the SampleApp v107 VirtualBox Virtual Machine image in order to create this exercise.

If we had you create an Essbase outline/cube from scratch, it would be several chapters in length; more if we wanted to do it justice. Instead, to save you time, we have provided the necessary Essbase outline (OTL) file, and data files necessary for you. In this way, you can focus on the quick load of the outline into your Essbase Server so that you can get right to the exercises. The set up is fairly straightforward and will be a breeze for anyone already familiar with developing an Essbase cube. Regardless, the steps for how to access the **Essbase Administration Services** (EAS) console, create a base Essbase application and database, and import the exercise files to the file system are explained in the section below.

### Prepping the VM image

It doesn't matter if you are using an existing Essbase server that you have access to or the Essbase Server on the SampleApp VirtualBox VM image, there are a few steps required to get going with the exercises. If you are using an existing Essbase Server on your network then you can skip to the section below called Prepping Essbase. Otherwise, complete this section for the SampleApp VM image.

---

**For More Information:**
Oracle Essbase and OLAP Integration

Downloading the VirtualBox application and setting up the SampleApp VM image is already explained in detail from the document that comes with the SampleApp VM download. Follow those instructions found in the PDF, which you may download from the following URL:

http://www.obillgbook.com/u/18

The following instructions assume that you are running the hosted VirtualBox application on a Windows operation system.

Starting the virtual machine image
The machine on which the VirtualBox application is installed is called the Host machine. The VM image, in this case the SampleApp VM Image, is referred to as the Guest. To start the VirtualBox guest image:

1. Navigate to Start | Programs | Oracle VM VirtualBox | Oracle VM VirtualBox.
2. Click on the SampleApp_<version> option in the VirtualBox Manager and then click on the Start button.
3. The VirtualBox guest VM image GUI will then start and should automatically login the user, Oracle. Once the Linux desktop is visible, continue to the next section below. If the Oracle user does not automatically get logged in, the password is oracle.

Starting up Essbase
The VM image makes starting up the standalone Essbase Server a breeze. From the Linux desktop:

1. Double-click on the Startup Scripts folder on the desktop. A folder list of executable files appears.
2. Double-click on the 4-StartESSB.sh option to begin the process to start the Essbase Server and the Analytic Provider Services (APS) server.
3. Click the Run Terminal button that appears in the resulting execution confirmation prompt.

An alternative to the above approach for starting the Essbase server is to open a terminal window and enter startESSB followed by pressing the Enter key. The startESSB command is an alias in the ~/.bashrc file that acts as a shortcut command to the startup shell script.

For More Information:
The terminal window will open and run. Once the terminal is started, continue to the next step.

**Starting up Essbase Administration Services (EAS)**

By default, EAS is not started. To start it:

1. Open a Terminal prompt on the Linux desktop via **Applications | Accessories | Terminal**.
2. Change the directory to the EAS directory path using the command:
   ```bash
   cd/epm/Middleware/user_projects/epmsystem1/bin
   ```
   Alternatively, depending on the SampleApp release:
   ```bash
   cd/home/oracle/epm/Middleware/user_projects/epmsystem1/bin
   ```
3. Run the Start EAS command: `./startEssbaseAdminServices.sh`. This command will run in the console and complete silently, returning to the command prompt, as shown in the following screenshot. In the next step, you’ll prepare the Essbase server to import the database outline and files required for this exercise:

```
[oracle@demo bin]$ ./startEssbaseAdminServices.sh
Starting EssbaseAdminServices.properties
Apache Ant version 1.7.0 compiled on December 13 2006
Buildfile: /epm/Middleware/EPMSystem11R1/common/config/11.1.2.0/resources/instance/start.xml
Finish EssbaseAdminServices.properties
[oracle@demo bin]$
```

Ultimately, for the communication with Essbase to perform properly the following three Essbase related components must be started:

- Essbase Server
- Analytic Provider Services
- Essbase Administration Services

The scripts `startEssbase.sh`, `startAnalyticProviderServices.sh` and `startEssbaseAdminServices.sh` must be executed in the order listed.

---

For More Information:

Prepping Essbase

As mentioned previously, we've developed the Essbase outline and ancillary files for you as a jumpstart to this exercise. Your initial job is to load these files onto your Essbase Server. There is really no automated means to import this information so follow the manual steps discussed in the following sections.

Creating the base Essbase application and database

In order to create the base Essbase application and database, carry out the following steps:

1. Open the Essbase Administration Console by navigating to the URL, http://<server_name>:10080/easconsole/console.html if using the SampleApp VM. If using an Essbase system on your network, your URL may vary slightly. You can also use the full EAS fat-client application for these steps, if it is available to you. You can download the EAS fat-client from the OTN by selecting the Essbase Clients Release option, http://www.obi11gbook.com/u/20, but the EAS thin web client will work fine for this exercise:

   If prompted with a launch screen, click on the Launch button.
   If prompted with the Java Web Start Opening easconsole. jnlp prompt, click on the OK button.

For More Information:

2. Accept any other prompts or warnings and continue.

3. Log in with the Administrator credentials and if you are using the SampleApp VM image, the username is admin and the password is password:

4. Expand the Essbase Servers node to ensure the server is started and you are properly connected.

If a TCP/IP error appears and you are unable to access the Essbase Servers list, even though you have started Essbase, APS, and EAS and you are working on the SampleApp VM image, the problem may be due to the VirtualBox Network Adapter settings for the VM image. Change the network settings to a Bridged configuration so that the VM image obtains its own DHCP assigned IP address, restart the VM and start again.

5. Expand the Essbase Servers node to show all available applications.

6. Right-click on the Applications node.
7. Go to **Create application | Using aggregate storage:**

8. In the name prompt, enter **OBIBook** and accept all of the defaults.

9. The application will be created and should show up under the Essbase Server’s Application list hierarchy on the left.

10. Right-click on the Essbase application, **OBIBook**, that you just created and select **Create database...**

11. Enter **OBIBook** in the field and click on the **OK** button.

12. Right-click on the **OBIBook** application once again and this time, go to the **Stop | Application** option. Accept the **Confirm Application Stop** prompt.

It is important that the application is stopped because in the next steps you will overwrite the very important `.otl` (outline) file that holds the dimensional structure and metadata for the Essbase cube.

---

For More Information:  
Migrating the Essbase files

As promised, we have the necessary Essbase files available from the Oracle BI 11g Book website or find the link on the Oracle BI 11g Book Forum, http://www.obillgbook.com/forums/.

1. Download the Essbase files from the links mentioned above or from Packt's website at http://packtpub.com.

2. Copy the .zip file to the following location on the Essbase Server, <EPM_FMW_HOME>/user_projects/epmsystem1/EssbaseServer/essbaseserver1/app/; for example, /epm/Middleware/user_projects/epmsystem1/EssbaseServer/essbaseserver1/app/.

You clearly need to have access to the physical server on which the Essbase Server resides in order to conduct this copy process. On the VM Image, you can either FTP the files or create a network share to place the files in the path.

3. Notice, that the folders in this directory correspond with the Essbase applications on the Essbase server. OBIBook is the main folder for the application you just created, but a subfolder by the same name exists which represents the database.

4. Unzip the downloaded file using the unzip command directly into this location. Several files will be overwritten in the subfolder, OBIBook, as well as some new ones will be added.

On Linux, the unzip command-line syntax would be: unzip <file_name>.zip

5. Accept any overwrite prompts so that the files are moved from the .zip file to the OBIBook folder and subfolder correctly.

Validating the Outline

The easiest way to check that the files were correctly copied into the new Essbase Server OBIBook application is to refresh the outline in EAS. Follow these steps:

1. Return to the EAS console.

2. Expand the Essbase Servers applications list and locate the OBIBook application.

3. Right-click on the OBIBook application and select Start Application.
4. Click on the Yes button to confirm the application start, if prompted.
5. Expand the OBIBook application node.
6. Expand the OBIBook database node.
7. Right-click on the Outline node and select the View option.

In the right-hand pane, the outline should render and contain the OBIBook outline that has been created for you. It should look like the following screenshot:

```
Outline: OBIBook (Active Alias Table: Default)
  |- Measures Accounts Dynamic: Compression <7>
  |   `- Wins (+)
  |   `- Set Wins (-)
  |   `- Set Loses (-)
  |   `- Game Wins (-)
  |   `- Game Loses (+)
  |   `- Aces (-)
  |   `- Break Points (-)
  `- Years Stored # Default # <1> (Label Only)
  `- Time Periods Time Stored # Default # <4>
  `- Time Series Dynamic <3>
  `- Scenarios Stored # Default # <1>
  `- Stadiums Stored # Default # <76>
  `- Court Types Stored # Default # <6>
  `- Players Stored # Default # <186> [Dexterity, Sox]
  `- Sox Attribute [Type: Text] # Default # <2>
  `- Dexterity Attribute [Type: Text] # Default # <3>
```

### Loading data into the cube

Load the data into the cube by referencing a load rule file and data file that were delivered with the downloaded .zip file you unzipped:

1. Right-click on the OBIBook database and select Load Data....
2. Click on the Find Rules File button at the bottom of the Data Load prompt.
3. Select the datatxt.rul file and click the OK button.
4. Click the Find Data File button at the bottom of the Data Load prompt.
5. Select the Essbase Server tab.
6. Select the OBIBook application level (star icon not database icon) option from the Look in: drop-down menu.
7. Select OBIBook.txt and then click on the OK button.
8. Click on the OK button in the Data Load prompt to begin the data load.
9. Click on the Close button to close the Data Load process when complete.
The data load will complete with warnings which is an expected behavior. This is because the text file used in the data load and Essbase outline were purposefully developed with inconsistencies for future training on cleaning and maintaining Essbase cubes. Data will still be successfully loaded to the OBIBook cube and you can use an MS Excel add-in or SmartView to verify this if you desire. Otherwise, continue on with the exercises below to model the Essbase cube into Oracle BI and create a report to view the Essbase data.

Anything needed to prep the Oracle BI Server?

Out-of-the-box, Oracle BI 11g installs with enough information so that it can recognize many metadata sources without the need for an additional driver or API reference configurations. The current Oracle BI version, 11.1.1.6 as of this writing, ships with the Essbase client API libraries, so that no additional configuration is required to connect to and model Essbase data. The Oracle BI server makes reference to any additional drivers and libraries it requires by calling an initialization command file whenever the Administration Tool is launched. Additional library references are required to communicate with separate applications such as Essbase and Hyperion Financial Management (HFM). The two files that assist in the reference of these client libraries are bi-init.bat and opmn.xml. However, the opmn.xml file does not get installed with the Oracle BI client tools only installation; only the full server installation.

The bi-init.bat file sets certain environment variable paths once the Administration tool is launched, so that the tool has a sense of where to locate certain libraries and configuration paths specific to the Oracle BI 11g client tools. Typically, there is no additional configuration required to this file.

The other configuration file, opmn.xml, is available under the Oracle BI 11g installation path, <FMW_HOME>\instances\instance1\config\OPMN\opmn\. The opmn.xml file is the configuration file which sets the locations for the Essbase or HFM client libraries amongst other settings specific to the OPMN. In the latest Oracle BI 11g version, the Administration tool is ready to import an Essbase cube as a data source out-of-the-box with no configuration necessary. Additional configuration of the opmn.xml file is required for HFM as it is a Microsoft Windows-based software product and the HFM client must be installed separately from the Oracle BI product.

For More Information:
Modeling Essbase into Oracle BI

The techniques that you will learn in this chapter follow best practices and will provide the result of front-ending Essbase data through Oracle BI. After the exercise, you'll get the right direction to further your learning so that eventually you can talk about this high-demand integration like a pro.

A bit of Essbase to Oracle BI knowledge

Importing a multidimensional data source such as Essbase into the Oracle BI RPD requires that Oracle BI is able to represent the multidimensional data, just as it would a relational data source. This provides the potential for federation, or joining, of heterogeneous data sources logically. As a multidimensional source is already well-structured based on the OLAP modeling techniques using facts and dimensions (star schema), the imported metadata about that source get translated cleanly to the Physical layer in the RPD. Extra columns and properties unique to Essbase are immediately available. These include Member Aliases, User Defined Attributes (UDAs), Member Number Ranking (memnor — numeric column sorting information instead of the default in Essbase which is alphabetical), and most importantly for hierarchical lineages, information about root, leaf, and parent levels. The hierarchy lineage information may take many extra steps to develop when using a relational source, but it is immediately included with a multidimensional source. When you import the Essbase source in the exercise below and expand a dimension hierarchy in the Physical layer, you will explore some of these columns and properties just described.

Something else to keep in mind is that the Oracle BI repository brings over almost all of the metadata inherent in an Essbase data source. Substitution variables (similar to variables in the RPD), at any scope, are imported, converted into Session variables, and an Initialization Block is created to dynamically cycle changes in those variables to reflect anticipated changes. Essbase generation names are reflected in the Physical layer which is why the best practice is to modify the Essbase outline generation names in advance for consistency. And, although the first Generation of an Essbase dimension is merely the dimension itself, an option exists in the Administration Tool to remove it from the Logical layer when modeling the repository, as it is typically not needed for analysis efforts.

For More Information:
The best practice of creating Essbase generation names can be learned from http://www.obi11gbook.com/u/25.

Hopefully, this section gave you a bit more information into how much effort has gone into ensuring that native Oracle Essbase information is reserved and leveraged as a data source in Oracle BI. Now, let’s begin the effort to bring the OBIBook cube into Oracle BI.

Importing Essbase as a data source

Now that the Essbase data source is alive and well on our Essbase server, we will need to open the Oracle BI Administration Tool to import the cube into our repository model.

1. Open the Oracle BI Administration Tool.
2. Open the Repository (Tennis.rpd) file in online mode using File | Open Online....
3. From the File menu, select Import Metadata....
4. Select Essbase from the Connection Type drop-down menu.
5. In the Essbase Server field, enter the IP address, or the server name, of the machine where your OBIBook cube is running.
6. Enter the Essbase Server administrator's credentials in the User Name and Password fields (admin/password).
7. Click on the Next button.
8. Expand the server name node under the Data source view column.
9. Select the OBIBook Essbase cube from the options and click the right-arrow from the center area between the Data source view and Repository View columns to import the cube.
10. Make sure that the Import UDAs checkbox is checked and leave all other default options in place.
11. Click on the Finish button.
The Essbase data source now appears in the Physical layer of the RPD. The import creates a Physical Database named after the Essbase Server from where you imported the source, a Physical Catalog named after the Essbase application, OBIBook, and a Cube Table named after the Essbase application database. Expanding the first level of the Court Type dimension, we can see that the Essbase generation naming (All Courts, Court Types, and so on) best practice is applied and is pulled from the Essbase server as anticipated:

By default, the dimension-level naming convention given to an Essbase data source imported into the RPD is very generic. To remedy this, in advance of importing the Essbase data source into the RPD, open the Essbase cube outline in EAS and change the name of the Essbase generations to reflect a more canonical form or business name. With this best practice, once the metadata import takes place, the generation names you've modified overwrite the default level names making the metadata identifiable and consistent with the source metadata.

Take a few seconds to review how the Physical layer structure of the OBIBook source looks in the RPD. Look at the measure and dimensions and compare them to the outline in EAS briefly to see that they are basically one-to-one.

A few OLAP adjustments before modeling
The Essbase data source in Oracle BI can be modified at the Physical layer so that its underlying metadata can be correctly interpreted by the Oracle BI Server engine at runtime. Of course, modifications to generation (translated to table column) names can be altered, and more just like any other source elements in the Physical layer. However, there are several other OLAP specific modifications that should be made, or at a minimum, assessed before beginning the full repository modeling process.

For More Information:
These include adjusting the **Hierarchy Type** and **Dimension Type** of the OLAP dimension, flattening a measure dimension's members to individual fact columns, creating columns for **User-Defined Attributes (UDAs)**, and creating columns for Alias Table member variations. Let's begin making these modifications in the following sections.

### Flattening the Measure dimension

Oracle BI pulls in an Essbase measures (Accounts) dimension as a measure hierarchy with a single measure representing all measures/accounts. In some basic reporting cubes, the number of measures or metrics that are used in the cube should be represented in a columnar structure, rather than in a hierarchy. In Essbase, having a measures dimension for the purpose of a financials-based data source, such as a **General Ledger (GL)** cube, usually means a **Chart of Accounts (COA)**. For example, the GL cube could have a measures hierarchy to identify Margin which would have children elements of Sales and **Cost of Goods Sold (COGS)**. That means Sales - COGS = Margin. Flattening the dimension hierarchy for a Chart of Accounts would not be ideal and would functionally break the desired GL reporting structure. However, for the OBIBook cube you will want to flatten the measures dimension due to the small number of metrics that are used and knowing that no real measure hierarchy exists in the cube design. To flatten the measures dimension, follow these steps:

1. Right-click on the **OBIBook** cube table from the Physical layer and select **Convert measure dimension to flat measures**.

2. As a result, the OBIBook cube table refreshes. Expand the cube table and notice that all of the elements in the measures dimension now appear as columns, as seen in the following screenshot:

   ![Flattened Measure Dimension Screenshot](image_url)

For More Information:

You will have noticed that before the measure dimension was flattened, only a single fact measure, **OBIBook - measure**, existed. It was replaced by the other seven measures in this process. Lastly, the **Measures** dimension still exists. However, right-clicking on it no longer shows the option to flatten the dimension. As a best practice, you’ll keep the **Measures** dimension intact for the life of the model. It does not get brought into the logical layer after you drag the cube table from the physical layer so there is no harm keeping it right where it is. You’ll drag over the physical layer objects to the logical layer very soon.

**Getting the UDAs**

The UDAs that are assigned to any level of a dimension can be pulled in as separate columns. You can do this for all UDAs across the entire cube, or you can focus on leveraging only the UDAs in one or more dimensions. The OBIBook cube has only one dimension using UDAs - **Players**.

1. Right-click on the **Players** dimension in the Physical layer and go to **Create columns for UDA | Top Country**:

   ![Create columns for UDA](image)

   *The option for **All UDAs** exists but here we are trying to highlight a minimalist's approach to getting at the Essbase data.*

2. Expand the **Players** dimension and you’ll notice how the **Top Country** UDA has become a column at each level in the Players dimension hierarchy.

**Dimension and hierarchy types**

All Essbase dimensions hierarchies are imported with the default hierarchy type of Unbalanced. An **Unbalanced** hierarchy type is what is referred to as a "ragged" hierarchy where not all levels within an Essbase hierarchy have the same depth. This hierarchy type will work for most dimensions but the other two hierarchy types allowed for Essbase modeling are **Fully Balanced** and **Value**. Fully Balanced refers to a level-based structure where all levels roll-up nicely to their parent levels.

---

**For More Information:**

An example would be a time dimension hierarchy of Year, Half-Year, Quarter, and Month. The Value hierarchy type truly refers to a parent-child structured dimension where all members have the same type such as an employee dimension, regardless if its hierarchy is balanced or not. The hierarchy type property helps the Oracle BI Server understand how to define the relationship of members within levels, such as when a user is drilling within a hierarchy in a report.

A guide to understanding OLAP hierarchies, dimensions, and levels is here:
http://www.obi11gbook.com/u/21

Dimension types are used to define the dimension as it was imported from Essbase. The dimension types are **Unknown, Time, Measure, Attribute, and Other**. By default, as you'll see by examining the OBIBook dimension hierarchies, Oracle BI correctly aligns Essbase attribute dimensions with the Attribute dimension type, the time dimension with Time dimension type, and so on.

To assess and begin modifying these properties, follow these steps:

1. In the Physical layer, double-click on the **Players** dimension to open the Physical Dimension properties prompt.
2. In the **Hierarchies** tab, select the **Players** hierarchy available from the pick-list, and then click on the pencil icon to edit the hierarchy.

   Notice that in the **Physical Hierarchy** prompt that appears, the **Dimension Type** is set to **Other** and that the **Hierarchy Type** is set to **Unbalanced**. This is typical for a standard Essbase dimension.

3. Repeat the same steps as above for the **Sex** and **Time Periods** dimensions. Notice how the **Hierarchy Type** for the Sex and Time Periods dimension are also **Unbalanced** but the Sex dimension's Dimension Type is **Attribute dimension** and the **Time Periods** dimension's Dimension Type is **Time dimension**.
4. Close all the prompts.

This quick look at dimension and hierarchy types should have given you a good idea of how Essbase dimension metadata gets translated into the RPD. No changes were made to the dimension or hierarchy types but now you know where to go in order to ensure that Essbase specific dimension types are being properly construed in Oracle BI.

---

For More Information:
Getting a quick win

Modeling Essbase has its learning curve complications. However, to get a quick win, such as in a POC, and start seeing data from the Essbase cube right away, it only takes a few steps. Once you complete those steps you'll be able to login to the Oracle BI/analytics application portal and begin pulling data from the Essbase data source via a new Analysis request.

Let's continue with a best practice first. One option that you can set in order to make your model much cleaner is the option to skip the inclusion of Essbase generation 1 levels from showing in the Logical layer. Remember, the first Essbase generation is just the name of the dimension and it is simply redundant in the RPD. To set this option, follow these steps:

1. Select Tools | Options... from the file menu bar.
2. Select the General tab and check the checkbox for Skip Gen 1 levels in Essbase drag and drop actions.
3. Click on the OK button to close the Options prompt and let's continue on with the quick win.
4. In the Physical layer left-click on the OBIBook Essbase cube table icon, representing the Essbase database data source, and drag it onto anywhere in the white space of the Business Model and Mapping layer. This creates a Business Model for the OBIBook Essbase database.
5. Double-click on the OBIBook Business Model to open the properties prompt and uncheck the Disabled checkbox so that the model is enabled.
6. Expand the OBIBook Business Model in the Logical layer.
7. Inspect how the OBIBook source was translated from the Physical layer to this logical layer.

   Look at the logical fact table, OBIBook. It was given the name of the Essbase database by default. You can rename this to Measures or Facts if desired.

8. Click and drag the OBIBook Business Model from the BMM onto anywhere in the white space of the Presentation layer.
9. Save the RPD and check in your changes if prompted. If prompted for a Global Consistency Check, confirm that option as well.

For More Information:
10. Log in to the Oracle BI /analytics application and view this new Subject Area. You may need to click on the Refresh the Metadata option in the Administration page of the /analytics application to see the new Subject Area appear in the list of available Subject Areas. Alternatively, you can restart the Presentation Services System Component using Enterprise Fusion Control options.

11. Create a simple query via an Analysis Request using the skills you learned in Chapter 10, Creating Dashboards and Reports.

For example, use the following columns and metrics to create a report that looks like the following screenshot, with **Year**, **Quarter**, **Set Wins**, and **Set Loses** then view the Results tab:

The steps you've covered is often enough to get a Proof-of-Concept (POC) out the door for a department at your organization, such as the Finance Department, so that they can become believers in the power of Oracle BI's technology.

**Incremental importing of Essbase metadata**

Like any other data source you may leverage in your Oracle BI repository, from time to time you will need to adjust the structure of the underlying data source and have it represented correctly within your RPD. If the metadata of an Essbase data source changes, you can re-import the metadata without much issue. This is referred to as an Incremental Import; where after you've conducted an initial metadata import you conduct subsequent imports of the same Essbase source regardless of the intensity of changes to the underlying source system or RPD.

Here are some caveats to keep in mind when requiring an incremental import of your Essbase cube and how it affects your existing Oracle BI model:

- All initial import objects and modifications are retained in the case of an incremental update.
- Removing an element/object from the Essbase cube source (via EAS, for example) does not delete the corresponding item from the Physical layer upon incremental import.

For More Information:

Oracle Essbase and OLAP Integration

- Renaming an object in the Essbase cube source re-imports the changed object as a new object. It does not modify the existing object in the Physical layer.
- Mainly all objects and new columns (UDAs, Alias Table, and so on) are retained in the Physical layer post incremental import. You must delete the objects in the Physical layer then re-import if wishing to revert, or go with the new source system changes.
- The addition of an Essbase source's dimension or generation within a dimension will require some remodeling effort if the source has already been used to create a Business Model. This effort would be similar to incorporating a new dimension table in a relational model where joins to the logic fact table, and so on must take place.

To re-import metadata for the Essbase cube:

1. Right-click on the **Connection Pool** object of the Essbase Server Physical Database in the Physical layer and select **Import Metadata**:

2. Select the Essbase cube from the **Data Source view** column and move it using the center arrow in the same way you imported the data source earlier.

3. A confirmation prompt will appear informing you that "the following cube(s) already exist". Click the **OK** button to proceed:

4. Click on the **Finish** button to close the prompt.

For More Information:
That's all it takes to do a basic incremental import. After an incremental import it is always a good idea to look through the Essbase source's dimension hierarchies and other properties to ensure everything you meant to be preserved is indeed so and that the incremental changes did indeed come through. You should always make a backup copy of your RPD before conducting this type of operation.

**Federation of data**

As discussed in previous chapters, the federation of data sources through Oracle BI is one of the most flexible features of the tool. The ability to take two or more disparate data sources and blend the data in a way that allows for enterprise data interoperability in order to satisfy analytic goals is amazing. Make no mistake about it; Oracle BI is no substitute for an ETL tool but just think of the possibilities for looking at Essbase data and Oracle database data in the same query, from a single reporting system, referencing the same level of granularity. All of that without the need to rework the underlying data source structure of Essbase or the Oracle database—in Oracle BI it just works.

**Oracle by Example** has a great tutorial on how to federate an Essbase and relational database source in Oracle BI at the following URL:

http://www.obi11gbook.com/u/19

Try to combine the OBIBook Essbase cube with the Tennis database.

**Oracle BI/EPM roadmap**

The integration between Oracle BI 11g and the Oracle EPM tools can only get better. In the Oracle BI 10g version, an attempt was made to bridge the Oracle BI dashboards with that of the Hyperion Workspace. This was a tricky integration and ultimately not the path Oracle wished to continue on. However, several successful integration points between the two still exist and work very well. One such integration point is that of Hyperion Shared Services security, one of Essbase's security provider options which, when integrated with Oracle BI, allows for existing Essbase cube row-level security to persist for any users accessing Essbase data via Oracle BI.

For More Information:

Workspace integration

Hyperion Workspace is the portal for Hyperion Products that became very popular in System 9 (version 9.x) of the Hyperion tool suite. Introducing Oracle BI users to the Hyperion product suite was a great initial concept, but was not carried over in the first minor releases of Oracle BI 11g. The next slated attempt at connecting these two applications will be via Fusion Middleware and is set to be provided in a later version release of Oracle BI 11g.

Software license combo

Oracle believes that their Business Intelligence and Enterprise Performance Management tools are the best in-class offerings on the market. They believe that an organization can get a great amount of business insight by using these two classes of tools in tandem. In an effort to continue that message strategy, Oracle has provided a special licensing package referred to as the Oracle BI Foundation Suite. This combo package of BI tools includes the Oracle BI 11g Enterprise Edition license, as well as licenses for Oracle Essbase, Oracle Essbase Analytics Link (EAL), Oracle BI Mobile, and Oracle Scorecard and Strategy Management. Providing this new discounted option of licenses encourages users to partake in the Oracle BI roadmap by giving organizations a chance to leverage the power of the tools Oracle believes will provide an even greater boost in decision making analytics than using OBIEE itself.

A review – what I should now know!

For self-review and a recap of the chapter, here are a few questions. There is no answer key. These questions are for your own reflection on the chapter:

1. Will adding one or members to an existing generation in the Essbase outline affect the re-import of an Essbase data source in the RPD?
2. What is the best practice for readying the Essbase data source into Oracle BI?
3. What are the differences between MOLAP, HOLAP, and ROLAP?

Additional research suggestions

For additional information, refer to the following sources:

- **Financial Management Analytics**: [http://www.obi11gbook.com/u/24](http://www.obi11gbook.com/u/24)
Summary

What has been discussed in this chapter answers many of the questions that we get from clients regarding the integration between Oracle BI 11g and Essbase. The details discussed in this chapter will provide any organization the means to begin exploring the integration with their Essbase systems. Or, perhaps it will provide enough clarity to commission a new project in order to bring their Essbase environment up-to-date and take advantage of this splendid Fusion Middleware integration. In this chapter you learned about the many OLAP types, how to import an existing Essbase outline to a different Essbase server, and you learned about the Oracle SampleApp Virtual Machine image's Essbase options. Exploring the Oracle BI Administration Tool, you learned how to import an Essbase cube as a data source and how to quickly model it through from the Physical layer to the Presentation layer. Finally, you examined several best practices for the Essbase and Oracle BI 11g integration that have slight to major impacts on development and maintenance efforts. We hope you've enjoyed this chapter and this book and look forward to your feedback, questions, and comments on the Oracle BI 11g Book forums, http://www.obi11gbook.com/forums/.
Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

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