CHAPTER 1

Oracle Business Intelligence Overview and Architecture
Oracle Business Intelligence 11g Release 1 is a platform for delivering business intelligence across a wide range of data sources and to a wide range of audiences within the enterprise. You can consider it a “toolkit” in that, in itself, it does not come with any prebuilt reports, data, or other content, although as we will see later in this chapter, you can license content from Oracle Corporation and other providers that can be used with Oracle Business Intelligence.

Introducing Oracle Business Intelligence

As an end user, your first encounter with Oracle Business Intelligence would be when logging in to, and interacting with, a web-based dashboard such as that shown in Figure 1-1. Oracle Business Intelligence dashboards are made up of pages of analyses, displayed as tables, pivot tables, charts, gauges, or other views using data from potentially many sources. These can be interacted with, allowing the end user to, for example, start with a set of summarized figures and then progressively drill into more detail. Oracle Business Intelligence dashboards are typically highly graphical and provide a familiar, point-and-click environment for users to explore their data.

FIGURE 1-1. An Oracle Business Intelligence dashboard
Chapter 1: Oracle Business Intelligence Overview and Architecture

For end users who wish to create their own reports, or those who wish to investigate their data in more detail, Oracle Business Intelligence allows users to create ad-hoc reports using data items taken from subject areas described using familiar business terms. Figure 1-2 shows a typical ad-hoc report, with a typical subject area made up of tables, columns, and hierarchies on the left-hand side of the screen that can be used to create table, chart, and other data views displayed on the right-hand side.

As well as providing an ad-hoc query environment suitable for data exploration, Oracle Business Intelligence also comes with tools for publishing reports in formats such as Adobe PDF and distributing them to large numbers of recipients through technologies such as e-mail.

Analyses created using Oracle Business Intelligence can also be accessed through collaboration and office products such as Microsoft Outlook, Microsoft Word, Microsoft PowerPoint, and Microsoft Excel, or can be embedded directly into applications such as Oracle E-Business Suite, Siebel Customer Relationship Management (CRM), or in the new Fusion Applications from Oracle Corporation.

The 11.1.1.6 release of Oracle Business Intelligence 11g, on which the examples in this book are based, introduces new capabilities and visualization options, including the ability to create scorecards and key performance indicators, display data in the form of maps, and integrate with applications and business processes through a feature called the Action Framework. We will look at these capabilities in more detail in later chapters of this book.

**FIGURE 1-2.** An Oracle Business Intelligence ad-hoc query
Chapter 1: Oracle Business Intelligence Overview and Architecture

The Development Toolkit
As developers, you have a number of tools within Oracle Business Intelligence that you can use to develop business intelligence solutions. The main tool that you will use is the Oracle Business Intelligence Administration tool, a Microsoft Windows-based tool that is used to define and then maintain the business metadata layer known as the Oracle BI Repository, used for creating reports and analyses. Several chapters of this book are devoted to this tool, and as a developer you will need to understand in detail the functionality of this tool and the Oracle BI Repository.

Figure 1-3 shows the Oracle Business Intelligence Administration tool, with a repository open for editing. The tool is also used for managing connection details through to data sources, defining security policies that control users’ access to data, and performing a number of administration tasks such as defining variables, managing caching, and checking the status of the cluster.

From the 11g release of Oracle Business Intelligence, a number of systems administration tasks previously carried out using the Oracle Business Intelligence Administration tool are now performed using Oracle Enterprise Manager Fusion Middleware Control. These tasks include enabling and disabling caching, setting the cache size, and managing the status of the system components in a cluster.

Other tools for developers provided as part of Oracle Business Intelligence 11g Release 1 include the Catalog Manager, a Java application used for managing the catalog of reports, dashboards, and other business intelligence objects; Oracle Enterprise Manager Fusion Middleware Control,
for administering the business intelligence platform; and Oracle WebLogic Server Administration Console, for controlling the functionality of the WebLogic Server application server.

Reports, analyses, and dashboards are created using web-based authoring tools that do not require any separate software to be installed on developers’ desktops. Oracle Business Intelligence 11g Release 1 brings together all report-authoring tools into a single integrated environment using the same semantic model, and later chapters in this book will show you how easy it is to create compelling, interactive dashboards using these tools.

**Platform Support**

At the time of writing this book, Oracle Business Intelligence is at release 11.1.1.6 and can be installed on a number of Microsoft Windows, Linux, and Unix platforms, running on both 32-bit and 64-bit processors. Because the report and dashboard-authoring tools within Oracle Business Intelligence are mostly web-based, any operating system that supports these web browsers can be used to create reports. There are separate certifications for server and client tools within Oracle Business Intelligence, and you should refer to the Oracle Technology Network web site, and in particular the System Requirements and Supported Platforms for Oracle Business Intelligence Suite Enterprise Edition 11gR1 document, to obtain the latest list of supported platforms and operating systems.

**NOTE**

The list of certified operating systems and platforms can change from release to release, and you should check the System Requirements and Supported Platforms for Oracle Business Intelligence Suite Enterprise Edition 11gR1 document, available on the Oracle Technology Network web site (http://otn.oracle.com), for your particular version of Oracle Business Intelligence.

**How Does Oracle Business Intelligence Work?**

So you now know that Oracle Business Intelligence comes with a number of end-user tools for developing and viewing reports, together with developer tools for administration, creating the semantic model, and maintaining the system. But how does Oracle Business Intelligence work, how does it access your various data sources, and what use does it make of other business intelligence systems such as data warehouses, online analytical processing (OLAP) servers such as Oracle Essbase, or data that might be of interest in your applications or company databases?

At a high level, Oracle Business Intelligence uses a four-tier architecture that provides access to your data through two main servers and a semantic model. Figure 1-4 shows a high-level schematic for Oracle Business Intelligence, with your data being accessed through two servers—the Oracle Business Intelligence (BI) Server and the Oracle BI Presentation Server—before it is presented to end users through a web browser.

Considering this four-tier architecture from the perspective of an end user requesting a dashboard of business information, the components within Oracle Business Intelligence perform the following high-level functions to return data to the user:

1. The web browser requests a dashboard of data, consisting of analyses, published reports, and other BI content.

2. This request is received by the Oracle BI Presentation Server, which translates requests for individual analyses and reports into logical SQL queries. These logical queries are then passed to the Oracle BI Server.
Oracle Business Intelligence Overview and Architecture

3. The Oracle BI Server takes these logical SQL queries, written against the semantic model contained in the Oracle BI Repository, and translates them into native SQL and MDX queries that are then sent to the underlying data sources.

4. The underlying data sources process the native SQL and MDX queries, and return results to the Oracle BI Server.

5. The Oracle BI Server returns a data result set to the Oracle BI Presentation Server. In instances where more than one data source is needed to satisfy the query, the BI Server is capable of combining multiple data sets into a single set of results.

6. Finally, the Oracle BI Presentation Server presents the results back to the end user, in the form of analyses, published reports, dashboards, and other BI content.

Unlike many other business intelligence tools that combine data presentation with query generation in a single server, Oracle Business Intelligence splits these functions into two separate servers:

- **Oracle BI Server**  This server provides simultaneous connectivity to heterogeneous data sources, a calculation and data federation engine, access to a semantic (metadata) model, and a security layer.

- **Oracle BI Presentation Server**  This server connects to the Oracle BI Server and provides users with a catalog of analyses, reports, and dashboards that they can use to analyze their data.

**FIGURE 1-4.** High-level Oracle Business Intelligence schematic

---

**Oracle BI Presentation Server**

**Oracle BI Server**

Data sources (data warehouse, OLTP databases, OLAP servers)

Dashboard and ad-hoc querying through web browser
Chapter 1:  Oracle Business Intelligence Overview and Architecture

Oracle Business Intelligence 11g actually makes use of other servers to handle clustering, scheduling of reports, and other services; but, for now, consider these two servers the “core” of the Oracle Business Intelligence’s functionality.

So Where Does the Data Come From?
As you will have seen from the above schematic, Oracle Business Intelligence does not itself hold data; instead, it uses a metadata layer to create a “virtual dimensional model” over one or more data sources and then generates SQL and MDX queries to retrieve data, on demand, from these data sources for presentation back to the user. As such, it leverages any investment you have made in data warehouse technology such as Oracle Database Enterprise Edition, or in OLAP technology such as Oracle Essbase, rather than replacing the need for them.

While Oracle Business Intelligence can optionally be configured to hold a cache of data to enable faster display of results, queries that it generates are otherwise sent directly to your underlying data sources. It follows, therefore, that your underlying data sources should be as optimized for queries as possible. Furthermore, the Oracle BI Server can also take advantage of any analytic functionality that is available on a particular data source to enable more efficient processing by “passing down” calculations to the underlying data source. Therefore, the recommended, optimal data source for Oracle Business Intelligence would be an enterprise data warehouse running on a database platform such as Oracle Database Enterprise Edition, potentially supplemented or enhanced by an OLAP server such as Oracle Essbase or the OLAP Option for the Oracle Database Enterprise Edition, to provide fast access to aggregated data.

However, Oracle Business Intelligence also has the ability to connect to more than one data source and “join together” results from each one into a single data set, giving you the ability to create “virtual” data warehouses made up of data taken in real time from separate databases, which can be departmental data warehouses, data marts, or even online transaction processing (OLTP) databases or nonrelational database sources such as OLAP servers, files, or sources such as Microsoft Excel spreadsheets. This ability to work with “federated” data sources gives you great flexibility in how you design your reporting system, allowing you to, for example, source the majority of your reporting data from a data warehouse but supplement it with data sourced in real time from a range of applications, file sources, and OLAP servers, as shown in Figure 1-5.

This approach becomes possible due to two key features provided by the Oracle BI Server:

- The semantic model, which can create a metadata model over multiple data sources from different vendors, presenting users with a single, unified view over their data regardless of the data source
- The ability of the BI Server to generate native, optimized queries for each data source and to combine the results returned into a single result set

So, it’s clear that the BI Server and the semantic model that it uses are key to how Oracle Business Intelligence provides access to data. With this in mind, let’s take a look at how the semantic model works and how it structures data so that it is optimized for querying.

The Oracle Business Intelligence Semantic Model
The Oracle Business Intelligence semantic model has three main objectives:

- To represent your enterprise’s data as a logical dimensional model
- To map this logical dimensional model onto the data sources used by the enterprise
To provide personalized views over the logical dimensional model for subsets of users giving them access to just the data they need while preserving, “under the covers,” a single unified business intelligence data model.

Semantic models in the Oracle Business Intelligence repository therefore have three distinct layers:

- **Physical layer**  This layer contains metadata on the physical databases and other data sources that provide data for the semantic model.

- **Business Model and Mapping layer**  This layer contains the logical dimensional model defined for the business.

- **Presentation layer**  This layer provides personalized subsets of the logical dimensional model tailored for different audiences.

Data flows through the semantic model, as shown in Figure 1-6, from the Physical layer, through mappings into the Business Model and Mapping layer, and is eventually accessed by end
users through the Presentation layer. The semantic model is defined and maintained using the Oracle Business Intelligence Administration tool and stored in the Oracle BI Repository, and Chapters 3 and 4 show you how semantic models can be created that access data from relational, OLAP, files, and other nonrelational sources.

Packaged Business Intelligence Solutions

The fact that Oracle Business Intelligence is, in effect, a toolkit that does not in itself provide any data or reports that you can use “out of the box” means that, realistically, you will need to spend a significant amount of time developing a solution before your users can start analyzing their data. In addition, because many organizations have standardized on packaged enterprise resource planning (ERP) systems such as Oracle E-Business Suite and PeopleSoft Enterprise or customer relationship management (CRM) suites such as Siebel CRM, the work you would be doing might essentially be “reinventing the wheel,” as many organizations would have had requirements similar to yours in the past and created similar business intelligence solutions to deliver similar dashboards and reports.

Oracle Corporation, as well as third-party vendors such as Noetix, have addressed this opportunity by developing packaged sets of dashboards, data models, and data extraction routines that you can install, along with Oracle Business Intelligence, to provide dashboards and reports within days or weeks rather than the usual months that are required to create a custom solution. The Oracle Business Intelligence Applications, from Oracle Corporation, are a suite of
Oracle Business Intelligence 11g Developers Guide

packaged BI products built around Oracle Business Intelligence that provides applications such as the following:

- Financial Analytics
- Human Resources Analytics
- Project Analytics
- Procurement Analytics
- Supply Chain Analytics

In addition to these, it provides other, industry-specific “vertical” packaged applications for the financial services, pharmaceuticals, and other industries. Figure 1-7 shows the relationship between Oracle Business Intelligence Applications and Oracle Business Intelligence, and how data for this combined system is either accessed directly from application and database data sources, or through a prebuilt data warehouse fed by predefined data extraction routines.

These packaged business intelligence applications provide a prebuilt and extensible data warehouse; data extraction routines that provide preconfigured access to Oracle E-Business Suite, SAP, Oracle PeopleSoft, and Oracle Siebel applications; as well as content for use with Oracle Business Intelligence. As these data models and extraction routines are based on industry-standard tools and databases, they can be customized after installation; many customers use them as the starting point for their entire business intelligence solution, using the packaged data model and

![FIGURE 1-7. The Oracle Business Intelligence Applications architecture](image-url)
Chapter 1: Oracle Business Intelligence Overview and Architecture

Data extraction routines as the starting point, and then customizing and extending them to cover the full range of their enterprise data.

The use and deployment of packaged solutions is outside the scope of this book, which instead focuses on the Oracle Business Intelligence platform itself and the creation of custom solutions. However, once you are familiar with these topics you should take time to investigate these packaged solutions, which you may wish to deploy along with your custom development in order to bring down the total time, and cost, of your business intelligence deployment.

**Oracle Exalytics In-Memory Machine**

For customers looking for “speed-of-thought” analysis of very large sets of detail-level data, Oracle Exalytics In-Memory Machine is a combination of hardware and Oracle Business Intelligence software that uses an in-memory database cache and special management tools to manage the cache.

Figure 1-8 shows the architecture for Oracle Exalytics In-Memory Machine, which comes with Oracle Business Intelligence and Oracle Essbase installed on the Exalytics hardware device, along
with Oracle TimesTen for Exalytics as the in-memory database. Exalytics is typically used by those customers looking to interactively analyze large volumes of data in a very graphically rich environment and is sold as an “appliance” that you can add to your data center and connect, via InfiniBand to Oracle Exadata or via Ethernet, to your corporate databases and data warehouses.

We will look in more detail at Oracle Exalytics In-Memory Machine in the final chapter of this book, including how it is configured, how the in-memory cache is used, and the new visualization types that are enabled by this high-performance platform for your BI system.

What Does Oracle Business Intelligence “Not Do”?

When considering a new software tool, it’s worth understanding also what it does “not do.” While Oracle Business Intelligence is a suite of products that provides a wide range of analysis tools, data source adapters, and—with the Oracle Business Intelligence Applications—prebuilt content for a wide range of business applications, it is worth understanding what it is not:

- It is not a replacement for Microsoft Excel. Microsoft Excel provides a free-form environment for analyzing and reporting on data, is programmable, and places few restrictions on how data is presented, input, analyzed, and distributed. Oracle Business Intelligence, in contrast, provides a structured analysis environment based around a dimensional model, primarily providing analytical views of data through a web-based dashboard. Individual analyses can, however, be imported into Microsoft Excel, either through an export from the dashboard or through a plug-in to Microsoft Excel, and the BI Server can also act as an ODBC data source for Excel (and other clients).

- Because it uses a logical dimensional model for presenting data to users, it is not generally suitable for querying unstructured data, being better suited to reporting on transactional, data warehouse, and OLAP data sources.

- While it can report against the transactional databases used by applications such as Oracle E-Business Suite, due to the complexity and existing workload on these databases, repositories created directly against them are generally not recommended; instead, data from the transactional databases used by these applications is generally extracted into a data warehouse, or OLAP cube, and then queried from that location by Oracle Business Intelligence.

- Although it can access data in Oracle Essbase or other OLAP servers, Oracle Business Intelligence is not itself a multidimensional OLAP server; instead, it could be thought of as a relational OLAP (ROLAP) server, providing access to a dimensional model through (virtual) relational data structures, themselves mapped to either relational, file or multidimensional physical data structures.

A History of Oracle Business Intelligence

There have been several “business intelligence” products released by Oracle Corporation over the years, and you may have come to this book planning an upgrade from a previous generation of tools such as Oracle Discoverer or Oracle Reports. These tools were developed in-house by Oracle Corporation’s developers and were designed to work primarily with Oracle’s own database, application server, and security products. As Oracle Corporation moved from being solely focused on database technologies and started to make acquisitions in the middleware, applications, and
Chapter 1: Oracle Business Intelligence Overview and Architecture

infrastructure industry sectors, it became clear that it needed a “next-generation” business intelligence platform that did not have such a dependency on Oracle database technologies but that could still take advantage of them if, as is often the case, the customer had used Oracle technology for their data warehouse.

The acquisition of Siebel CRM Systems, Inc., though primarily for their Siebel CRM platform and their extensive customer base, presented Oracle with an opportunity to update their business intelligence platform through another, lesser-known product that Siebel offered; Siebel Analytics. Though well regarded in the industry, Siebel Analytics was not as well known as similar products from vendors such as Business Objects (now part of SAP) and Cognos (now part of IBM), but the Siebel Analytics platform met many of Oracle Corporation’s requirements for a next-generation business intelligence platform in that it could access data from many different data sources, had an industry-leading metadata layer (the “semantic model”), very user-friendly dashboards and reporting tools, and also came ready-integrated with popular ERP and CRM systems such as SAP, Oracle E-Business Suite, PeopleSoft—and, of course, Siebel CRM—in a package called Siebel Business Analytics.

Oracle announced in 2005 that what was previously called Siebel Analytics would now be adopted by Oracle Corporation as their strategic business intelligence platform, and they renamed it Oracle Business Intelligence Enterprise Edition. The existing Oracle Discoverer and Oracle Reports tools would be packaged as Oracle Business Intelligence Standard Edition, and while customers would not be forced to upgrade from the older toolset to the new one, in time upgrade tools and services would be made available to make this process easier for those customers who chose to do so.

Siebel Analytics was itself, though, developed outside of Siebel CRM Systems, Inc., and was in fact originally developed by a technology startup out of Minneapolis, MN, called nQuire. Led by Larry Barbetta and a number of ex-Platinum Software engineers and product managers, nQuire released the nQuire Server, the predecessor to what eventually became the Oracle BI Server, back in the late 1990’s as a stand-alone analytics and search server that featured connectivity to a wide range of data sources. The nQuire Query Server featured a metadata model (which eventually became the semantic model) that provided a virtual logical dimensional model over these data sources, and over time the nQuire Query Server was joined by nQuire Answers and nQuire Delivers, giving us the core of what is now Oracle Business Intelligence. nQuire was itself acquired by Siebel CRM Systems, Inc., in October 2001, and it developed the product further and licensed data models and data extraction routines from Informatica Corporation that now form the core of the Oracle Business Intelligence Applications. So while what you know as Oracle Business Intelligence may be a product that is only a few years old, the core of the product itself can be traced back to groundbreaking work done by the nQuire team back in the mid-1990’s.

Oracle Product Release History

Shortly after Oracle Corporation acquired Siebel Systems, what was Siebel Analytics was renamed Oracle Business Intelligence Enterprise Edition, whilst Siebel Business Analytics was renamed Oracle BI Applications. The initial release of Oracle Business Intelligence was the 10g 10.1.3.2 version, with subsequent major releases of Oracle Business within the 10g timeline:

- **Oracle Business Intelligence 10.1.3.2**  First “Oracle-branded” release of Oracle Business Intelligence; introduced Oracle BI Publisher as a replacement for Actuate, a new Oracle “look and feel” with 64-bit support; time-series functions and features for multiuser development.
Oracle Business Intelligence 11g Developers Guide

- **Oracle Business Intelligence 10.1.3.3.x**  MS Office integration, support for metadata import through the Oracle Call Interface; support for embedded database functions; initial support for Oracle Essbase as a data source.

- **Oracle Business Intelligence 10.1.3.4.x**  Integration with Hyperion Workspace; integration with Oracle Smart View and Oracle Smart Space; the introduction of a utility to upgrade Oracle Discoverer End-User Layers to Oracle Business Intelligence repositories.

In addition, at the time of writing this book there have been three major releases as part of the 11g Release 1 timeline:

- **Oracle Business Intelligence 11.1.1.3**  Initial 11g release, provided new “look and feel,” support for KPIs and scorecards, the Action Framework, and other new features. Platform support limited to Microsoft Windows, Linux, and IBM AIX, with Oracle WebLogic Server as the sole JEE (Java Platform, Enterprise Edition) application server.

- **Oracle Business Intelligence 11.1.1.5**  Oracle Extension of platform support to HP/UX and Sun Solaris, introduction of iOS (Apple iPhone, Apple iPad) native clients, and restoration of data sources temporarily desupported in the 11.1.1.3 release.

- **Oracle Business Intelligence 11.1.1.6**  Support for the Exalytics In-Memory Machine platform and integration with version control tools. New visualization options and new certified data sources, including Oracle TimesTen.

In addition, there are three editions of Oracle Business Intelligence, the first two of which in the following list are within the scope of this book, and the third that is not:

- **Oracle Business Intelligence Enterprise Edition**  The full set of business intelligence tools and servers. This book will concern itself primarily with this edition.

- **Oracle Business Intelligence Standard Edition One**  “Departmental” or budget version of Oracle Business Intelligence that comes with certain restrictions on, for example, the number of allowable users and CPUs. Check with your Oracle representative or http://www.oracle.com for up-to-date details on this and other product packages.

- **Oracle Business Intelligence Standard Edition**  Somewhat confusingly, a different family of products altogether. This is a container for “legacy” Oracle Business Intelligence tools such as Oracle Discoverer and Oracle Reports. You can use upgrade tools to migrate Discoverer metadata to Oracle Business Intelligence Enterprise Edition and Standard Edition One; but, otherwise, this edition is outside the scope of this book.

Now that you know a little more about Oracle Business Intelligence's background and a little of its history, let’s take a look in more detail at the individual products within the platform, its architecture, and how it works “under the covers.”

**Oracle Business Intelligence 11g Release 1 Architecture**

Earlier in this chapter, we looked how the “heart” of Oracle Business Intelligence is the Oracle BI Server and the Oracle BI Presentation Server. The Oracle BI Server provides native, federated access to data sources, together with security, calculations, and data navigation. The Oracle BI
Chapter 1: Oracle Business Intelligence Overview and Architecture

Presentation Server connects to the BI Server to obtain data, which it presents to users in the form of analyses, reports, and dashboards.

In addition to these two servers, there are three other servers that work with them to provide core Oracle Business Intelligence functionality:

- **Oracle BI Cluster Controller** This server provides a central point of access for the Oracle BI Presentation Server when two or more BI Servers are working together in a cluster, together with load-balancing, failover, and other cluster services.

- **Oracle BI Java Host** This server works alongside the BI Presentation Server to provide connectivity to Java tasks and the Java-based Oracle BI Publisher, as well as to support chart generation.

- **Oracle BI Scheduler** This server is used to schedule and automate the production and distribution of analyses, as well as to automate workflow tasks based around business intelligence functionality.

These three servers, together with the Oracle BI Server and the Oracle BI Presentation Server, are known in Oracle Business Intelligence 11g Release 1 terminology as *system components*, and they run as services and servers directly on the host platform. They are operating system executables written in C-based languages.

To create the link between an end user’s web browser and the dashboards, analyses, and reports provided by the Oracle BI Presentation Server, a Java application called the Oracle BI Analytics Plug-In runs in a Java application server and routes incoming requests through to the BI Presentation Server. (Currently, only Oracle WebLogic Server is supported, but later releases of Oracle Business Intelligence should support other application servers.) A simplified schematic of the Oracle Business Intelligence 11g system components, together with the Oracle BI Plug-In, is shown in Figure 1-9.

This basic, internal architecture has stayed consistent since the days of nQuire and Siebel Analytics and is still at the core of the 11g Release of Oracle Business Intelligence. It, together with two additional Java server applications for publishing reports and connecting to Microsoft office, is largely the architecture of the 10g release of Oracle Business Intelligence and would run fairly comfortably on a smaller server, desktop computer, or laptop. Because the product has been adopted within Oracle Corporation as their strategic business intelligence platform and, in particular, because it has been integrated over time into their wider Oracle Fusion Middleware platform due to customer requirements for BI to integrate into wider business processes and applications, these core components have been built on and enhanced with additional Java components to form the more complete architecture used in the 11g Release 1 version.

**Oracle Business Intelligence 11g and Oracle Fusion Middleware**

While the core components within Oracle Business Intelligence remain the Oracle BI Server and Oracle BI Presentation Server, supported by the Oracle BI Scheduler, Oracle Java Host, and Oracle BI Cluster Controller, these have been supplemented in the 11g Release 1 release by Java-based Oracle Fusion Middleware technologies based around the Oracle WebLogic Server application server (with plans to extend this to other non-WebLogic application servers in future releases). While the previous, 10g release of Oracle Business Intelligence made limited use of application server technology to, for example, host the Oracle BI Plug-In and Java-based applications
such as Oracle BI Publisher and Oracle BI Office, the 11g release of Oracle Business Intelligence leverages Fusion Middleware and WebLogic technologies in areas such as the following:

- **Security and authentication**, which are now delegated to Oracle Fusion Middleware 11g, with users and groups now held, by default, in the WebLogic Server LDAP directory rather than the Oracle BI Repository
- **Systems administration**, now centralized using Oracle Enterprise Manager Fusion Middleware Control
Chapter 1: Oracle Business Intelligence Overview and Architecture

- **Connectivity** to outside applications and processes through the Java-based Action Service, with security and credentials handled via Fusion Middleware’s credential and policy stores
- **Administration scripting** using the Oracle WebLogic Scripting Tool and JMX MBeans (Java managed beans that provide the core administration functionality behind Oracle Enterprise Manager Fusion Middleware Control)
- **Process management** of the system components (BI Server, BI Presentation Server, and so on) through the Oracle Process Manager and Notification Server
- **High availability** through clustering of WebLogic Managed Servers within a WebLogic domain

Figure 1-10 shows the logical architecture for Oracle Business Intelligence 11g Release 1, which together is called an Oracle BI domain.

**FIGURE 1-10.** The Oracle Business Intelligence 11g logical architecture
This logical architecture is made up of a number of components:

- **Oracle BI domain**  The complete set of Java and non-Java components that make up a single Oracle Business Intelligence environment
- **WebLogic domain**  Houses the Java components within the architecture
- **WebLogic Server Admin Server**  A JEE container (server) that contains a dedicated Java Virtual Machine used for monitoring and managing the system
- **WebLogic Server Managed Server**  Another dedicated JEE container that, in this case, is used to house the Java applications used by Oracle Business Intelligence
- **Java components**  Java applications such as the Oracle BI Analytics Plug-In, Oracle BI Publisher, the Action Service, and Oracle BI Office that work alongside the traditional system components described previously
- **System components**  The term now used for the original BI Server, BI Presentation Server, and other servers used in the core Oracle Business Intelligence architecture
- **WebLogic Server Administration Console**  An application that runs in the Admin Server and is used for controlling WebLogic Server
- **Oracle Enterprise Manager Fusion Middleware Control**  For managing the Oracle Business Intelligence system across a single or multiple cluster nodes
- **Supporting database schemas**  Created using the Oracle Fusion Middleware Repository Creation Utility, which contains relational tables used for storing additional Oracle Business Intelligence metadata (BIPLATFORM) and metadata used by Oracle Fusion Middleware’s Metadata Services (MDS)

Note that it is possible to install Oracle Business Intelligence in a “simple” architecture configuration that does away with the WebLogic Managed Server and Node Manager, to reduce the software and memory footprint for laptop-style installations. This type of installation does, however, restrict you in how you can “scale out” your installation afterward, so it should only be used for demo or proof-of-concept scenarios. Chapter 2 details the various installation options and what these mean in terms of installed components on your server.

This distinction between Java components that are written in Java and managed by WebLogic Server, and System Components that run as operating system services and are written in C++, is due to the history of Oracle Business Intelligence. The core server components (system components, in 11g terminology) were written in C++, but most new development of distinct new functionality carried out by Oracle Corporation has been carried out in Java as part of Oracle Fusion Middleware. Rather than convert all of the legacy servers to Java applications from C++, however, these have been “lifted” into the new 11g architecture but allowed to run stand-alone, outside of the Oracle WebLogic Server domain structure, and are now called system components. This approach is actually common to many Oracle Fusion Middleware products that need to combine long-standing C-based server products with more recent, Java-based applications, and allows both sets of products to benefit from the same management and integration infrastructure based around Oracle Enterprise Manager Fusion Middleware Control and Oracle WebLogic Server.

While the distinction between system components and Java components becomes obvious to experienced Oracle Business Intelligence developers, both types of components are managed at a system level by the same application: Oracle Enterprise Manager Fusion Middleware Control. While each component and type of component has its own unique configuration tools, Fusion
Chapter 1: Oracle Business Intelligence Overview and Architecture

Middleware Control provides a centralized, web-based console for controlling the business intelligence system.

In addition to the Java and system components, two additional servers are used to control and monitor the running of these component categories:

- **WebLogic Server Node Manager**  This server is used to start, stop, restart, and monitor the Managed Server, or multiple Managed Servers if WebLogic Server is running clustered (not available for “Simple” install types, which do not need to stop and start any Managed Servers).

- **Oracle Process Manager and Notification Server (OPMN)**  This server performs a similar role for the system components—stopping, starting, and restarting them, and reporting their status to monitoring applications.

Let’s now take a look at some of these components in detail.

**The Oracle BI Domain**

The Oracle BI domain is the overall container for all Java and non-Java (in most cases, C++) applications and servers that make up a single Oracle Business Intelligence environment, together with the repositories, catalogs, and configuration files that these components use. An Oracle BI domain contains a single WebLogic domain (containing an Administration Server and one or more Managed Servers), and a typical customer may have several Oracle BI domains, for example:

- A development Oracle BI domain
- A test/QA Oracle BI domain
- A production Oracle BI domain

Each Oracle BI domain requires its own separate installation, with, in general, its own binaries and configuration files, though it is possible to share a single WebLogic Server installation among multiple domains, to aid patching and management.

**The WebLogic Domain**

A WebLogic domain is the basic administrative unit for a WebLogic installation. In the context of Oracle Business Intelligence 11g, each Oracle BI domain contains a single WebLogic domain, which in turn contains a single Administration Server and one or more Managed Servers (optionally clustered), containing the Oracle Business Intelligence Java components.

**The Oracle WebLogic Server Administration Server**

Every WebLogic domain contains a single Administration Server, which you use to configure, and then manage, all other server instances and resources in the domain. In terms of Oracle Business Intelligence 11g, the Administration Server is the first component that you start when starting an Oracle BI domain, and the Managed Servers (via the Node Manager) contact the Administration Server as they start up, to obtain their configuration information.

You work with the Administration Server either graphically, through the Administration Console Java application that runs in the Administration Server JVM, or from the command line. Once you have successfully set up and configured your Oracle BI domain, you rarely interact with the Administration Server except when you configure security and when you stop, start, and restart the Oracle BI domain.
Chapter 1: Oracle Business Intelligence Overview and Architecture

The Oracle WebLogic Server Managed Server(s)
While it is possible to install the Oracle Business Intelligence Java components directly into the Administration Server (and this is what happens when you choose the Simple Install option described in Chapter 2), generally Java applications such as these are installed into a separate WebLogic Server Managed Server. In a WebLogic domain, servers other than the Administration Server are referred to as Managed Servers, which refer to the Administration Server to obtain their configuration information. Each Managed Server is based around its own Java Virtual Machine, and typically each host running Oracle Business Intelligence will have a single WebLogic Server Managed Server installed on it that can take advantage of the full resources of the host.

Two or more Managed Servers can optionally be configured as a cluster. This is distinct and separate to the clustering used by Oracle Business Intelligence’s system components (the BI Server, the Presentation Server, and the Java Host), although these two types of clustering are often used together in a scaled-out Oracle BI domain. Chapter 13 goes into clustering, high-availability, and scale-out of the Oracle BI domain in more detail.

The Oracle WebLogic Server Node Manager
As your Oracle BI domain may be spread over several hosts and several locations, Oracle Business Intelligence uses another WebLogic component called Node Manager to manage the Managed Servers that you have configured for your domain. Node Manager is a Java utility, separate from the Administration Server, that Oracle Business Intelligence uses to start up, shut down, and restart Managed Servers across hosts and locations.

Node Manager, along with Oracle Process Manager and Notifications Server, is one of the few services that has to be running on a Microsoft Windows–based environment for your domain to start successfully, and under Linux/Unix, you would need to ensure the Node Manager service is running before you attempt to start your Managed Servers. Beyond ensuring that it is running, though, you do not generally have to interact with this WebLogic utility.

Java Components
Most new functionality added to Oracle Business Intelligence since the original Siebel and nQuire days has been added as Java-based applications, or “Java components.” Java components are installed into the WebLogic server Managed Server and Administration Server, and those installed in the Managed Server can be clustered through standard WebLogic clustering.

Compared to the non-Java system components, Java components apart from Oracle BI Publisher generally do not require much maintenance and do not, for example, require the same level of stopping, starting, and restarting due to configuration changes that system components require.

The following Java components are installed as part of Oracle Business Intelligence 11g Release 1.

The Oracle BI Analytics Plug-In As the Oracle BI Presentation Server system component cannot, in itself, communicate over the Internet with a web browser, the Oracle BI Analytics Plug-In runs as an application in the Managed Server; receives requests for dashboards, analyses, and other BI content; and routes it to the Presentation Server. The Oracle BI Analytics Plug-In handles requests such as these via HTTP and HTTPS, and performs a similar role for web service requests via the Simple Object Access Protocol (SOAP).

Oracle BI Publisher Oracle BI Publisher is a report authoring, publishing, and distribution server that was initially developed separately from Oracle Business Intelligence as a reporting platform for Oracle E-Business Suite. With the 10.1.3.2 release of Oracle Business Intelligence,
Chapter 1: Oracle Business Intelligence Overview and Architecture

Oracle replaced Actuate, which until then, with Oracle BI Publisher, performed a similar role for Siebel Analytics and across other Siebel products. As a Java-based application, it was installed into a Java application server rather than running as an operating system service.

Oracle BI Publisher runs as a Java component in the Managed Server, and reports can be authored and viewed either stand-alone through a Java-based front-end or integrated into Oracle BI Presentation Services dashboards. If installed and configured together with Oracle Business Intelligence, Oracle BI Publisher shares security with the rest of your business intelligence deployment and stores report and data model definitions in the same catalog.

**Oracle BI Office**  
Oracle Business Intelligence comes with a plug-in to Microsoft Office 2003, 2007, and Oracle Office that allows analyses to be embedded in spreadsheets and other office documents. The Oracle BI Office Java component provides connectivity between these office tools and Oracle Business Intelligence, allowing these tools to act as a client for Oracle Business Intelligence and reuse the security that you have applied to your business intelligence objects.

**Oracle BI Action Services**  
Oracle BI Action Services is part of the Action Framework (detailed in Chapter 7 which covers the wider topic of “Actionable Intelligence”) and is used when invoking external web services and processes. Oracle BI Action Services contains a set of web services that provides directories of external services that users can browse, as well as call from their analyses, dashboards, and agents.

**Oracle BI Security Services**  
Previous releases of Oracle Business Intelligence had authentication and authorization provided directly by the Oracle BI Server system component, which could connect to external directories such as Microsoft Active Directory and Oracle Internet Directory to authenticate users against existing enterprise security systems. The 11g release of Oracle Business Intelligence instead delegates this process to an Oracle Fusion Middleware service called Oracle Platform Security Services, and Oracle BI Security Services acts as the interface between it and the Oracle BI Server system component.

**Oracle Real-Time Decisions**  
Oracle Real-Time Decisions is an optional component of Oracle Business Intelligence that provides automated decision making based on data mining–style models that it creates. Oracle Real-Time Decisions was originally developed by a company called Sigma Dynamics, which in turn licensed the technology to Siebel Systems, Inc., for use within Siebel CRM. Sigma Dynamics itself was later acquired by Oracle Corporation, shortly after it acquired Siebel Systems.

Oracle Real-Time Decisions consists of a server element, installed as a Java component within the Managed Server, and client tools that can be downloaded from the Oracle Technology Network. Oracle Real-Time Decisions is outside the scope of this book; however, you may wish to research it if you need to add a real-time, self-learning, decision-making process to your CRM or other applications.

**Oracle BI SOA Services**  
In addition to calling external services and processes through the Action Framework and Oracle BI Action Services, BI objects such as analyses, conditions, and agents can also be invoked by external applications through Oracle BI Service-Oriented Architecture (SOA). Oracle BI SOA Services provide a web service framework for invoking analyses, conditions, and agents from Business Process Execution Language (BPEL) processes, giving applications the ability to connect to business intelligence functionality to add intelligence and analytics to your applications.
Administrative Components  While most of the Java components used by Oracle Business Intelligence run in the Managed Server, some run in the Administration Server instead. The Managed Server Java components are directly related to business intelligence functionality, while the ones that run in the Administration Server are used for administering the system. These include the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control, and Java JMX Mbeans used for programmatic access for managing the domain.

System Components
The system components are the “core” of Oracle Business Intelligence and consist of C++ and J2SE (Java 2 Platform, Standard Edition) components that provide the central functionality of Oracle Business Intelligence. They were largely developed by nQuire and Siebel prior to the Siebel acquisition and have been enhanced over time by developers at Oracle Corporation.

The Oracle BI Server  The Oracle BI Server provides several areas of functionality for Oracle Business Intelligence:

- It receives incoming logical SQL queries from the Oracle BI Presentation Server (and other clients via the Oracle BI ODBC Client), and through the semantic model turns these into one or more physical SQL and MDX queries, which are sent to the underlying data sources.
- It provides the connectivity through to these data sources, through OCI, ODBC, Essbase Client, and other native adapters.
- It provides load-balancing data to the Oracle BI Cluster Controller, to enable Oracle Business Intelligence to be run in a scaled-out, highly available configuration.
- It applies row-level and subject-area security for data described using the semantic model.
- It can maintain a data cache, storing the results of previously executed queries locally to avoid unnecessary round trips to the underlying data sources.
- Where user queries require data from more than one data source, it can join results returned from each data source together and also apply calculations and analytics to the resulting data set.

As such, the Oracle BI Server performs the role of an extra, “analytic” server that sits over your data warehouse and data sources, providing additional integration features, an additional layer of calculation and analysis, and data security, before passing it to the Oracle BI Presentation Server for display as dashboards, analyses, and reports.

Figure 1-11 shows a diagram of the internal architecture of the Oracle BI Server, made up of a number of logical components.

The key components within the BI Server logical architecture include:

- **Logical SQL/ODBC interface and business model**  This receives from the client tool requests for information against a logical business model (semantic model).
- **Navigator**  This takes these logical requests and, using the information in the semantic model, turns these into individual physical requests to be sent to the underlying data sources.
Chapter 1: Oracle Business Intelligence Overview and Architecture

- **Optimized query rewrite engine**  This takes these physical requests and the BI Server’s knowledge of the calculation capabilities of the underlying data sources, passes function requests to these sources where possible and, where not, rewrites the query to obtain the basic data set and has the BI Server perform the calculations instead.

- **Execution engine**  This passes these physical queries to the underlying data sources and collects the results back.

- **Cache management**  This stores the results of previous queries to avoid future round trips to the underlying data sources for the same queries.

- **Security services**  This applies row-level and subject-area security to the logical business model (semantic model).

- **Query governance**  This applies limits to the number of rows returned by queries and the times that users can access data.

FIGURE 1-11. The Oracle BI Server logical architecture
Business intelligence systems built using Oracle Business Intelligence can use the features of the BI Server to varying degrees. A system built on a single, integrated data warehouse as a dimensional “star schema” with built-in summaries, indexes, and derived calculations may end up using very little of the BI Server’s capabilities, perhaps using it only to provide load balancing, connection pooling, and row and subject-area security over the data.

Other systems, where data is spread over several source systems and is not naturally organized into a dimensional model, where aggregations, calculations, and other analytics need to be applied post-query, may use more of these features. The way you use the BI Server will depend on the nature of your data and how much additional processing is required after data retrieval takes place, but you will find that the BI Server provides a very powerful additional analytic layer to your business intelligence system, providing some functionality faster and more flexibly than if you tried to implement everything in extraction, transformation, and loading (ETL) processes or tried to materialize all calculations or aggregations in advance.

The Oracle BI Presentation Server In contrast to the Oracle BI Server, the Oracle BI Presentation Server performs a more traditional role within your architecture—that of a presentation and visualization server that takes incoming data and renders results in the form of dashboards, analyses, reports, alerts, and scorecards. The Oracle BI Presentation Server receives requests for BI content through the Oracle BI Plug-In, installed in a JEE application server (WebLogic Server as of Oracle Business Intelligence 11.1.1.6), and then communicates with the Oracle BI Server via ODBC to authenticate incoming users, pass to it logical SQL queries, and then receive results back for rendering on the users’ screens.

The BI Presentation Server also performs a number of additional roles in support of its role in providing dashboards and other visualizations for users:

- In a similar way to the BI Server, the Presentation Server caches its output (tables and pivot tables of data, charts, and other graphics) so that when the user switches between dashboard pages, results are displayed immediately rather than the user having to wait while the analysis is rerun.
- The BI Presentation Server also caches the repository metadata provided by the BI Server so that users can quickly select between subject area items when creating analyses.
- The BI Presentation Server also maintains its own repository, called the “catalog,” which stores definitions of analyses, dashboards, alerts, and other BI content.
- The BI Presentation Server also provides two SOAP interfaces that can be used by external applications to request the output of analyses, alerts, and conditions and integrate them into standards-based portals and other applications. One of these SOAP interfaces, Oracle Business Intelligence Web Services for SOA, provides a simplified API for executing analyses and other BI objects typically used by other SOA processes such as BPEL workflows, while the older Oracle Business Intelligence Session-Based Web Services also found in the 10g release provide a more fine-grained API that is more suited for programmatic use.

While the 11g version of the Oracle BI Server supports OLAP analysis against any data source, more granular caching, and a wider range of hierarchy types, the BI Presentation Server also had a number of significant changes compared to the previous 10g release, including the following:
Chapter 1: Oracle Business Intelligence Overview and Architecture

- It no longer maintains its own set of users and groups (referred to as “webgroups” in the 10g release); instead, it uses the same users and application roles as the BI Server, and these are defined using Oracle Fusion Middleware 11g. These webgroups are, however, still available for backward-compatibility purposes when upgrading systems from the 10g version to 11g.

- The BI Presentation Server now uses the underlying data source to calculate as many totals and subtotals as possible, rather than calculating them itself after the main data set has been returned from the source database.

While developers would benefit from having a close knowledge of the workings of the Oracle BI Server, the BI Presentation Server generally does not require a large amount of understanding or maintenance. There are some tasks and configuration changes that you may wish to apply over time (detailed in later chapters of this book), but for the most part the BI Presentation Server can be considered a “black box” that does its required job without much intervention on your part.

**The Oracle BI Cluster Controller**  The Oracle BI Cluster Controller provides load-balancing and query-routing functionality in a clustered Oracle BI Server environment. While the BI Cluster Controller was an optional server in the 10g version of Oracle Business Intelligence, it assumes a more significant role in the 11g version, as clustering is enabled by default, albeit with a single BI Server node.

As discussed earlier in this chapter, when the BI Cluster Controller is used (by default in 11g installations, but optional in 10g ones), it is the Cluster Controller rather than the BI Server itself that the BI Presentation Server communicates with to pass requests and receive results back. Of the BI Servers it is responsible for, the BI Cluster Controller maintains a record of which ones are actually online and available for queries, and routes requests to them on a round-robin basis. Individual BI Server components contain load-balancing functionality that reports their current load to the BI Cluster Controller, and the BI Cluster Controller treats the BI Server components under its control as “active-active” resources, meaning that all available BI Servers are used if available, with failover happening dynamically as BI Server resources become unavailable and then available again.

In contrast, the BI Cluster Controller (along with the BI Scheduler) is considered an “active-passive” resource; if the primary BI Cluster Controller becomes unavailable, a secondary BI Cluster Controller is then activated (if previously configured for this role), and user queries use the secondary controller for subsequent queries. Only primary and secondary BI Cluster Controllers can be configured (in contrast to BI Servers, BI Presentation Servers, and Java Hosts, which effectively have no limit to how many you can configure actively in a cluster), and the secondary controller only becomes active if the primary one becomes unavailable.

**The Oracle BI Scheduler**  The Oracle BI Scheduler is enabled by default in Oracle Business Intelligence 11g and provides the ability to schedule analyses, agents, and other BI content to be delivered to users via a number of channels. For clustering purposes, the Oracle BI Scheduler (like the Oracle BI Cluster Controller) is considered “active-passive,” with a secondary BI Scheduler often defined to provide backup and failover if the primary one becomes unavailable.

The BI Scheduler stores details of job executions in relational tables within the supporting database schema, although definitions of the jobs themselves are held in the Oracle BI Presentation Server Catalog. Note that the BI Scheduler is not included in the Standard Edition One license for Oracle Business Intelligence.
The Oracle BI Java Host  The Oracle BI Java Host is a component that you can generally leave to work in the background. As a developer, you do not usually interact with this component, but it works in the background with other components to provide support for Java tasks. In particular, it provides or supports the following functionality with Oracle Business Intelligence 11g:

- Graph generation
- SVG rendering
- Java task support for the Oracle BI Scheduler
- Oracle BI Publisher
- Advanced reporting
- URL Connect (issues an HTTP request to another component)
- Integration Service calls (used by the Oracle BI Server to execute Java code)
- Authentication to external systems such as Hyperion Financial Management and Hyperion Shared Services

In general, your only interactions with the Oracle BI Java Hosts are first to ensure that it is running and, second, to configure additional clustered BI Java Hosts to work in “active-active” mode with the default one, to provide failover and load balancing.

Oracle Process Manager and Notification Server  Oracle Process Manager and Notification Server (OPMN) is a standard piece of Oracle Fusion Middleware Technology used to stop, start, and monitor the status of Oracle Business Intelligence system components across a network. It is used by Oracle Fusion Middleware Control, and by yourself manually from the command line, to control the status of the BI Server, BI Presentation Server, and other system components. It works alongside the WebLogic Server Node Manager, which performs a similar role for the Java components.

OPMN is used by Oracle Business Intelligence because key servers such as the BI Server and BI Presentation Server aren’t written in Java and can’t have their status controlled by the usual Java-based management servers found within Oracle Fusion Middleware. OPMN provides a server for managing the status of non-Java components and is used by Fusion Middleware Control to provide the interface between the Java-based Enterprise Manager JEE application and the operating system executables that make up the Oracle Business Intelligence system components. When you view the status of the BI Server, BI Presentation Server, or other system components in the Fusion Middleware Control management screens, it is OPMN that is being used to report back their status and pass on your requests to stop, start, and resume them.

Supporting Database Schema  The supporting database schema is a set of database tables that can be installed via the Oracle Fusion Middleware Repository Creation Utility into an Oracle, Microsoft SQL Server, or IBM DB/2 database. The database schema, typically called BIPLATFORM and accompanied by another one called Metadata Services (MDS) that contains metadata used by Oracle Fusion Middleware, contains tables that support the following system and Java components:
Chapter 1: Oracle Business Intelligence Overview and Architecture

If you use a tool such as Oracle SQL*Developer or Oracle SQL*Plus to look at the contents of this schema, you will also note that there are other tables to support products such as Oracle Real-Time Decisions, Oracle Calculation Manager, and Oracle Essbase; however, you can ignore these tables, and they are outside the scope of this book.

So now you understand a little more about the architecture of Oracle Business Intelligence, and we will return to the topic of this architecture, and management of the various components, in later chapters of this book. For now, though, let’s move on to what is probably the single most important topic that you need to understand as a developer: creating repositories against relational data sources.