

# Business Intelligence Technical Overview

*An Oracle White Paper*  
*July 2005*

# Business Intelligence Technical Overview

Executive Overview .....	3
Introduction .....	4
Phase 1: Consolidation .....	4
Step 1: Map transactional sources to a target data warehouse .....	5
Step 2: Generate the code to extract, transform and load data.....	5
Step 3: Generate the business area.....	6
Warehouse lifecycle management.....	6
Phase 2: Discovery .....	6
Business areas defined automatically .....	7
Leveraging Database Analytic Capabilities.....	7
Authenticity is verified on demand .....	8
Phase 3: Sharing Data .....	8
Query results are shared.....	8
Generate once and deploy in any format .....	8
Generate once and deploy anywhere .....	9
Expanding Business Insight.....	9
Rapid Development of Custom BI Applications.....	9
Data Mining.....	10
Personalization .....	11
Portal.....	11
Conclusion.....	11

# Oracle Business Intelligence Technical Overview

## EXECUTIVE OVERVIEW

In an increasingly competitive world, as a business executive, you are constantly in need of business intelligence (BI) that informs and guides your decision making to keep your products and services competitive. No matter what your arena, you need complete business intelligence about your company's operations, competitors and customers to assess the potential opportunities and threats associated with various types of customer and competitive response. To build a BI system for the enterprise, IT professionals would typically purchase point solutions piecemeal. Unfortunately, such piecemeal solutions result in proliferating complexity and escalating maintenance costs. For example, each upgrade would require dismantling and recalibrating the system. Furthermore, when multiple disparate vendors upgrade their respective products they rarely account for any impact on the other products that are part of the overall solution. Often the customer is left stuck in the middle with a lot of finger pointing between vendors each time the system undergoes an upgrade, or maintenance cycle. The net result is more time and money to stitch together and maintain point solutions compared with a truly integrated solution.

More importantly, there can be a real business impact for un-integrated systems beyond just the maintenance costs. With various tools using different copies of metadata, you could have multiple users viewing the same data, but arriving at separate conclusions. Consider for a moment the impact of sales territory reorganizations. If the CRM application is updated with the latest territories, but the financial application still points to the outdated territories, the projected revenue reports would yield two different results. Imagine the spirited dialog between the VP of Sales and VP of Finance as they discuss future business directions based on their own reports.

Oracle Business Intelligence offers an integrated BI solution that provides the business user with a complete picture across the entire organization. With an integrated solution from a single vendor, customers benefit with:

- A cost effective deployment that's faster, simpler, and easier
- Superior functionality
- Improved scalability and reliability
- Reduced administration and maintenance costs

- A single vendor responsible for support and upgrades.

Oracle Business Intelligence is designed to easily and quickly integrate with diverse data sources, automatically populate metadata, find information from the database, share the newfound information, and exploit the data to learn more about the business and its customers.

## **INTRODUCTION**

There are typically three phases in building a BI system: consolidation, discovery, and sharing. In addition, the BI system should be designed with robustness to include expansion, as the business needs change. At first glance, the three phases may seem simple, but a closer look at each phase reveals potentially complex issues as the system is integrated. For example in the first phase, consolidation, the number of complications escalates as the number of diverse data sources increases. In addition, the data usually requires a degree of transformation, or name and address scrubbing during data consolidation. Once the consolidated data is prepared for discovery, a myriad of reporting tools must be integrated to deliver business intelligence to the appropriate business decision maker. Such reporting tools include analytical applications, query & analysis, enterprise reporting and lineage. Finally, each tool may have its own high performance engine that must be integrated into the BI system.

By integrating such an intricately complicated system that includes data management, user administration, reporting tools, and performance engines from multiple vendors, the organization faces:

- Very high software costs, as individual vendors charge premium prices for their contribution to the overall system
- Expensive training to learn all the different reporting, data management and administrative tools
- A wide variety of technology expertise required to maintain the system
- Protracted and complex implementations
- Complicated system upgrades and maintenance from different vendors

Oracle Business Intelligence is designed as the single end to end integrated platform for business intelligence. This paper provides an outline of the integrated business intelligence functionality available from Oracle. Included in this paper are illustrations of the tight integration between individual Oracle BI components as they are applied to each phase in building a complete and integrated BI solution from a single vendor.

## **PHASE 1: CONSOLIDATION**

Consolidating data is becoming increasingly common as businesses merge, reorganize, and globalize to stay competitive. The data tends to become spread throughout the entire organization in multiple data sources, making it increasingly

difficult to derive business intelligence from a single source of truth. The Oracle Warehouse Builder component is designed to meet this challenge by consolidating disparate data sources, performing any required data transformations, and managing the entire warehouse lifecycle, plus integration with the Oracle reporting, analysis and publishing tools.

### **Step 1: Map transactional sources to a target data warehouse**

Once the business questions and data requirements of the end-users have been determined, the IT professional starts identifying and connecting to the data sources containing the data to meet these requirements. Connections to flat file sources (including spreadsheets), or relational data sources such as Oracle, DB2, Informix, Microsoft SQL Server, Sybase, and SAP R/3 are made possible by simply selecting the appropriate Oracle Warehouse Builder (OWB) integrator. The IT professional then creates source modules that store data definitions containing both connection information and table definitions. Once connected to the data source, the specified integrator is used by OWB to extract data and metadata from that source. The other type of module the user creates with OWB is the warehouse module, which contains the logical design of the facts, dimensions, staging tables, etc. that make up the data warehouse. Together, these two types of modules form the central elements used in warehouse design. The user can simply drag and drop objects from the modules to use as building blocks in designing a target warehouse. After the objects of interest are selected, OWB provides a graphical mapping environment where users can model all aspects of a mapping in the form of a dataflow diagram. These mappings include participating source and target tables and all other operations that apply to the extraction, transformation, and loading (ETL) process. OWB can perform the ETL operation in one complete step, making warehouse deployment fast and easy.

Using OWB to map data sources to targets is easy with a graphical Mapping Editor, where the user visually designs, or models all aspects of the ETL operation. Developer productivity is enhanced as the user designs complex transformations, inline expressions, multiple joins, aggregations, etc. with an intuitive interface, without requiring specialized SQL programming knowledge.

### **Step 2: Generate the code to extract, transform and load data**

Once the mapping models are completed, OWB can generate the SQL and PL/SQL code to instantiate and populate the data warehouse. This saves time and reduces the level of expertise required to program SQL code. Furthermore, OWB is open to allow users to inspect the code it generates at any step in the ETL operation. This adds to the flexibility in the event a customized operation is required. In addition, the OWB Code Generator can generate code that's optimized with respect to the data source. For example, if the data is clean with no anticipated conflicts, the user simply clicks the set based code generation option and high performance code is generated to load the data as an entire set. On the other hand, if the data has high potential for constraint violations, or other load

errors, the user clicks on the row based code generation option and instantly the code is rewritten to load data on a row-by-row basis. Because the PL/SQL packages are automatically generated and do not require any specialized PL/SQL programming knowledge, the tool is available to more users, reducing the cost of ownership and increasing the productivity of the department. Note that OWB is flexible enough to allow the user to specify if they wish to deploy their logical warehouse design created in step 1 above as a star schema relational target, or as an OLAP multidimensional target to take advantage of the Oracle database OLAP option. In either case, the appropriate PL/SQL code is generated automatically by the tool. Further, OWB users can define if-then-else workflows for any given ETL process. For example, a user could define an e-mail alert be sent to a particular system administrator should a given table fail to load or else e-mail a success message to the warehouse designer. This allows for quick responsiveness and helps with smooth system operation.

### **Step 3: Generate the business area**

Now that the data is consolidated and loaded into a target warehouse, the multidimensional design is easily shared with Oracle reporting tools. Oracle Warehouse Builder can generate the metadata necessary for reporting with the Oracle reporting tools. This means that Oracle reporting tools can “understand” the dimensions and hierarchies as well as all table names and column headers. As a result, valuable programming time is saved because the user is not required to separately create the reporting metadata when deploying the Oracle reporting tools as would be the case with point solutions.

### **Warehouse lifecycle management**

Once the warehouse is deployed, managing changes to the warehouse is essential. Inefficient maintenance of the warehouse can strain any IT budget. Changes, such as tables, columns or views in the source are added, dropped or modified. When these changes occur, most ETL tools today can only detect that something changed between the source and target, but leave it to users to reconcile the differences. Oracle Warehouse Builder, on the other hand, reconciles any changes in the source and target objects. For example, if a change occurs after a source module was created in the OWB repository, the repository definitions are out of sync with their corresponding source objects. The re-import function of Oracle Warehouse Builder automatically reconciles and synchronizes the definitions in the repository with the source objects. Or, the warehouse upgrade function synchronizes the definitions in the repository with the target objects. The user saves time maintaining the warehouse, constantly trying to keep track of source data changes and extending those changes to the target database.

## **PHASE 2: DISCOVERY**

Now that the data is located in one place, the powerful ad hoc query and analysis tool, Oracle Business Intelligence Discoverer, another component of Oracle’s

integrated Business Intelligence solution, delivers insight associated with the data captured about your products, customers, and marketplace, and allows you to quickly disseminate this information across the enterprise.

### **Business areas defined automatically**

Discoverer hides the complexity of underlying relational and OLAP database structures like tables, columns, joins, dimensions etc. by presenting a business oriented view of the data. Business users create reports by opening up a subject oriented business areas represented by file drawers and folders, moving selected items onto a worksheet. Logical hierarchies, calculated Items, pre-built conditions, custom sort orders, exception highlighting and more enable users to perform otherwise complex tasks by simply clicking and dragging. You can create a business area in Discoverer Administrator using the Bulk Load Wizard or alternately by using OWB as part of your datawarehouse deployment.

The tight integration between OWB and Discoverer allows users to easily populate business areas by using a wizard-based interface. The dimensions and hierarchies created in OWB are understood by Discoverer, which improves productivity and provides faster deployment to end users for ad hoc query and analysis.

### **Leveraging Database Analytic Capabilities**

One of the most important improvements to relational databases has been the introduction of analytic functions. Discoverer leverages the Oracle Database relational Analytic Functions that allows users to answer sophisticated business questions on demand. Powerful analytics like ranking, period-to-period comparisons and moving averages are now available to administrators and end users. In addition, users can sort, pivot, and drill on the data to meet their analysis needs. Discoverer further extends analytical flexibility by permitting users to nest functions like calculating the growth rates between two data sets, then ranking them at the same time. For example, the user can rank in descending order the fastest growing products this year compared to last year in a single calculation. Nesting calculations dramatically reduces the time users spend seeking answers to compound business questions. The latest release of Discoverer offers new templates that help users to set up analytic formulas, using basic business terms. These wizard based templates are developed in conjunction with our award winning usability laboratory and allow users to simply apply these templates for their most commonly used business questions without knowledge of SQL. Further Discoverer users can also create reports directly against OLAP structures in the database, with the same interface that they would use when creating a relational report. This unique capability lowers your training costs, as users do not need to learn more than one reporting tool depending on which technology is being used in the database. Discoverer connected to the OLAP option can take full advantage of the powerful analysis, modeling and forecasting capabilities that the Oracle OLAP engine is renown for.

### **Authenticity is verified on demand**

Providing lineage reports extends the integration of Discoverer with OWB even further. Discoverer users can drill from their queries to the metadata definition in OWB and can view the objects impacted by a particular data transformation. For example, if the aggregated revenue from a region abroad were questionable, the user could drill on the item to view a graphical representation of all the calculations, or other transformations used to calculate that revenue. Now, users can create their own queries and easily verify the validity of the results in the same session.

### **PHASE 3: SHARING DATA**

Often much time and effort by the users are expended in the discovery phase where just the right query and analysis is performed. Once the right information is generated, sharing the information across the enterprise before it becomes stale can be challenging.

### **Query results are shared**

Discoverer promotes sharing the queries with the enterprise via its tight integration with OracleAS Reports. The export feature of Discoverer allows for a full fidelity transfer. Information about layout, formatting, exceptions, parameters, and even the SQL used for the query are passed from Discoverer to Reports via XML. This gives users access to the full publishing and distribution capabilities of Oracle Reports without recreating the query. In addition, integration between Discoverer and OracleAS Portal empowers users to publish their favorite reports, list of reports or even gauge dials to OracleAS Portal without any programming. Other users in the organization easily access the information by browsing these portal pages. The "List of workbooks portlet" provides a list of workbook names and the "Worksheet Portlet" can display a tabular or cross-tabular report, a graph, or both. Finally the "Gauge portlet" can be used to visualize any crosstab as a dial within the Portal. Each portlet also comes with personalization options. Users can create their own custom highlighting, change graph types or other attributes directly on the portlets, which each personalization saved for that individual user. This makes it easy to deploy personalized dashboards across the enterprise without an IT manager having to maintain different copies of the dashboard. The Discoverer portlets present the same look feel and behaviour to users regardless of which technology is used to deliver the underlying report – OLAP or relational. This transparency makes it easy for users to focus on the business rather than having to learn different reporting tools.

### **Generate once and deploy in any format**

Oracle Reports has always been a powerful tool to publish data in a variety of formats: PDF, XML, HTML, HTMLCSS, Postscript, PCL, Delimited text, and RTF. Now, users can publish data using industry-standard JSP's. This means that developers can create a Web page template using an HTML editor, then bring it into the Oracle Reports environment and integrate data from various data sources



into that Web page. This includes using business logic and your own corporate styles to fashion and control a report's appearance. Of course, a JSP doesn't have to return HTML, as it can be used to create any text-based output format (e.g. WML). Finally, because JSP's are used, when a report is updated, all the Web pages tagged with that report are updated instantly.

### **Generate once and deploy anywhere**

Printer, Web, file-system, Email, and Oracle Portal are all destinations that Oracle Reports has supported for some time now on a wide variety of platforms. With Oracle Reports, as well as greatly enhancing the product's out-of-the-box emailing capabilities, developers are now able to create their own java-based extensions to the Reports Server to open it up to previously unsupported destinations such as Fax, FTP, WebDAV, etc. The broadcast of this information can be done on-demand or scheduled, or even as a reaction to an event that has occurred within an Oracle Database, Oracle Workflow, or an application that communicates via Web Services. Email and wireless notifications of the report success/failure are available, and once again, developers can plug-in their own java-based notification mechanisms. Of course information security is of paramount importance, and with Oracle Reports, administrators have the flexibility to secure Reports, Report Servers, and printers for fine-grade control of who can do what, where, and when.

### **EXPANDING BUSINESS INSIGHT**

Reacting quickly to business needs is critical. Sometimes it's necessary to build custom applications to allow insight into real-time business performance in order to make better business decisions. The Oracle Business Intelligence Beans (BI Beans) delivered with Oracle JDeveloper (JDeveloper) is designed specifically for development of powerful business intelligence applications that are integrated with all Oracle business intelligence tools. In fact, these same BI Beans are used by Oracle to build the Discoverer and OLAP Spreadsheet tools, so you have access to the same development foundation that Oracle uses for our products. Developers can quickly assemble an integrated BI application from high level, reusable components while taking advantage of Oracle Business Intelligence OLAP capabilities.

### **Rapid Development of Custom BI Applications**

JDeveloper and BI Beans provide the most productive development environment for building business intelligence applications. BI Beans provide high-level components that encapsulate OLAP business logic for querying and presenting data. Data can be presented in tables, crosstabs, and graphs. Using BI Beans QueryBuilder and CalculationBuilder, developers easily access the advanced analytic capabilities of Oracle OLAP for generating queries. The combination of these flexible query and presentation options make it very easy to visually design OLAP objects. Wizards are provided in JDeveloper that package these components into

applications in the form of either servlets or Java applications. A BI Beans tag library is also available enabling the rapid development of JSPs.

Advanced analytics are the heart of any business intelligence application. Oracle OLAP provides centralized analytic processing in a scalable and secure environment. The BI beans make it easy to formulate those business questions involving complex multidimensional queries. The results of such queries are then displayed in both crosstab and graphical presentations using the presentation beans. The QueryBuilder is a powerful, patented tool that enables users to specify query properties without expertise in SQL. Complex multidimensional queries are made simple by presenting the query definition in business terms, which end users can modify to meet their needs. For example, a default query definition called “Top 5 Products based on Sales” can be modified to “Bottom 10 Products based on Sales” by selecting the appropriate query commands from a drop down list. With unlimited query possibilities, users quickly leverage advanced database analytics to ask complex business questions of their data. Utilizing the BI Beans Catalog, users save and share their discoveries with others across the enterprise. **Note that the BI Beans are themselves used within Oracle Discoverer, so developers have access to the same Java technology used by Oracle.**

## **Data Mining**

Oracle Data Mining allows companies to build advanced business intelligence applications that mine corporate databases to discover new insights and integrate those insights into business applications. In addition to OLAP, the Oracle Database has embedded data-mining functionality like which can perform pattern matching classifications, predictions, and associations. This allows application developers to integrate data-mining capabilities into their business intelligence applications to support such activities as:

- Preventing customer attrition
- Identifying best candidates for cross-selling
- Acquiring new customers
- Fraud detection
- Identifying the most profitable customers
- Customer targeting and segmentation

Oracle Data Mining opens the door for integrating sophisticated data-mining capabilities, once the domain of specialized servers, into mainstream business intelligence applications.

Because Oracle Data Mining is embedded in the database, there is an immediate savings in administration and maintenance, as security can be applied in one place and ETL processes to load data into a separate data mining server are not necessary.

## **Personalization**

OracleAS Personalization is more than “business rules” or collaborative filtering. It delivers individualized recommendations, advertisements, and content with the touch and timing of a close personal friend. Its scope extends beyond recommending similar items based on a collection of clicks from potential buyers. Personalization brings together the entire customer experience, which includes “clicks”, purchases, returns, complaints, support requests, ratings, wish lists, and demographics. An automated, timely, and non-intrusive recommendation based on a complete view of the customer is true personalization.

Developers use OracleAS Personalization’s Recommendation Engine API to equip a Web site to tag, or capture a visitor’s “clicks” and to request real-time recommendations. These “click” data are combined with historical data, if available within the OracleAS Personalization schema, and passed to the OracleAS Personalization Recommendation Engine. The Recommendation Engine searches for “rules” or recommendations that best fit the current session and historical data scenario and passes the recommendations to the Web application in a fraction of a second. Integration with the Oracle Database delivers power and scalability with a minimum of data redundancy. OracleAS Personalization collects and stores the data, then builds predictive models all within the Oracle Database. Next, the powerful data mining technology embedded in Oracle Database automatically discovers individualized behavior patterns to generate highly accurate personalized recommendations in real time.

## **Portal**

OracleAS Portal is a complete framework for development and deployment of Web-based portals. It includes user administration, security, content customization, and development features to create and maintain basic reports, charts, and form-based applications. Creating a Business Intelligence dashboard personalized by job role is easy with OracleAS Portal. Charts and/or reports representing key performance indicators (KPI’s) can be rapidly developed. These charts and reports are deployed as Portlets. Individual users may customize their portal presentation by selecting the KPI portlets that are most relevant to their management focus.

Out of the box Discoverer provides three flavors of portlets ready for use within OracleAS Portal: list of worksheets, worksheet and gauges. These allow end users to embed their Discoverer reports directly into the portal environment via a simple wizard. Once located within Portal, users have access to customization options for these reports and can also take advantage of the single sign on capabilities.

## **CONCLUSION**

Today’s managers and data analysts are inundated with information and under time constraints to make key business decisions. They need visual cues to identify the latest business trends and then make both operational and strategic management decisions. All of which must be shared with the right people throughout the

enterprise. When the tools used for data consolidation, data analysis and enterprise reporting are integrated and available from one vendor, Oracle, it's clear that companies deploying Oracle Business Intelligence are armed for success. They have the capability to deliver the right answers and enjoy a superior ownership experience.

Oracle Business Intelligence Technical Overview

**ORACLE FUSION MIDDLEWARE** July 2005

Oracle USA Inc  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2005, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.