Advanced Development Techniques using OBIEE Plus

Mark Rittman, Director, Rittman Mead Consulting
UKOUG Conference & Exhibition
Who Am I?

- Oracle BI&W Architecture and Development Specialist
- Co-Founder of Rittman Mead Consulting
  - Oracle BI&W Project Delivery Specialists
- 10+ years with Discoverer, OWB etc
- Oracle ACE Director, ACE of the Year 2005
- Writer for OTN and Oracle Magazine
- Longest-running Oracle blog
  - http://www.rittmanmead.com/blog
- Chair of UKOUG BIRT SIG
- Co-Chair of ODTUG BI&DW SIG
- Second year of OU BI Masterclasses
  - 18 countries visited in 2006-7
Rittman Mead Consulting

- Oracle BI&DW Project Specialists
- Consulting, Training, Support
- Works with you to ensure OBIEE project success
- Small, focused team
- OWB, Oracle BI, DW technical specialists
- Clients in the UK, Europe, USA
Oracle BI Suite Enterprise Edition Plus

- Oracle “Next-Generation” BI tools platform
- Based on Siebel Analytics
- Originally developed by nQuire
- Standards (J2EE) based
- Heterogeneous connectivity
- Server and Web-based
  - Oracle BI Server
  - Oracle Presentation Server
  - Oracle BI Answers
  - Oracle BI Delivers
  - Oracle BI Administration
  - Job Manager, Catalog Manager etc
Oracle BI Server

- Calculation and Data Integration Engine
  - “Virtual Data Warehouse”
  - Allows reporting across multiple sources
  - Data accessed through metadata layer
- Connects to RDBMS and OLAP data
  - Oracle Database (including Oracle OLAP)
  - IBM DB2
  - Microsoft SQL Server
  - Teradata
  - Microsoft Analysis Services (OLAP)
  - SAP BW 'info cubes'
- Security, Summary Management
- Translates incoming “logical SQL” into optimized, physical SQL

Analytical and Operational Data Sources

Oracle BI Server

Analytical and Operational Data Sources

Oracle BI Server

Logical SQL ODBC/OLEDB (Logical Business Model)

Load Balancer

Security Management

Intelligent Request Generation

Logical Request Generation

Navigator

Cache Services

Optimized Query Rewrites

Execution Engine

Data Source Adaptors

ODBC, CLI, OCI, XML, MDX

Analytical and Operational Data Sources
Oracle BI Presentation Server

- “BI Application Server”
- Single application that provides Oracle BI Answers, Dashboards
- Metadata stored in Web Catalog
- User Profiling, Security and Session Management
- SOAP Web Services, XML and URL Interface

Diagram:
- Web Server
  - (IIS, Tomcat, Websphere, iPlanet)
  - SAW Bridge (J2EE/ISAPI)
- Oracle BI Presentation Services
  - SOAP Web Services, XML and URL Interface
  - Oracle Interactive Dashboards
  - Oracle Answers
  - User Profiling, Security and Session Mngmt
  - Cache Services (Web) & Connection Mngmt
- Web Browser
  - Javascript for Usability & Interactivity
  - HTML, XML, XLS, PDF, TXT over HTTP/HTTPS
  - TCP/IP (SSL)
- External Applications and Portals
  - HTML, SOAP over HTTP/HTTPS

Rittman Mead Consulting Ltd  telephone +44 (0)7866 568246  http://www.rittmanmead.com
Oracle BI Delivers Server

- Scheduling and distribution
- Uses iBots (“Intelligent Bots”)
- Integration (through Java) with BPEL
- iBot chaining, calling from BPEL, calling BPEL
- Permits alerts, through execution of Answers reports
  - If report returns rows, alert triggered
  - Alert appears on dashboard, via email, SMS, cell phone
  - Can trigger narrative and conditionally-appearing reports

Oracle Delivers Server

- Scheduling/Event Services
- Agent Execution Logic
- Device Adaptive Content
Oracle BI Publisher Server

- New in 10.1.3.2, based on Oracle XML Publisher
- Separate application, integrated with Oracle BI Presentation Server
  - Initially usernames and passwords integrated
  - Thereafter separately administered
- Provides high-fidelity layout, positioning of data, charts etc
- Multiple data sources
  - OBIEE Metadata Layer
  - Oracle BI Answers reports
  - Any source accessible via SQL
  - XML over HTTP
Oracle BI Administration

• MS Windows application for managing BI Server
• Creates and manages Common Enterprise Information Model
  • Physical, logical and presentation layers
• Manage other aspects of BI Server
  • Security
  • Scheduler (Delivers server)
  • Jobs
  • Joins
  • Marketing
  • Sessions
  • Cache
• Utilities and Tools
• Connect to BI Server via ODBC DSN

Oracle BI Administration

- Metadata Management Services
- Multi-User Development Services
- Metadata Documentation Services
- Server Management Services
Core BI Server Functionality

- Data Integration through Business Model View
- Calculation and analytical engine
- Support for RDBMS and OLAP data sources
- Intelligent Request Generation and Optimized Data Access
- Security and Data Scoping
- ODBC Interface out, native RDBMS and OLAP interfaces in
- Clustering, replication, support for Windows and Unix/Linux hosts, 32-bit and 64-bit
Data Access

- Reporting data may be sourced from multiple databases, applications
  - Oracle Database (including Oracle OLAP)
  - IBM DB2
  - Microsoft SQL Server
  - Teradata
  - Microsoft Analysis Services (OLAP)
  - SAP BW 'info cubes'
  - SAP, Peoplesoft, Siebel, E-Business Suite
- Oracle BI Server joins this source data together and presents a logical business model
  - Facts
  - Dimensions
  - Calculations
Common Enterprise Information Model

- Metadata layer for OBIEE tools
- Three layers of metadata
  - Physical layer, connections to sources
  - Logical layer, denormalized facts and dimensions
  - Presentation layer, “report-centric” folders
- Maintained using BI Administrator
- Model is the “center” of the OBIEE universe
Data Modeling using Oracle BI Server

- Design goal for the BI Server metadata layer is to create a Star Schema
  - Fact tables
  - Dimension tables
  - Drill paths
- Source data can either be pre-integrated (data mart, data warehouse) or integrated using the BI Server ("federated")
  - Federated can be faster to deliver (report in place)
  - Assumes data sources share common key values
  - Data marts are simpler and faster to query
- MDX (OLAP, Multi-dimensional) data is converted into RDBMS row-sets
- Logical layer is main integration point
- Presentation layer allows “report centric” folders
“Model First” Development

- Business Model is centre of the OBIEE “universe”
- First element of design that is produced
- Integrated view of the organization’s data
- Option 1: Star with model first, then map on physical sources
  - “purest” approach, uses theoretical model as basis
  - Tricky to get joins and integration working
  - Lots of manual logical table and column definition steps
- Option 2: Produce model on paper, then construct it using elements of physical model
  - Quicker and faster, just drag and drop
  - May end up with redundant columns
  - Still needs a design phase at the start
Sample Business Model
Data Modeling Workflow Steps

1. Define logical model (on paper or using BI Administrator)
2. Import physical data sources
3. Create physical primary and foreign keys
4. Either drag physical tables to create logical tables, or map
Step 1: Create Physical Layer

- Define ODBC connection from client or server to database
  - ODBC only needed to define initial connection, import metadata
  - Native connection (OCI etc) takes over from then
- Import tables, views, keys etc from data source
- Define keys, foreign keys if needed
- Update row counts
- Define table aliases
Physical Join Types

- If source data has foreign key joins, they will be imported
- If not, they can be defined in the physical model
- Two types of join
  - Foreign key joins (e.g., CUSTID = CUSTID)
    - 1:N joins only
  - Complex joins (TRAN_DATE between STARTDATEID and ENDDATEID)
    - also supports 1:1, O:1, 1:O etc
- Important to create physical joins before importing into logical layer
  - Logical joins rely on physical joins to define paths
  - Makes life a lot simpler
- Create joins using Join Manager, or Diagrammer
Step 2: Create Logical Layer

• Firstly, create new logical model
• Then create logical tables and columns
  • Either create from scratch, then connect to physical
  • Or define by dragging and dropping physical tables
• Start off with single source for each table, then add connections
• Define dimensions to create drill paths
• Add calculations
• Use Rename Wizard to clean up table and column names
Calculations

- OBIEE has its own SQL syntax
  - Cross-platform
  - Same SQL across all sources
  - Calculations, analytics, time-series
- Calculation Wizard for time-series calculations
  - AGO, TO_DATE
- Oracle BI Answers features SQL pass-thru
  - Has to be enabled per user
  - Allows native functions to be used
- Future: direct support for native SQL functions
Time-Series Calculations

- OBIEE 10.1.3.2 supports limited time-series functions
- AGO – value of a measure x months, quarters, years ago

\[ \text{AGO(Videostore."Sales fact view".Sales, Videostore.TimesDim."Month", 1)} \]

- TODATE – value of a measure to the end of a month, quarter, year

\[ \text{AGO(Videostore."Sales fact view".Sales, Videostore.TimesDim."Quarter", 1)} \]

- Requires at least one “time dimension” in the metadata layer
  - Requires at least one or more Chronological keys
  - Monotonically increasing value (DAY etc)
Create Calculations in the Optimal Location

- Whilst the BI Server can create calculations, it may be more appropriate to do this in the underlying database
  - Oracle Database analytic functions (LAG, LEAD, RANK etc)
  - Oracle Database OLAP functions (time-series, forecast, allocation)
- Use the available architecture efficiently
  - Perform calculations as close to the source data as possible
  - Use dedicated OLAP technology if available
  - Use BI Server to provide features not found in the sources
- Just because the BI Server *can* do something, doesn’t mean it *should* do something
Logical Join Types

- Logical layer supports both foreign key and complex joins
- Foreign key joins not recommended
  - Restrict choices that BI server can make over join paths
  - Only there for backward compatibility
- Complex joins are preferred
  - Indicates relationship between tables
  - Allows BI Server to determine best join path
  - Useful when more than one Logical Table Source
Logical Table Sources

- Logical tables can have more than one physical source
- Can create >1 Logical Table Source
  - BI Server creates two physical queries
  - Useful when mapping data from different levels
  - and when using “fragmentation”
- Each Logical table source can have >1 table mapped
  - BI server creates one query per LTS, joining as per physical foreign key joins
  - Used when you need columns from other joined tables in your logical table (join only happens if additional columns requested)
Consider Where to Join Data

- Again, consider where to integrate and join data
- Whilst the BI Server can join data, it may be faster to load into a data mart
  - ETL tools can integrate data at time of load
  - Denormalized data structures can be faster than joining at report time
  - Indexing, Summary Management, Query Optimization
- Creating a data warehouse still is the optimal solution
  - BIEE data source joining features are most appropriate when bringing new data in that’s not yet in the data warehouse
Step 3: Create Presentation Layer

- Business models themselves are not visible to users
- Users see presentation catalogs whose columns are defined in business models.
- Presentation catalogs appear as “subject areas” in the web UI
- Presentation catalogs are “databases” to client tools using ODBC
- To create a presentation catalog, drag business model to left panel. Then customize – re-order, re-folder, re-name.
- Can create multiple Presentation Catalogs from single Logical Model
Initial Reporting Scenario

- Global Company has a data warehouse, containing order and customer data
- Requirement is to take the existing warehouse (normalized) and turn it into a star schema (denormalized)
- For now, OBIEE will be used to map the data
Demonstration

Creating the initial Common Enterprise Information Model
Reporting Across Multiple Data Sources

- OBIEE can combine and join data from multiple data sources
- Data is retrieved in separate SQL, MDX calls and joined using the BI Server
- Allows you to perform federated queries
- Watch out for performance though
  - End goal should be to eventually move reporting data to a data warehouse
Federated Query Scenario

- Additional data exists in an Excel spreadsheet
- Contains quotas, plus information on products and dates
- Quotas are at the product category and month level
  - Existing data is at the product and day level
- Requirement is to join data, but only show quotas the relevant level
Joining A Second Data Source : Step 1

- Import the new schema into the physical model, ensure FK and PK constraints are set
Joining A Second Data Source: Step 2

- Use the Join Manager, or Physical Model Diagrammer, to create joins between tables in the two schemas.
Joining A Second Data Source : Step 3

- Add the new dimension tables as additional logical table sources to the existing logical dimension tables
- Set the dimension hierarchy level at which they are applicable
Joining A Second Data Source: Step 4

- Add the additional measures to the fact table
- Set the dimension levels at which the measures become relevant
Demonstration

Adding a Second Data Source
Incorporating Real-Time Data

• Initial data set consists of historical Customer Data Warehouse data
• Data also exists for real-time Order Bookings system
• Two logical models could be created, but ideal is to combine historical and real-time data in one model
  • Known as “Fragmentation”
Fragmentation

• Logical tables have >1 Logical Table Source
  • Logical Table Source 1 = Historical Data
  • Logical Table Source 2 = Real-Time Data
Fragmentation Step 1

• Add new logical table source to Items logical table
Fragmentation Step 2

- Ensure mapped tables are added to new LTS
Fragmentation Step 3

- Add Fragmentation Content conditions

Fragmentation content:

```
"Rmdemo - Final".Items.Orderdate >= '01-jan-2000' AND "Rmdemo - Final".Items.Orderdate < '01-jun-2007'
```

This source should be combined with other sources at this level

Fragmentation content:

```
"Rmdemo - Final".Items.Orderdate >= '01-jun-2007' AND "Rmdemo - Final".Items.Orderdate < '31-dec-2020'
```

This source should be combined with other sources at this level
Fragmentation Step 4

- Repeat process for Dimensions
- Do not add fragmentation condition if same data in both dimension versions
  - Add “distinct” clause instead
Demonstration

Adding real-time data using Fragmentation
Oracle BI Presentation Server

- “BI Application Server”
- Single application that provides Oracle BI Answers, Dashboards
- Metadata stored in Web Catalog
- User Profiling, Security and Session Management
- SOAP Web Services, XML and URL Interface
Oracle BI Answers

- Query authoring tool for BI EE
- Report against business model
  - Tables, Cross-tabs
  - Graphs, Gauges
  - View and column selectors
  - Narratives
- Publish to Oracle BI Interactive Dashboards
- Database-neutral
- Connects through BI Server
Answers Views

• Each report (“request”) contains a single query
  • Possible to union, minus etc with similar queries
• Query has one or more views associated with it
  • Table, pivot-table
  • Chart, gauge, funnel chart
  • Title, narrative
  • View and column selectors
• Views are combined in the compound view
• Compound view generally displayed in dashboard
Answers Formatting Options

- Columns can be formatted on column, or other column value
- Colors, fonts,
- Images and icons
  - Can replace or supplement figures
- Images can be added to report titles
- Comes with a library of stock images
Oracle BI Interactive Dashboards

- Main method of interacting with reports
- Fully interactive
  - Drill on data, graphs
  - Switch between views
  - Drill out to detail
- Incorporate images, web content
- Role-based security
- Lightweight, DHTML
Dashboard Interactivity

- Prompts can be added to dashboard
- Drop-down lists, multi-select
- Passes parameters to reports
- Switch between different views
- Change columns used in reports
Oracle BI Publisher

- Publishes Answers reports as PDF, XLS, RTF etc
  - Also direct access to metadata layer
- “Pixel-perfect” reports
- Used for distribution-quality reporting
- Integrated into dashboard
Oracle BI Delivers

- Scheduling and distribution
- Uses iBots (“Intelligent Bots”)
- Integration (through Java) with BPEL
  - iBot chaining, calling from BPEL, calling BPEL
- Permits alerts, through execution of Answers reports
  - If report returns rows, alert triggered
  - Alert appears on dashboard, via email, SMS, cell phone
- Can trigger narrative and conditionally-appearing reports on dashboard
Dashboard Alerts & Guided Analytics

- Enables content and layout of dashboard to change based on information analyzed.
- Sections or links can be defined to only appear when there is “interesting” information in the data.
Types of Guided Analytics

1. Adding a link to a dashboard, that only appears if a condition is met
2. Making reports appear on the dashboard, when a condition is met
   • Requires two things
     • A report to run, that determines whether the event has happened
     • A report, or link, that you want to appear
Step 1: Create a “Trigger” Report

- Create a report that checks for your condition
  - Has a branch fell below its sales threshold?
  - Has a product’s year on year sales fallen?
- Use filters to check for the condition
- If the report returns rows, the trigger will be met
- Save to the Web Catalog
Step 2: Add Guided Analytics Link to Dashboard

- Drag and drop from Dashboard Objects palette
- Add to dashboard, just under the report to which it is related
Step 3 : Edit the Guided Analytics Link Properties

- Select the report that is run to determine whether the link is displayed
- Select whether *any* rows, or *no* rows, is the trigger
- Enter link text
- Select report, dashboard or URL to link to
Step 4: View Link in Dashboard (if Triggered)

- Link only appears if condition is met
- Guides user to the next action they should take
Making Sections Display Conditionally

• Individual sections can be displayed conditionally
• Select “Guided Navigation” from the section properties
• Pick the report to run
Demonstration

Guided Analytics and Conditional Display
Oracle Data Integrator

- Comprehensive data integration platform
- Extraction, transformation and load of data, to and from databases, events and Web services
- Batch and real-time loading
- Java-based, hot-pluggable, database independent
ODI and Oracle BI EE

• Complements the data integration capabilities of the BI Server
  • More complex integration
  • Larger data volumes
  • Create offline data mart
• Integration of data from services, events
• Cross-platform
• Creating history tables, SCDs
• Real-time integration of new data
Previously Known as “Sunopsis”

• Sunopsis were an Data Integration tool vendor originally from France, with offices in the USA and UK
• Two major products
  • Sunopsis Data Conductor (database integration)
  • Sunopsis Active Integration Platform (event and serviced-based integration)
• Sold on it’s platform independence, Java architecture and EAI capabilities
• Acquired by Oracle in 2006
Part of Oracle Fusion Middleware

- Data Integration tool, part of the Fusion Middleware family
- Cross-platform
- Hot-pluggable
- SOA-Enabled
Data Integrator Overview

- Same philosophy as OWB – Use the Database as the ETL engine
  - ODI supports heterogeneous databases, not just Oracle
- Built for SOA environments
  - Support for Web Services, EII etc
- Supports batch, event-based and real-time integration
- Data Integrity Controls create a data “firewall”
  - Reduces data prep time by not processing erroneous data
- Extensible through “Knowledge Modules”
  - Change Data Capture
  - Slowly Changing Dimensions
  - Bulk load
- Java client application with server elements
Oracle Data Integrator Architecture
ODI Key Components

- **Designer**
  - Design data flows, processes
- **Operator**
  - View progress of jobs
- **Topology Manager**
  - Define physical database, SOA connections
- **Security Manager**
  - Define permissions on project elements
- **Agents**
  - Schedule and orchestrate mappings, process flows
- **Metadata Navigator**
  - Web-based reports on repository
Business Rules and Technical Details

- ODI splits data mappings into business rules and technical implementation.
- Business rules define what goes where and using which transformation rules.
- Technical implementation defines how data is moved.
  - Changed Data Capture
  - SQL to SQL
  - File to SQL
  - Slowly Changing Dimension
- Allows you to split mapping role into business analyst and technical specialist.
Extensible Knowledge Modules

- Provided “out of the box”, and are user extensible
- Written for specific source and target platforms
- User created example: Oracle 10g Data Pump
Knowledge Module Categories

• RKM: Reverse-Engineering Knowledge Modules
• JKM: Journalizing (CDC) Knowledge Modules
• LKM: Load (Extract) Knowledge Modules
• CKM: Check (Constraint) Knowledge Modules
• IKM: Integrate (Load) Knowledge Modules
• SKM: Service (Web Service) Knowledge Modules
Event and Service-Based Integration

• Integrates data from events (files arrive, data arrives on queue) and from services (SOA, Web services)
• Perform integration in real-time (Enterprise Information Integration)
• Reflects today’s reality that not all data lives in databases
• Integrates with Oracle SOA Suite
Oracle Data Integrator and OWB

- Complements OWB by adding advanced data integration functions
  - Works in the staging/integration layer
  - Pre-build packaged solutions for CDC, Web Services etc
  - Supports event-based, service-based and real-time integration
- Complements the BI Server by handling large data volumes and data marts
  - Provides heterogeneous functionality - OWB may be more appropriate for Oracle data warehouses
  - Create offline store (data mart) with history tables (SCD etc)
  - Perform complex integration tasks
  - Integrate SOA and event-based data
- Adds the ability to target heterogeneous platforms
- Supports real-time data integration
ODI Role in a Typical Oracle Data Warehouse

- **Oracle BI Admin**
  - OWB Core ETL
  - OWB Enterprise ETL

- **ODI**
  - OWB Core ETL
  - OWB Data Quality
  - OWB Enterprise ETL

- **Presentation**
  - Reports, Dashboards, Alerts, KPIs
  - Star schemas, OLAP, Spatial, Data Mining
  - Normalised integrated tables
  - External tables, Staging Tables, Profiling
  - CDC Subscriber Views, Work area, Corrected Schemas
Usage Scenario: Real-Time Integration

• Requirement to create a sales data mart on Microsoft SQL Server 2000
  • With loading in real-time (trickle-feed)
• Source data held in an Oracle 10.2 database, and a flat file
  • Oracle data to be captured using Changed Data Capture
  • Leverages Oracle 10.2 CDC feature (asynchronous hotlog CDC)
• Three main challenges
  • Load a non-Oracle database
  • Capture data using CDC
  • Integrate in real-time

• Scenario published on Oracle Technology Network
An Introduction to Real-Time Data Integration

by Mark Rittman

A introduction to Oracle Data Integrator, Java-based middleware that uses the database to perform set-based data integration tasks in an SOA.

In these days of complex, “hot-pluggable” systems and service-oriented architecture (SOA), bringing data together and making sense of it becomes increasingly difficult. Although your primary applications database might run on Oracle Database, you may well have other, smaller systems running on databases and platforms supplied by other vendors. Your applications themselves may intercommunicate by using technologies such as Web services, and your applications and data may be hosted remotely as well as managed by you in your corporate data center.

Oracle Data Integrator, a member of the Oracle Fusion Middleware family of products, addresses your data integration needs in these increasingly heterogeneous environments. It is a Java-based application that uses the database to perform set-based data integration tasks but extends this capability to a range of database platforms as well as Oracle Database. In addition, it gives you the ability to extract and provide transformed data through Web services and messages and to create integration processes that respond to and create events in your service-oriented architecture.

Oracle Data Integrator Product Architecture

Oracle Data Integrator is organized around a modular repository that is accessed by Java graphical modules and scheduling agents. The graphical modules are used to design and build the integration process, with agents being used to schedule and coordinate the integration task. When Oracle Data Integrator projects are moved into production, data stewards can use the Web-based Metadata Navigator application to report on metadata in the repository. Out-of-the-box Knowledge Modules extract and load data across heterogeneous platforms, using platform-specific code and utilities.
Why Use ODI for this Integration Task?

- Target DB is MS SQL Server, ODI has better tools (currently) for populating this target
- ODI provides a user interface for Oracle Change Data Capture
Data Integration Steps

1. Define data sources and targets
2. Import table metadata in to ODI repository
3. Set up journal (CDC) on Oracle data source, through ODI
4. Start the journal, check it’s working
5. Define interface, selecting table journal as source
6. Create a package (process flow) that listens for new journal data,
1. Defining Data Sources and Targets

- Connect to Oracle via OCI
- Connect to MS SQL Server via JDBC/ODBC bridge
- Connect to file via ODI JDBC driver
2. Import Table Metadata into Repository

- Reverse-engineer Oracle and MS SQL Server data in using ODI
  use RKMs where possible
- Create file metadata manually
- Metadata available as models, in Designer module
3. Set Up Table Journal (CDC)

- Oracle, IBM DB/2 support native change data capture
  - Oracle synchronous (triggers) or asynchronous (hotlog)
  - Other RDBMS use ODI functionality (triggers etc)
- Set up within ODI, defined at table (Data Store) level
- Single tables, or groups
4. Start the Journal, Check it’s Working

- Automatically creates CDC control tables first time around
5. Create Interface using Journal Data

- Only additional step to use CDC data is to check the “Journalized Data Only” checkbox in interface (mapping)
6. Add Package that Listens for New Data

- Add `odiWaitForLogData` event listener at the start of package
- When event occurs (set number of log entries, or timeout)
  - Interface is executed (set number of rows found, or timeout)
  - Interface is skipped (timeout and no rows)
  - `odiStartScen` then restarts package, to listen for more log data
7. Execute the Package through an Agent

- Runs continuously, executes interface when set number of journal log entries are found, or when event detector times out
- Ensures that process does not wait too long
- Monitored through Operator Module
Demonstration

Oracle Data Integrator
Summary

• In today’s masterclass, we have looked at three main areas
  • The Oracle BI Server architecture
    • Data Modeling
    • Integrating multiple data sources
    • Incorporating Real-Time Data
  • The Oracle BI Presentation Server
    • Basic Answers and Dashboard functionality
    • Guided Analytics
  • Oracle Data Integrator
    • Integrating heterogeneous data in real-time
And Finally...

• Thank you all for attending
• I hope the two hours have been of interest to you
• Please remember to complete the session evaluations
• Leave your contact details with me to forward on the slides
• Have a safe journey home
• Goodbye!