

Oracle OBI EE: The Case for Self Service



“Enabling Self-Service BI with OBI EE”

Oracle Business Intelligence (BI) Applications are prebuilt BI solutions that deliver, role-based intelligence for everyone in an organization from front line employees to senior management, to enable better decisions, actions, and business processes. Designed for heterogeneous environments, these solutions enable organizations to gain insight from a range of data sources and applications.

Oracle BI Applications are built on the Oracle BI Suite Enterprise Edition, a comprehensive, leading BI platform in the Oracle applications market. The OBIEE presentation catalog contains pre-built metrics, analytic decision workflows, reports, dashboards and alerts (across functional areas) used to deliver information, stored in the Business Analytics data warehouse, for end user consumption.

However, Oracle advises, wherever possible, to use the out-of-the-box content/functionality and keep customizations to a minimum. Even with the robust functionality built in to this solution, most customers find that changes are required. Key reports may be missing content and dashboards may not adequately reflect key performance indicators. Some common areas that require customization are discussed below:

AREA 1 – Simple entity attribute extensions

The data warehouse contains a large number of entities and attributes. However, not all are exposed in the repository and thus not available for reports and dashboards. One of the simplest customizations is to make these attributes available in the logical and presentation models.

Effort Required:

Change OBI EE to add the attribute(s) to the physical/logical/presentation models

AREA 2 – Complex entity attribute extensions

Adding attributes to existing dimension entities, and measures to existing fact entities. The complexity of this customization depends on the nature of the information intended in the attribute. If the attribute is part of the same table in an existing ETL query then the inclusion is relatively simple.

Effort Required:

1. Identify and add column(s) to the physical staging and target tables.

NOTE: Assumption is that the dimension/fact grain remain the same.

2. Modify the source/target ETL mappings to add intended attribute(s) to the extraction logic and the target table.

NOTE: Source Dependent Extract mappings (SDE), Source Independent Load routines (SIL)

Oracle OBI EE: The Case for Self Service



3. Change OBI EE to add the attribute(s) to the physical/logical/presentation models.

AREA 3 – New Fact tables, Parent Child Hierarchies, Bridge tables

Adding new facts, parent child hierarchies or bridge tables (a technique for modeling many-to-many relationships) will require adding new extraction mappings to the staging area and new loading routines from the staging area to the data warehouse.

Effort Required:

1. Additional physical tables created in the staging and data warehouse.
2. Add new extraction mappings (SDE) to populate the target staging table.
3. Add new load routines (SIL) to populate the data warehouse target table.
4. OBIEE changes to add new entities, attributes to the physical/logical/presentation models.
5. Adding new SDE and SIL to the DAC dependency.

AREA 4 – Support localizations

Supporting localization in OBIA requires the incorporation of localizations in the repository metadata, data warehouse data and reports/dashboards.

Effort Required:

1. Additional translation tables/ modification to existing columns in the staging and data warehouse areas.
2. Add/Modify source extraction mapping (SDE) to populate the target staging table.
3. Add/Modify target load mapping (SIL) to populate the data warehouse target table.
4. Session variables and configuration settings are changed, and then supporting metadata objects are created.
5. OBIEE changes to add new entities, attributes to the physical/logical/presentation models.
6. Adding new SDE and SIL to the DAC dependency

AREA 5 – Fact entity grain modification

To add new Dimensions to an existing Fact requires changes to existing SDE and SIL mappings for the Fact table. In addition, SDE and SIL mappings are required to implement new dimensions.

Effort Required:

1. Additional physical dimension table created in the staging & data warehouse.
2. Additional key fields to the staging and target fact tables.
3. Add new SDE source mapping to populate the target staging table.

Oracle OBI EE: The Case for Self Service



4. Add new SIL target mapping to populate the data warehouse target table.
5. Change the existing SDE source mapping to add the additional keys to the staging fact table. Create requisite workflows.
6. Change the existing SIL target mapping to add the additional key column(s) to the data warehouse target fact table. Also, include the key lookup to the new dimension. Create requisite workflows.
7. Create DAC tasks with required dependencies.
8. OBIEE changes to add new entities, attributes to the physical/logical/presentation models.

AREA 6 – Create an entirely new Star Schema

Create a new fact entity with some new dimension entities along with some existing ones.

Effort Required:

1. Additional physical dimension entity and fact entities created in the staging, data warehouse.
2. For each new physical dimension entity and fact entity a new SDE source mapping is needed to populate the new target staging tables. Create requisite workflows.
3. For each new physical dimension entity and fact entity a new SIL target mapping is needed to populate the data warehouse target table. Create requisite workflows.
4. Create DAC tasks with required dependencies.
5. OBIEE changes to add new entities, attributes to the physical/logical/presentation models.

AREA 7 – Support for additional source system

It is possible that there might be a requirement to integrate additional data sources (home grown legacy systems or other ERP systems) into the Business Analytic Warehouse (BAW). Additional sources may include simple spreadsheets to interlacing data into existing data models.

Effort Required:

1. Additional columns created for additional attributes in dimension and fact table(s) created in the staging and data warehouse.
2. Modify the existing SDE and SIL mappings to include additional attributes for the new source. Create corresponding workflow(s).
3. Create DAC tasks with required dependencies.
4. OBIEE changes to add new entities, attributes to the physical/logical/presentation models.

Oracle OBI EE: The Case for Self Service



Summary

The changes described above are not uncommon and, as described, can be incorporated into OBI EE and the BI Applications. The question is, how much time and resource is required to implement the changes. Each change must follow a process, a process that involves the originating source data or application, the process for extraction, translation and loading into a data warehouse and then the delivering of that data into the physical, logical and finally presentation layers of the solution. IT involvement is not optional, rather, it is mandatory.

Requirements Conundrum

Given the process involved to make changes, it follows that it is imperative that requirements be well understood and well documented. Anyone that has been involved in requirements definition knows that this can be a real challenge. Not only is it difficult for end users to effectively communicate their requirements, requirements will change. The data analysis process, turning raw data into actionable information, is cyclical. Business or change drives new information needs. Requirements may be well documented initially, however, if the implementation takes weeks or months, the requirements may change and new requirements may arise.

Documentation of Data Needs vs. Self-Service Prototyping

Business users have firsthand knowledge of business changes and hence the best understanding of the requirements for specific data. These requirements can be documented and sent forth for further interpretation and implementation as described above. However, it would be more efficient if users could "prototype" using actual data, viewing it in a variety of ways, collaborating with peers, without going through the time and resource consuming processes described above. Some analyses may never need to be implemented for the enterprise audience and those that do, will be well thought out and prototyped. Given the effort required to implement change in the OBI EE environment, prototyped analyses would be more effective and efficient versus "documented" requirements. To enable fast track prototyping, requires a BI platform that is simple and intuitive for end users on one hand, and requires minimal IT support on the other.

Self-Service Prototyping BI Platform

CarbonBI is a "NO ETL" BI platform with the full functionality to fill the gaps in a typical life cycle of enterprise report and dashboard development. By removing the necessity of an ETL process along with the requirement for defining facts and dimensions, changes to reports, analytics and dashboards can occur very quickly. Better yet, these changes can be executed by users themselves, removing the time delay in communicating changing requirements to IT, and allowing the freedom to model and discoverer different views of the data. IT is then released from the process, allowing more focus on strategic, enterprise level requirements.

Oracle OBI EE: The Case for Self Service



CarbonBI enables users to quickly source data in a self-service manner, supporting a range of data analysis requirements including:

- Ad hoc and operational reporting
- "On-the-Fly" analytics, for trending, ranking, and period comparisons
- Analysis and comparisons of data from heterogeneous data sources
- Visualizations or Dashboards
- Collaboration with peers for feedback
- Scheduling daily reports and dashboards
- Output options including Excel and pixel perfect PDF formats

Complementary Use Cases:

1. OBI EE is under evaluation but not yet implemented.
 - a. Use CarbonBI to prototype reports, analysis and dashboards
 - b. Enable users to better communicate requirements that can later be implemented in OBI EE.
2. OBI EE is installed but the BI Apps implementation is stalling.
 - a. Use CarbonBI as a prototyping platform, enabling users to drive the BI Apps implementation methodology
 - b. IT can then more easily roll-in proven, accepted prototypes
 - c. IT gains much better visibility into end user requirements and leverage this throughout the implementation
3. OBI EE is implemented but the change request backlog is growing.
 - a. Use CarbonBI to provide self-service, no "ETL", data analysis. Some requirements may be addressed "locally" at the department level and never require implementation at the enterprise level.
 - b. Enable users to address the reporting backlog in a self-service, collaborative manner
 - c. Enable IT to implement BI Apps following user led, prototyped analysis.

Oracle OBI EE: The Case for Self Service



Mashboards



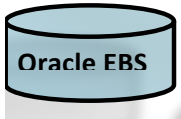
Analytics - Ad Hoc
& Operational Report
Viewer



CarbonBI Platform



Data Sources



Output Formats



Oracle OBI EE: The Case for Self Service



Conclusion

The investment in OBI EE and the BI Apps can provide value to organizations willing to accept the large upfront investments required to institute standardized reporting, dashboards and KPI's. However, data can be leveraged to a much greater extent, if access is easier for end users. Prototyping, collaborating and socializing data, analysis and dashboards, can provide tremendous benefits at a relatively low cost.

CarbonBI provides an opportunity to address short term, ad hoc and analytical reporting needs in a more efficient and cost effective manner. As analysis is prototyped in a self service manner, candidate reports and dashboards for enterprise use can then be put forth, with the knowledge they have been well socialized, tested and documented.

Leveraging a technology that has been designed to enable user self-service, requiring no ETL for rapid implementation, and minimal IT governance, dramatically reduces costs and streamlines the processes involved in providing actionable business intelligence throughout an organization. Perhaps most importantly, opportunities for lowering costs or increasing revenues, identified at the "field" level, have a much better chance of being realized.

"Drive more value from YOUR Data!"