



Manage the DW Lifecycle using OWB10gR2

Mark Rittman, Rittman Mead Consulting

mark.rittman@rittmanmead.com



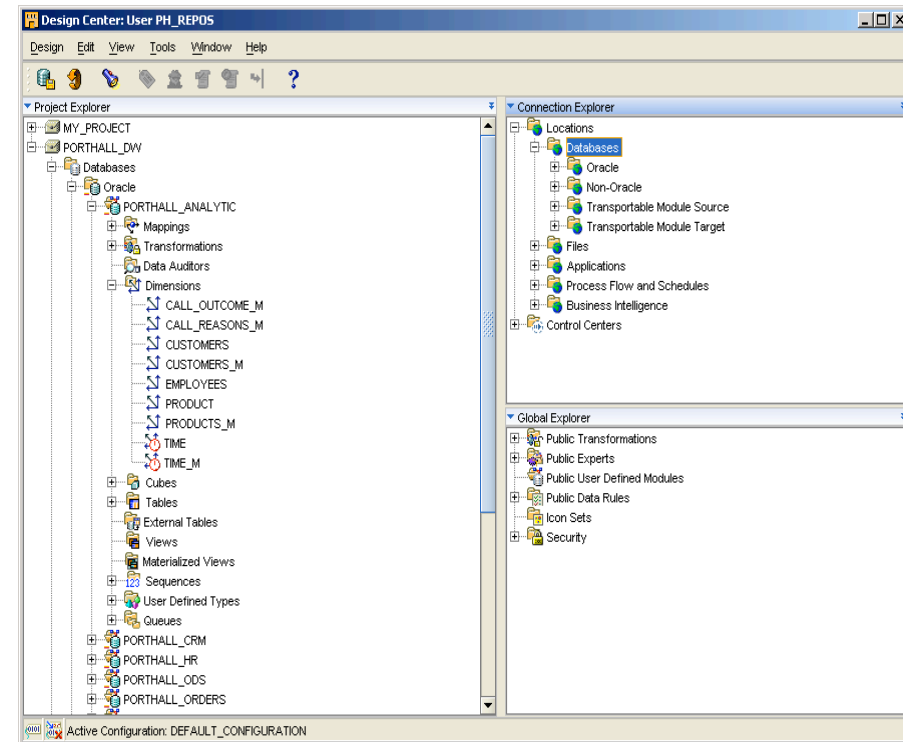
Who Am I?

- Oracle BI&W Architecture & Development Specialist
- The “Rittman” of “Rittman Mead Consulting”
- 10+ years with Discoverer, OWB etc
 - Now working with BI EE, BIP, Data Integrator
- Oracle ACE, blog at <http://www.rittmanmead.com/blog>
- Articles at OTN, Oracle Magazine etc
- Chair of UKOUG BI & Reporting Tools SIG
- Author of forthcoming Oracle Press book
“Oracle BI Suite Developers Guide”



Oracle Warehouse Builder 10gR2

- Oracle's Data Warehouse design & build tool
- Now at version 10g Release 2, 11g due late 2007?
- Data Modeling
- Data Mapping
- Data Migration
- OLAP
- Data Quality





Getting Started with Warehouse Builder

- It's easy to get started with Warehouse Builder
- Free download from <http://otn.oracle.com>
- Simple install for demo or POC environments
 - **Single database:** single repository + target database
 - **Laptop installation:** software & DB together
 - **Single environment:** all development
 - **Single developer:** you!





Typical Use of Warehouse Builder

The screenshot displays the Oracle Warehouse Builder (OWB) interface, which is used for designing and managing data warehouses. It consists of several main components:

- Data Object Editor:** Located at the top, it provides a visual environment for defining data objects and their relationships.
- Mapping Editor:** Below the Data Object Editor, it is used for creating and editing mappings between source and target data objects.
- Control Center:** The central component for managing the warehouse. It shows a hierarchical tree of objects under the 'PORTHALL_DW' schema, including mappings like 'MAP_CUSTOMERS', 'MAP_EMPLOYEES', 'MAP_PRODU', 'TIME_MAP', and 'TIME_M_MAP'. A context menu is open over 'MAP_EMPLOYEES', showing options like 'Deploy', 'Start', 'Configure...', 'Register...', 'Unregister', and 'Set Action'.
- Object Details:** A panel on the right that provides detailed information about the selected object. It includes a table with columns for Object, Design Status, Deploy Action, Deployed, and Deploy Status. The 'MAP_EMPLOYEES' object is listed with a 'New' design status and 'None' deploy action, and is currently 'Not Deployed'.
- Control Center Jobs:** A section at the bottom right showing a list of deployment jobs. The table below details these jobs:

Job	Id	Status	Finished	Owner
DEPLOY_PLAN	315	✓	2/2007 6:18 PM	PH_REPOS...
DEPLOY_PLAN	303	✓	2/2007 5:42 PM	PH_REPOS...
DEPLOY_PLAN	291	✓	2/2007 5:42 PM	PH_REPOS...
DEPLOY_PLAN	279	✓	2/2007 5:41 PM	PH_REPOS...

Additional interface elements include a left-hand palette with various operators (Aggregator, Anydata Cast, Constant, etc.), a top toolbar, and a status bar at the bottom indicating the active configuration is 'ORA10G'.



Sounds easy, doesn't it....?



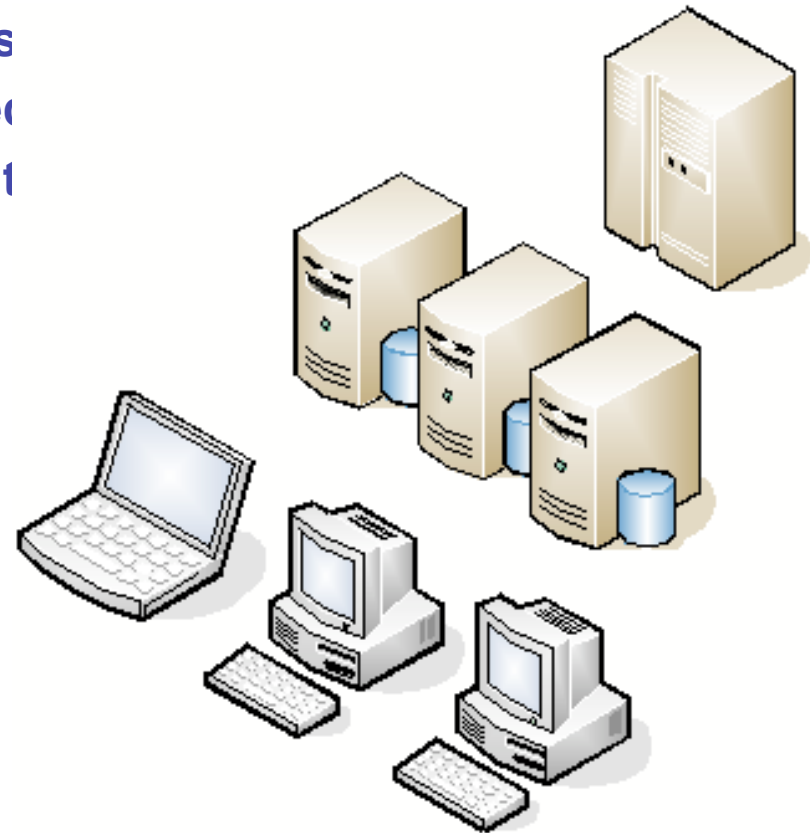


However, Real Life is Messier...

- Your data quality is unknown, and/or dirty
- You need to base your warehouse on user requirements
- Your ETL is complicated, and needs
- The warehouse needs to be released
- You need to create a layer of security
- Things need to be automated

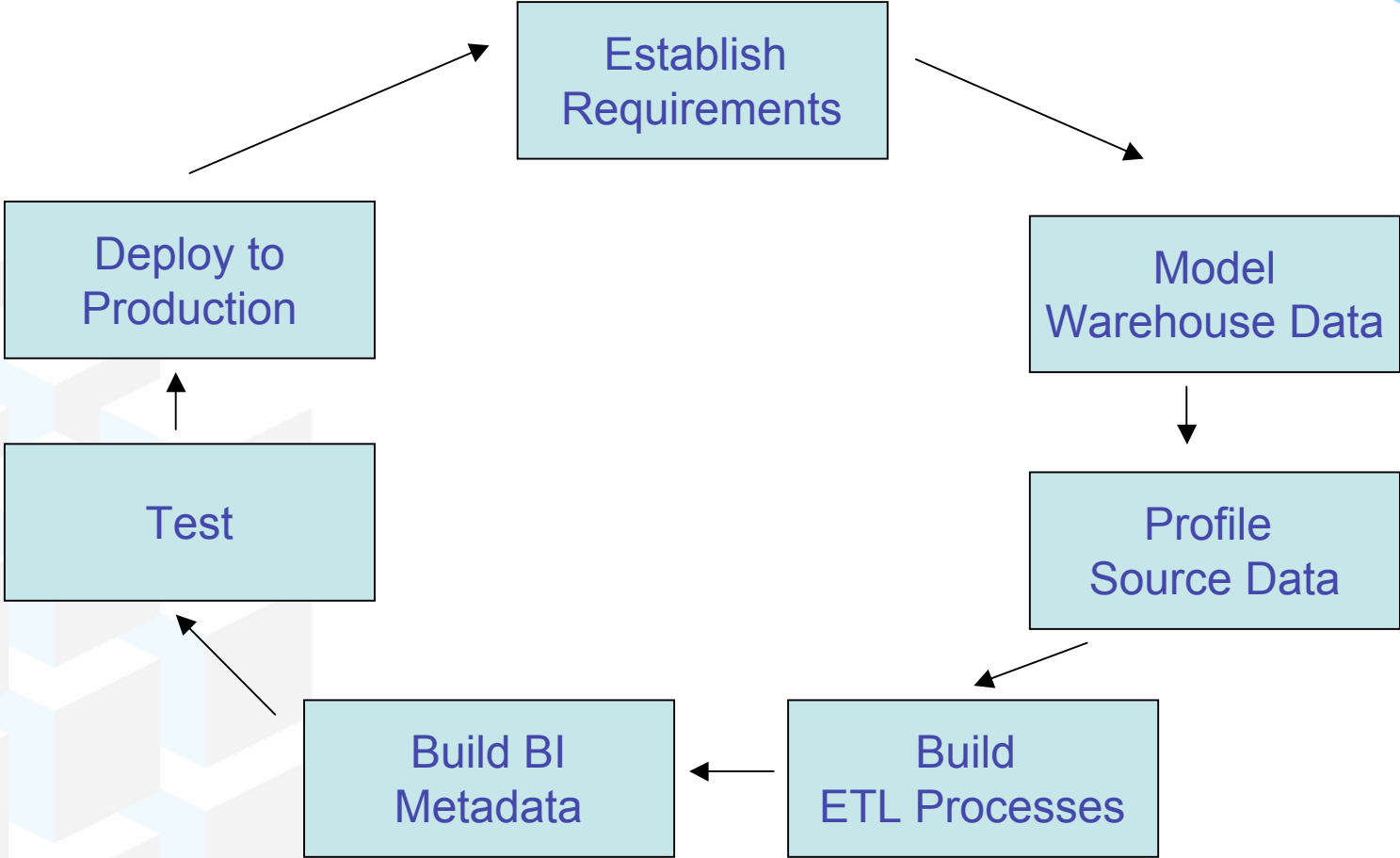
.. In short

You have to define and manage a project lifecycle.





Typical DW Project Lifecycle





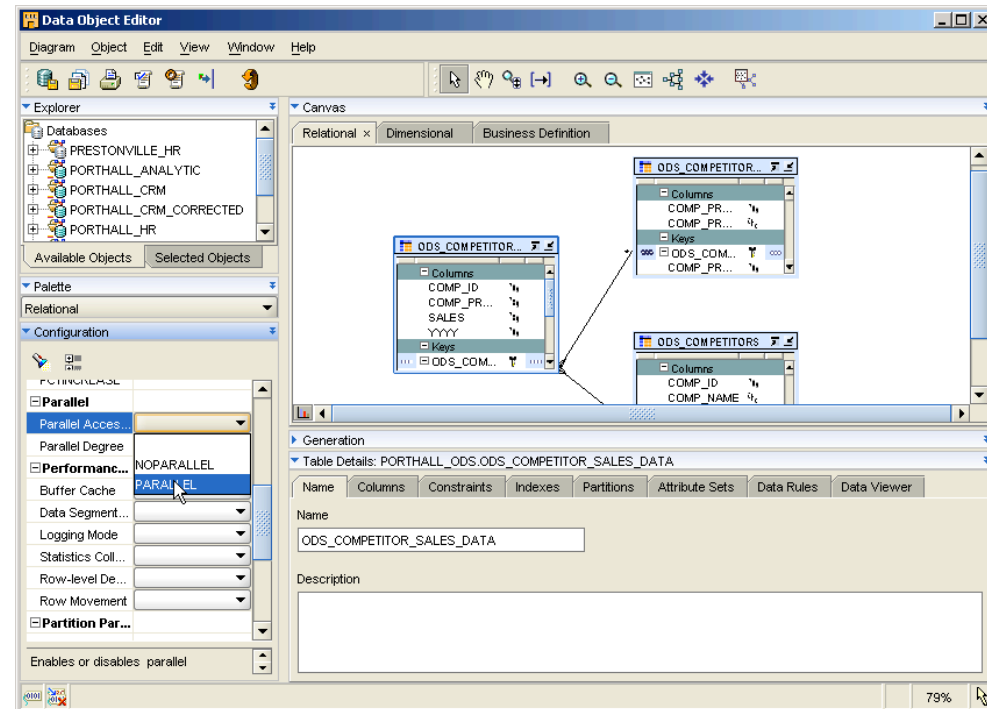
OWB DW Lifecycle Features

- **Modeling data warehouse structures**
 - Objects are held in a repository, can be versioned and controlled
- **Profiling source data**
 - Interactive data profiler, data rules and data corrections
 - Keeps data quality part of the DW development process
- **Designing ETL processes**
 - Versioning, Impact Analysis & Change Propagation
- **BI Metadata Creation**
 - Automatic derivation of Discoverer BAs from OWB project
 - Creation of Discoverer Catalog objects
- **Deployment**
 - Ability to deploy to multiple environments with different DB characteristics
 - Automation of build process



Repository-Based Data Modeling

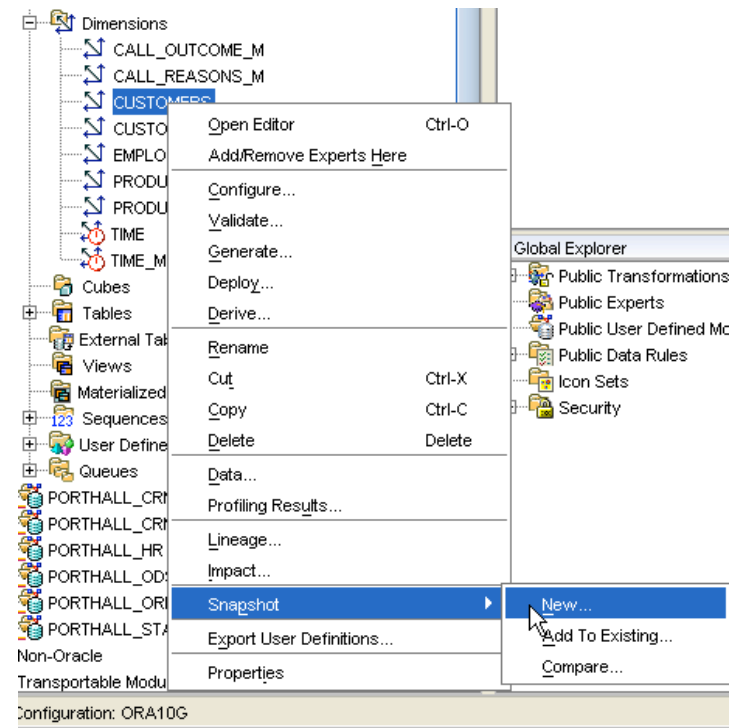
- Tables, views, dimensions, cubes etc are build in a repository
- Model your data warehouse, then deploy
- Multi-user
- Diagramming
- Import / Export from other data modeling tools
- Logical and Physical
 - Logical column design
 - Physical storage characteristics





Project Versioning

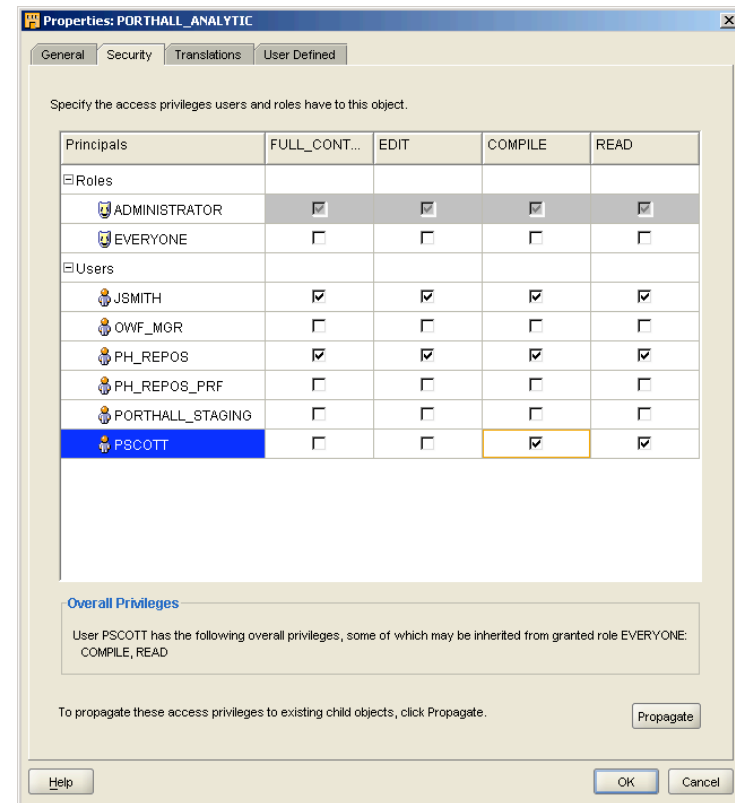
- **Need to “snapshot” project elements at certain times**
 - Records project at a certain stage
 - Allows restoration to previous point
 - Useful for establishing diffs.
- **Versioning done through Snapshots**
- **Managed using Change Manager**
- **Important: Make sure you backup snapshots!**
 - Export to MDL file
 - Backup database

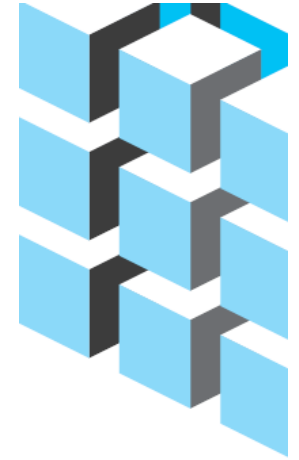




OWB10gR2 Project Security

- New in OWB10gR2 – “Fine Grained” project security
- Database accounts created on repository DB
- Access rights defined at project, module, object level
- Switched off by default
- Used to control access to project elements





Demonstration

Snapshots and Repository Security in OWB





Data Profiling

- Full user interface for profiling and understanding source data
- Keep data profiling part of the overall OWB project
- Faster and simpler than SQL*Plus
- Results stored in a repository
- Can rerun when new source data comes in
- However ... requires Data Quality Option
 - (\$20k per CPU)

The screenshot displays the 'Data Profile Editor: CRM_PROFILE' window. The main area shows 'Profile Results Carvases' with tabs for Domain, Unique Key, Functional Dependency, Referential, Data Rule, Profile Object, Aggregation, Data Type, and Pattern. The 'Aggregation' tab is active, showing a table of aggregation analysis results for the CUS_INTERACTIONS table (9 columns, 36 rows).

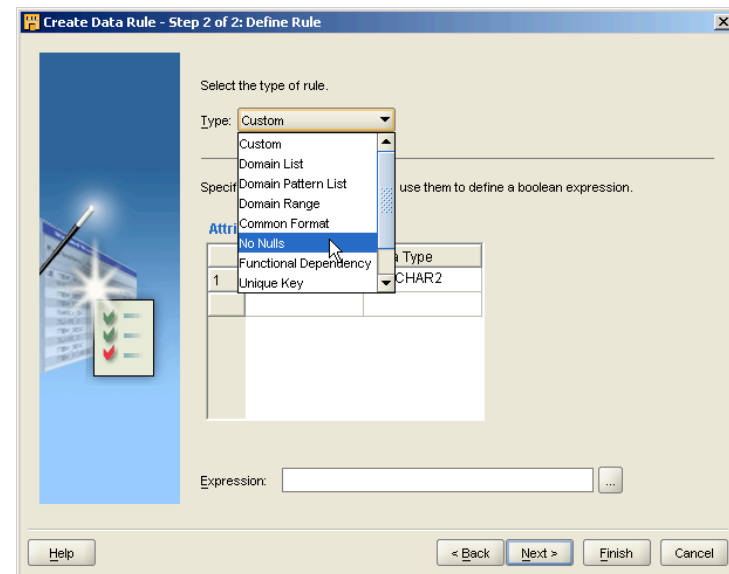
Columns	Minimum	Maximum	# Distinct	% Distinct	NOT NULL	Recommen...	# Nulls	% Nulls	Six<
CALL_DATE	10-AUG-...	31-DEC-...	36	100%	No	Yes	0	0%	7.00
CALL_OUTC...	1	4	4	11.1%	No	Yes	0	0%	7.00
CALL_REAS...	1	Brighton	6	16.7%	No	Yes	0	0%	7.00
CITY		Walforde	21	58.3%	No	Yes	0	0%	7.00
CONTACT_N...	Adams	Walter	12	33.3%	No	Yes	0	0%	7.00

Below the main table, there are sections for 'Data Drill Panel' (showing drill results for CALL_DATE) and 'Data Rule Panel' (showing applied rules and bindings).



Creation of Data Rules

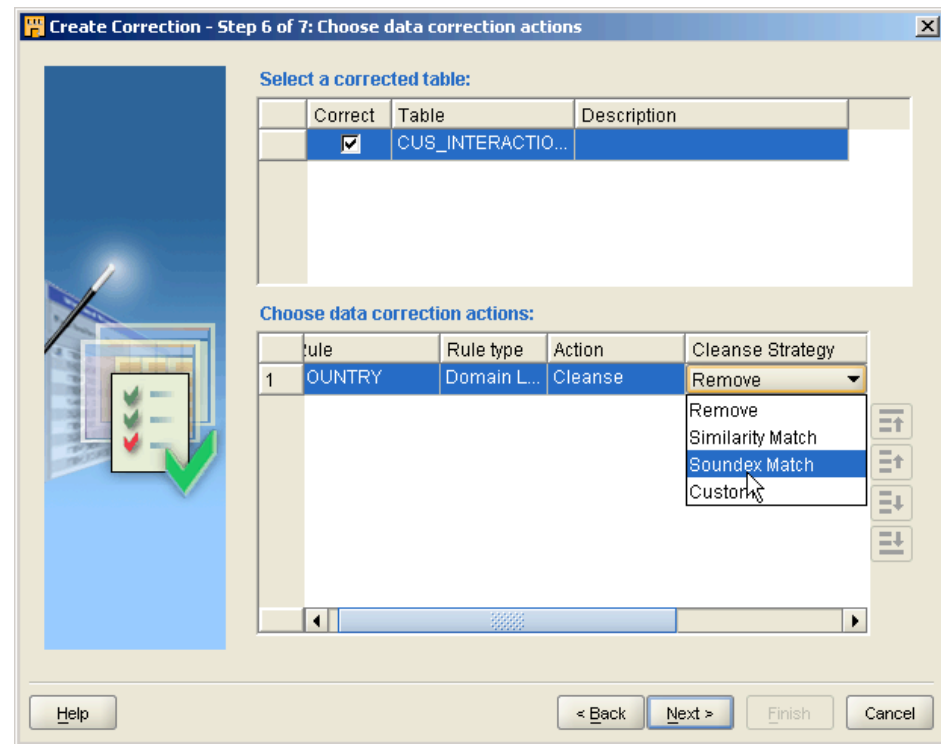
- **Automatically generate data rules based on profile results**
- **95% of rows not null**
 - Add NOT NULL data rule
- **All values are numbers**
 - Change datatype to NUMBER(10,2)
- **CUST_ID has functionality dependency on CUSTOMERS table**
 - Create a foreign key data rule
- **What happens to data that fails the data rule?**
 - Automatically correct it using OWB





Metadata-Driven Data Corrections

- Correct source data based on data rules
- All based on project metadata
- Creates a continuous process
 1. Profile data
 2. Create data rules
 3. Correct “raw” data
 4. Present it to DW load process





Data Profiling Example

- **Oracle Technology Network article:**
 - “Data Profiling and Automated Data Correction using OWB10gR2”





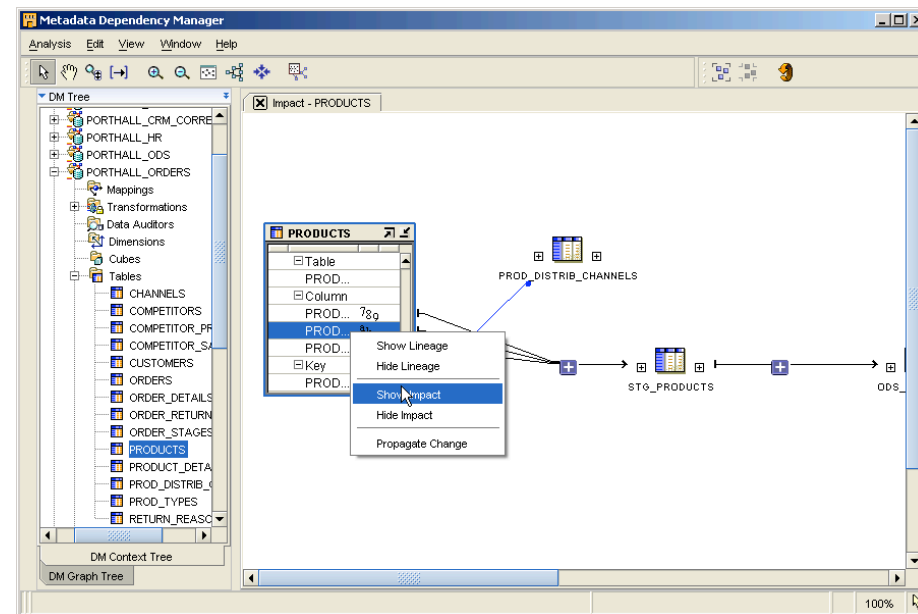
Data Mappings in OWB10gR2

- Mappings from source to target objects
- Mapping can have multiple source, multiple targets and multiple stages
- Typical DW load process has many mappings
- Process flows (OWF) used to orchestrate mappings
- Can potentially get complicated when source or DW objects change
 - Need to establish impact of change
 - Need to propagate change



Interactive Impact Analysis

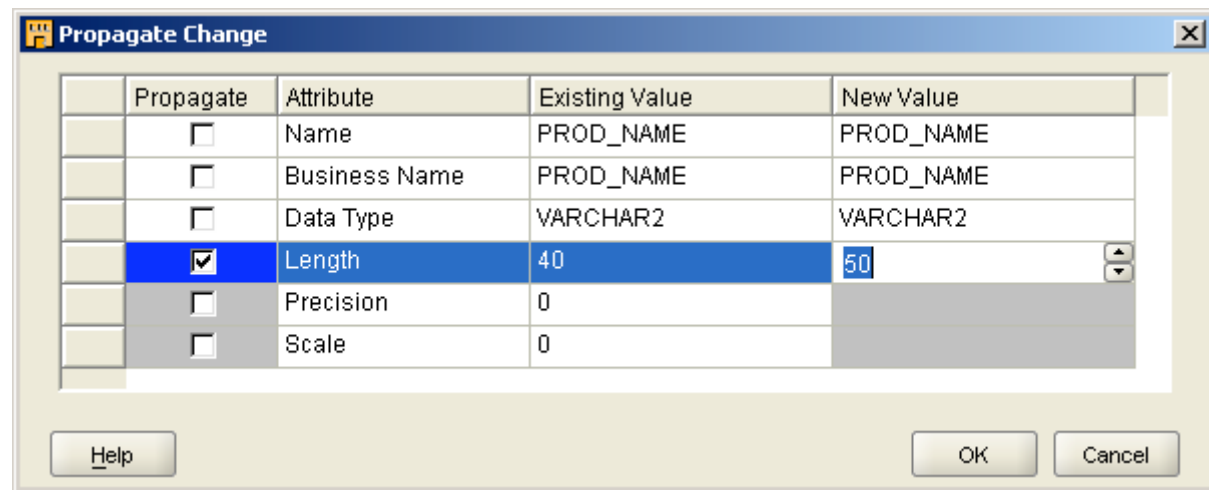
- **Right-click on any object, establish impact or lineage**
 - Impact displays dependent objects and mappings
 - Lineage shows what objects were used to populate
- **Enterprise ETL feature (\$10k per CPU)**
- **Metadata Dependency Manager provides alternate UI**

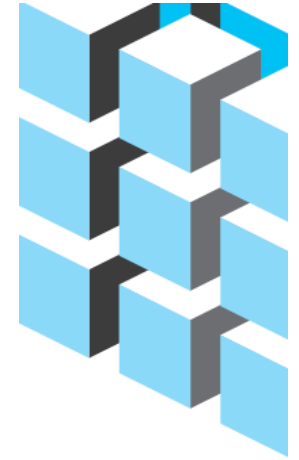




Change Propagation

- **Introduce changes to source objects, propagate change through dependencies**
 - Highlight column, select “Propagate Change”
 - Change required attributes + propagate change
- **Gotchas:**
 - Ensure you select “Show Full Impact” before propagating
 - Some transformations (PL/SQL mostly) block propagation





Demonstration

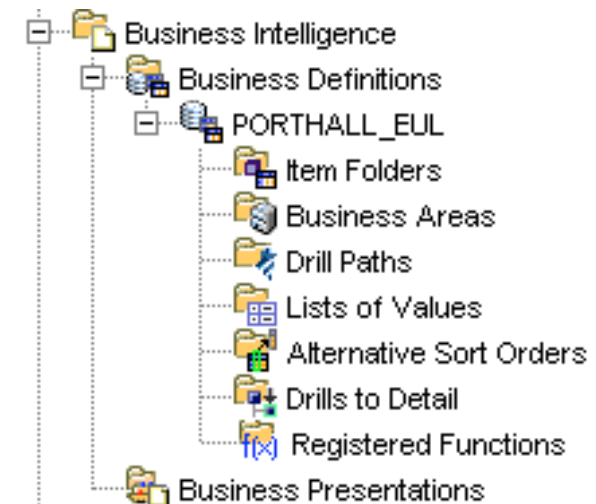
Impact Analysis & Change Propagation

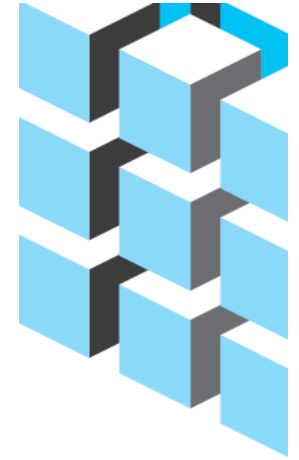




BI Metadata Creation

- **Create OracleBI Discoverer EUL elements within OWB**
 - Business Areas
 - Folders
 - Drill paths (hierarchies)
 - Item Classes
 - Lists of values
 - Alternate Sort Orders
 - Registered Functions
- **Create manually, or derive**
 - Turns dimensions in to drill paths
 - Keeps all BI elements in one repository
- **Requires Enterprise ETL Option**
- **Still need to create EUL using Discoverer Admin**





Demonstration

Oracle BI Discoverer Integration





Deployment to Production

- **Most projects have a number of environments**
 - Development
 - Test
 - Production
- **Best practice is to create a repository and target schema(s) for each environment**
 - Allows each environment to hold it's own project definition
 - Separate target schemas for each environment, with their own locations
 - Code promotion through MDL file exports
 - Script using OMB*Plus



Automation through OMB*Plus

- **Robust projects rely on build automation**
 - Quickly integrate changes
 - Enable regression testing
 - Allows creation of daily build
- **OMB*Plus is the OWB scripting language**
 - Based on TCL
 - Access to all repository features
 - Create/amend objects
 - Deploy objects and mappings
 - Import/Export between repositories
 - Combine with WSH or Unix shell scripts to automate build process



OMB*Plus Deployment Scripting Steps

1. Export development module to an MDL file
2. Import it into the production repository
3. Connect to the default control center
4. Alter the module so that the production location is it's default location
5. Alter the module so the production location is where it's deployed to
6. Register the location
7. Loop through the module...
 1. Deploy all the tables
 2. Deploy all the views, sequences, dimensions etc
 3. Deploy the mappings and process flows
8. Disconnect and mark as available for use.



Scripted Deployment to Production

```
OMBCONNECT owb_repos/password@winxpvm:1521:ora10g
```

```
OMBEXPORT MDL_FILE 'c:\customers.mdl' FROM PROJECT 'PRODUCTION_EXAMPLES'  
COMPONENTS (ORACLE_MODULE 'CUSTOMERS') OUTPUT LOG TO 'c:\customers.log'
```

```
OMBDISCONNECT
```

```
OMBCONNECT owb_repos_prod/password@winxpvm:1521:ora10g
```

```
OMBIMPORT MDL_FILE 'c:\customers.mdl' USE UPDATE_MODE MATCH_BY_NAMES  
OUTPUT LOG TO 'c:\customers.log'
```

```
OMBCC 'PRODUCTION_EXAMPLES'
```

```
OMBCONNECT CONTROL_CENTER
```

```
OMBCOMMIT
```

```
OMBALTER LOCATION 'CUS_PROD_LOCATION' SET PROPERTIES (PASSWORD) VALUES  
( 'PASSWORD' )
```



Scripted Deployment to Production

```
OMBALTER ORACLE_MODULE 'CUSTOMERS' ADD REFERENCE LOCATION  
  'CUS_PROD_LOCATION' SET AS DEFAULT
```

```
OMBALTER ORACLE_MODULE 'CUSTOMERS' SET PROPERTIES (DB_LOCATION) VALUES  
  ('CUS_PROD_LOCATION')
```

```
OMBCOMMIT
```

```
OMBREGISTER LOCATION 'CUS_PROD_LOCATION'
```



Scripted Deployment to Production

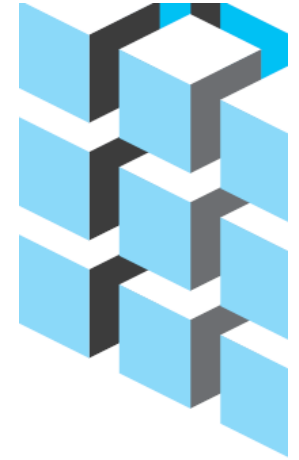
```
OMBCC 'CUSTOMERS'
```

```
set tabList [ OMBLIST TABLES ]  
  foreach tabName $tabList {
```

```
    OMBCREATE TRANSIENT DEPLOYMENT_ACTION_PLAN 'DEPLOY_PLAN'\  
    ADD ACTION 'TABLE_DEPLOY' SET PROPERTIES (OPERATION)\  
    VALUES ('REPLACE') SET REFERENCE TABLE '$tabName'  
    OMBDEPLOY DEPLOYMENT_ACTION_PLAN 'DEPLOY_PLAN'  
    OMBDROP DEPLOYMENT_ACTION_PLAN 'DEPLOY_PLAN'  
    OMBCOMMIT  
  }
```

```
set mapList [ OMBLIST MAPPINGS ]
```

```
  foreach mapName $mapList {  
    puts "deploying: $mapName"  
    OMBCREATE TRANSIENT DEPLOYMENT_ACTION_PLAN 'DEPLOY_PLAN'\  
    ADD ACTION 'MAPPING_DEPLOY' SET PROPERTIES (OPERATION)\  
    VALUES ('CREATE') SET REFERENCE MAPPING '$mapName'  
    OMBDEPLOY DEPLOYMENT_ACTION_PLAN 'DEPLOY_PLAN'  
    OMBDROP DEPLOYMENT_ACTION_PLAN 'DEPLOY_PLAN'  
    OMBCOMMIT }  
}
```



Demonstration

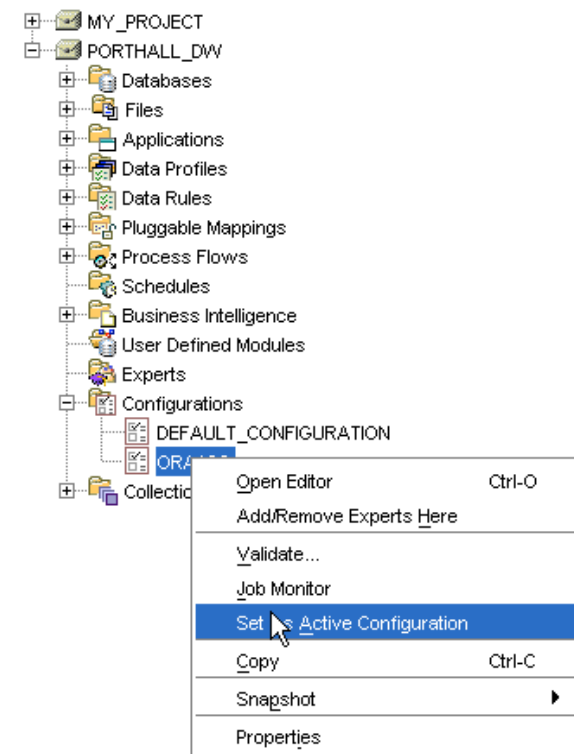
Deployment to Production





Multi-Configuration

- **Allows a project to store more than one set of physical object properties for an object**
 - Development properties (dev tablespaces)
 - Production properties (production tablespaces)
- **Allows you to quickly switch between active Control Centers**
 - Module location can switch as well
- **Alternative to multi-repositories when working with test, prod etc environments**
 - Additional configurations point to test, prod etc locations, contain test, prod object properties





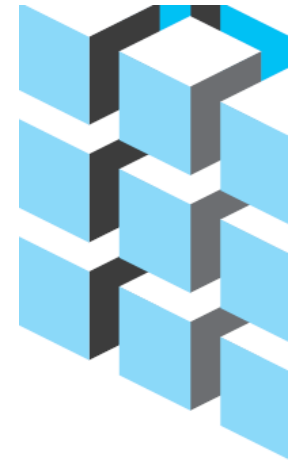
Summary

- **Real-Life Oracle Warehouse Builder projects need proper management of their lifecycle**
- **OWB10gR2 provides a number of lifecycle features**
 - A repository & security
 - Impact analysis, data lineage and change propagation
 - End-to-end data profiling and data cleansing
 - BI Tools integration
 - Ability to handle multiple deployment configurations
 - Build scripting
- **Like all new products, still a number of gotchas**
 - Useful to have a bit of experience under your belt



How We Can Help...

- **Rittman Mead Consulting's team have many years of OWB experience**
 - Large data migration projects
 - DW projects “pushing the envelope” from OWB2.1 to 10gR2
- **We can provide experience and reassurance for your project**
 - Review and QA your development plan
 - Architect and design the OWB project
 - Provide expert assistance with build and project lifecycle
 - Troubleshooting
 - Training and mentoring your team
 - Support for your OWB installation going forward
- **Contact Mark Rittman for more details**
 - mark.rittman@rittmanmead.com
 - +44 7866 568246



Questions & Answers

