

Oracle Utilities Advanced Spatial and Operational Analytics

Administration Guide

Release 2.4.1

E40760-01

May 2013

Copyright © 2000, 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007).

Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Contents

Preface	i-i
Audience	i-i
Pre-Requisite Knowledge	i-i
How This Guide Is Organized	i-i
Related Documents	i-ii
Conventions.....	i-ii
 Chapter 1	
What's New in Oracle Utilities Advanced Spatial and Operational Analytics version 2.4.1	1-iii
 Chapter 2	
Overview of Data Warehouse	2-4
Source Application	2-4
Staging Area.....	2-4
Data Presentation Area.....	2-4
ETL/ELT	2-4
 Chapter 3	
Overview of Oracle Utilities Advanced Spatial and Operational Analytics	3-6
What is OUASA	3-6
Overview of ETL	3-7
Oracle Warehouse Builder Based ETL.....	3-7
Oracle Data Integrator ELT	3-8
Overview of Analytics.....	3-10
 Chapter 4	
Configuring Oracle Utilities Advanced Spatial and Operational Analytics	4-12
Configuring the Source System	4-12
Configuring ETL/ELT.....	4-12
Oracle Warehouse Builder Based ETL.....	4-13
Oracle Data Integrator Based ELT	4-25
Configuring Analytics.....	4-28
Administration Dashboards	4-28
Configuring Dashboards	4-29
Performance Recommendations	4-37
Parallelisms in Oracle Warehouse Builder Mappings	4-38
Parallelism in Materialized Views.....	4-39
Parallelism in Materialized Views and Facts	4-40
Optimizing Top N Answers.....	4-42
 Chapter 5	
Extending the Oracle Utilities Advanced Spatial and Operational Analytics	5-44
Sample Use Case	5-44
User Defined Columns	5-46
User Defined Field.....	5-46
User-Defined Measure	5-46

User Defined Degenerate Dimension.....	5-47
User Defined Foreign Key Dimensions	5-47
User Defined Dimension	5-47
Extending Start Schemas	5-47
Oracle Warehouse Builder Based Extension	5-48
Oracle Data Integrator Based Extension	5-52
Customizing Analytics	5-54
Modifying the RPD file	5-54
Customizing Answers	5-54
Customizing Report Labels	5-54

Chapter 6

Adding New Components	6-56
Creating a New Star Schema	6-56
Defining ETL for New Start Schema	6-56
Creating New Oracle Warehouse Builder Workflows	6-58
Auto Cache Refresh	6-59
ODI Based ELT for New Objects	6-59
Define and Configure	6-59
Set Up Replication	6-60
Model for New Facts	6-60
Model for New Dimensions	6-60
Model for Replicated Objects	6-60
New Interface	6-60
New Package	6-61
Creating New Analytics	6-62
Creating New Answers	6-62
Adding New Labels	6-62

Chapter 7

Maintaining Environments	7-63
Overview of Environment Maintenance	7-63
Moving Oracle Warehouse Builder based Code	7-63
Database Cloning for Oracle Warehouse Builder	7-65
Oracle Data Integrator Based Products	7-67
Moving Oracle Data Integrator Repository	7-67
Moving Metadata	7-67
Purging	7-67
ODI Session Logs	7-70
Staging Data	7-70
Replication Data	7-70

Chapter 8

Licensing and Optional Features	8-72
Oracle Warehouse Builder Licensing and Optional Features	8-72
Disabling the Optional Features in Oracle Warehouse Builder	8-72

Appendix A

Oracle Utilities Application Framework Extractors	A-1
Trigger-Based Extractors	A-1
Oracle Utilities Customer Care and Billing	A-1
Change Detect Mechanism	A-2
Fields on the Change Log	A-2
Typical Structure of Triggers	A-2
Rows in the Change Log	A-3
Extracting and Transforming Data	A-3
Two Modes of Execution	A-3
Basic Parameters Supplied To Extract Processes	A-3

Appendix B

Oracle Utilities Advanced Spatial and Operational Analytics Administration	B-6
Product Instance	B-10
Target Entity.....	B-11
Job Configuration	B-14
Source Table.....	B-15
Job Executions	B-16
Golden Gate Configuration	B-18

Appendix C

Oracle Utilities Customer Care and Billing Extractor Details	C-1
--	-----

Appendix D

Oracle Utilities Work and Asset Management Extractor Details	D-1
--	-----

Appendix E

Oracle Utilities Meter Data Management Extractor Details	E-1
Synchronization BO-based Extract Details:	E-1
Idiosyncratic Batch Extract Details:	E-3

Appendix F

Oracle Utilities Mobile Workforce Management Extractor Details	F-1
.....	F-2

Appendix G

Oracle Utilities Network Management System Extractor Details	G-1
--	-----

Appendix H

Oracle Utilities Validation Functions and Error Identification Procedure Names.....	H-1
---	-----

Appendix I

Oracle Warehouse Builder ETL Components	I-1
External Table.....	I-1
Mapping	I-1
Workflow	I-1
Packages	I-2
File Manager	I-3

Appendix J

Oracle Warehouse Builder Deployment.....	J-1
Oracle Warehouse Builder Code Generator.....	J-1
Deploying TCL Files.....	J-3
Scheduling New Workflows.....	J-4

Appendix K

Terminology.....	K-1
Data Source Indicator.....	K-1
Late Arriving Dimension.....	K-1
Referencing Foreign Keys.....	K-2
Oracle Data Integrator.....	K-2
Context	K-2
Logical Schema	K-2
Physical Schema	K-3
Model.....	K-3
Agent	K-3
Reverse Engineer	K-3
Journalization	K-3
Change Data Capture	K-3
Scenario	K-4
Knowledge Module	K-4

Appendix L

Oracle Data Integrator ETL Components L-1
 Packages L-1

Preface

This guide provides instructions for configuring and administering Oracle Utilities Advanced Spatial and Operational Analytics (OUASA).

This preface contains these topics:

- **Audience**
- **Pre-Requisite Knowledge**
- **Related Documents**
- **Conventions**

Audience

This administration guide is intended for anyone interested in the process of configuring and administering Oracle Utilities Advanced Spatial and Operational Analytics.

Pre-Requisite Knowledge

Oracle Utilities Advanced Spatial and Operational Analytics uses several technologies. The person reading this guide should have the knowledge of the following:

- Oracle Data Warehouse concepts
- Business Intelligence concepts
- Oracle Warehouse Builder
- Oracle Data Integrator
- Oracle Business Intelligence Enterprise Edition

How This Guide Is Organized

Chapter 1 describes the new features and functionalities and new changes added in Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) v2.4.1 release.

Chapter 2 and **Chapter 3** give you the overview of Data Warehouse fundamentals and Oracle Utilities Advanced Spatial and Operational Analytics product respectively. Make sure you make yourself familiar with these chapters before you proceeding to configure the product.

Chapter 4 explains how to configure Oracle Utilities Advanced Spatial and Operational Analytics once it is installed at your end. This configuration is mandatory for proper and expected functioning of Oracle Utilities Advanced Spatial and Operational Analytics.

Chapter 5 explains how to extend Oracle Utilities Advanced Spatial and Operational Analytics. OUASA provides you out of the box star schema, extract- transform - load programs, and analytics. This chapter explains how you can extend the OUASA product by adding user-defined fields. This is a very important concept to understand and helps you extend the product easily. This is the only supported means of extending OUASA product.

Chapter 6 explains that you can add new requirements. To add a new star schema to Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) product, this chapter explains how it is done to remain as close as possible to the base product. However, any new requirements are required to be maintained by the customer, and Oracle does not support changes to the base products (other than the one explained in chapter 6) or additional star schema or related functionality.

Chapter 7 describes methods to maintain the environment. The methods explained in this chapter help customers maintain their environments easily. It is recommended that you refer to the related product documentation first and then read this chapter for the complete understanding.

Chapter 8 describes various licensing restrictions while using some of the Oracle products, that is, Oracle Warehouse Builder, Oracle Data Integrator, and Oracle Golden Gate. Any out of the box functionality is covered under the license agreement described as part of this product licensing. However, if you want to create any additional functionality using these products, please make sure it is covered under the licensing agreement, and does not require any additional licenses.

Related Documents

Refer to the following documents for complete information:

- Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide
- Oracle Business Intelligence Enterprise Edition Documentation
- Oracle Golden Gate documentation
- Oracle Data Integrator documentation
- Oracle Warehouse Builder documentation
- Oracle Weblogic server documentation

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1

What's New in Oracle Utilities Advanced Spatial and Operational Analytics version 2.4.1

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) is a set of star schemas, graphic templates, and data processing programs that allows you to build a Business Intelligence (BI) solution to meet your organization's analytic requirements. Oracle Utilities Advanced Spatial and Operational Analytics used Oracle Warehouse Builder (OWB) along with the source application programs to Extract-Transform-Load (ETL) data from the source application to the Data Warehouse until release v2.4.0. With this new release of Oracle Utilities Advanced Spatial and Operational Analytics v2.4.1, the product has started the process of migrating from Oracle Warehouse Builder (OWB) based ETL to Oracle Data Integrator (ODI) based ELT. This migration is planned in a phased manner, spanning across several releases over a period of time. During this migration cycle, ETL for some of the source applications will continue to use the Oracle Warehouse Builder, while others will use the Oracle Data Integrator.

Oracle Utilities Advanced Spatial and Operational Analytics version 2.4.1 introduces new products supporting Oracle Utilities Operational Device Management (ODM) analytics. This is the first of its kind product, which uses the Oracle Data Integrator (ODI) ETL methodology.

The following two products have been added to Oracle Utilities Advanced Spatial and Operational Analytics set of products in this release:

- Oracle Utilities Operational Device Management Extractors & Schema
- Oracle Utilities Operational Device Analytics

Chapter 2

Overview of Data Warehouse

Data Warehouse is a database primarily used for reporting and analysis purpose. It is a central repository for the current as well as historical data from various operational applications. Data from transactional or operational applications is extracted, transformed, and loaded into the star schema in data warehouses.

Various components of a typical data warehouse environment comprise of the following:

- Source application from where data is transferred to the data warehouse
- Extract, Transform, and Load (ETL) tools to move data from the source application to the data warehouse
- Data storage architecture in the data warehouse, such as the star schema
- Tools to query and present data to the user

Source Application

The day to day transaction of the system is captured in the source application. The data in the source application is stored in a way that allows for fast reads and updates. Queries are not expected to be in bulk form, but typically arranged, row by row. To achieve this, the data is stored in a normalized form in the source application. Generally, the source system does not store historical data.

Staging Area

Data warehouse stores data from various operational sources. The staging area is a database that stores the raw data extracted from the source application. The replication area is similar to the source application in structure. This area can be used for cleansing the data, resolving domain conflict, merging or combining data from multiple source applications, data duplication, etc.

Data Presentation Area

Once data is cleansed in the staging area, it is ready to be moved to presentation area. The data is organized in such a way that it is easily available for the presentation. The data is stored in dimensional model, typically in the star schema.

ETL/ELT

The above process of identifying data to be transferred from the source to the data presentation area, extracting the data, transforming and loading into the data warehouse is called Extraction,

Transform and Load (ETL). There are many tools available in the industry. Based on the complexity of the ETL, you need to choose the tools or write the ETL programs.

Chapter 3

Overview of Oracle Utilities Advanced Spatial and Operational Analytics

What is OUASA

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) consists of pre-built analytics applications as well as a collection of Extractors and Schema products. Oracle Utilities Advanced Spatial and Operational Analytics products help customers extract, transform, load, and analyze data generated in Oracle Utilities Applications such as Oracle Utilities Customer Care and Billing (CC&B), Oracle Utilities Meter Data Management (MDM), Oracle Utilities Network Management System (NMS), Oracle Utilities Work & Asset Management (WAM), Oracle Utilities Mobile Workforce Management (MWM), and Oracle Utilities Operational Device Management (ODM).

Built on world class Oracle Business Intelligence Enterprise Edition (OBIEE) 11g platform with integrated spatial features, Oracle Utilities Advanced Spatial and Operational Analytics supports end to end analytic workflows including the ability to drill back to the source applications. The extractors and schema products are designed with pre-built mapping between source and target, and supports schema extendibility with built in user-defined fields, dimensions and measures.

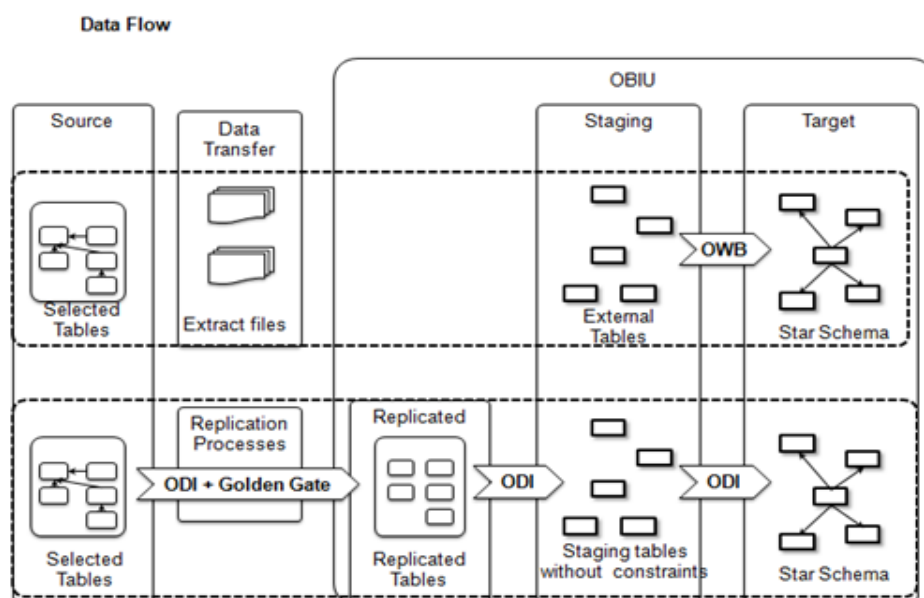
The Oracle Utilities Advanced Spatial and Operational Analytics data warehouse is a separate database from the operational database. Data Warehousing involves large volumes of data used primarily for analysis. The data warehouse has the following features:

- Data structures are easily accessible by end users for their reporting needs.
- Large volumes of data can be retrieved quickly. This in turn allows for fast rendering of graphics that showcase the Key Performance Indicators (KPIs).
- Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) contains star schemas and graphics suited for data retrieved from various Oracle Utilities Edge Applications.
- The Oracle Business Intelligence (BI) application also provides you with the ability to add additional star schemas and graphics as per your requirement using the required development tools.
- Oracle Utilities Advanced Spatial and Operational Analytics can be divided into two broad components, namely, the Extract, Transform, and Load (ETL) process and the Analytics. Oracle Utilities Advanced Spatial and Operational Analytics uses OBIEE as its Analytics tool. Up until Oracle Utilities Advanced Spatial and Operational Analytics version 2.4.0 release, the ETL was done using Oracle Warehouse Builder. This new Release of 2.4.1 has introduced ODI based ELT for Operational Device Analytics. In upcoming future releases of Oracle Utilities Advanced Spatial and Operational Analytics, most of the existing ETL will be moved to ODI in a phased manner.
- In release 2.4.1, the following products use the Oracle Warehouse Builder based ETL:

- Oracle Utilities Customer Care and Billing Analytics
- Oracle Utilities Work and Assets Analytics
- Oracle Utilities Outage Analytics
- Oracle Utilities Meter Data Analytics
- Oracle Utilities Mobile Workforce Analytics
- The following product uses the Oracle Data Integrator based on ELT:
 - Oracle Utilities Operational Device Analytics

Overview of ETL

The architecture for Oracle Warehouse Builder based ETL and Oracle Data Integrator based ELT is different. The diagram below illustrates the overall data processing pipeline using two different methodologies.



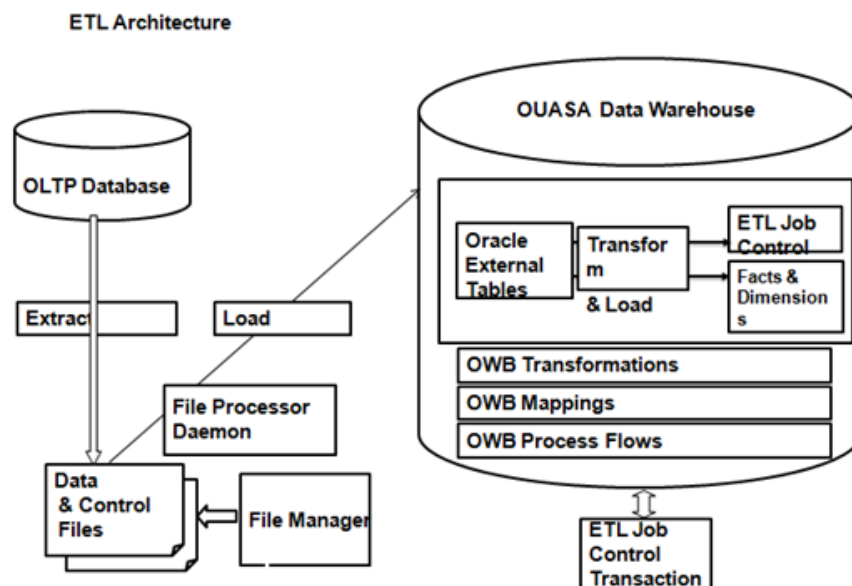
The following sections give an overview of both the architectures.

Oracle Warehouse Builder Based ETL

Oracle Utilities Advanced Spatial and Operational Analytics uses Oracle Warehouse Builder to store the following items:

- Table designs of the star schemas.
- Data mappings used to generate batch jobs that perform the extract, transform, and load operations.
- Process flows that validate the extracted information, load the star schemas, and perform exception handling.

The OWB based methodology uses a three phase approach, where the source system extractors extract data into flat files, these files are transferred to the target system and from there the OWB process loads the data into the target start schema.



Extract

Extraction process executes on the source system and performs the transformations to convert data into a format that is accepted by the target data warehouse.

Extractors are provided for the Oracle Utilities suite of products. The extract programs execute in the operational database as they are extracting operational data. Oracle Utilities Advanced Spatial and Operational Analytics uses flat files as the source to load data into the data warehouse. The flat files are generated through an extraction process in the edge applications. Every fact and dimension in the data warehouse schema has a corresponding extract batch process. These batch processes extract data from the source system and transfer it to flat files. Along with each data flat file containing the extracted data, a single-record control file containing the batch information about the extractor program is also generated. The data and the control flat files, in turn, are loaded into the Oracle Utilities Business Intelligence data warehouse.

Note: Extractor programs are packaged with edge applications.

Transfer

Data files generated on the source can be copied over to the target using FTP/SFTP/SCP or utilize a shared file location.

Load

External tables in the target database point to these files and the Oracle Warehouse Builder process loads the target data warehouse. Data validation and reference key transformations are performed by the load process.

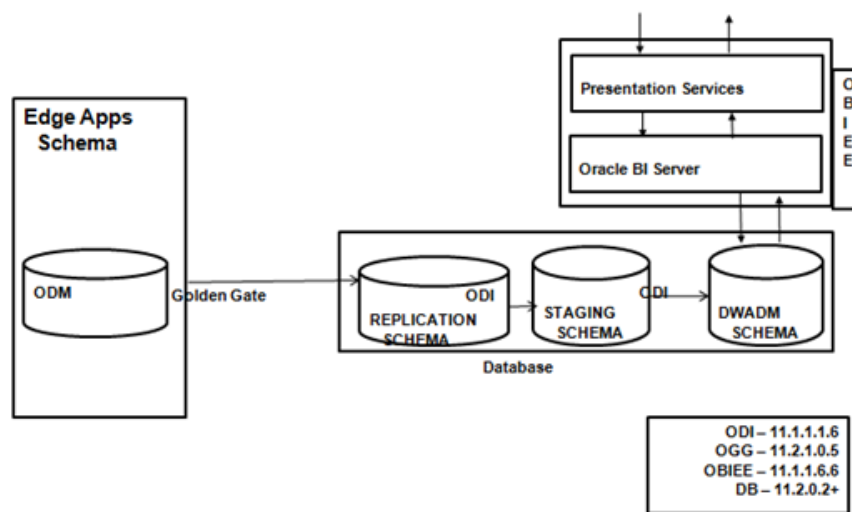
The flat files produced by the extract programs serve as input to the load programs. The load programs use this data to populate the star schemas in the data warehouse.

Refer to **Appendix L** for complete information on Oracle Warehouse Builder ETL and its components.

Oracle Data Integrator ELT

Oracle Data Integrator (ODI) is a comprehensive data integration platform that covers all data integration requirements ranging from high-volume, high-performance batch loads to event-

driven, trickle-feed integration processes to Service-Oriented Architecture (SOA) enabled data services.



The new methodology utilizes three key technologies from Oracle:

- Oracle Data Integrator
- Oracle Golden Gate
- Oracle Weblogic

Replication

Oracle Golden Gate is utilized to replicate selected tables from source system into a replicated schema. Each source system instance has a dedicated schema associated for its replicated data. The replicated tables are similar in structure to the source tables with a few additional columns added for tracking history and audit.

The Golden Gate processes capture any changes on the source tables and transfer them to the replicated tables. The changes are available in the replication layer within few seconds, and the data can be used to load the target data warehouse. Also, later on, if for any reason, you want to reset and reload data in the data warehouse, the data in replication layer is available, provided they were not purged.

Staging

The staging tables are structurally similar to the target table and contain a few additional columns for data transformations. There are no constraints on the staging tables. Foreign key mapping, data validations, and other transformations are performed in the staging area. Data retention in staging area is controlled by configuration stored in the metadata.

Target

The final target is the data warehouse entities, namely, Dimensions and Facts.

For additional information, refer to the Oracle Utilities Advanced Spatial and Operational Analytics Data Mapping guides for the respective source application to star schema mapping.

Data Cleansing

The current release of OUASA v2.4.1 paves the way to enable custom data cleansing routines to be implemented. To support this feature a column "**cleansed_flg**" is provided on the replicated tables. The value is set to '**Y**' as default. Future release versions of the Oracle Utilities Advanced

Spatial and Operational Analytics product will provide a framework to utilize the cleansing feature to implement custom cleansing rules. At present, the data load process does not check this flag while loading the data.

Data Lineage

When data is loaded from the source systems into the data warehouse using the Oracle Data Integrator (ODI), it is possible to use the Oracle Data Integrator Lineage for Oracle Business Intelligence feature to consolidate ODI metadata with OBIEE and expose this metadata in a report-to-source data lineage dashboard in OBIEE.

Refer to the document Oracle Data Integrator 11g 11.1.1.6 New Features > Overview section *Oracle Business Intelligence Enterprise Edition Data Lineage*.

Materialized Views

In general, most facts have high volume of data and multiple dimensions for slicing and dicing data. This can potentially lead to performance issues in summarized dashboards. To mitigate risks associated with performance, Oracle Utilities Advanced Spatial and Operational Analytics utilizes the Oracle Materialized views to pre-aggregate the data.

Overview of Analytics

Oracle Utilities Advanced Spatial Analytics (OUASA) comes with a rich set of pre-built analytics created using Oracle Business Intelligence Enterprise Edition (OBIEE). These analytics offer a detailed insight into vital Key Performance Indicators (KPI) for the various source applications.

These analytics are built on top of various star schemas that are delivered in the data warehouse for different source application. For better performance, these analytics have been optimized to retrieve data from specially designed materialized views in the data warehouse. These materialized views are highly optimized to summarize data efficiently and make data retrieval much faster.

The pre-built analytics are created for each source application and grouped as separate catalog files as per their application. These product based catalogs are licensed individually.

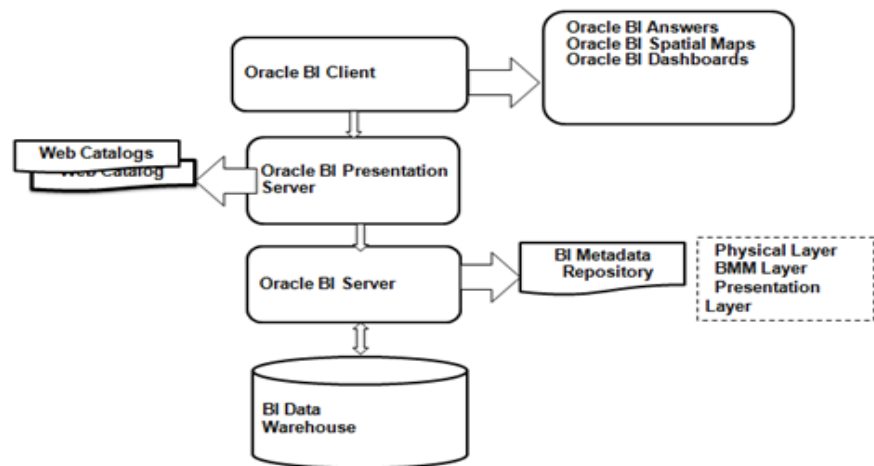
Using the OBIEE tool, a metadata repository is first built using the star schemas available in the BI data warehouse. This repository file consists of three distinct layers

- **Physical Layer:** Physical table representation
- **Business Layer:** Contains the business logic for the fields
- **Presentation Layer:** Tables grouped as subject areas for fact based reporting

Once the repository file is created and deployed on the Oracle Business Intelligence (BI) server, dashboards are created based on subject areas from the presentation layer. These dashboards are saved in various web catalog files.

Below is a high level architecture diagram for the OBIEE components.

OBIEE Architecture



When the Oracle Utilities Advanced Spatial and Operational Analytics product is shipped for the analytics portion, the metadata repository and the source application specific catalog files will be delivered.

For details on Oracle Business Intelligence Enterprise Edition, refer to Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition.

Chapter 4

Configuring Oracle Utilities Advanced Spatial and Operational Analytics

Before you start using the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) product, it needs to be configured.

Follow the below-mentioned sections to configure the Source System, ETL, and Analytics. It is extremely critical that you configure your system correctly for the product to behave as expected.

- **Configuring the Source System**
- **Configuring ETL/ELT**
- **Configuring Analytics**
- **Performance Recommendations**

Configuring the Source System

For each dimension and fact in the data warehouse, there is a batch program in the source system that extracts and transforms the required data based on the type of extract and parameters passed to the batch program. These batch programs extract data and place them in the flat files. Make sure that these batch programs are configured on the source system. A list of all the batch programs for the target entities is provided in Appendices provided with this guide (starting from **Appendix C** to **Appendix G**).

Other than the batch configuration, there may be additional configuration required on the source application. For details on how to setup each of the source applications for data extraction, refer to the Configuration chapter in the *Oracle Utilities Advanced Spatial and Operational Analytics Data Mapping Guides* for the respective source application.

Configuring ETL/ELT

Depending on which product you use, you may need to configure Oracle Warehouse Builder (OWB) based ETL or Oracle Data Integrator (ODI) based ELT.

This section describes the following:

- **Oracle Warehouse Builder Based ETL**
- **Oracle Data Integrator Based ELT**

Oracle Warehouse Builder Based ETL

Once the extractor programs have generated the flat files for facts and dimensions, Oracle Warehouse Builder (OWB) jobs pick these files and load the data into the data warehouse. This section describes various processes that are useful for scheduling the load jobs, capturing errors, monitoring jobs, and resubmitting the failed jobs.

This section describes the following:

- **Scheduling Jobs Using the File Processor Daemon**
- **Log File**
- **Monitoring Jobs**
- **Handling Load Errors**
- **Capturing Fact Load Errors**
- **Fixing Load Errors**
- **Resubmitting a Failed Job**
- **Custom Error Handling**
- **Multiple Data Source Indicator Configuration**

Scheduling Jobs Using the File Processor Daemon

The File Processor Daemon is a simple java based utility that mimics the capabilities of a job scheduler. This persistent process runs continuously in the background and periodically monitors the extract folder. When new data files arrive, it processes them and triggers the appropriate OWB process flows for loading the data.

It has the intelligence to determine the fact dimension dependency. When a fact data file arrives in the extract file directory, it scans the extract directory to see if there are any data files present for any of the dimensions associated with this particular fact. If so, the loading of this particular fact file is skipped to let the dimension data load first into the data warehouse.

The File Processor Daemon also scans the error folder to verify whether any of the dimension load jobs have failed. If there are failures, all the related fact data files are skipped from processing until the related dimensions are loaded successfully. Fact dimension dependency is determined through the database constraints table present in the data warehouse.

Mappings in the parameter file control how the file processor daemon can determine which OWB process to trigger for which data file and which table name to check while querying the constraints table for fact/dimension dependencies. For more details refer to the detailed description of the parameter `extract.file.mappingN` mentioned below.

The installation of the standard setup for the File Processor Daemon is documented in *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide*. All standard process flows are configured to run with the base installation. For information on how to install and run the File Processor Daemon, refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide*.

The File Processor Daemon reads a parameter file, `SchedulerParm.properties` existing in the directory where the File Processor Daemon is installed. The released version of this properties file includes entries for all standard process flows. Hence, if any of the base extract files are present in the extract load directory, they will be processed automatically. No extract configuration needs to be done if only base extracts are being implemented.

The following required parameter entries are present in the delivered `SchedulerParm.properties` file and can be modified if needed for an implementation by using the `(configureEnv.sh/cmd)`

command as documented in the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide*.

Parameter Name	Description
execution.switch	Determines if the file processor daemon is active or inactive. As long as this parameter is set to 1, the file processor daemon will continue to run. To stop the file processor daemon without killing the process, modify this file, and set the execution.switch parameter to 0.
scheduler.poll.duration	This parameter determines whether the File Processor Daemon is active or not. As long as this parameter is set to 1, the File Processor Daemon will continue to run. To stop the File Processor Daemon, modify this file, and set the execution.switch parameter to 0.
extract.dir.path	This parameter tells the File Processor Daemon where to look for new extract files. This path must match the path that the Oracle Warehouse Builder process flows have been configured to check for extract files.
extract.max.load	This parameter tells the File Processor Daemon how many extract files to load at a single time. If there are more files in the extract directory than the number specified by this parameter, the first set of files found are loaded with the current run of the File Processor Daemon. This parameter can be modified based on the size of the machine/system and on the number of files that can be handled at once by the Oracle Warehouse Builder.
extract.file.mapping.count	This parameter tells the File Processor Daemon how many process flows are listed in the parameter file. This count must match the largest extract.file.mapping(N) present in the parameter file. If this count is set to a lower number than the number of mappings, then only the set number of mappings are processed by the File Processor Daemon.

Parameter Name	Description
extract.file.mappingN	<p>N is a number between 1 and the extract.file.mapping.count parameter, or extract.file.mapping.override.count parameter if this is specified. These parameters tell the File Processor Daemon which extract files to look for, which process flow to run when an extract file is found, and the actual table name in the data warehouse. The format of this parameter is: Extract File Name, Process Flow Name, Table Name. The extract file name should be just the base name without the data source indicator, batch number, thread number values and without the .DAT or .CTL extensions.</p> <p>For example, this entry will look for the Account Extract Files and run the SPLWF_D_ACCT process flow when a new account extract file is found: extract.file.mapping1= D_ACCT_EXT,SPLWF_D_ACCT,CD_ACCT</p> <p>The table name is used for determining the dependency between the facts and dimensions by query the database constraints table. This dependency check is required to avoid processing of any fact data files, if any of its dimension data files are required to be loaded first. It is important when entering values that the B numbers increase sequentially and that no numbers are skipped. The largest Bvalue should match the mapping.count or mapping.override.count parameter specified in the parameter file.</p>

Note: The following below-listed optional parameter is not present in the delivered SchedulerParm.properties file but can be added if needed:

Parameter Name	Description
extract.file.mapping.override.count	<p>This parameter provides a way to override the mapping count specified in the extract.file.mapping.count parameter. The default parameter file does not include this parameter. If this parameter is specified, then the extract.file.mapping.count value is not used by the file processor daemon.</p>

The important fields in the parameter file to look at when implementing new loads are the mapping.count and the mappingN parameters. New loads can be added to the parameter file if they are implemented by a project. It is recommended that the new mapping **N** values be entered at the end of the list; so, that during an upgrade process, it is easy to identify the added records when updating the newly released SchedulerParm.properties file.

After updating any of these parameters using the configureEnv program, you must run the initialSetup.sh/cmd script to create the cm_schedulerParm.properties.exit_1.include parameter file. You must place this file in the <INSTALL_DIR>/templates directory, and then restart the file processor daemon. Refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide* for more information on this process.

Configuring Additional Parameters

The following optional parameter is not present in the delivered SchedulerParm.properties file but can be added if needed:

Parameter Name	Description
extract.file.mapping.override.count	This parameter provides a way to override the mapping count specified in the extract.file.mapping.count parameter. The default parameter file does not include this parameter. If this parameter is specified, then the extract.file.mapping.count value is not used by the file processor daemon.

The important fields in the parameter file to look at when implementing new loads are the **mapping.count** and the **mappingN** parameters. New loads can be added to the parameter file if they are implemented by a project. It is recommended that the new **mappingN** values be entered at the end of the list, so that during an upgrade process, it will be easy to identify those added records when updating the newly released SchedulerParm.properties file.

After updating any of these parameters using the configureEnv program, you must run the initialSetup.sh/cmd script to create the cm_schedulerParm.properties.exit_1.include parameter file. You must place this file in the <INSTALL_DIR>/templates directory, and then restart the file processor daemon. Refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide* for complete information.

Log File

The instructions on how to start the File Processor Daemon are explained in *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide*. Once started, a log file called **FileProcessorDaemon.log** is written by the File Processor Daemon. Messages from the File Processor Daemon are written here for various normal activities. Error messages are also written to this file. If Oracle Warehouse Builder process flows are not triggered properly, review this log file to possibly identify the cause of the problem.

The log file is deleted once it reaches a size of 100 megabytes, and a new file is started. Hence, if older errors are not showing up, then it is possible that the file may have been recently purged by the File Processor Daemon.

Monitoring Jobs

Note: You must have full license of the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) to use this feature.

The process flows that are run by the file processor daemon are set up in two different ways, depending on whether a fact or a dimension extract file is being loaded. For a dimension load, only the dimension file is loaded into the dimension table of the data warehouse. However, when a fact file is loaded, the fact table is updated, and then, any associated materialized views are refreshed.

The load jobs are visible in the ETL Job Control Administration Portal. The Job Complete (JC) status indicates that the data file loaded successfully. If a fact is loaded, the Job Complete Status does not indicate that the materialized views were successfully updated. An e-mail is sent to the data warehouse administrator if the materialized view refresh fails.

The following attributes are available for ETL Job Control process:

- **Job No.:** Indicates Unique number identifying the load process. A link is presented on the value if the job status is 'ERROR'

- **Batch Code:** Unique code identifying the extraction routine used to generate the extract files on the source system.
- **Description:** A descriptive name for the extract process.
- **Batch No.:** The batch number indicates the batch in which the extract was run.
- **Batch Thread:** A batch can generate multiple files. Each file is identifiable by a thread number.
- **Start Date/Time:** Indicates the start time of the load process.
- **End Date/Time:** Indicates the End time of the load process.
- **ETL Job Status:** The current status of the job.
- **Data Source Indicator:** Unique identifier for each source system.
- **Load Audit ID:** A reference to the Oracle Warehouse Builder (OWB) audit log tables.
- **ETL Map:** The process name that contains the code to load the data into the target.
- **Extract Record Count:** The number of records in the extract file.
- **Load Record Count:** The number of records loaded into target.
- **Load Error Count:** The number of records that failed.
- **Load Error Message:** Indicates the summary error message identifying the cause of the job failure.

In addition, the load jobs and any associated errors can be viewed in the Oracle Warehouse Builder Control Center. In general, if an e-mail message indicating a load failed is not received, then it indicates that the load of a data file (and any materialized view refresh) was successful.

Handling Load Errors

While loading data from flat files to target schema using OWB, there are various reasons while load fails. This section describes attributes to check when trying to find out why an extract does not load or why an error is generated during a load. For details regarding resolving OWB load problems, refer to the Oracle Warehouse Builder User's Guide. You should turn off file processor daemon before debugging.

- **Job Status:** Make sure that jobs are not in an In Progress (IP) or Error (ER) state. The status of a job is stored in the JOB_STATUS_FLG field in the B1_ETL_JOB_CTRL table, and you can view the status on the ETL Job Control Administration Portal. If the data in error has Job Control Portal as well.
- **OWB Errors:** There are two views that you can query to see errors from a process flow: ALL_RT_AUDIT_EXEC_MESSAGES and BIREPOWN.WB_RT_ERRORS. These errors should be present in the e-mail messages sent when a mapping fails. However, you can run the following SQL statements to view the errors stored from the last four hours, if the e-mail messages are lost or do not contain any error messages:

```
begin
owbsys.wb_workspace_management.set_workspace('SPLBIREP','BIREPOWN');
end;
-- where workspace name is SPLBIREP and workspace owner is
BIREPOWN;
-- replace if necessary.
To find the Workspace Name and Owner, run:
select * from owbsys.WORKSPACE_ASSIGNMENT;
select to_char( created_on, 'dd-mon-yyyy hh24:mi:ss - ' ) ||
message_text
from all_rt_audit_exec_messages
where created_on > sysdate - .2
order by message_audit_id;
```

- **Error RPE-02248:** If you get this error, then change the Runtime.properties file in the \$ORACLE_HOME/owb/bin/admin directory:

```
-- Change these settings from DISABLED to NATIVE_JAVA
property.RuntimePlatform.0.NativeExecution.FTP.security_constraint
= NATIVE_JAVA
```

```
property.RuntimePlatform.0.NativeExecution.Shell.security_constraint
= NATIVE_JAVA
```

```
property.RuntimePlatform.0.NativeExecution.SQLPlus.security_constraint
= NATIVE_JAVA
```

- **Viewing Data in Oracle:** Sometimes it helps to view the data in an extract file from Oracle. This can be done by copying the Extract file to the Staging file used during the load. This file has the same name as the Extract file, without the numbers in the file name. For example, if the Account extract generates these two extract files, D_ACCT_EXT000004000000001001.CTL and D_ACCT_EXT000004000000001001.DAT, then the staging files are named D_ACCT_EXT.CTL and D_ACCT_EXT.DAT.
- After copying the files, the data can be viewed in Oracle using the following two Staging Tables: STG_ACCT_CTL_EXT and STG_ACCT_EXT.
- **Strange Characters in Data Files:** If you view the data in Oracle and there are strange characters when you run a query, then the character set may be specified incorrectly for the external file. The character set can be changed by running the EditFFCS.TCL file in OWB. You can see the character set for a specific external file by running the following query:

```
select * from dba_external_tables
where table_name = 'STG_ACCT_CTL_EXT';
```

Note: For viewing the character set for a specific external file, specify the appropriate external table instead of the one mentioned in the example given above.

Capturing Fact Load Errors

During the fact load, users may want to capture various exception scenarios for which data should not be loaded. The load routine captures the exceptions and displays those to the users. Users can then correct the exceptions and reload the correct data.

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) provides out of the box validation for checking the referential integrity between the facts and corresponding required dimensions. During a fact load, if a required dimension record is not found in the dimension table, the fact record is not loaded, and an error notification is sent to the user indicating this.

Apart from this out of the box validation, OUASA also allows users to add additional validations to capture their business requirements.

In general, this process of capturing Fact load errors is divided into two steps:

1. Validation:

- A validation function is invoked to execute the validation check. Oracle Utilities Advanced Spatial and Operational Analytics out of the box validation validates the number of records loaded into the target file by comparing it with the number of records specified in the control file. Users can write their own validation check if and when required.
- In case, if validation fails, (as in case of out of the box validation, if number of records loaded into the target file is less than the number of records specified in the control file), the load is marked as an invalid load and a process to identify such records is executed.

- ### 2. Error Identification:
- This process executes certain queries to identify the records that failed to load and also the reason for the failure (missing reference records in a required dimension). All such records are inserted into a table B1_ETL_DATA_ERR. If users want to write a

custom check, they should write the logic to identify such records and make an entry into the B1_ETL_DATA_ERR table.

The following screenshot displays user interface records that failed to load:

Select	Job No.	Batch Code	Description	Batch No.	Batch Thread	Start Date / Time	End Date / Time	ETL Job Status	Data Source Indicator	Load Audit Id	ETL Map	Extract Record Count	Load Record Count	Load Error Count	Load Error Message
<input type="checkbox"/>	206	EXTCP	WRT Customer Outage Fact (SUPPLY_LOAD_LOAD)	1		1/10/2012 2:52:13 AM	7/10/2012 2:55:15 AM	ERROR		4	558 SRMAP_F_CUST_RECENT_OUTG	18554	0	0	ORA-20011: INVALID_LOAD: Invalid number of records loaded into CF_CUST_RECENT_OUTG. 0 Inserted, 18552 Merged and 0 Updated when 18554 changes expected. Transaction rolled back.
<input type="checkbox"/>	216	EXTFED	OWS Feeder Load Extraction (EDM_FEEDER_LOAD_INF)	1		1/11/2012 12:08:10 AM	7/11/2012 12:08:13 AM	ERROR		4	680 SRMAP_F_FEEDER_LOAD_LOAD	20	0	0	ORA-20011: INVALID_LOAD: Invalid number of records loaded into CF_FEEDER_LOAD_LOAD. 0 Inserted, 0 Merged and 0 Updated when 20 changes expected. Transaction rolled back.
<input type="checkbox"/>	216	EXTFVIRT	Extract Work Order Task	0		1/11/2012 3:05:11 AM	7/11/2012 3:05:11 AM	ERROR		3	715 SRMAP_F_WORKORD_TK	4	0	0	ORA-30726: unable to get a table set of rows in the source tables
<input type="checkbox"/>	216	EXTVNC	OWS Call Transaction Fact (INCIDENTS)	1		1/11/2012 3:02:11 AM	7/11/2012 3:02:13 AM	ERROR		4	717 SRMAP_F_BST_CALL	2096	0	0	ORA-00001: unique constraint (EDWADM.VP4005) violated

In the above screenshot, the jobs have been filtered by the status = 'ERROR'. The report shows the number of rows in the data file and the number of rows loaded. Click on the **Job No.** field to navigate to that particular error details page. It shows error details recorded by the Oracle Warehouse Builder (OWB) as shown in the screenshot below:

Sequence Number	Fact Natural Key	Error Description	Dimension Table	Dimension Natural Key	Update Date/Time	Data Source Indicator	Validation Procedure Name
4734798	SRC_ORDER_ID=073602468886	Foreign Key Validation Error	CD_PER	SRC_PER_ID=2270059960	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734799	SRC_ORDER_ID=010729467647	Foreign Key Validation Error	CD_ORDER_STATUS	ORDER_STATUS_CD=40	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734800	SRC_ORDER_ID=010729467647	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-B	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734801	SRC_ORDER_ID=029629236090	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-A	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734802	SRC_ORDER_ID=073602468886	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-A	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734803	SRC_ORDER_ID=074673453728	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-A	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734804	SRC_ORDER_ID=081903783811	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-B	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734805	SRC_ORDER_ID=127894876175	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-A	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734806	SRC_ORDER_ID=218333629927	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-A	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734807	SRC_ORDER_ID=227072417307	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-A	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER
4734808	SRC_ORDER_ID=315237500330	Foreign Key Validation Error	CD_CAMPAGN	CAMPAGN_CD=CAMP-A	12/15/2011 12:00:00 AM	170593	B1_ERR_F_ORDER

The figure above shows individual record level details.

The report provides the following details:

- **Fact Natural Key:** Indicates the unique key combination for the staging table.
- **The Error type description:** B1FK indicates a foreign key reference issue.
- **Dimension Table:** The dimension table for which a reference could not be resolved.
- **Dimension Natural Key:** The natural key columns of the dimension which are used in to fetch the reference key.
- **Update Date/Time:** The Update date time in the extract file.
- **Data Source Indicator:** The Data source indicator in the extract file.

- **Validation Procedure Name:** The procedure name that was used to identify these issues.

In the example, there are missing references to the following dimensions:

- CD_MSRMT_COND
- CD_CONS_TYPE

Fixing Load Errors

This section describes ways to determine why records fail to load. If a data file fails to load, it is moved to the error directory just below the load directory. The load directory can be accessed from the database server.

You can determine the directory path using the following query from the DWADM account, and with the Control Table Name that is included in the failed record:

```
SELECT directory_path
FROM user_external_tables a, all_directories b
WHERE a.table_name = UPPER( '&Control_Table_Name' ) AND
b.directory_name = a.default_directory_name AND
b.owner = a.default_directory_owner;
```

For example, if the load directory is /spl/BIDevelopment/bi221prf, the error file will be located in /spl/BIDevelopment/bi221prf/error.

The easiest way to examine the data file that has errors is to copy it into the Load directory, so that Oracle queries can be run against it. The extract file replaces the .DAT file in that directory with the File name. For example, the /spl/BIDevelopment/bi221prf/error/D_ACCT_EXT01793300000001001.DAT file would replace the /spl/BIDevelopment/bi221prf/D_ACCT_EXT.DAT file. The .CTL file in the processed directory can also replace the .CTL file in the load directory.

Once you find the missing records, you can then determine if a required dimension key is missing from the dimension table. Note that it is possible that a later dimension load could have added the dimension records after the fact table was loaded, so if no records are returned by any of these queries, then reloading the fact records may solve the problem.

There are several different ways to fix the data so that it can be loaded. The method you use to successfully load the data depends on how the data is fixed:

- Make sure that the dimension files were loaded successfully. If there were errors during the dimension load or the number of records loaded did not match the number of records in the file, then you may need to figure out why the dimension records did not load. Fix these and reload the dimension files, before you can start working on the reloading fact records.
- Fix a fact data problem in the source system. This allows the data to be re-extracted the next time the fact data is extracted. However, some fact extracts only extract new records, not changed records; so, this may not always fix the data in the data warehouse.
- Modify the fact records to have valid dimension keys. If the fact records are modified, then they must be added to a new extract file to get reloaded. A manual extract can be done to create a new extract file, and then, the modified records can be loaded to this file. The modified records are then loaded when the next load process is run.
- Change effective start and end dates on the dimension records. If a fact record does not load when the update time on the fact record is not within the range of the effective start and end date on the dimension, the effective start or end date on the dimension record can be changed manually to allow the load to occur. If this is changed, then the fact records should be added to a new extract file as described in the preceding section.

Resubmitting a Failed Job

Note: You must have the full license of the Oracle Utilities Advanced Spatial and Operational Analytics Application to use this feature.

Resubmitting a Failed Job

Follow these steps to resubmit a failed job:

1. View the **ETL Job Control** screen under the **Admin** menu of Dashboards.

Business Intelligence

Search

Advanced

Administration

Help

ETL Job Control

Home

Catalog

Dashboards

New

Open

Signed

ETL Job Control

Please select jobs in ERROR and click on Update to Re-Submit

Select	Job No.	Batch Code	Description	Batch No.	Batch Thread	Start Date / Time	End Date / Time	ETL Job Status	Data Source Indicator	Load Audit Id	ETL Map	Extract Record Count	Load Record Count	Load Error Count	Load Error Message	
<input type="checkbox"/>	2006	NRTC0F	NRT Customer Outage Fact (SUPPLY_NODE_LOG)	1	1	7/10/2012 2:52:13 AM	7/10/2012 2:55:15 AM	RE-INITIALIZE		4	558	SPUMAP_F_CUST_RECENT_OUTG	19854	0	1946	ORA-20001: INVALID LO number of re into CF_CUST_RE 0 Inserted, : Merged and when 19854 expected. Tr Rolled back.
<input type="checkbox"/>	2009	EXT0RIVA	OMS Crew Activity Extraction (CREW_EVENT_HISTORY)	1	1	7/10/2012 2:53:11 AM	7/10/2012 2:53:13 AM	JOB COMPLETE		4	561	SPUMAP_F_RST_CREW	40	40	0	
<input type="checkbox"/>	2010	EXTJ0BT	OMS Job Transaction Extraction (JOBS)	1	1	7/10/2012 2:54:10 AM	7/10/2012 2:54:18 AM	JOB COMPLETE		4	562	SPUMAP_F_RST_JOB	1436	1436	0	

Rows 1 - 25

Update

Revert

Refresh

Print

Export

2. Select the Job that needs to be restarted. This will change the ETL Job Status on the screen to **RE_INITIALIZE**.
3. Move the data and control files that errored out from the error folder. This is very important as the file processor does not process the files that are in the error folder.
4. Place the corrected files in the data folder and switch on the File Processor Daemon. The corresponding process flow is triggered during the next polling of the file.

Custom Error Handling

Note that the out of the box validation is always going to be executed during every Fact load. In addition to this out of the box validation, a user exit hook has been provided to allow customers to write their own Customer Modification (CM) validation functions and error identification procedures. The out-of-box fact load procedure automatically calls these functions/procedures if these objects exist in the database.

The list of all the existing Validation functions and Error Identification Procedures is provided in **Appendix G**. It also lists the custom function and procedure names to be used when extending the data load validation.

Note: The user must use the exact name provided in the **Appendix G** for their custom validations to be called while loading the fact.

In general, it is advisable to follow closely the out of the box routine for the custom code. The following guidelines have to be followed when creating custom validation functions:

- The validation function name should be derived from the built in validation function by replacing 'B1' with 'CM' in the name.

Refer to **Appendix H** for a complete list of validation functions for all facts. For example for

the fact CF_CC, the name of the out of the box validation function is B1_VAL_F_CC, then the custom validation function name should be CM_VAL_F_CC.

- The custom validation function should have same input parameters as those for the custom validation function. Follow the below template to create customer validation function:

```
CREATE OR REPLACE
FUNCTION << Custom Validation Function Name >> ( p_num_inserts IN
NUMBER, p_num_merged IN NUMBER, p_num_updates IN NUMBER)
RETURN NUMBER
AS
lv_ret_value NUMBER;
BEGIN
/* Your Logic Here */
EXCEPTION
WHEN OTHERS THEN
lv_ret_value := 1 ;
RETURN lv_ret_value;
END B1_VAL_F_ARREARS;
/*
```

The input parameters to the custom Validation function are explained as below:

- p_num_inserts:** Indicates the actual number of records inserted into target fact table by the mapping
- p_num_merged:** Indicates the actual number of records merged into target fact table by the mapping
- p_num_updates:** Indicates the actual number of records updated into target fact table by the mapping

*/

- The function should return **1** in case the validation fails and the load needs to be cancelled and **0** otherwise
- The following guidelines have to be followed when creating custom error identification procedure. Always use the built in procedure as a starting point:
- The procedure name should be derived from the built in procedure by replacing 'B1' with 'CM' in the name.

Refer to **Appendix G** for complete list of error identification procedures.

For example for the fact CF_CC, the name of the out of the box Error Identification Procedure is B1_ERR_F_CC, then the custom Error Identification Procedure name should be CM_ERR_F_CC.

- The custom procedures should have the same input parameters as those of the out of the box procedures.
- Follow the below template to create the customer error identification procedure:

```
CREATE OR REPLACE
PROCEDURE << Custom Error Handling Procedure Name >> ("IN_MAP_NAME"
IN VARCHAR2, "IN_JOB_NBR" IN NUMBER)
IS
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
/* Your Logic Here */
EXCEPTION
WHEN OTHERS THEN
ROLLBACK;
```

```
END << Custom Error Handling Procedure Name >>;_
/*
```

The input parameters to the custom Error Identification Procedure are explained as below:

- **IN_MAP_NAME:** Indicates the name of the mapping for which the procedure is getting executed for custom error record identification.
 - **IN_JOB_NBR:** Indicates the job number of the particular mapping (as in B1_ETL_JOB_CTRL table.) for which the procedure is getting executed for custom error record identification.
- ```
*/
```
- The procedure must be executed as an autonomous transaction.
  - The insert statement should be appropriately modified as per the custom validation implemented.

## Multiple Data Source Indicator Configuration

Oracle utilities customers can implement multiple utilities application leveraging the integration supported by these systems.

For example, Oracle Utilities Meter Data Management (MDM) & Oracle Utilities Customer Care and Billing (CC&B) joint implementations can leverage the MDM-CCB integration and customers can use Oracle Utilities Customer Care and Billing as the master repository for customer related data (Account, Premise, Person, etc). So the Oracle Utilities Meter Data Management instance does not store any customer related information but refers to the information in CC&B instead. Similarly, the Oracle Utilities Mobile Workforce Management (MWM) can be integrated with Oracle Utilities Meter Data Management and Oracle Utilities Customer Care and Billing, use MDM as the master repository for all meter and measurements related data. In the Oracle Utilities Business Intelligence data warehouse, the data for these objects is loaded from the master repository only. E.g. Account, Person, Premise dimensions are loaded only from CC&B and MDM fact may have a reference to the dimensional data loaded from CC&B.

In this case, while loading the MDM data, the load process should look-up for correct records in the dimension table loaded from CC&B. Hence, there is a need to identify the Data Source Indicator (DSI) of CC&B environment. Data Source indicator is a unique value corresponding to each instance of an edge application feeding data into the data warehouse. The edge applications default the data source indicator from an environment ID. To support this requirement the MDM and MWM extract programs are enhanced to accept additional parameters for CC&B DSI and MDM DSI. These additional DSI values are appended into the control file of extract files. During data load in the warehouse, the load process use CC&B DSI to lookup and join to CC&B specific dimensions and MDM DSI to lookup and join to MDM specific dimensions.

Refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Readme* document for the required MDM and MWM patches to support this feature.

## Multiple Data Source Indicator Changes for Oracle Utilities Meter Data Management Application

This section describes how to configure the Oracle Utilities Customer Care and Billing Data Source Indicator (DSI) on MDM Applications. To configure, follow these steps:

### Configuring Oracle Utilities Customer Care and Billing (CC&B) DSI On Oracle Utilities Meter Data Management (MDM) Application

1. Add a new feature configuration for the feature type '**Business Intelligence Configuration**'.
2. Select the Option Type '**External Data Source Indicator 1**' and enter the CC&B DSI value. The CC&B DSI value can be retrieved by running the below query in the CC&B database:\

```
select ENV_ID from F1_INSTALLATION;
```

During the ETL process, if Oracle Utilities Customer Care and Billing (CC&B) Data Source Indicator is set in the extract Staging Control File, it is used to join with the matching Data Source Indicator in the dimension table to match the dimension key. If the Customer Data Source Indicator has not been extracted in the Staging Control File, then the default data source indicator will be used.

For the MDM source application, CC&B DSI joins with CC&B related dimension keys such as Account Key, Person Key, Premise Key, and Service Agreement (SA) Key for the following Facts:

- CF\_CONSUMPTION
- CF\_DEVICE\_ACTIVITY
- CF\_DEVICE\_EVT
- CF\_INSTALL\_EVT
- CF\_SP
- CF\_SP\_SNAP
- CF\_SP\_UT\_AGE
- CF\_VEE\_EXCP

### Multiple Data Source Indicator Changes for Oracle Utilities Mobile Workforce Management

This section describes how to configure Oracle Utilities Customer Care and Billing Data Source Indicator (DSI) and Oracle Utilities Meter Data Management (MDM) DSI on Oracle Utilities Mobile Workforce Management (MWM) Applications:

#### Configuring DSI on Oracle Mobile Workforce Management (MWM) Applications

1. Add a new feature configuration for the feature type '**Business Intelligence Configuration**'.
2. Select the Option Type '**External Data Source Indicator 1**' and for the Value, enter the CC&B DSI. The CC&B DSI can be retrieved by running the below query in the CC&B database

```
select ENV_ID from F1_INSTALLATION;
```

3. Add one more entry with the Option Type as '**External Data Source Indicator 2**' and for the Value, enter the MDM DSI. The MDM DSI can be retrieved by running the below query in the MDM database

```
select ENV_ID from F1_INSTALLATION;
```

The MWM application can be integrated with CC&B and MDM and therefore two separate data source indicator columns have been added to the MWM Fact Staging Control tables.

Oracle Utilities Customer Care and Billing Data Source Indicator joins with CC&B related dimension keys such as Account Key, Person Key, Premise Key, and Service Agreement (SA) Key. Oracle Utilities Meter Data Management Data Source Indicator joins with MDM related dimension keys such as Contact Key, Meter Device Key, SP Key, and US Key for the following Facts:

- CF\_CREW\_TASK
- CF\_FLD\_ACTIVITY

**Note:** The Multiple Data Source Indicator support is a new enhancement included in Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) release 2.4 Service Pack 4. This feature is not supported in earlier releases of OUASA. In previous releases, it was suggested to make MDM DSI same as CC&B DSI, to achieve the same functionality. After this enhancement, it is not required to match MDM DSI with CC&B DSI. Therefore, the MDM

and MWM DSI values are defaulted from environment ID and are no more configurable. For customers who are upgrading to this enhancement, a special upgrade script is provided. For more details refer to the *Oracle Utilities Advanced and Operational Spatial Analytics Installation Guide*.

## Oracle Data Integrator Based ELT

Oracle Data Integrator (ODI) based ELT uses Oracle Golden Gate and ODI combination to Extract, Load and Transform data from the source database to the target database. Golden Gate captures the changes in the selected tables of the source schema, and transfers these data to the replication schema. Tables in the replication schema are similar in structure to the source tables with a few additional columns added for tracking history and audit. Data retention in replication schema is controlled by configuration stored in the metadata.

Based on the scheduler configuration, data from replication schema is transferred by ODI to the staging schema. Tables in the staging schema are similar to the tables in the target schema. It contains a few additional columns for data transformations. There are no constraints on the staging tables. Foreign key mapping, data validations and other transformations are performed in the staging area. Data retention in staging schema is controlled by configuration stored in the metadata.

From the staging schema, ODI transfers data to the Facts and Dimensions in the target schema. ODI also loads and refreshes the Materialized Views in the target schema.

The required configuration for the Golden Gate and ODI is done during the OUASA installation process. OUASA provides an Administration Tool to maintain these configurations. The following user interfaces are available as part of the Administration Tool for maintaining various configurations.

- Product Instance
- Golden Gate Configuration
- Job Configuration
- Target Entity
- Source Table
- Job Execution (Display Only)

For further information on how to use the Administration Tool, please refer to **Appendix B**.

## Scheduling

In any data warehouse, the basic challenge is to get the data loaded quickly and efficiently whether it is the initial load or the incremental load. Data volumes are high during initial load and during incremental loads the volumes are considerably smaller. The ODI ELT processes utilize a time based slicing mechanism to split the load volumes into more manageable slices to process the initial as well as incremental loads. The metadata configurations allow you to control the size of the slice, parallelism of the loads and much more. To ensure that the data loaded is always consistent, the process executions are governed by a set of rules. The following criteria are considered for Job execution:

- There should be no errors which needs reprocessing.
- The maximum retries limit for the day should not be exceeded or reached.
- Tasks will wait for the configured retry interval before submitting a retry for the job.
- Number of parallel jobs will always be limited to the maximum parallel configured.
- Jobs are not executed beyond the time of the most recent golden gate sync.

- If a job is dependent on tables, which are being synced by separate golden gate processes, then the common sync time of both processes are considered.
- If a scenario does not exist, then the jobs are not executed.
- An interface should be active.
- All dependencies should be run.
- The number of running/error instances of the job should be less than the maximum parallel executions allowed.
- If the instance is configured as run once, then it should not execute once it is successfully executed.
- If a job fails, it should be retried again until the maximum retries per day is reached. The interval between successive retries should be based on configuration.
- The configured scenario should exist.
- The entity should not be disabled.
- The golden gate models comprising of the source tables used in the entity should have been synced up. In case, the sync timestamps vary across multiple models, then the least common sync timestamp is used.
- Snapshot entities are executed on or after the snapshot end period.
- The schedule time can be used to stagger loads and distribute processing. A job is not be executed until the current time crosses the scheduled time.

Refer to **Target Entity** section in **Appendix B** for where to schedule jobs.

All jobs executions are internally managed by the scenario named B1\_RUN\_ALL. This should be scheduled using the Oracle Data Integrator to run every 5-10 minutes.

## Job Monitoring

As jobs keep running on a regular basis, they should be monitored or tracked to ensure that the required performance parameters are met. Jobs are created so that they are capable of automatic re-execution and retries. However, reasons for failure should be looked into and appropriate actions should be taken to resolve those issues. The following views are provided for achieving this:

| View                  | Purpose                                                                                                                                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| b1_jobs_vw            | Provides a list of jobs executed, with details of the slice start and end, scheduled execution time, actual start and end times, status and record counts.                                                                    |
| b1_config_vw          | Consolidated view of the current configuration for all entities in the data warehouse                                                                                                                                         |
| b1_replication_cfg_vw | Consolidated view showing which objects will be replicated, are available for replication and the methodology that will be used for replication. this also provides details on whether the tables have been set up for purge. |



| View               | Purpose                                                                                                                                                                                                     |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| b1_wait_reasons_vw | There are possibilities of erroneous configuration or other reasons due to which entities may not be running. To figure out what may be preventing a job from executing have a look at this view's results. |

## Debugging

The Oracle Data Integrator jobs are designed to make it easy to look at the data that was processed to figure out issues in the processing pipeline. To do this, we utilize the staging tables that retain data for a configured duration. Each execution results in some data that has been processed from the replication layer into the staging area, and then, further on to the target entity. Each such set is stored in the staging table with the session number to identify the executing session.

At any point of time if a job fails or you want to look at how data was before it was loaded into the target, you can query the associated staging tables to look at the data.

Staging tables names are derived by prefixing the target entity name with "STG\_".

## Data Reload

Data warehouses are usually designed based on the assumption that data is added only once to the warehouse. It is rare, although, sometimes to reset specific entities, the data is reloaded again. Oracle Utilities Advanced Spatial and Operational Analytics provides the functionality to reset and reload individual entities or all data associated with a specific instance of a source system.

The following packages in Oracle Data Integrator are provided to support this functionality.

### Resetting an entity

Perform the following steps to reset an entity

1. Log in to ODI Client.
2. Go to variables section under the project folder.
3. Open the variable **B1\_RESET\_ENTITY\_NAME** and in the "refreshing" tab change the query to return the entity name to be reset.  
For example, to reset the fact CF\_OPR\_DEVICE the query would be  

```
select 'CF_OPR_DEVICE' from dual
```
4. Save and close the variable.
5. Navigate to **Load Plans & Scenarios** section. Select the scenario named **B1\_RESET\_ENTITY**.
6. Right-click and execute.
7. In the popup menu, select the context.  
For example, if you want to reset the fact for the Oracle Utilities Device Management product instance 1, you should select the context **ODM1**.
8. The entities automatically start reloading after the reset is completed.

### Resetting an Instance

Perform the following steps to reset an instance:

**Note:** The steps shown below to reset an instance should be used with care. The replication layer data should not be purged already, or else, the target does not load with the expected amount of data.

1. Log in to the Oracle Data Integrator Client
2. Navigate to **Load Plans & Scenarios** section. Select the scenario named **B1\_RESET\_INSTANCE**.
3. Right-click and execute.
4. In the popup, select the context.  
For example, if you want to reset all entities for the Oracle Utilities Operational Device Management (ODM) product instance 1, you should select the context **ODM1**.
5. After verifying the configurations, enable all entities you wish to load.

## Configuring Analytics

This section describes the various Administration Dashboards and the details about configuration of Dashboards.

- **Administration Dashboards**
- **Configuring Dashboards**

### Administration Dashboards

This section describes the various reports available as part of the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) Administration Dashboards.

The following dashboard pages are described in this section.

**Base Field Maintenance**

**Custom Field Maintenance**

**Configuration**

**ETL Job Control**

#### Base Field Maintenance

This dashboard allows users to provide an override description for the fields delivered along with the product. These fields contain descriptions, which appear on report titles and column titles for dashboards delivered with the product.

For complete details on the field labels, refer to **Label Configuration** in section **Configuring Dashboards**.

#### Custom Field Maintenance

This dashboard allows users to define additional fields and their description, which they may want to add as part of customization.

This dashboard has the following pages to allow users to Add, Edit, and Delete their own fields:

- Update
- Insert
- Delete

For complete details on the field labels, refer to **Label Configuration** in section **Configuring Dashboards**.

#### Configuration

This dashboard provides pages for users to configure the drill back URL and map related configuration. The two pages in this dashboard are described below.

- **Source Drill Back**
- **Map Profile**

### Source Drill Back

This dashboard page allows users to provide the source application details for Oracle Utilities Advanced Spatial and Operational Analytics product integration. The source application details like hostname, port number, and context is entered in this dashboard page. These details can be entered for the Oracle Utilities Customer Care and Billing, Oracle Utilities Meter Data Management, and Oracle Utilities Mobile Workforce Management edge applications depending on which application is being used.

This information is used by reports in the detail pages for providing a link back to a specific source application page.

For more detailed information on drill back configuration, refer to section [Configuring Drill Back](#).

### Map Profile

This dashboard page allows users to provide an override value for the map profile configuration. These configurations are used by maps available in the Outage Analytics Dashboard.

For complete information on drill back configuration, refer to section [Spatial Data Configuration](#).

### ETL Job Control

This dashboard page allows users to monitor the Oracle Warehouse Builder ETL Jobs. Users can track the status of the jobs along with its various attributes. During its lifecycle, when a job fails, users can resubmit the job for reprocessing using this page.

For more detailed information on ETL Job Control page, please refer to the section [Scheduling and Monitoring](#) in **Chapter 5: Oracle Warehouse Builder Components**.

## Configuring Dashboards

This section describes the configuration dashboards in Oracle Utilities Advanced Spatial and Operational Analytics (OUASA), including:

- **Configuring Drill Back**
- **Oracle Utilities Meter Data Management Analysis Configuration**
- **Label Configuration**
- **Spatial Data Configuration**
- **About Page**

### Configuring Drill Back

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) provides multiple drill back functionality from various reports in OUASA to the source applications like Oracle Utilities Meter Data Management (MDM), Oracle Utilities Mobile Workforce Management (MWM), and Oracle Utilities Customer Care and Billing (CC&B). You must configure drill back by providing required information for this functionality to work.

The Configuration dashboard under Administration group can be used for configuring various options in Oracle Utilities Advanced Spatial and Operational Analytics (OUASA). The **Configuration** tab contains drill-back settings for source applications.

You can update the host name, port, and the context root folder for various edge applications with Oracle Utilities Advanced Spatial and Analytics (OUASA) application through this page. After updating the values for the environment, the drill back links on various dashboard pages use these new values when an item is selected.

Currently, this drill back configuration supports the following edge applications:

- Oracle Utilities Customer Care and Billing (CC&B)
- Oracle Utilities Meter Data Management (MDM)
- Oracle Utilities Mobile Workforce Management (MWM)
- Oracle Utilities Operational Device Management (ODM)

**Note:** Customers are required to configure only those source applications for which they have implemented the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA).

The drill back for the first three products are done in the Configuration Dashboard, while for the ODM, it is configured in the Administration Tool available with the release 2.4.1 of OUASA.

## Oracle Utilities Meter Data Management Analysis Configuration

The following Oracle Utilities Meter Data Management Answers should be configured before viewing the data:

- Tamper Events answer (Overview dashboard)
- Usage Unreported for > 30 Days (Overview dashboard)
- Percent of Normal Intervals (Overview dashboard)
- Percent of On-Time Intervals (Overview dashboard)
- Degree Days (Overview dashboard)

After customizing the answers, save the reports in a separate CM catalog.

Refer to the *Oracle Utilities Meter Data Management Metric Reference Guide* for details.

## Label Configuration

This section describes how to create and customize the labels that appear in answers and dashboards.

**Note:** You must have the full license of the Oracle Utilities Advanced Spatial and Operational Analytics to use this feature.

This section includes the following topics:

- **Overview of Labels**
- **Overriding Base Labels**
- **Supporting Multiple Languages**

### Overview of Labels

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) uses labels for columns and tables in the delivered Oracle Business Intelligence Enterprise Edition (OBIEE) repository file when displaying the columns in the presentation folders for users. These labels are displayed in the answers for the columns on the dashboards. In addition, the answers are also titled based on labels stored in metadata displayed in report titles, sub-titles, and other dashboard strings.

The main reason the application uses labels instead of hard coding the text values in the answers and RPD file is to support translation of dashboards into different languages and allow easy overriding of labels for customers, if they wish to customize the field label.

For example, within an answer, labels can be referred to by biServer variables. For example, the Device Activity - Distributions report uses this variable in the title section of the answer:

```
@{biServer.variables['NQ_SESSION.B1_RPT_DEV_ACTI_DISTRIBUTION']}
```

The B1\_RPT\_DEV\_ACTI\_DISTRIBUTION label is defined in the B1\_MD\_FLD table in DWADM schema.

For columns in the fact and dimension tables, labels exist for every field. For example, the UDF1\_DESCR column in the CD\_ACCT table has the Description of Customer Class, and the Customer Class label is displayed in the presentation folder for this field.

## Overriding Base Labels

There are several reasons that an implementation may want to update an existing label:

- A field may contain data that does not match the default extracted data for that field.  
In the CD\_ACCT example in the above section, you may elect to store information other than customer class in the UDF1\_DESCR field. If an extract change is made to the default CD\_ACCT extract, then an implementation change in the label for the UDF1\_DESCR field of the CD\_ACCT table at one place will change the label in all dashboards and answers that display that field. This reason also applies if data is extracted to a User Defined Field (UDF) field that is not already having a default population.
- Even if you use the default extract code, you may choose to use some other name for the extracted data other than the default name.  
In the CD\_ACCT example, if you call the field extracted into the UDF1\_DESCR field Account Class instead of Customer Class, you can make this change at one place and have it updated on all dashboards and answers.
- You may want to provide multilingual labels for your users. The Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) application provides the labels to a user based on the language selected when logging into the Oracle Business Intelligence Enterprise Edition (OBIEE), assuming that the language is present in the B1\_MD\_FLD table. An implementation can add its own translated fields or can download supported language packs from the Oracle Software Delivery Cloud.

**Note:** Multilingual support is only provided for labels and not for the underlying data in the data warehouse. The data displayed in all database tables is not translatable from extract language.

## Supporting Multiple Languages

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) is released with default support for English labels on all dashboards and answers. Oracle Business Intelligence Enterprise Edition (OBIEE) and OUASA both provide support for multiple languages.

The default language on the login page is English. However, users can select any of the supported language on the login page or can change the preferred language under the Administration menu to view Dashboards in a different language. If you have not purchased and applied the specific language pack and if you select a language other than English, the default Oracle Business Intelligence Enterprise Edition (OBIEE) labels will still be translated in the selected language but the Oracle Utilities Advanced Spatial and Operational Analytics product specific labels will appear in English.

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) may release various language packs depending on user demands. Hence, for the language that is already released, installing the language pack is sufficient for creating the labels needed by the dashboards.

To view the list of Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) language pack applied on an environment, you can navigate to the About OUASA dashboard under the About group in the OBIEE Dashboards menu.

Contact your Oracle support representative to purchase a OUASA Language Pack for additional language support.

To update a label for a base field, use the Base Field Maintenance dashboard in the Administration portal.

- **Table Labels:** For records that have the Table Name field populated but not the Field Name, this label is shown in the presentation folder for the fields available in this table. For example, the CD\_ACCT table has the label 'Account Dimension' displayed in the Presentation folders wherever it is used.
- **Field Labels:** For records that have both the Table Name and Field Name fields populated, this label is shown in the presentation folder and on answers whenever that field is used. For example, the UDF1\_DESCR field in the CD\_ACCT table has the label 'Customer Class' displayed whenever it is used in an answer or when a user selects it from the presentation folder when creating a new answer.
- **Report Labels:** Records that have a field name like 'B1\_RPT%' and no table name value are used for the titles of answers in the dashboards. For example, the B1\_RPT\_DEV\_ACTI\_DISTRIBUTION label is defined to be 'Device Activity Distribution', and this is displayed on the MDM dashboard when the answer is displayed.
- **Other Labels:** All other non-report labels that have a field name but no table name are used for calculations that are computed in the RPD Logical layer for display on answers. For example, the B1\_APPT\_IND\_SUM label is defined to be 'Number of Appointments', and is used in MWM answers that compute the number of crew appointments based on the Appointment Indicator field in the CF\_CREW\_TASK fact table.

If a base field label should be changed, then the implementation team can query the required record on the Base Field Maintenance dashboard, populate a value in the Override Description field, and click Update. Once populated, the OBIEE Server must be restarted for the changes to take effect.

## Spatial Data Configuration

This section describes how to configure mapping for Oracle Utilities Advanced Spatial and Operational Analytics. It includes the following topics:

- **Loading Geographical Data**
- **Integrating Mapping with the Network Management System Network Model**
- **Map Profile Configuration (For NMS Maps)**
- **Configure Google Map Tile Layer**
- **Implementing Maps**

### Loading Geographical Data

In order to place information on a geographic map, data in the data warehouse must match geographic data (themes) that are configured in Oracle Map Viewer.

The standard map answers delivered with Oracle Utilities Advanced Spatial and Operational Analytics include maps that query state, city, county, postal codes, and network model summary data. As Oracle Utilities Advanced Spatial and Operational Analytics does not have access to this spatial data (and as each customer may require different spatial data), you must set up the geographic themes used in the maps.

For details regarding setting up these standard spatial themes, refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide*.

The installation instructions refer to shape files downloaded from the US Census bureau. However, shape files can also be used for the state, city, county, and zip code boundaries. The only requirement is that the names of the geographic entities in the shape file should match the corresponding name in the CD\_ADDR table. This is not usually a problem for postal code data, but can be an issue for city and county names, as different sources may use different names to refer to geographic places. Make sure after loading the MapViewer shapefiles that the names in the geographic tables match the names that are in the CD\_ADDR table. If they do not match, then the maps may not display correctly.

## Integrating Mapping with the Network Management System Network Model

Oracle Utilities Network Management System (NMS) provides a mechanism to create spatial themes in Oracle MapViewer for the entire electrical network model. The default implementation of Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) does not provide links to these MapViewer themes; however, the maps can be modified in the Outage Analytics dashboards to show the various elements of the network model on the Outage Maps. This section provides an overview of the steps needed to provide the network model on the outage maps. For details regarding the steps required to set up the Network Management System (NMS) Model Build process, refer to the NMS Installation Guide.

To build the network model in Oracle Utilities Network Management System (NMS), the model build process must be set up to populate the geometry columns. Detailed instructions for this can be found in the NMS Installation Guide.

To ensure that the network model can be displayed without coordinate translations during runtime, one of the geometry columns should use the same projection coordinate system as the base map used by the Outage Analytics Outage maps. If Oracle eLocation is being used as the base map, then this would be srid 54004.

Once the model build is set up to populate the geometry columns, themes for the various network model components must be built. For example, there may be a theme for transformers, another theme for conductors, and other themes as required to build the network model. NMS provides templates that helps in setting up these themes, a base product installation may have ten or more themes. For details, refer to the *Oracle Utilities Network Management System Installation Guide*.

After the Network Management System (NMS) Network Model themes are set up, they can be either accessed directly from the outage Analytics Outage Maps or can be copied to the MapViewer instance used by the Oracle Business Intelligence Enterprise Edition (OBIEE) dashboards.

There following are the advantages of using either of the two options discussed above:

- If the Oracle Utilities Network Management System (NMS) themes are accessed directly, then near real-time device status information can be displayed in the Outage Analytics Outage Maps. Caching affects the lag time of the status information. This can be a good or bad, depending on how the refresh frequency of the Near Real Time (NRT) outage information in the data warehouse is set. There may remain some information in the device status that is yet to be extracted. This may result in a mismatch between the database data and the spatial data.
- Display of the Network Management System (NMS) themes require access to the NMS database; hence, if the database is down or network access to the database is not available from the data warehouse database, then the NMS themes is displayed on the maps.
- There can be a performance impact on the NMS database, if a large number of users are displaying the outage maps in the Oracle Business Intelligence Enterprise Edition (OBIEE) dashboards.
- If the themes are accessed through the Business Intelligence (BI) database, then a mechanism to periodically copy any changes to the Network Model from the Network Management System (NMS) to Business Intelligence (BI) needs to be setup. The DIAGRAM\_OBJECTS table along with the Theme Metadata should also be copied.

For each theme defined in the Network Management System (NMS) Network Model that should be displayed on an Outage Map, the static text for the answer should be updated to access that theme and provide a check box that will turn the theme on or off.

Note that if this change is done, the process described in the Oracle Business Intelligence Enterprise Edition (OBIEE) customization section above should be followed. First create a copy of the answer that needs to be modified, edit that copy of the answer, and then modify the dashboard to access the copy of the answer instead of the released copy of the answer.

Consider the following example. You want to add two Oracle Utilities Network Management System (NMS) themes named 'Transformers' and 'Conductors' to the Outage Map answer on the Overview tab of the Outage Dashboard. If you edit the dashboard page, you will see that this answer is called 'outage demo' and exists in the / Shared Folders/Outage Analytics/Spatial Requests folder. To edit modify this answer, follow these steps:

1. Create a folder called '**Project Customizations**' or some other unique name in the /Shared Folders folder, and copy the outage demo answer into it.
2. Open the outage demo answer and edit the static text box.
3. In the static text box, edit the JavaScript code to add the new Transformers and Conductors themes. The following code should be added just before the addLegendFilter code. Set the legendSelected parameter to '**Y**' to display the network model themes when the map first opens up.

```
var parm11 = new Array();
parm11['nodeId'] = 'MapNode1';
parm11['legendLabel']='Show Transformers';
parm11['theme'] = 'Transformer';
parm11['legendSelected'] = 'N';
addStaticThemeParameter(parm11);

var parm12 = new Array();
parm12['nodeId'] = 'MapNode1';
parm12['legendLabel']='Show Conductors';
parm12['theme'] = 'Conductors';
parm12['legendSelected'] = 'N';
addStaticThemeParameter(parm12);
```

**Note:** Using the Firefox Browser to edit static text sometime results in issues like not being able to view the complete static text. Hence, it is recommended to use Internet Explorer (IE) browser instead when editing the static text for the Map answers in Oracle Utilities Advanced Spatial and Operational Analytics.

### Map Profile Configuration (For NMS Maps)

A separate page for the map configuration is available as part of the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) Administration Dashboards. Under the OBIEE Dashboards menu, users can access the Configuration dashboard and navigate to the Map Profile page. The Map Profile page contains configuration options for the 10g version of the maps used in the NMS dashboards. The various options are broadly classified into Map Attributes and the Theme Attributes. Out of the box these configuration options are supplied with a default value. But if users want to specific a different value, they can do so by providing an override value in the column provided.

### Configure Google Map Tile Layer

Out of the box Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) is configured to fetch the map tiles from Navteq. However, users can switch to Google as an alternative source for the map tiles.

#### Configuring Google Map Tile Layer

Perform the following steps:

1. Logon to Map-viewer Console  
Sample URL:  
`http://<host-name>:<port>/mapviewer`
2. Click **Admin** and enter the credentials
3. On the **Management** tab, click **Manage Map Tile Layers**.
4. Create a new tile layer by selecting **Google**.



Location, Location, Location

Manage MapViewer | Manage Map Tile Layers

• Create  
• **Manage**

✓ TIP You must first specify where the map tile images will c  
Select type of map source: Google Maps

Continue

- Enter the key fetched from Google in the key field and edit the default lib\_url to include the key value and choose appropriate data source.

Select and Add Delete

Select name value

Properties:

- ☐ lib\_url mp.sensor=false&key=\$key\$
- ☐ key your\_key
- ☐ map\_type\_values MVGoogleTileLayer.MAP\_TYPE\_RC
- ☐ map\_type\_names Road,Hybrid,Shaded,Satellite
- ☐ version 2

- Click on **Submit** to save the information.

Manage MapViewer | Manage Map Tile Layers

• Create  
• **Manage**

Managing Map Tile Layers

Refresh

Existing map tile layers

Select a map tile layer and Edit/View details View map/Manage tiles Bring online Take offline Delete

| Select Name | Data Source | Base map | Zoom levels | Internal | Tile width | Tile |
|-------------|-------------|----------|-------------|----------|------------|------|
| GOOGLE_MAP  | MAPCONN1    |          | 19          | false    | 256        |      |

- Click on **View map/Manage** tiles to verify the new Tile layer shows up properly.
- Click on **Show Map** button to view the map.
- Logon to OBIEE Analytics and navigate to **Administration > Manage Map Data** and click on the Background Maps tab.


Location, Location, Location

Home Management Rec

Area Selection Tool  
Draw a rectangular selection area on the map  
☐ Off

Zoom Level Selection

Center X 0.0 Center Y 0.0 SRID 3785 Zoom Level 0 Show Map



- Import the tile layer '**GOOGLE\_MAP**' created on the map-viewer console.
- Add Layers to the map and save it.

**Edit Background Map - GOOGLE\_MAP**

Name:

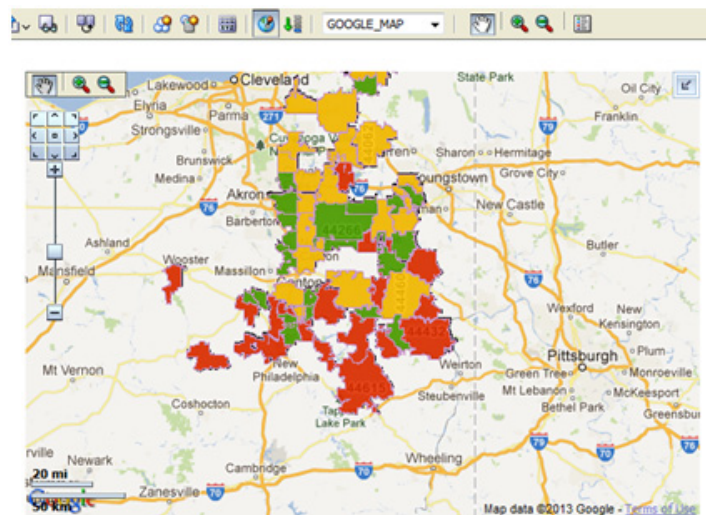
Description:

**Interactive BI Layers**  
For each layer, select the zoom levels at which associated BI data can display

|             | Zoom Level |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |
|-------------|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
|             | 0          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| City        |            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |
| City Name   |            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |
| Postal Code |            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |

12. Subject areas will now be associated to the map.

13. Create a new answer and add a new Map-view to have the data displayed on the Map.



## Implementing Maps

This section describes the two different methods of implementing maps in Oracle Utilities Advanced Spatial and Operational Analytics (OUASA), including:

- **Standard Implementation Method**
- **Custom Implementation Method**

### Standard Implementation Method

The standard implementation method is the default implementation method for Oracle Business Intelligence Enterprise Edition (OBIEE) 11g. This form of map can be seen in various dashboards like Meter Data Management (MDM), Mobile Workforce Management (MWM) and Customer Care and Billing (CC&B) Analytics. Using this method, you can create new answers using the Map View. This view uses the configuration defined under the Administration menu in Manage Map Data.

The layers, background maps, and the images being used in the map must be defined in this page. The Key column and geographical columns should be mapped for each Subject Area used in the analysis. This is a one-time setup unless new subject areas are added.

Note that you should not customize the map metadata until after you import the Spatial Catalog file.

For customizations that involve map analysis, all customer modifications must be done in a separate folder in order for those modifications to be preserved when upgrading Oracle Utilities Advanced Spatial and Operational Analytics.

### Custom Implementation Method

The custom implementation method has been used for Outage Maps from the Oracle Utilities Network Management System (NMS) dashboards and is similar to the stand-alone Map Viewer setup used in Oracle Business Intelligence Enterprise Edition (OBIEE) versions prior to OBIEE 11g. Sets of Map attribute and theme profiles are provided to support this method. Attribute profiles hold the data source information and the Application Programming Interface (API) keys. Theme profiles are used to map the Geographic column with the key column.

Using the above method you can:

- Create new answers using static text view with a call to the Standard APIs along with the theme profiles that should be applied.
- Update theme profiles from the Map Profile Page that is provided in the Configuration Dashboard. You can override the base values by using the Override Value column, as shown in the below image:

**Note:** No support is provided to create new theme profiles. Upgrade scripts are provided to load the custom themes into the new Configuration table.

### About Page

The About Page shows information about the product name along with the current release version and patch number. It also lists all of the languages that are currently installed in the product.

### Supporting Multiple Languages

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) is released with default support for English labels on all dashboards and answers. Oracle Business Intelligence Enterprise Edition (OBIEE) and OUASA both provide support for multiple languages.

The default language on the login page is English. However, users can select any of the supported language on the login page or can change the preferred language under the Administration menu to view Dashboards in a different language. If you have not purchased and applied the specific language pack and if you select a language other than English, the default Oracle Business Intelligence Enterprise Edition (OBIEE) labels will still be translated in the selected language but the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) product specific labels will appear in English.

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) may release various language packs depending on user demands. Hence, for the language that is already released, installing the language pack is sufficient for creating the labels needed by the dashboards.

To view the list of Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) language pack applied on an environment, you can navigate to the About OUASA dashboard under the "About group" in the OBIEE Dashboards menu.

Contact your Oracle support representative to purchase a OUASA Language Pack for additional language support

## Performance Recommendations

This section discusses how to properly configure the database for partitioning and parallelism for better system performance.

Partitioning helps to scale a data warehouse by dividing database objects into smaller pieces, enabling access to smaller, more manageable objects. Having direct access to smaller objects addresses the scalability requirements of data warehouses.

It takes longer to scan a big table than it takes to scan a small table. Queries against partitioned tables may access one or more partitions that are small in contrast to the total size of the table. Similarly, queries may take advantage of partition elimination on indexes. It takes less time to read a smaller portion of an index from disk than to read the entire index. Index structures that share the partitioning strategy with the table, such as local partitioned indexes, can be accessed and maintained on a partition-by-partition basis.

The database can take advantage of the distinct data sets in separate partitions if you use parallel execution to speed up queries, DML, and DDL statements. Individual parallel execution servers can work on their own data sets, identified by the partition boundaries.

Parallel execution enables the application of multiple CPU and I/O resources to the execution of a single database operation. It dramatically reduces response time for data-intensive operations on large databases typically associated with a decision support system (DSS) and data warehouses. You can also implement parallel execution on an online transaction processing (OLTP) system for batch processing or schema maintenance operations such as index creation. Parallel execution is sometimes called parallelism. Parallelism is the idea of breaking down a task so that, instead of one process doing all of the work in a query, many processes do part of the work at the same time. An example of this is when four processes combine to calculate the total sales for a year, each process handles one quarter of the year instead of a single process handling all four quarters by itself. The improvement in performance can be quite significant. Parallel execution improves processing for:

- Queries requiring large table scans, joins, or partitioned index scans
- Creation of large indexes
- Creation of large tables (including materialized views)
- Bulk insertions, updates, merges, and deletions

For details on parallelism, partitioning, and other performance enhancement options, refer to the Oracle Database VLDB and Partitioning Guide 11g Release 2.

The following topics are discussed in detail:

- **Parallelisms in Oracle Warehouse Builder Mappings**
- **Parallelism in Materialized Views**
- **Parallelism in Materialized Views and Facts**

## Parallelisms in Oracle Warehouse Builder Mappings

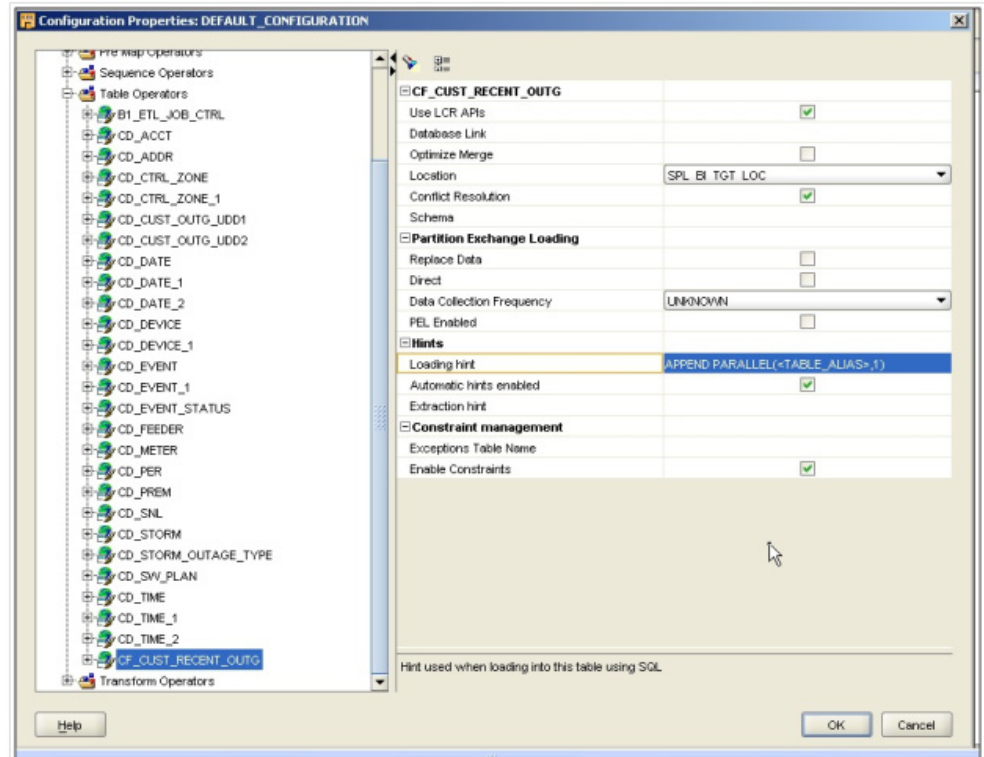
Oracle Warehouse Builder (OWB) mappings generate PL/SQL packages which utilize bulk load to populate the target entities. These jobs can make use of the parallel DML feature available with the Oracle database. All mappings have been configured for parallelism; however, the degree of parallelism has been set to a default of 1. Customers can appropriately change the degree of parallelism based on their hardware setup, data volumes for individual entities, and the performance gain obtained by increasing the degree of parallelism.

Installing or upgrading the Oracle Utilities Advance Spatial and Operational Analytics (OUASA) prompts you to set the degree of parallelism. However, if you want to change the degree of parallelism, follow these steps outlined in the Oracle Warehouse Builder (OWB):

### Changing the Degree of Parallelism

1. Open Oracle Warehouse builder designer.
2. Navigate to the mapping for which you would like to modify the degree of parallelism.
3. On the Context menu, click **Configure**.

4. Navigate to the target entity and change the loading hint (highlighted in the image below), by replacing the value 1 with an appropriate degree of parallelism.
5. Save the changes and redeploy the mapping



## Parallelism in Materialized Views

The materialized views can also utilize parallelism to refresh the snapshot data quickly. All materialized views have been preconfigured to a default degree of parallelism of 1. However, this can easily be changed during install or post install from the database or through the Oracle Warehouse Builder (OWB)

To change the degree of parallelism at the time of the installation, refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide*.

Use the following command to change the degree of parallelism through database:

```
ALTER MATERIALIZED VIEW <MVIEW_NAME> PARALLEL <DEGREE>
```

Specify the materialized view for which you want to change the parallelism by replacing <MVIEW\_NAME> with the actual materialized view name and <DEGREE> with an appropriate integer value greater than 1. This does not require the materialized view to be recreated.

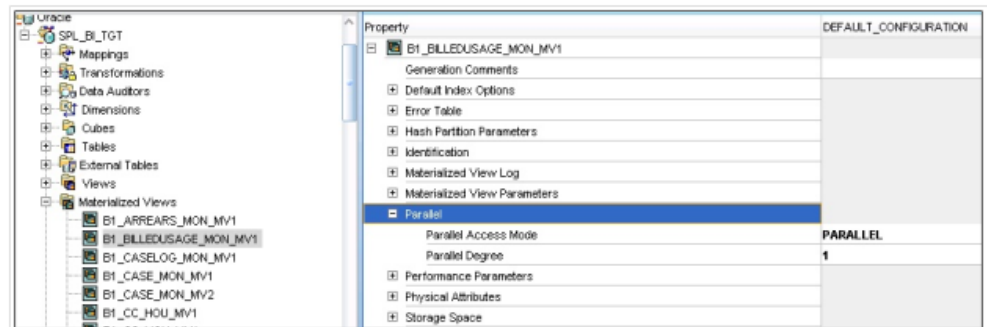
### Changing the Degree of Parallelism for a Materialized View

Follow these steps to change the degree of parallelism for a materialized view using OWB:

1. Open the Oracle Warehouse Builder (OWB) designer.
2. Navigate to the mapping for which you would like to modify the degree of parallelism.

3. On the Context menu, click on Configure.Extract, Transform, and Load Processes (ETL) 3-27 .
4. Change the property for the Parallel Degree to an appropriate value.
5. Save changes and redeploy the materialized view.

**Caution:** Before changing the degree of parallelism for mappings or materialized views, understand and analyze the implications of the change and the capability of the hardware to support the changes.



## Parallelism in Materialized Views and Facts

The materialized views and the facts which are components of the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) product are not partitioned by default. Customers who have the partitioning license can opt to partition the materialized views and the facts as another way to increase the overall performance of the product. The following topics are discussed in detail in this section:

- **Partitioning Recommendations**
- **Tables to Consider for Partitioning of Materialized View**

### Partitioning Recommendations

It is recommended that the materialized views be partitioned on the year and month columns. Most of the dashboards and analytics are month and year based; hence, partitioning on the month and year improves efficiency of the data fetches in the dashboards.

In particular, we recommend to partitioning the following materialized views used in the TopX reports be partitioned on the month and year keys:

- B1\_VEE\_EXCP\_TOPX\_MON\_MV1
- B1\_DEV\_ACT\_TOPX\_MON\_MV1
- B1\_DEVICE\_EVT\_MON\_TOPX\_MV1
- B1\_SP\_SNAP\_MON\_TOPX\_MV1
- B1\_SP\_UT\_AGE\_MON\_TOPX\_MV1
- B1\_CONSUMPTION\_MON\_TOPX\_MV1
- B1\_CREW\_TASKS\_MON\_MV1
- B1\_CMP\_SHIFT\_MON\_MV1
- B1\_FLD\_ACTIVITY\_MON\_MV1
- B1\_OPR\_DEVICE\_SNP\_MON\_TOPX\_MV1

- B1\_OPR\_DEVICE\_MON\_TOPX\_MV1

These materialized views can become huge quite quickly and partitioning based on the date key of the fact may help in improving the performance.

Note: The above-mentioned list is not a complete list of materialized views delivered with the Oracle Utilities Advanced Spatial and Operational Analytics product

## Tables to Consider for Partitioning of Materialized View

If the partitioning of materialized view does not prove sufficient to achieve desired performance you can consider partitioning Fact tables as well.

As the primary keys for all tables are sequential, it is possible to partition any table based on the primary key field. However, you should partition fact tables based on one of the date keys present in the table. Some of the date keys are optional, so it is important to pick a date key field that will always have a non-zero value. Also, because the recent fact tables should be purged daily, these tables do not need to be partitioned.

The following is a list of the tables and corresponding key columns that are candidates for partitioning. The partitioning key listed is the suggested date key field that should not have 0 values.

**Note:** The optimal partition key for a table may vary depending on the data. The list below is just a potential candidate keys for partitioning.

| Table Name           | Partition Column Name |
|----------------------|-----------------------|
| CF_ARREARS           | DATE_KEY              |
| CF_BILLED_USAGE      | BILL_DATE_KEY         |
| CF_CASE              | OPEN_DATE_KEY         |
| CF_CASE_LOG          | LOG_DATE_KEY          |
| CF_CC                | CC_DATE_KEY           |
| CF_CITY_OUTG         | CF_CITY_OUTG          |
| CF_COLL_EVT          | EVENT_DATE_KEY        |
| CF_COLL_PROC         | START_DATE_KEY        |
| CF_CTRL_ZONE_OUTG    | BEGIN_DATE_KEY        |
| CF_CUST_RECENT_OUTG  | BEGIN_DATE_KEY        |
| CF_CUST_RST_OUTG     | BEGIN_DATE_KEY        |
| CF_FEEDER_DLVRD_LOAD | SNAPSHOT_DATE_KEY     |
| CF_OP_ACTG           | TRANS_DATE_KEY        |
| CF_ORDER             | CREATE_DATE_KEY       |
| CF_OUTG              | SNAPSHOT_DATE_KEY     |
| CF_PAY_TNDR          | PAYEVT_DATE_KEY       |
| CF_RST_CALL          | CALL_DATE_KEY         |
| CF_RST_CREW          | ASSIGN_DATE_KEY       |

| Table Name                 | Partition Column Name                                  |
|----------------------------|--------------------------------------------------------|
| CF_RST_JOB                 | BEGIN_DATE_KEY                                         |
| CF_SA                      | START_DATE_KEY                                         |
| CF_STRM_INV                | SNAPSHOT_DATE_KEY                                      |
| CF_SW_PLAN                 | BEGIN_DATE_KEY                                         |
| CF_SW_PLAN_STATE           | BEGIN_DATE_KEY                                         |
| CF_TD_ENTRY                | CREATE_DATE_KEY                                        |
| CF_UCOL_EVT                | EVT_DATE_KEY                                           |
| CF_UCOL_PROC               | START_DATE_KEY                                         |
| CF_FLD_ACTIVITY            | SCHED_START_DATE_KEY,<br>STATUS_DATE_KEY, CRE_DATE_KEY |
| CF_CMP_SHIFT               | SHIFT_PLANNED_START_DATE_KEY                           |
| CF_CREW_TASK               | FROM_DATE_KEY                                          |
| CF_INSTALL_EVT INSTALL     | DATE_KEY                                               |
| CF_SP_SNAP                 | DATE_KEY                                               |
| CF_VEE_EXCP                | DATE_KEY                                               |
| CF_DEVICE_ACTIVITY         | START_DATE_KEY                                         |
| CF_DEVICE_EVT              | START_DATE_KEY                                         |
| CF_SP_UT_AGE               |                                                        |
| B1_OPR_DEVICE_SNAP_MON_MV1 |                                                        |
| B1_SERVICE_HIST_MON_MV1    |                                                        |
| B1_ASSET_LOC_MON_MV1       |                                                        |
| B1_OPR_DEVICE_MON_MV1      |                                                        |

**Note:** Before partitioning the facts or materialized views, understand and analyze the implications of the change and the capability of the hardware to support the changes.

## Optimizing Top N Answers

Top N charts have to go through millions of records to find out Top N objects that meets the criteria. Top N Materialized Views rearrange and partition the data so that the data reads are optimal. However, at times, depending on the amount of data, additional configuration may be required to reduce the size of the data set, which is being scanned in order to identify the Top N objects.

For example, a large sized utility with 6 million customers will have around 6 million service points in their Oracle Utilities Meter Data Management application. This means that the Consumption Fact in the OUASA MDM star schema will have several million records for every snapshot month. So, for each month level partition on the Consumption Detail level materialized view, there will be several million records. The detail level pages (Top N Analysis and Unreported Usage Details) on the Usage Summary dashboard under Oracle Utilities Meter Data Analytics tries to access a huge volume of data which can result in sluggish performance sometimes. The recommendation in



such scenarios is make a key prompt filter like '**City**' as mandatory. This ensures the report looks at a smaller data set; thereby, improving the report performance. Customer can choose to make any set of filters mandatory as per their data requirement.

# Chapter 5

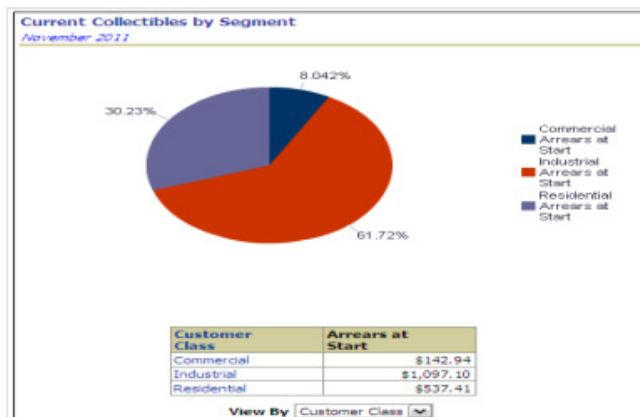
## Extending the Oracle Utilities Advanced Spatial and Operational Analytics

The data warehouse schema in Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) covers a wide range of reporting requirements. Customers often required to add additional data elements to meet site specific requirements. OUASA allows such extensions to the schema through the use of user defined constructs such as User Defined Fields, User Defined Measures, User Defined De-Generate Dimensions, and User Defined Foreign Keys. Using these constructs customers can extended the star schemas that are delivered along with the product. With the additional data now available in the warehouse custom reports can be created in OBIEE to leverage this additional data. Using these features customers can easily customize the product to suite their extended requirements. The following sections are described in this chapter:

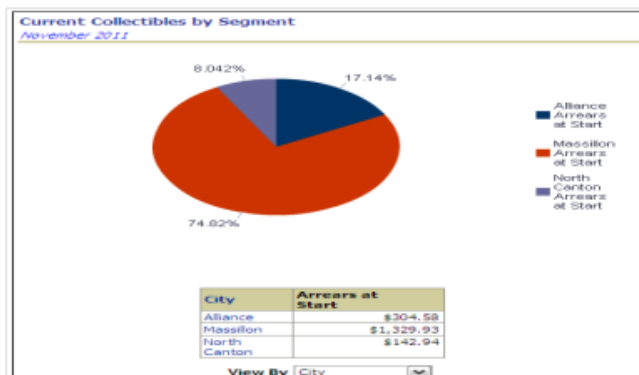
- **Sample Use Case**
- **User Defined Columns**
- **Extending Start Schemas**
- **Customizing Analytics**

### Sample Use Case

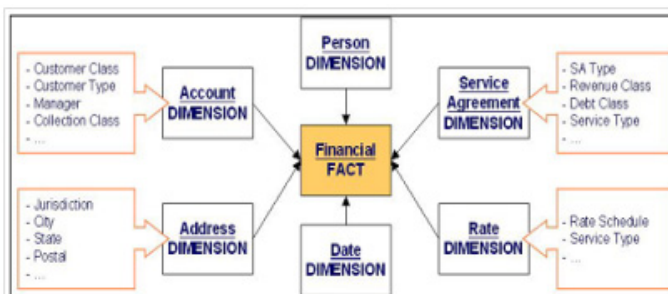
Users look at the facts in a data warehouse by slicing and filtering the analytic data by different dimensions. For example, the following graph shows collectible sliced by customer class (the Collectibles fact is sliced by the customer class field on the account dimension):



Whereas the below report slices the same fact by a different field on a different dimension (the city on the address dimension). In addition, it limits the analysis to a specific customer class, i.e., Commercial. The below figure shows how a single report can be sliced and filtered by different dimensional attributes.



Users can "slice and filter" a fact using any field on any dimension linked to the analytic's fact. For example, users can slice reports related to the financial fact by any field on its dimensions. The following simplified data model of the financial fact's star schema helps clarify this concept.



This data model shows that the financial fact has six dimensions. This means that graphs can be configured to allow end users to slice the financial fact by any of the attributes on the six dimensions. For example, you could set up a report to slice the financial fact by any combination of:

- The Account Dimension's Customer Class and Manager
- The Address Dimension's Jurisdiction
- The Service Agreement Dimension's Revenue Class
- The Rate Schedule Dimension's Rate

You can set up another report to slice and filter this fact by a different combination of fields. It should be noted that the above example is a simplified version. In reality, most facts have more than six dimensions and most dimensions have several fields.

While Oracle Utilities Advanced Spatial and Operational Analytics allows you to slice and filter a fact by any field on its dimension, it also enables you to limit the number of fields on your report to a discreet group. This helps as the materialized views that are configured may make the system slow if these views contain too many fields.

## User Defined Columns

The predefined facts and dimensions are provided with a set of user extensible columns which can be used for extending the existing entities. These columns include

- **User Defined Field**
- **User-Defined Measure**
- **User Defined Degenerate Dimension**
- **User Defined Foreign Key Dimensions**
- **User Defined Dimension**

### User Defined Field

User Defined Fields (UDFs) reside on the dimension tables in the star schemas. In general, all dimensions are provided 10 UDF columns. Customers can make use of these columns to extract additional information from the source system.

Once configured appropriately the ETL provided for the star schemas will automatically take care loading the data into the appropriate columns.

You can change the User-Defined Field (UDF) on your dimensions at a later date.

This UDF column feature allows all facts to be sliced and filtered by the newly added data in these User-Defined Fields. If you want to "slice and filter" historical facts by the new fields, you must update the historical dimensions to contain the current value of the new User-Defined Field.

### User-Defined Measure

The acronym User-Defined Measure (UDM) is used to reference the measures on the facts that you populate with implementation-specific measures.

A measure is a column on a fact that holds a measurable value. For example, the financial fact has a measure that holds the amount of revenue produced by a bill. Most facts in the Oracle Utilities Advanced Spatial and Operational Analytics data warehouse have several measures. For example, in addition to the revenue measure, the financial fact also has measures holding the total taxes and other charges on a bill.

For example, the following report shows several measures - Score, Revenue Amount for the current, last, and the last three periods.

Revenue by Segment  
2011 November - FY11

0 - 80 ■ 80 - 100 ■ 100 + ■

| Status | Customer Class | Score  | Revenue Amount | Last Year  | Average Revenue Last Three Periods |
|--------|----------------|--------|----------------|------------|------------------------------------|
| ■      | Commercial     | 70.36  | \$694.42       | \$986.90   | \$826.31                           |
| ■      | Industrial     | 114.09 | \$6,063.22     | \$5,314.54 | \$6,578.76                         |
| ■      | Residential    | 120.68 | \$1,241.93     | \$1,029.15 | \$1,287.65                         |

View By Customer Class ▼

The facts and their extract programs are delivered with all of the obvious measures populated. However, if your implementation requires additional measures, you can populate User-Defined Measure (UDM) on the facts. To do this, you can introduce logic to the fact's extract program (in a user exit) to populate one or more UDM accordingly. Note that no database or Oracle

Warehouse Builder (OWB) changes are necessary as both the data warehouse and OWB are delivered in a ready state to support the newly populated UDMs.

## User Defined Degenerate Dimension

User-Defined Degenerate dimension (UDDGEN) columns reside directly on the fact table and can be used to filter fact data in the same way as the User-Defined Field (UDF). For example, currency code columns are commonly used UDDGEN in the Oracle Utilities Advanced Spatial and Operational Analytics application. These columns exist on most of the fact tables and can be used to limit fact data shown in reports to a given currency code. Most fact tables in Oracle Utilities Advanced Spatial and Operational Analytics are delivered with multiple UDDGEN. These columns are populated by introducing user-exit code in the respective fact extract programs. The main benefit of using UDDGENs as opposed to using User Defined Dimension (UDDs) is that UDDGENs can be populated in the fact extract program and thereby, reduce implementation time.

## User Defined Foreign Key Dimensions

At times, there are requirements that can be easily satisfied by adding an existing dimension to a fact. For example, the case fact is not delivered referencing the service agreement dimension. If your users require analytics that "slice and filter" cases by service agreement dimensional attributes, you can configure the system to reference the existing service agreement dimension on the case fact. Facts that support this type of extension contain columns called User-Defined Foreign Key (UDDFK). If you do not see these columns on a fact, then it means that this functionality is not available in it.

## User Defined Dimension

User Defined Dimensions are empty dimensions that are delivered along with the star schemas in Oracle Utilities Advanced Spatial and Operational Analytics. Customers can make use of these dimensions to extend the existing star schemas.

As described earlier, you can set up analytic reports to "slice and filter" a fact using any field on the dimensions linked to the fact. Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) delivers facts referencing the obvious dimensions. However, your implementation may need you to link additional dimensions to some facts. For example, the financial fact is delivered assuming that the revenue, tax, and expense amounts should be aggregated regardless of the General Ledger (GL) accounts impacted by a financial transaction (for example, if a given adjustment references six revenue GL accounts, all six revenue amounts are summarized onto a single financial fact). This means that you cannot "slice and filter" revenue by specific general ledger accounts.

If customers have such a requirement to "slice and filter" an existing fact data using an additional group of related attributes, then they can opt to make use of these UDD tables. For each fact two User-Defined Dimensions are provided. These tables can be utilized for extending the star schemas. Also provided are five unbound UDD foreign key columns.

You can use user defined dimensions only if the dimension doesn't change the granularity of a fact.

## Extending Start Schemas

Customers can extend star schema without actually updating the table structure. This can be achieved by using the User Defined Columns. Specific details on how to extend the star schema for a particular source product is dependent on the ETL methodology it is based upon. The below-mentioned sections covers both Oracle Warehouse Builder and Oracle Data Integrator based approaches:

- **Oracle Warehouse Builder Based Extension**
- **Oracle Data Integrator Based Extension**

## Oracle Warehouse Builder Based Extension

For the source applications whose ETL is based on OWB the customization will have to start from the source application itself where the data extraction is done. The following source applications make use of OWB based ETL.

- **Oracle Utilities Customer Care and Billing and Oracle Utilities Work and Asset Management**
- **Oracle Utilities Meter Data Management and Mobile Workforce Management**
- **Oracle Utilities Network Management System (NMS)**

Since different source applications employ different approaches to extract their data, the sequence of steps that will have to be done for schema extension vary on the extraction approach as well. Once the extractors have been configured to pull the additional data, the OWB ETL delivered along with the OUASA product will automatically take care of loading these additional data into the warehouse.

The following sections will cover the necessary steps to be done for each of the source applications.

### Oracle Utilities Customer Care and Billing and Oracle Utilities Work and Asset Management

Oracle Utilities Customer Care & Billing (CC&B) and Oracle Utilities Work and Asset Management (WAM) use the trigger-based approach to detect the changes on the base table that needs to be populated in the data warehouse.

Most of the extract programs support populating User Defined Field (UDF) and User Defined Measure (UDM) fields on their flat file records with specific fields from the source system. For example, you can set up the premise extract program to populate the first User-Defined Field on its flat file with the premise's city, county, or any address-oriented field.

You should specify to the extract program which fields require to be populated on the flat file by populating the batch process parameters. The number and type of parameters differ depending on the extract and type of information being extracted. However, in general, there are two types of fields that can be transferred to the flat file:

- **Columns:** Many dimensional extracts allow predefined columns to be populated on the flat file. For example, you can set up the premise extract program to populate the first User-Defined Field (UDF) on its flat file with the premise's city, county, or any address-oriented column. An analogous concept is used to populate the User-Defined Measure (UDM) on the fact extracts.
- **Characteristics:** Many dimensional extracts allow characteristics to be populated on the flat file. For example, you can set up the premise extract program to populate the first User-Defined Field (UDF) on its flat file with the premise's tax characteristic, or/and premise-oriented characteristic.

**Note:** Most dimensional extracts support the population of their User-Defined Fields with characteristic values. A limited number of fact extracts allow characteristics to be used to populate User-Defined Measures. This is because most of the transaction files that trigger the fact extracts do not contain characteristics.

You can identify how an extract populates its User-Defined Fields and User-Defined Measures by populating parameters. Each User-Defined Field/ User-

Defined Measure supported by an extract has two parameters that must be populated:

- **Type:** This parameter is defined or populated if the field is a true column or a characteristic. Enter PROG if you want to populate a User-Defined Field (UDF) / User-Defined Measure (UDM) with a column. Enter CHAR if you want to populate the User-Defined Field (UDF) / User-Defined Measure (UDM) with a characteristic.
- **Value:** This parameter defines the respective column or characteristic.  
To define a column, the value should be in the formation Table.Column (for example, CI\_PREM.CITY\_UPR - would be used on the address extract to populate a UDF with the upper-case format of an address's city).  
To define a characteristic, enter the characteristic's type code. Note, as of now, in the current release only pre-defined value characteristics are supported.

**Note:** For details, refer to the relevant fact and dimension chapter in this guide for a description of each extract program and the various User-Defined Field (UDF) and User-Defined Measure (UDM) parameters that are supported in each.

### Extracting Additional Fields

While the extract programs are delivered to populate their flat files with commonly used information, you may want to populate the User Defined Field (UDF) and User Defined Measure (UDM) with information not supported by the base-package.

Refer to **Appendix C** and **Appendix D** for complete details regarding Batch Control and Trigger Names respectively while trying to customize the extracts.

**Note:** For additional details, refer to the Knowledge Article Extending the OUASA Data Warehouse (Using User Defined Constructs in Oracle Utilities Customer Care and Billing (CC&B) (Doc ID 1516838.1) available at the My Oracle Support website (<https://support.oracle.com/>)

## Oracle Utilities Meter Data Management and Mobile Workforce Management

Oracle Utilities Meter Data Management (MDM) and Oracle Utilities Mobile Workforce Management (MWM) generally use the Sync Business Object (BO)-based approach to detect the change in the source table that needs to be populated in the data warehouse. In some cases, where there are complex logic to perform the extract, there are idiosyncratic batch processes that does the job.

Both of these approaches support the population of User-Defined Field (UDF) and User-Defined Measure (UDM) on their flat file records with specific fields from the source system. For example, you can set up the metadata in the source system to populate the first User-Defined Field on a Service Point Dimension's flat file with the time zone or any Service Point-oriented field.

A particular fact or dimension may fall into one of these two extraction styles, both of them have their own ways of being extended. In both the methods, once the metadata has been configured appropriately, the respective batch controls have to be run again to generate the new flat files with the User-Defined Field (UDF)/User-Defined Measure (UDM) columns populated.

### Synchronization BO-Based Extract

Each fact and dimension that use the synchronization Business Object (BO) approach for extract has its own Sync Request BO. Each of these business objects have the following options:

- **Snapshot DA:** This is the data area which contains the elements sent to the data warehouse. The elements in this Data Area (DA) are listed in order in which the ETL expects them.
- **BO to Read:** This is the primary Business Object (BO). The contents of this business object are sent to the data warehouse. Its elements should match those in the Snapshot DA as it will be read and then its contents "moved by name" to the Snapshot DA.

- **Element Population Rule:** These options are used to populate an element on the Snapshot DA that cannot be "moved by name" from the BO to Read. The option value contains the following mnemonic:

| Option Type             | Sequence                   | Option Value                                                                      |
|-------------------------|----------------------------|-----------------------------------------------------------------------------------|
| Element Population Rule | <A unique sequence number> | sourceElement=XXX<br>populateTargetWithInfoString=true/false<br>targetElement=XXX |

The sourceElement attribute refers to an element in the Business Object (BO) specified in the option type 'BO to Read'. The targetElement attribute refers to an element in the Snapshot DA that needs to be populated. The populateTargetWithInfoString accepts a 'true' or 'false' value (a value of 'true' means that the sourceElement's Foreign Key (FK) reference in the BO to Read schema will be used to retrieve the element's information string).

- **Post Service Script:** This is the script which is executed after the Snapshot DA is populated with the BO to Read elements and the Element Population Rules (if any). It is responsible for populating elements on the Snapshot DA that cannot be derived directly from the BO to Read.  
You can extend the facts and dimensions by adding new options:
- **Element Population Rule:** To extend the Element Population Rules, you should add new Element Population Rule options following the mnemonic specified above. You can set up as many Element Population Rules as you need, and the sequences do not matter.
- **Post Service Script:** To extend the Post Service Script, you should create a new service script.

| Option Type                     | Sequence                   | Option Value               |
|---------------------------------|----------------------------|----------------------------|
| Post Service Script for Extract | <A unique sequence number> | Custom Service Script Name |

The option value to be supplied is the name of the customer service script which has the logic to populate the User-Defined Field (UDF) / User-Defined Measure (UDM) columns with the desired values. There are few things to note here. First, the schema of the service script should match the schema of the Snapshot Data Area used in the Sync Request BO (this can be done by including the Snapshot DA specified on the sync request BO's option in the Post Service Script's schema). Second, the Post Service Script with the highest sequence number will be the one executed. Hence, make sure to provide the appropriate sequence number on your custom service script. And lastly, because the highest-sequenced Post Service Script will override any existing scripts, if MDM or MWM has delivered a Post Service Script, your new custom service script should either call the existing Post Service Script (and avoid duplicating the same steps in your custom script), or not, depending on what you want to populate.

## Idiosyncratic Batch Extract

For snapshot facts and certain dimensions, there are idiosyncratic batch jobs that were created to handle complex extraction logic.

- **Batch Parameters:** Certain batch jobs delivered as part of the source system allows additional batch parameters for the end users to supply custom service script. The service script must include the same Snapshot DA defined as batch parameter on the batch job.
- **Algorithm Soft parameters:** Certain algorithms delivered as part of the source system allows additional soft parameters for the end users to supply custom element population rules and



service script. The usage of these parameter values is similar as explained for the 'Element Population Rule' and 'Post Service Script' above in the Synchronization BO-based style. The service script must include the same Snapshot DA defined as soft parameter on the algorithm.

- **New Algorithm:** In case where the user needs to drastically change the logic of populating the UDF/UDM columns or even the remaining set of fields in the flat files, users may wish to create a new algorithm and plug it in on the originating Business Object (BO). This algorithm should be plugged in with a higher sequence than the base product algorithm. Additional care will have to be taken since this will completely override the existing algorithm logic and this new algorithm will have to supply the logic of populating the entire flat file.

### Extracting Additional Fields

While the extract programs are delivered to populate their flat files with commonly used information, you may want to populate the User Defined Field (UDF) and User Defined Measure (UDM) with information not supported by the base-package. To do this, customers must use the "Sync BO Options" to the respective extract objects.

Refer to **Appendix E** and **Appendix F** for complete details regarding Batch Control and Sync BO Names respectively while trying to customize the extracts.

**Note:** For additional details refer to the Knowledge Article Extending the OUASA Data Warehouse (Using Sync BO in Oracle Utilities Meter Data Management) (Doc ID 1516859.1) available at the My Oracle Support website.

### Oracle Utilities Network Management System (NMS)

NMS uses the view-based approach to detect the change in the source table that needs to be populated in the data warehouse.

For any changes required in Extracts like populating the User-Defined Field and User-Defined Measure with new column values, follow the below approach:

- Create new modify and delete views that have the business logic to extract the required data.
- Create a new extract program/procedure to access the view from above step. Ensure that the existing views/scripts are not updated as it might lead to upgrade impacts.
- Run the new extract program to retrieve data into flat files.

Refer to **Appendix G** for complete details regarding extract program and view names for customizing the extracts.

### Configuring UDD Tables

If customers have a requirement to "slice and filter" the existing fact data using an additional group of related attributes, then they can opt to make use of the User Defined Dimension tables. For each fact two User-Defined Dimensions are provided. These tables can be utilized for extending the star schemas.

If you want to offer this option to your users, you must introduce these tables as additional dimensions to the existing fact (in addition, you must change the fact's extract program to extract as described in the earlier sections at this new level of granularity).

The following points below summarize how to set up a new UDD:

- You must create database trigger(s)/view(s) or sync Business Objects (BO) to cause new and changed dimensions to be interfaced to the data warehouse. There are many examples of dimensional triggers in the operating system that can be used as samples for the new triggers.
- You must create a new program to extract the new dimension's values. This extract will be executed in the operational system and will produce a flat-file containing the dimension's values. There are many examples of dimensional extract programs in the operating system that can be used as a basis of your new program.

- The flat-file produced by the extract is the input to the Oracle Warehouse Builder (OWB). Oracle Warehouse Builder is delivered preconfigured to load the data warehouse from the flat-file
- Run the new extract program in "extract everything" mode and let the Oracle Warehouse Builder populate the dimension's rows.
- Return to Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) and display the User-Defined Dimension (UDD) table by updating the Oracle Business Intelligence Enterprise Edition (OBIEE) .rpd file. Enter the appropriate Override Label of each User-Defined Fields (UDFs) on the table. Note these are the dimensional attributes that is used to slice and filter the dimension's facts. For example, if the dimension is meant to hold General Ledger (GL) accounts, it makes sense to define at least two User-Defined Fields as shown below:
  - The General Ledger (GL) account number
  - The General Ledger (GL) account type (for example, revenue, expense, tax etc.)
  - Transfer to the operating system (for example, Oracle Utilities Customer Care & Billing) and introduce user-exit code to the extract program to the appropriate User-Defined Dimension (UDD) values for the fact.

Refer to the source applications sections above for the various approaches for more information about the extract programs.

When you extract the facts after this point, the flat-file supplied to Oracle Warehouse Builder (OWB) will be populated with the appropriate references to the User-Defined Dimension (UDD).

## Oracle Data Integrator Based Extension

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) provides placeholder columns for user extension to out of the box star schemas. We have also provided a methodology that makes it easier for you to add logic to populate these user defined columns. Refer to the section **Extending Start Schemas** in this chapter for details which columns can be customized and what is the purpose of these columns.

To start populating the additional user defined columns, you should identify the source table which is used to populate the target column. If the table is not in the list, follow the steps in the section below **"Extending Replication"** to add this table for replication.

The next step is to write a user extension procedure that can populate these columns. The interfaces have been created so that they perform additional actions like translating the source foreign keys to appropriate dimension keys.

### Extending Replication

Out of the box Oracle Utilities Advanced Spatial and Operational Analytics ships a required set of tables marked for replication. For extensibility purposes, additional tables may be configured to be replicated. To configure additional tables for replication, follow the steps below:

#### Configuring Additional Tables for Replication

1. Check if the table to be replicated is listed in B1\_SOURCE\_TABLE
2. If the table is not listed in the table, add an entry in B1\_OBJECT\_MAP setting the SOURCE\_OBJECT\_NAME as the table name and the TARGET\_OBJECT\_NAME as the target fact or dimension which will have some attributes loaded from this table.
3. The SOURCE\_OBJECT\_NAME can also be set to an Maintenance Object (MO) or Business Object (BO) in which case all tables comprising the Maintenance Object (MO) is pulled in.

4. Execute the scenario B1\_OUAF\_CFG\_METADATA. Now an entry for this table will be present in the B1\_SOURCE\_TABLE
5. Set the CM\_REPLICATE\_FLAG to 'Y'
6. Execute the initiateSetup.cmd which will create a new model and the steps required for the replication process to be initiated.  
For details refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide* and check the Initiate Setup section under chapter Installation.)
7. Follow the instructions to set up the Golden Gate scripts generated to start.

## Populating User Defined Columns

Extending the data load process for populating the user defined columns (UDF, UDM, UDDGEN & UDDFK) consists of the following steps.

1. Create a user exit procedure for the target entity. The recommended naming convention is CM\_UDX\_<entity\_name>.
2. The User Exit procedure signature should support three parameters. This is to ensure that the user exit procedure works on a specific data set only. The user exist procedure should update the user defined columns in the UDX table.
  - **V\_SESS\_NO**: The session number for processing. All SQL statements processing on the UDX table should have a filter on the session number.
  - **V\_SLICE\_BEG\_TS**: This is the slice begin timestamp. To be used for filtering data for the particular slice or snapshot. Data processed should be greater than or equal to this value.
  - **V\_SLICE\_END\_TS**: This is the slice End timestamp. To be used for filtering data for the particular slice or snapshot. Data processed should be less than this value.

Refer to section **Job Configuration** in **Appendix B** to set the user exit procedure name.

For Example:

Below is a template code for extending an entity.

```
CREATE OR REPLACE CM_UDX_<entity_name>(
 v_sess_no IN NUMBER
, v_slice_beg_ts IN DATE
, v_slice_end_ts IN DATE)
AS
BEGIN
 UPDATE UDX_<entity_name> udx
 SET udf1 = (SELECT <some_column>
 FROM <TABLE_A> src
 WHERE src.<nk1> = udx.<nk1>
 AND src.<trx_date> >= v_slice_beg_ts
 AND src.<trx_date> < v_slice_end_ts)
 WHERE udx.sess_no = v_sess_no ;
END;
```

When writing the code for extending any target entity, take care of the following:

- **<entity\_name>**: replace with the appropriate target entity name, for example CF\_OPR\_DEVICE.
- **<TABLE\_A>**: replace with the source table or tables that are required for fetching the source data
- **<some\_column>**: the source column to be used for populating the user defined column
- **<nk1>** : In the case of CF\_OPR\_DEVICE the natural keys are SRC\_ASSET\_ID & DATA\_SOURCE\_IND.

- **<trx\_date>** : typically this is the update\_dttm column of the source table, however depending on functionality this can be a transactional date column.

Note that the above is just an example on how you can select a value from one of the replicated columns. You can replace this logic by substituting code of any complexity as long as it returns the above mentioned parameters.

## Customizing Analytics

This section describes how to use Oracle Business Intelligence Enterprise Edition (OBIEE) to customize Oracle Utilities Advanced Spatial and Operational Analytics. It includes the following topics:

- **Modifying the RPD file**
- **Customizing Answers**
- **Customizing Report Labels**

### Modifying the RPD file

All customer modifications must be done in a separate repository file which is separate from the product's out-of-the-box repository. Any customization done is merged into the upgraded repository file through the Merge utility of Oracle Business Intelligence Enterprise Edition (OBIEE) along with the product's out-of-the-box repository file.

Oracle recommends that you use a staging environment for the repository upgrade. However, as long as customer modifications are done on top of a copy of the base repository file, the OBIEE upgrade process is able to handle most customizations that may be made to the repository file. The simpler the changes, the less complex upgrade procedure; hence, you should try to limit the changes made to the repository file.

For information about managing, upgrading, and merging repository (.rpd) files, refer to *Oracle Business Intelligence Server Administration Guide*.

### Customizing Answers

All customer modifications must be done in a separate folder in order to preserve these modifications for upgrading the Oracle Utilities Advanced Spatial and Operational Analytics. If an existing answer needs to be changed to meet your requirements, a copy of the product report should be created, and changes should be made to the copy (not to the original report). The dashboard should be changed to point or refer to the new copy instead.

Note that dashboards are overwritten during the upgrade and any mappings between dashboards and customized answers are lost and must be re-mapped manually. Therefore, you should use a staging environment for upgrade and manually remap dashboards before moving the upgraded customized content into the production environment.

For more details about managing, upgrading, and merging presentation catalogs, please refer to Oracle Business Intelligence Presentation Services Administration Guide.

For more details on how to create or edit answers, please refer to Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition.

### Customizing Report Labels

Customers can choose to opt for a different label or caption for an existing report or any of the report columns in it. They can provide an override description instead of the product provided description. These details can be provided in the 'Base Field Maintenance Page', under the Administration Dashboard in the OBIEE Dashboards menu. Once the changes are saved and the

cache clear, upon the next login, the new descriptions can be seen on the report title or the column title.

For more details, refer to the '**Administration Dashboards**' section in **Chapter 4: Configuring Oracle Utilities Advanced Spatial and Operational Analytics**.

# Chapter 6

---

## Adding New Components

Note that this chapter is primarily for the source applications that are based on the Oracle Warehouse Builder (OWB) ETL methodology.

### Creating a New Star Schema

The star schema is perhaps the simplest data warehouse schema. It is called a star schema as the entity-relationship diagram of this schema resembles a star with points radiating from a central table. The center of the star consists of a large fact table. The end points of the star are the dimension tables.

A star query is a join between a fact table and a number of dimension tables. Each dimension table is joined to the fact table using a primary key to foreign key join; however, the dimension tables are not joined to each other. The optimizer recognizes star queries and generates efficient execution plans. It is not mandatory to have any foreign keys on the fact table for star transformation to take effect.

A typical fact table contains keys and measures. A star join is a primary key to foreign key join of the dimension tables to a fact table. The main advantages of a star schema are as follows:

- It provides a direct and intuitive mapping between the business entities analyzed by end users and the schema design.
- It provides highly-optimized performance for typical star queries.
- It is widely supported by a large number of business intelligence tools, which may anticipate or even require that the data warehouse schema contain dimension tables.

Star schemas are used for both simple data marts as well as very large data warehouses. Once the model has been designed, the Oracle Warehouse Builder (OWB) code generator can be utilized to generate the mappings and process flows.

For details regarding data modeling, refer to Oracle® Database Data Warehousing Guide 11g Release 2 (Chapter 19 Schema Modeling Techniques).

### Defining ETL for New Start Schema

#### Creating New Extractors

Since different source applications employ different approaches to extract their data, the sequence of steps that will have to be done for schema extension vary on the extraction approach as well. Once the extractors have been configured to pull the additional data, the OWB ETL delivered along with the Oracle Utilities Advanced Spatial and Operational Analytics product automatically takes care of loading these additional data into the warehouse.

- The following sections cover the necessary steps to be done for each of the source applications.

### Oracle Utilities Customer Care and Billing and Oracle Utilities Work and Asset Management

The source applications Oracle Utilities Customer Care and Billing (CC&B) and Oracle Utilities Work and Asset Management (WAM) use the trigger-based approach to detect the change in the source table that needs to be populated in the data warehouse.

To populate the new facts/dimensions, the following approach is to be followed:

1. Create new triggers that have the business logic to extract the required data from the new tables.
2. Create a new extract program/procedure to with the desired extraction logic from the change log tables.
3. Create new batch controls for the extract programs.
4. Run the new batch controls to extract data into flat files.

Refer to **Appendix C** and **Appendix D** for existing Batch Control and Trigger Names while trying to create the new extracts.

For additional details, refer to the Knowledge Article Extending the OUASA Data Warehouse (Using User Defined Constructs in CC&B) (Doc ID 1516838.1) available at the My Oracle Support website.

### Oracle Utilities Meter Data Management and Mobile Workforce Management

The source applications of Oracle Utilities Meter Data Management (MDM) and Oracle Utilities Mobile Workforce Management (MWM) make use of the Sync Business Object (BO) based approach for data extraction. For all the facts and dimensions other than those of snapshot type and certain dimensions with complex extract logic, it is advisable to use the Synchronization BO mechanism provided by the Oracle Utilities Application Framework (OUAF). There are two options to extract new Dimension or Fact data

Perform the following steps to create the new synchronization request Business Object (BO):

#### Creating New Synchronization Request Business Object:

1. Copy an existing Sync Request BO.
2. Modify the following BO Options in the newly created Sync Request BO:
  - **BO to Read:** This needs to be populated with the BO which has elements that need to be extracted.
  - **Snapshot Data Area:** This defines the schema which will be exactly extracted in the flat file, including the order of elements and the element types.
  - **Post Service Script:** If you want to extract data which is not available in the schema BO to Read BO, you will have to write a processing script to extract such elements.
  - **Element Population Rule:** If you want to move data from an element defined on BO to Read BO to another element defined in the Snapshot Data Area (DA), you can add the Element Population Rule.
  - **Batch for Extract:** This is the name of the batch that needs to be executed to extract the flat files. It is recommended that you create a new extract batch control for your new fact or dimension so you can define and default the parameters specific to your fact or dimension, but you should be able to use the existing extract java batch program.
  - **Star Schema Type:** Mention whether this Business Object (BO) is for a Fact or Dimension.

3. To handle initial synchronization, it is recommended that you create a new initial load batch control for your new fact or dimension if there is no existing one for the Maintenance Object (MO) of your ETL source. If there is an existing initial load batch control for the MO, add the newly created Sync Request BO as additional Sync Request BO batch parameter.
4. To handle incremental sync, create an Audit algorithm to define the logic to control the creation of pending Sync Request record. This should be plugged in on the MO of your ETL source. Refer to an existing audit algorithm delivered in the edge application. Depending on the audit algorithm created above, add the newly created Sync Request BO as an option on the MO of your ETL source.

For snapshot type facts and certain dimensions with complex extraction logic, you can create an idiosyncratic java batch program to extract the data without using the Sync Request BO.

### Creating an Idiosyncratic Java Batch Program to Extract Data

Perform the following steps to create such program:

1. Define the Data Area (DA) that reflects the structure of the flat file, including the data type and order of elements.]
2. Write the batch extract program that retrieves each record that needs to be extracted and performs the following:
  - Populate the Data Area with the information to be extracted as appropriate.
  - Invoke the BusinessService "F1-ConvertXMLToFileFormat" to transform Data Area into the fixed length string. This string would be written to the extract file when the program is executed.
  - Write the record to the extract flat file.
3. Create new batch control for the new extract program

Refer to **Appendix E** and **Appendix F** for details regarding existing Batch Control and Sync Business Object (BO) Names respectively while trying to create the new extracts.

For additional details refer to the Knowledge Article Extending the OUASA Data Warehouse (Using Sync BO in Oracle Utilities Meter Data Management) (Doc ID 1516859.1) available at the My Oracle Support website (<https://support.oracle.com/>).

### Oracle Utilities Network Management System

Oracle Utilities Network Management System (NMS) uses the view-based approach to detect the change in the source table that needs to be populated in the data warehouse.

To populate the new facts / dimensions, the following approach is to be followed:

1. Create new modify and delete views that have the business logic to extract the required data.
2. Create a new extract program/procedure to access the view from above step.
3. Run the new extract program to retrieve data into flat files.

Refer to **Appendix G** for details regarding Extract Program and View Names while try to create the extracts.

## Creating New Oracle Warehouse Builder Workflows

The steps below outline the high level steps to be carried out for creating the Oracle Warehouse Builder (OWB) ETL code:

1. Create a fact or dimension.
2. Specify the external table name.
3. Map a column from external table to target.



4. Specify the join conditions.
5. Generate the Oracle Warehouse Builder (OWB) code. For specific details refer to Appendix J: OWB Deployment

## Auto Cache Refresh

OBIEE provides a mechanism called Event Polling, which allows OBIEE to query a Database Table to find out when data has been updated in fact or dimension tables. By modifying the OWB load process to populate an Event Polling table, we can let OBIEE know when data has been updated in the data warehouse, and enable OBIEE to know when to refresh the cache data that has been queried before.

A new event polling table has been made available as part of Oracle Utilities Advanced Spatial and Operational Analytics B1\_OBIEE\_EVENT\_POLLING.

The use of an Oracle BI Server event polling table (event table) is a way to notify the Oracle BI Server that one or more physical tables have been updated and then that the query cache entries are stale.

Each row that is added to an event table describes a single update event, such as an update occurring to a Product table.

The Oracle BI Server cache system reads rows from, or polls, the event table, extracts the physical table information from the rows, and purges stale cache entries that reference those physical tables.

For new requirements, new extractors will be created along with new OWB load processes to load data into the new fact or dimension tables. Here in order to ensure that the OBIEE cache data is automatically refreshed, an additional step has to be included in the OWB process flow to ensure that an entry is made in the available event polling table B1\_OBIEE\_EVENT\_POLLING.

This will ensure that whenever new data is loaded by the OWB process flow, the OBIEE cache is automatically refreshed and made available for the analytics reports.

Refer to an existing base product supplied OWB process flow for samples.

## ODI Based ELT for New Objects

Users attempting to utilize ODI for extending the star schemas should be familiar with using ODI for development activities. The following sections list out the activities that need to be performed to add new facts and dimensions using ODI.

1. All custom interfaces, packages or procedures should be created in a separate folder under the main project folder.
2. All custom models should be created under a separate folder in the models section.

## Define and Configure

Once you have designed the new facts and dimensions you need to follow the following steps to start your development.

1. Create the table structures in DWADM schema. Recommended practice is to prefix customer owned objects with CM\_.
2. Use Oracle Business Intelligence for Utilities Administration to create entries in Target Entity and Job Configuration for the new objects that you have created. Refer to **Appendix B** for further details.

## Set Up Replication

While developing the requirements for a new star schema objects, identify the source entities that will be required to populate the target entities. Using the admin UI check if these entities are available in the table B1\_SOURCE\_TABLES. If yes follow the steps mentioned in the section "Extending Replication" to enable replication for additional tables.

Assuming that the source system maintenance objects or specific tables that are sources for your new interfaces are not available in the table B1\_SOURCE\_TABLES, follow the steps below:

1. In the current release the metadata table B1\_OBJECT\_MAP is not exposed via the admin UI. You should not modify any entries already present in this table. For a new MO or Table source you will need to create a new entry mapping your source MO/Table to the target entity.
2. Once configured, execute the package B1\_OUAF\_CFG\_METATADA. B1\_OUAF\_CFG\_METATADA is an implementation accelerator for Oracle Utilities Application Framework products, and it pulls additional information for these objects into the metadata tables. This package is automatically executed during the addition of a new instance as source for Oracle Utilities Business Intelligence. Refer to **Appendix B** for a list of the metadata tables and the purpose of each of these metadata.
3. Follow the steps for extending replication to enable these new entities to be replicated

## Model for New Facts

Perform the following:

1. Create a new model for facts.
2. Set the logical connection as "Target".
3. Reverse engineer your newly created facts into this model.

## Model for New Dimensions

Perform the following:

1. Create a new model for dimensions.
2. Set the logical connection as "Target".
3. Select the custom RKM Oracle (BI) for reverse engineering the dimensions. You can set the prefix used for the effective date columns in Type II dimensions.

The custom RKM auto configures type I & II dimension configurations based on unique and primary keys on the database tables. The dimension should have a primary key constraint on the surrogate key column and a unique key on the natural key.

## Model for Replicated Objects

Once the required objects have been replicated follow the instructions below to set up a source model for the replicated tables.

1. Create a new model, set the logical schema to "Replication".
2. Reverse engineer all objects you need as source for your target entities.

## New Interface

To create a new interface for your target entities, follow the steps below:

1. Under your custom folder right click -> new interface

2. Drag and drop your target table to the target section of the interface.
3. Drag and drop your source tables (from the replication model) into the source section of your interface.
4. Set up the appropriate join conditions between the tables.
5. Map the columns from source to target as per your design.
6. Add new filter.
  - Add the condition on slicing\_ts column using the variables, as below
 

```
(src.slicing_ts >=
to_date('#B1_SLICE_BEG_TS', 'yyyymmddhh24miss')
(src.slicing_ts <
to_date('#B1_SLICE_END_TS', 'yyyymmddhh24miss'))
```
  - For multiple tables add an "OR" condition to ensure that all changed records are picked up even if the change occurred in only one table.
7. Mark the columns other than the PK & UK columns for update and all columns for insert.
8. In the Flow tab select the appropriate KM based on the target entity. KM's supplied with the product should be used and they have been named so that it is easy to identify the fact/ dimension KM's to be used.
9. Save the interface.

The following expression should be used for mapping the target dimension key columns:

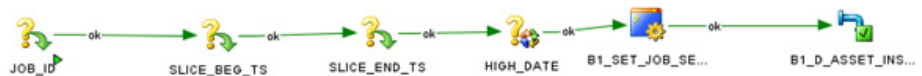
```
(CASE WHEN SRC.<src_nk> IS NULL
 THEN #B1_DEF_NULL_KEY
 WHEN <dim>.<dim_pk> IS NULL
 THEN #B1_DEF_MISSING_KEY
 ELSE <dim>.<dim_pk>
 END)
```

In the sample code above SRC is the alias of the primary source table that contains the source foreign key "<src\_nk>" to the dimension. <dim> is the alias for the dimension and <dim\_pk> is the primary key of the dimension.

## New Package

A new package has to be created for each target entity. The easiest way to create a package for your entity is to copy an existing package based on the target object type and replace the interface with your new interface. While copying the package remember to move it to the custom folder.

Below is a sample package for a type II dimension.



Below is a screenshot for a fact package.



Once the package has been created, right click and generate the scenario. Ensure that the new package name has been configured appropriately in the Job configuration section using Admin UI.

## Creating New Analytics

This section describes how to use Oracle Business Intelligence Enterprise Edition (OBIEE) to add new analytics Oracle Utilities Advanced Spatial and Operational Analytics. It includes the following topics:

- **Creating New Answers**
- **Adding New Labels**

### Creating New Answers

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) provides out-of-the-box dashboards with rich and varied set of analytics for Credit & Collection Analytics, Customer Analytics, Distribution Analytics, Meter Data Analytics, Mobile Workforce Analytics, Outage Analytics, Revenue Analytics, Exception Analytics, and Work & Assets Analytics. However, if required, customers can create new answers or dashboards.

As noted in the above section regarding customization of existing answers, new answers should also be saved in a separate folder so that they are not overwritten while upgrading the Oracle Utilities Advance Spatial and Operational Analytics (OUASA).

A customer implementation can create field labels for use in their answers or the labels can just be created directly in the answer, if there is no multilingual/localization requirements. If product labels are used in an answer, they can be modified during an upgrade (unless you have entered an override label). At best, it is recommended to limit the changes to the existing labels; however, there can be certain situations, when they are updated.

### Adding New Labels

If an implementation wants to use the label mechanism for new answers, then the Custom Field Maintenance dashboard can be used to add, update, and delete custom labels. These custom labels can then be used in answers as well as in the logical and physical objects in the repository or RPD file.

Note: Only custom field labels, identified by a Customer Modification (CM) owner flag, can be updated or deleted. New labels are created with a Customer Modification (CM) owner flag. A label that already exists cannot be created, so if a base labels already exists, an implementation can update the override label as described in the preceding section Creating New Answers.

For more details refer to the '**Administration Dashboards**' section in **Chapter 4**.

# Chapter 7

---

## Maintaining Environments

This section describes how to maintain your Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) environments, including moving code changes from one environment to another.

This section includes the following topics:

- **Overview of Environment Maintenance**
- **Moving Oracle Warehouse Builder based Code**
- **Database Cloning for Oracle Warehouse Builder**

### Overview of Environment Maintenance

You should implement processes to maintain code control over various environments. The following components of the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) should be handled separately.

- OBIEE Web Catalog
- OBIEE Repository File
- Field Label Metadata
- OWB Repository
- OUAF Metadata (only required if there are OWB customizations)
- Mapviewer Spatial Data

Assuming that custom changes are made to any of these objects, then a mechanism must be put in place to develop, test, and move these customizations to the production environment.

### Moving Oracle Warehouse Builder based Code

During the development phase of coding, you usually do not need to move code from a development environment to any other environments. However, once a project moves to the Quality Analyst (QA) or production phases, code changes may be need to be migrated.

For example, in an internal development process, there may be two Development environments, one for Oracle Business Intelligence Enterprise Edition (OBIEE) Dashboard creation and one for the Oracle Warehouse Builder (OWB) Development. You can build a QA environment from scratch, that is, an empty database is created, the Oracle Warehouse Builder (OWB) objects are imported and deployed, and the OBIEE web catalog and repository file are created fresh.

To do this, use the installation process for creating an empty Oracle database, Oracle Warehouse Builder repository, and WebLogic environment. Then, use the OWB Export process to create two

MDL files from the Development environment, one for the locations, and one for all of the other OWB objects. You should then copy the OBIEE Repository file, and use the OBIEE export process to create the OBIEE web catalog files.

Once you have these files, follow the install process to load the MDL files in the OWB repository, copy the OBIEE Repository file into the Quality Analysis (QA) WebLogic environment, update the user name, password, and database connections for the Quality Analysis databases, and import the web catalog export files into the same location as they were in the development environment.

This process works well for a development move to QA, but will not work once a system goes into production, because parts of this process require the creation of an empty database, which is not something that should be done in a production environment.

In situations, where bug fixes have been made in a development environment and they need to be moved to a production environment, you can to export the entire OWB repository and OBIEE web catalog and replace this in the production environment. Use this method to move the code if it is not known exactly which objects have changed.

For OWB though, if the modified objects are known, then it is possible to export only the changed objects, import them in the QA environment, and then, once QA is successful, do the same import process into the production environment. Another option is to save the TCL files that were created by the OWB Code Generator, and then load them into the QA and production environments.

### Creating an MDL File

To create an MDL file for a known set of OWB objects, follow these steps:

1. Log on to the Workflow Development Database using Design Center as Repository Owner (BIREPOWN).
2. Review the modified the Oracle Warehouse Builder (OWB) objects and then, select the OWB objects:
3. Export them by navigating to **Design > WareHouse Builder Metadata**.
4. Review the path for MDL and log file and select Export all dependencies.
5. Click **Export**.
6. Use the created MDL file to move objects from a Development environment to a QA or Production environment.

This process assumes that database changes are handled outside of the Oracle Warehouse Builder (OWB). Hence, new tables, modifications to existing tables, or materialized view log changes should be handled via SQL scripts created by the Development team.

For Oracle Business Intelligence Enterprise Edition (OBIEE), a full move is not recommended; there are ways of merging code in the web catalog and also in the repository file. These methods are documented in the Oracle Business Intelligence Enterprise Edition (OBIEE) Administration guide.

Refer to Chapter 24, "Moving Between Environments," in the System Administrator's Guide for Oracle Business Intelligence Enterprise Edition 11g Release 1. This chapter shows how you can Maintaining Environments 8-3 move Oracle Business Intelligence to a new environment or from a test to a production environment.

For moving modified override labels from development to QA to production, for MapViewer data, it should be added to QA or production environments the same way that it is added to development. If a shapelier is downloaded and added to the Development environment, the same shapelier should be added to the QA and production environments.

Finally, it is very important that the process of moving code from development to QA is exactly the same as the process that moves the code to production. It is only by following the same sequence of steps in both cases that the movement process is also tested. If one process is

followed to move code to QA and another process is followed to move code to Production, then problems can arise in the move to production that were not seen in the move to QA.

## Database Cloning for Oracle Warehouse Builder

You can use database cloning to move the entire development database to a QA environment or a new production environment. This cannot be used to move a development database to an existing production environment, as the existing production database tables will be overwritten. But for QA purposes, this can move the development database quicker than a fresh install. Note that this does not move the OBIEE objects, but does handle all of the Oracle Warehouse Builder (OWB) code, field label changes, Marvin mediated, and any new database objects.

### Cloning a Development Database

To clone a development database, follow these steps:

1. Clone the existing database.
2. Go to \$ORACLE\_HOME/owe/Underpays directory.
3. Execute reset\_owbcc\_home.sql with OWBSYS user and remote\_owb\_install.sql scripts with says user.
4. Create the password file for database if not exists.
5. Go to \$ORACLE\_HOME/owe/reposess/up directory and execute upg112to11203.sql as scalpels /no-load @upg112to11203.sql if database is cloned from lower Oracle Home version to Oracle 11.2.0.3 version home and this script is already not executed.
6. Please wait till execution of this script. It will take time.
7. connect as OWBSYS user and execute
  - @\$ORACLE\_HOME/owb/reposasst/secHelper.pks
  - @\$ORACLE\_HOME/owb/reposasst/secHelper.plb
  - ORACLE\_HOME/owb/reposasst/upg/load\_java.sql OWBSYS  
<OWBSYS\_PASSWORD>
8. Navigate to \$ORACLE\_HOME/owb/bin/admin and rename the rtrepos.properties.
9. Connect as OWBSYS user and execute @\$ORACLE\_HOME/owb/rtp/sql/reset\_repository.sql.
10. Submit the **SELECT \* FROM OWBRTPS** query.  
It shows updated oracle home in value column.
11. Submit select SERVER\_SIDE\_HOME from WB\_RT\_SERVICE\_NODES query It will show updated oracle home.
12. Log into to design repository and get name of all the control center with which locations are registered.
13. Connect with OWBSYS and execute UpdateControlCenter.sql for all control centers with which locations are registered. Provide below inputs:

Enter Workspace Name: SPLBIREP

Enter Workspace User Name: BIREPOWN

Enter Control Center Name: <Control Center Name>

Host: <Hostname>

Port: 1521

Service Name: <DB name>

New Net Service Name: <DB name>

14. Run @UpdateLocation.sql for all location. Make sure to provide correct version.

For example:

Enter Workspace Name: SPLBIREP

Enter Workspace User Name: BIREPOWN

Enter Location Name: SPL\_BI\_TGT\_LOC

New Host: <Hostname>

New Port: 1521

New Service Name: <DB name>

New Net Service Name: <DB name>

New Version: 11.2

Please provide version 0 for SPL\_BI\_FF\_LOC and SPL\_BI\_LOG\_LOC locations and 2.6.4 for SPL\_BI\_WF\_LOC location.

You can also use below block to update Workflow location.

```

Declare
v_result boolean := FALSE;
Begin
v_result := wb_rt_reset_location.fixDTLocation
('SPL_BI_WF_LOC', '<Hostname>', '1521', '<DB name>', '2.6.4', '<WF
user>', '<WF user's password>');
END;
/

```

15. EXECUTE UPDATE WB\_RT\_SERVICE\_NODES SET  
CONNECT\_SPEC='<hostname>:1521:<DB name>'; commit;
16. Connect to control center manager in design repository .
17. Register all locations.
18. Unregister all location.
19. SAVE ALL OBJECTS. and exit from control center manager
20. Double click the control center and move all selected location to available locations.  
Click **OK**.
21. Rename the control center as required.
22. Set the other control center, with which locations are registered as default control center for  
default\_configuration and connect to control center.
23. Register all locations.
24. Unregister all locations. SAVE ALL OBJECTS. and exit from control center.
25. Double click the control center and move all selected location to available locations.  
Click **OK**.
26. Rename the control center as required
27. Repeat steps from 22 to 26 for all registered control center.
28. Doing so we are removing all control center info from registration tab of all locations.
29. If you are not removing control center dependency, you are able to register location with any  
control center but not able to update location info after you unregister the location.



30. Move the locations from available section to selected location in required control center and set the required control center as default control center for default\_configuration and log in to control center.
31. Now all location are available for update and can be registered followed by objects deployment.

## Oracle Data Integrator Based Products

### Moving Oracle Data Integrator Repository

To move an ODI repository perform the following steps.

1. Log into ODI client using a supervisor user.
2. Export the master and work repository using the ODI client.
3. Log out of the ODI client.
4. Use the master repository creation wizard and create a new repository if it does not exist.
5. Log into the new master repository and import the xml's exported earlier.
6. Modify connections in the topology section as required.
7. Delete/Drop the older ODI master and work repository schemes or alternatively delete all connections, users from the Topology & Security sections respectively. This is a precautionary action to prevent any users from accidentally logging into the older repository and executing jobs.

### Moving Metadata

The metadata objects reside in the DWADM schema by default. In case you have made some configuration changes and want to change to a different database, you will need to export the metadata objects and import them into the new environment.

### Purging

A data warehouse is designed to accumulate data across many years. Over a period of time data is accumulated in the data warehouse. The data can consist of the following

- Star Schemas
- Logs (Audit/Process/Error)
- Replicated Data
- Staging Data

Target data warehouse data is usually expected to be retained for periods longer than seven years and even when the threshold is reached older data is expected to be archived. Since the archival routines and rules may differ for different implementers, archival of target data warehouse is not covered here.

Depending on the policies in place it is always advisable to take a backup before purging data. A routine backup should be set up for the data warehouse.

This section provides details on what can be purged and how to purge. Not all topics are applicable in all situations. The following sections are covered.

- NMS/BI Recent Outage Data
- Near Real Time Data

- OWB Audit Logs  
ETL Job Control Tables
- ODI Session Logs
- Staging Data
- Replication Data

## NMS/BI Recent Outage Data

The NMS RDBMS is generally not intended to be in the long term (longer than a year or two) repository for historical customer outage data. Rather Network Management System (NMS) is intended to track the current and relatively recent status of the utility load area infrastructure. The NMS RDBMS generally includes a record of relatively recent customer outages - for operational reference/review purposes. Beyond this project specific "reasonably recent" window it is expected that some type of Network Management System (NMS) outage data archive/purge process will be executed on a regular (daily or weekly) basis to keep the NMS RDBMS instance relatively lean and performing optimally.

The Oracle BI RDBMS is intended to be the long term home for historical customer outage data. However, the Oracle BI RDBMS tracks two different types of outages in two sets of BI RDBMS tables and only one of them is considered a long term repository. The Oracle BI RDBMS tracks both recent and restored outages:

- **Recent outages** - sometimes called Near Real Time (NRT) outages. The primary purpose of the NRT data store is to support tracking current (active) and relatively recent completed outages. The NRT data store can also be used to help gauge the ability of existing resources to deal with a current storm - to help determine if external/foreign (crew) resources will or will not be required. Just like the Oracle NMS RDBMS data store the Oracle BI RDBMS "recent outage" data store is not intended to be a long term repository.
- **Restored outages** - sometimes called historical outage data. The primary purpose of the restored outage tables is for long term reporting purposes. The restored outage data store should be the most accurate data store and is intended to support required regulatory customer impact reports. This data store has no periodic purge requirements. Historical outage data is intended to be held in this data store indefinitely.

## Near Real Time Data

Near Real Time (NRT) data can be purged two ways:

- Purging data manually using Oracle supplied SQL script
- Purging BI NRT data through extract files generated on the NMS side based on the contents of the oms\_delete\_log table

In both methods described above, the database function `SPL_OMS_SNAPSHOT_PKG.SPL_PURGE_RECENT_FNC` is being called to purge the data. The database function physically deletes the data from the following tables

- cf\_recent\_call
- cf\_recent\_crew
- cf\_recent\_job
- cf\_cust\_recent\_outg
- cf\_recent\_td\_entry

When data is purged manually through Oracle supplied script, the function deletes data older than a customer defined retention period. When NRT data is being purged using extract files generated on the NMS side, the BI import job reads the NMS extract file and executes the workflow package `SPLWF_F_PURGE_RECENT`. The workflow package calls the same database function

(SPL\_OMS\_SNAPSHOT\_PKG.SPL\_PURGE\_RECENT\_FNC) to physically purge the no longer desired BI NRT data.

Following things are to be noted while purging BI data:

- Purging BI data using either of the above method requires no down time. Purging jobs can be scheduled while other BI import jobs are running.
- If the data to be deleted does not exist in BI RDBMS, no SQL error will be generated since the DELETE statements will find nothing to delete.
- Purging BI NRT data can be done independent of NMS data purging process. That said, retention period of data for NMS and BI can be different. The retention period can be defined based on customer's business needs.
- It is worth noting that having smaller data volume in BI NRT tables will generally improve performance for import jobs and building Materialized Views that are based on NRT tables.
- It is generally advised that periodic purge of BI NRT data should be done on a regular basis - at least weekly, possibly daily. This can be achieved via an automated process (Cron or similar).

## OWB Audit Logs

Note: this is applicable to the OWB based ETL only

Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) utilizes Oracle Workflow when running Oracle Warehouse Builder (OWB) Process flows to load extract files into the data warehouse. Even if extract files are not present, records are created in Audit tables each time a process flow is run. Depending on the frequency with which process flows are scheduled, these audit tables can grow to become unmanageable and can cause upgrades or process flow changes to fail when deployed.

The following topics are discussed in this section:

- Purging Extract, Transform, and Load (ETL) Job Control Tables
- Configuring Oracle Warehouse Builder to Enable Purging

A few of these audit tables include the run-time Oracle workflow audit tables and can grow very large:

- **WF\_ITEM\_ATTRIBUTE\_VALUES:** This table stores the run-time values of the Item attributes for a particular Process flow.
- **WF\_ITEM\_ACTIVITY\_STATUSES:** This table, along with the **WF\_ITEM\_ACTIVITY\_STATUSES\_H** contains all of the activities executed by a specific occurrence of a Process flow.
- **WF\_NOTIFICATION\_ATTRIBUTES:** This table contains the run-time values of all the Message Attributes for a specific Notification.
- In addition, Oracle Warehouse Builder (OWB) also contains audit tables that can also grow very large if not purged periodically.

Oracle Utilities Advanced Spatial and Operational Analytics includes a Purge process flow that calls the Oracle Warehouse Builder (OWB) and Oracle Workflow Application Programming Interfaces (APIs) to purge these audit tables as well. The **OUBIWF\_PURGE\_RT\_AUDIT** process flow in the **INIT\_PKG** Package is set up to purge audit data that is older than one month.

The **OUBIWF\_PURGE\_RT\_AUDIT** process flow is not run from the file processor daemon, so you must schedule it using a scheduler tool that can run OWB Process Flows. You can also schedule the procedure that this process flow calls, **OUBI\_PURGE\_RT\_AUDIT\_DATA\_PRC**, using a tool that can call a PL/SQL command. This procedure requires no parameters, and can be called directly from a PL/SQL block, like this:

```
BEGIN
OUBL_PURGE_RT_AUDIT_DATA_PRC;
END;
```

You should run either this purge routine, or the OWB and OWF purge routines at least monthly, so that the audit tables remain small. It is recommended to purge these tables at least once a month.

## ETL Job Control Tables

**Note:** this is applicable to the OWB based ETL only

It is recommended that the Extract, Transform, and Load Job Control tables be purged on a regular basis. For analysis purposes it is suggested to retain 30-90 days of data but depending on your need this value should be appropriately adjusted.

The ETL Job Control table resides in the dwadm schema on the BI instance. A sample script to purge the ETL Job Control table is shown below. Provide an appropriate value for the number of days for which data needs to be retained in the ETL Job Control table.

```
delete from bl_etl_job_ctrl
where start_dttm < sysdate -&days_to_retain and end_dttm is not
null
and job_status_flag = 'JC';
commit;
```

## ODI Session Logs

**Note:** This is applicable to ODI ELT only

ODI session logs that are older than a calculated date are purged periodically. The date is calculated as below

purge date = Least of (current date - b1\_retain\_days, Oldest session start where job is in error and has not been reprocessed)

The ODI Variable b1\_retain\_days is set to 1 by default. It can be configured using ODI client to a different value.

## Staging Data

**Note:** This is applicable to ODI ELT only

There is one staging table for each target entity. Staging data is useful for tracing the data propagation and debugging in case of issues. The retention period is configured in the Metadata table B1\_TARGET\_ENTTTY using column STG\_RETAIN\_DAYS. Default value is 7.

The staging data older than configured retention period is automatically purged as part of the execution of the interface.

## Replication Data

**Note:** This is applicable to ODI ELT only

Each source table that has been marked for replication has a replica table created in the replication schema. Tables that are used as source for Type II dimensions retain history for each change. Over a period of time the volume in the replication area can grow huge. It is recommended to retain the data in the replication layer for as long as it is feasible to do so considering the data storage requirements as this will enable you to reload the warehouse at any given point of time. This comes in handy when additional user defined columns are being added and you want to reload all entities where the new user defined column has been configured for load.

The table B1\_SOURCE\_TABLE has a column REPL\_RETAIN\_DAYS which can be modified to control the purge in replication area.

Note:

- A minimum retention period of 30 days is enforced.
- If there are multiple target entities dependent on the source table, then the least of last sync timestamp of the entities is considered.
- If the least of last\_sync\_dttm is older than the retention period then data older than this data is purged

# Chapter 8

---

## Licensing and Optional Features

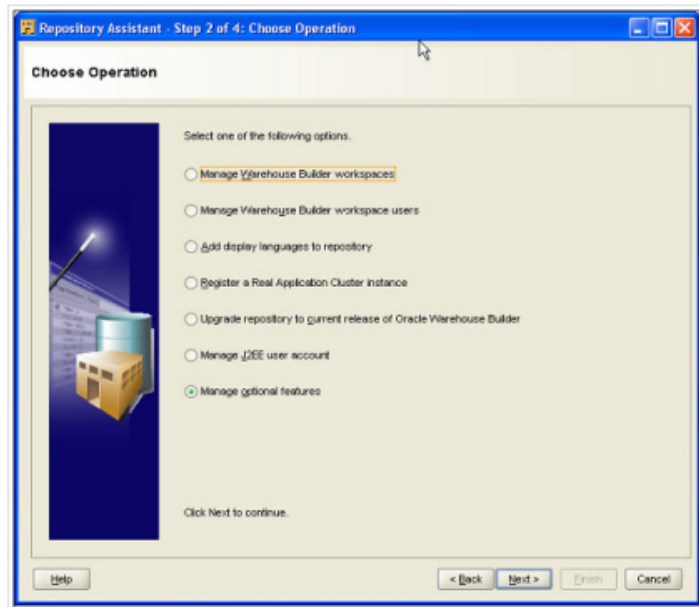
### Oracle Warehouse Builder Licensing and Optional Features

Oracle Warehouse Builder (OWB) provides various optional features which are not included in the basic Extraction, Transformation, and Loading (ETL) feature group. The basic ETL feature group is included in the Oracle Database Enterprise Edition License, hence; there is no additional license cost required to use or install the basic features. The standard ETL processes included in Oracle Utilities Analytics (OUASA) uses only the features that are included in the basic ETL feature group.

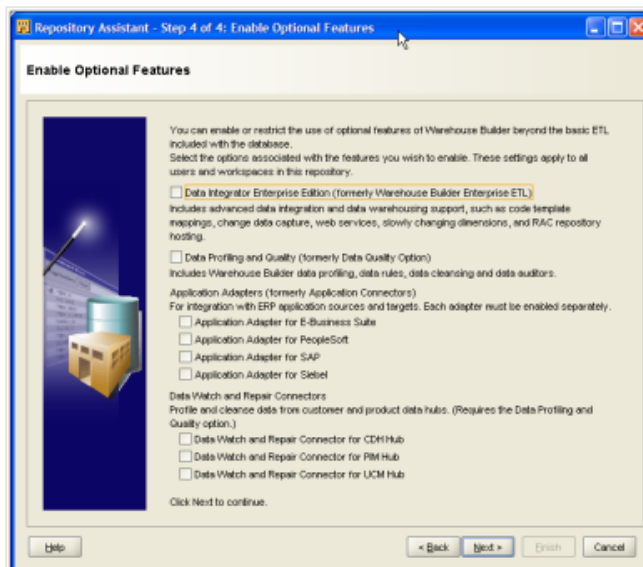
In addition, the Oracle Warehouse Builder (OWB) Code Generator does not create any code that requires the use of optional OWB features; hence, any additional Extraction, Transformation, and Loading (ETL) code created by an implementation using the OWB Code Generator does not require any additional OWB License costs. However, if Oracle Warehouse Builder (OWB) is used to create other ETL code outside of the OWB Code Generator, then using some of these optional features may require additional OWB licenses.

### Disabling the Optional Features in Oracle Warehouse Builder

In order to ensure that optional features are not used, the Oracle Warehouse Builder (OWB) provides a means to disable the use of optional features. After starting the Warehouse Builder Repository Assistant, choose the "Manage optional features" operation, as shown in the following image.7-2.



After entering the password for the OWBSYS user, deselect all of the licensed option names on the Enable Optional Features page.



Once the options are deselected, the new selections will take effect for any new connections to Oracle Warehouse Builder, and if options are used that are not available, an error dialog will be displayed.

For further details regarding the feature groups and licensing of Oracle Warehouse Builder (OWB), visit the OWB page on OTN at this location:

<http://www.oracle.com/technetwork/developer-tools/warehouse>.

# Appendix A

---

## Oracle Utilities Application Framework Extractors

This section contains the following topics and describes the general structure and process followed for the extractors for Oracle Utilities Application Framework (OUAF) based applications:

- Change Detect Mechanism
- Fields on the Change Log
- Typical Structure of Triggers
- Rows In the Change Log
- Extracting and Transforming Data
- Basic Parameters Supplied To Extract Processes
- Configuration changes for User defined columns

### Trigger-Based Extractors

This section contains the following topics:

- Change Detect Mechanism
- Fields on the Change Log
- Typical Structure of Triggers
- Rows In the Change Log
- Extracting and Transforming Data
- Basic Parameters Supplied To Extract Processes
- Configuration changes for User defined columns

### Oracle Utilities Customer Care and Billing

This section describes the data extraction methods used in Oracle Utilities Customer Care and Billing and Oracle Utilities Work and Asset Management Edge Applications. This section contains the following topics:

- **Change Detect Mechanism**
- **Fields on the Change Log**



- Typical Structure of Triggers
- Rows In the Change Log
- Extracting and Transforming Data
- Basic Parameters Supplied To Extract Processes

## Change Detect Mechanism

Every production database table used to populate the data warehouse must be monitored for changes so that these changes can be reflected in the data warehouse. Triggers insert a row into the Change Log when the source tables change. The topics in this section describe the Change Log and the triggers that populate it.

**Note:** This section applies to the Oracle Utilities Customer Care and Billing and Oracle Utilities Work and Asset Management.

## Fields on the Change Log

The sole job of triggers is to populate the change log. Therefore, You must understand the fields of the change log table in order to understanding the triggers. The change log contains the following primary fields:

| Field                | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Change Log ID        | This is a random prime key of the change log and is generated by the trigger.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Batch Code           | This is the code for the extract process that will process this change.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Batch Number         | This is the current run number for the extract process.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Change Date and Time | The date and time of the change                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Change Type          | This indicates if a row in the table was inserted, updated, or deleted.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Table Name           | The name of the table that was changed                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Prime Key 1 - 5      | The prime key of the object that was affected. The change log accommodates prime keys with up to five parts. The prime key stored on the change log is not the prime key of the record that was changed but the prime key of the object. For example, if the phone number of a person was changed, these prime key fields would contain the prime key of the person object, not the prime key of the phone number record. When any field on an object is changed, the entire object must be re-extracted. |

## Typical Structure of Triggers

Because all triggers populate the change log, they are similar in the following ways:

- They determine if a row needs to be inserted into the change log. Not all table changes need to be reflected in the data warehouse, so not all changes need to be noted in the change log. For example, if an unfrozen financial transaction is created, a change log record does not need to be inserted if the data warehouse only tracks frozen financial transactions.
- They generate a prime key for the change log.
- They know the codes for the appropriate extract processes that will handle the table change.

- They retrieve the current run numbers for the extract processes.
- They determine the prime key of the main object.

## Rows in the Change Log

A record in the change log is processed by only one extract process. If multiple extract processes are needed to handle a single change in a source table (for example, if a new object requires the addition of multiple facts or dimensions) then multiple rows must be inserted into the change log. This can be accomplished with one trigger inserting multiple rows into the change log or with multiple triggers on the same table, each trigger inserting one row.

## Extracting and Transforming Data

Both Oracle Utilities Customer Care and Billing and Oracle Utilities Work and Asset Management use batch controls with an underlying extract program to generate the flat files based on the change log tables populated by the triggers.

## Two Modes of Execution

Most extract programs support two modes of execution (you control the mode by a parameter supplied to the extract process):

- **Extract everything mode, or Initial Extract:** This mode extracts every row on the operational table. You would use this mode to instantiate the data warehouse. For example, if you run the extract accounts program in "extract everything mode", every account will be extracted.
- **Extract recent changes mode, or Incremental Extract:** This mode only extracts data that was added or changed since the last time the extract was executed. For example, if you run the extract accounts program in "extract recent changes mode", every account that was added or changed since the last execution will be extracted.

## Basic Parameters Supplied To Extract Processes

All extract processes are submitted in their source system (e.g., programs that extract data from Oracle Utilities Customer Care and Billing are submitted in Oracle Utilities Customer Care and Billing). The following points describe the hard parameters that are supplied to these processes for Oracle Utilities Customer Care and Billing.

- **Batch code:** Batch code is the unique identifier of the extract process. The batch code for each extract process is identified in the description of the various facts and dimensions. Refer to the appropriate fact and dimension chapter for the details in Oracle Utilities Data Mapping Guides.
- **Batch thread number:** Thread number is only used for extract processes that can be run in multiple parallel threads. It contains the relative thread number of the process. For example, if the arrears process has been set up to run in 20 parallel threads, each of the 20 instances receives its relative thread number (1 through 20). Refer to Optimal Thread Count for Parallel Background Processes in the background process chapter of the source system for more information.
- **Batch thread count:** Thread count is only used for extract processes that can be run in multiple parallel threads. It contains the total number of parallel threads that have been scheduled. For example, if the billing process has been set up to run in 20 parallel threads, each of the 20 instances receives a thread count of 20. Refer to Optimal Thread Count for Parallel Background Processes in the background process chapter of the source system for more information.
- **Batch rerun number:** Rerun number should only be supplied if you need to download an historical run (rather than the latest run).

- **Batch business date:** Business date is only used for extract processes that use the current date in their processing. For example, the Oracle Utilities Customer Care and Billing arrears extracts use the business date to extract arrears as of a given date. If this parameter is left blank, the system date is used. If supplied, this date must be in the format YYYY-MM-DD. This parameter is only used to test how processes behave over time.
- **Override maximum minutes between cursor re-initiation:** This parameter is optional and overrides each extract process's Standard Cursor Re-Initiation Minutes. Each extract process re-initiates cursors every 15 minutes. You can reduce these values, for example, if you are submitting a job during the day and you want more frequent commits to release held resources (or more frequent cursor initiations). You may want to increase these values when an extract process is executed at night or on weekends, and you have sufficient bandwidth and memory available on the servers. The maximum minute between cursor re-initiation parameter is relevant for Oracle implementations only. Most of the system extract processes contain an outermost loop/cursor. The cursor is opened at the beginning of the process and closed at the end. If Oracle detects that the cursor is open for too long, it may incorrectly interpret this as a problem and will display an error that the snapshot is too old. The processing for the extract processes is designed to refresh the cursor based on the minutes between cursor re-initiation in order to prevent this error.
- **User ID:** Following must be ensured with respect to a User ID:
  - The **User ID** is a user who should have access to all application services in the system as some batch processes call application services to perform maintenance functions (e.g., when an account is updated, the batch process may call the account maintenance application service). The display profile of the User ID controls how dates and currency values are formatted in messages.
  - **Password:** Currently, Password is not used.
  - **Language Code:** All language-sensitive data is extracted in this language. In addition, all error messages are presented in this language.
  - **Trace program at start (Y/N), trace program exit (Y/N), trace SQL (Y/N) and output trace (Y/N):** These switches are only used during Quality Analysis (QA) testing and benchmarking. If trace program start is set to **Y**, a message is displayed whenever a program is started. If trace program at exist is set to **Y**, a message is displayed whenever a program is exited. If trace SQL is set to **Y**, a message is displayed whenever an SQL statement is executed. If output trace is set to **Y**, special messages formatted by the extract process are written.

The information displayed when the output trace switch is turned on depends on each extract process. It is possible that an extract process displays no special information for this switch.

- **Initial Load Switch:** This switch controls whether the extract program is run in extract everything mode or extract recent changes mode.
- **File Path and File Name:** These parameters define the file path and/or file name for the output file. When supplying a FILE-PATH variable, the directory specified in the FILE- PATH must already exist and must grant write access to the Oracle Utilities Business Intelligence administrator account. You may need to verify a proper location with your system administrator. The syntax of the FILE-PATH depends on the platform used for your Oracle Utilities Business Intelligence application server. Contact your system administrator for verification. For example, if the platform is UNIX, use forward slashes and be sure to put a trailing slash, for example /spltemp/filepath/.

**Note:** The control file is created with the same name as the data file but with a fixed extension of CTL. For this reason, do not use CTL as the extension when defining value for FILE-NAME parameter.

In order to avoid overwriting the flat files generated during the previous execution, the extractor programs insert a string containing the concatenated values of data source indicator, batch number, and the batch thread number in the name of the generated data and the control file. The value is inserted just before the extension of the file name specified.

- **Maximum Errors:** This parameter is not currently used.
- **User-Defined File UDF and UDMs:** Refer to Extending extractors for the details on how to extend the various UDF and UDM fields.

# Appendix B

## Oracle Utilities Advanced Spatial and Operational Analytics Administration

The ODI based ELT architecture is a metadata driven framework that allows users to configure the out of the box ELT jobs, as well as making it easy to integrate custom ELT jobs to the OUASA product. The out of the box ELT jobs are set up during the installation of the product. However, user can alter or maintain some of the parameters using the OUASA Administration Tool. Please refer to the Chapter *Installing Admin Tool* in the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide* for details on how to install and access the Administration Tool.

The following table lists the metadata tables used to store this metadata, and the purpose of each.

| Entity           | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Product          | This table is used to identify which products are supported, and to identify the ODI scenario (accelerator) to be used to import source metadata and configurations. This table is for the internal use.                                                                                                                                                                                                                                                                                                                                                                               |
| Product Instance | A customer may have multiple instances of the same product that they want to integrate with the BI solution. The Instance object represents each instance of the same product that can be utilized as a source. The objective is to enable development of a single interface that can be utilized across multiple instances. Instances differ by the source database connections and possibly by the differing configurations. The data source indicator is unique across instances and products as this allows traceability to identify the source of the data in the data warehouse. |

---

| Entity            | Purpose                                                                                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Target Entity     | This table holds the configurations for target entities in the data warehouse. The target entities can be dimensions or facts. The same target entity may be loaded from multiple source instances. Configuration that is common across multiple instances should be stored in this table.           |
| Job Configuration | This table is used to provide configuration for the package (the executable logic) for populating the target entities for each instance. This table stores the current progress and an override package for use when the logic for one instance differs from the logic used for the other instances. |
| Source Tables     | This table contains configurations controlling which tables are replicated and the mode of replication.                                                                                                                                                                                              |
| Job Executions    | This table is used for tracking the execution of the ELT processes. An entry will be created for each execution. Some attributes are populated from the SNP_SESSION table which is used by ODI to track sessions.                                                                                    |

| Entity       | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dependencies | <p>This table is used to map a job to the jobs it is dependent on. Dependent objects will only be executed up to the minimum sync timestamp of all the dependencies. For example:</p> <ul style="list-style-type: none"> <li>• The Operational Device Fact is associated with three Type - Two dimensions (Operational Device Dimension, Utilities Asset and Address).</li> <li>• Assuming that Operational Device Fact is scheduled to run at 11 AM, and Address dimension is scheduled to run at 7 AM.</li> <li>• Address dimension would have processed all changes to address till 7AM. Assume an address location exists for a warehouse location in NY with ID101, effective start date of 01/10/2010 and effective end date of 31/12/4000.</li> <li>• A device was added to the system at 9 AM and a change was made to the NY location 101.</li> <li>• At 11 AM when the fact is processed, since the Address dimension is loaded only till 7 AM, the new history starting 7/30/2013 09:00 AM is not yet in the dimension and this will end up referencing the older record.</li> <li>• To avoid this, the Fact will be loaded only up to the last load timestamp of the Address dimension, which in this example will be 7 AM.</li> </ul> |

| Entity                    | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DSI Mapping               | A source system may be integrated with other source systems. When pulling information from multiple source instances it is possible that the data in one instance refers to master data from another instance. This table allows such cross references to be configurable. by default the configuration is at an instance level that is a product & instance number level. However more fine grained control is possible by specifying the entity for which the configuration should be applied. Note that users will have to configure data for this entity on the source application. For details refer to the 'Configuration' chapter in the Oracle Utilities Advanced Spatial and Operational Analytics Data Mapping Guides for the respective source application. |
| Golden Gate Configuration | This table is used to provide configuration details to be utilized for the Golden Gate script deployment on the source and target environments. An entry will need to be created for each source instance providing details, such as host, port, SID, Oracle home path etc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Golden Gate Checkpoint    | This table is used by golden gate replication process to track its processing activities. This table will be used by the scheduler process to ensure that the warehouse loading tasks only process data that has been synced. The structure is controlled by Golden Gate.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Extract Parameters        | This table stores configurations that control the extraction of data. Each source system may have different codes used for the different purposes. This table allows users to configure the codes so that the extraction routine can utilize it appropriately. Note that users will have to configure data for this entity on the source application. For details refer to the 'Configuration' chapter in the Oracle Utilities Advanced Spatial and Operational Analytics Data Mapping Guides for the respective source application.                                                                                                                                                                                                                                   |

Many of the metadata tables mentioned above can be configured / maintained with the help of the Administration Tool. The Administration tool is delivered along with the Oracle Utilities Advanced Spatial and Operational Analytics (OUASA) product to allow users to maintain these tables. This Administration tool has been built on top of Oracle Application Express feature which is available with the Oracle Database. The following sections explain the details of each of



the metadata tables and the details of how customers can maintain them using the Administration Tool.

## Product Instance

The Product Instance represents a specific instance of a source application that can be configured as a source for the BI solution. A record is created for every instance of the product for which ELT is setup to extract data. OUASA delivers a record out of the box for the product which uses ODI for the ELT. Customers may have multiple instances of the same source application that they want to integrate with the OUASA data warehouse.

The following attributes are available on the Product Instance Metadata table.

| Attribute             | Purpose                                                                                                                                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Product Code          | A reference to the product code, identifying the product for this instance                                                                                                                                                                      |
| Instance Number       | A unique number starting with 1 to uniquely identify the instance.                                                                                                                                                                              |
| Context Code          | A unique code comprised of the product code and instance number which is used to identify the connections in ODI. Due to limitations imposed by Golden Gate (used for replication) the context code cannot be more than 5 characters in length. |
| Data Source Indicator | A unique value representing the instance. For OUAF products this will be the environment id of the source instance.                                                                                                                             |
| Journal Indicator     | A flag identifying the methodology to be used for replicating source tables. Default should be Golden Gate.                                                                                                                                     |
| Drill back URL        | A URL to be used to allow users to drill back to the source system from analytics.                                                                                                                                                              |
| Currency code         | The currency used in the source product instance. Multiple currencies are not supported.                                                                                                                                                        |
| Time Zone Code        | The time zone of the source product instance. This will allow the interfaces to be built so that dependencies on job execution for multiple time zones is handled correctly.                                                                    |
| Language Code         | The primary language supported by the data warehouse. This will be used to filter language specific data in the data warehouse.                                                                                                                 |

| Attribute               | Purpose                                                                                                                                                                                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source Instance Version | The current version number of the source product instance. This can be used to validate whether the version is supported by the data warehouse or provide suggestions in case upgrades or patches need to be applied to enable support for the source product. |

The Product Instance UI in the Admin Tool allows users to maintain existing records. To add a new product instance, users will have to use the script delivered as part of the installation package. For more details refer to the *Oracle Utilities Advanced Spatial and Operational Analytics Installation Guide* and check the Add Instance section under Post Installation Tasks of ODI ETL Installation in Chapter 5.

**Maintain Product Instance**

Main

Cancel

Delete

Save

Source Product \*

Oracle Operational Device Management

Instance Number \*

1

Context Code

ODM1

Change Data Capture \*

Oracle Golden Gate ☒

Drillback URL \*

http://slc01linz.us.oracle.com:8500/ouaf/

Note : Enter the URL to access the source application in the following format http://(host-name):(port)/ouaf/cis.jsp

Information Retrieved from Source

Time Zone Offset

-08:00

Note : Specify in the following format 'HH:MM:SS'

Currency

USD

Language

ENG

Data Source Indicator

273498

Product Version

V2.0.1.1.0

## Target Entity

This table holds the configurations for target entities and their attributes in the data warehouse. The target entities can be dimensions or facts. The same target entity may be loaded from multiple source instances. Configuration that is common across multiple instances should be stored in this table.

| Attribute | Purpose                                                                     |
|-----------|-----------------------------------------------------------------------------|
| Name      | Name of the entity which needs to be scheduled for loading into the target. |

| Attribute                   | Purpose                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Type                        | Type of the entity supported by OUASA: <ul style="list-style-type: none"> <li>• Slowly changing dimension Type I</li> <li>• Slowly changing dimension Type II</li> <li>• Accumulation Facts</li> <li>• Snapshot Facts</li> <li>• Replication Entities</li> </ul>                                                                                        |
| Maximum Parallel Executions | To efficiently load data, it may be necessary to execute multiple instances, each instance working on a different data set. This attribute will control how many parallel executions can be spawned for a single entity load.                                                                                                                           |
| Retry Interval              | The base architecture has been designed for automatic retries. In case of failures jobs will be retried and this attribute controls the interval between successive retries in case of a failure. Default is 30 minutes but can be configured as per the requirement.                                                                                   |
| Maximum Retries per day     | This attribute will control the maximum number of retry attempts in a day. Once this limit is reached and the load is still failing it will be retried next day.                                                                                                                                                                                        |
| Schedule Type               | Three different modes of schedules are supported: <ul style="list-style-type: none"> <li>• Daily Incremental load; loads are executed as soon as data is available for load</li> <li>• Near Real time Load; loads are executed within a configured interval. This is for future use.</li> <li>• One time Load: A load is executed only once.</li> </ul> |
| Schedule Interval           | This is applicable for NRT loads only and the value controls the frequency of execution of the load process.                                                                                                                                                                                                                                            |
| Schedule Time               | A job will be executed only after the scheduled time each day. Default is 00:00. This can be changed as per the requirement                                                                                                                                                                                                                             |

| Attribute                | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Slice Duration Type      | <p>In any data warehouse, the basic challenge is to get the data loaded quickly and efficiently whether it is the initial load or the incremental load. Data volumes are high during initial load and during incremental loads the volumes are considerably smaller. A slice is a volume of data bound within a duration of time. Different objects have differing data distribution and load processing requirements. This attribute controls the duration between two slices. The following slicing intervals are supported</p> <ul style="list-style-type: none"> <li>• Day(s)</li> <li>• Week(s)</li> <li>• Month(s)</li> <li>• Quarter(s)</li> <li>• Year(s)</li> <li>• Hour(s)</li> <li>• Minute(s)</li> </ul> |
| Slice Duration           | The number identifying the slice duration based on the slice duration type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Package                  | Name of the ODI scenario that should be executed. Refer to Appendix L for a list of packages pre configured for ODI.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Staging Retention Period | Number of days to retain data in the staging tables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Owner Flag               | Indicates whether the record is owned by the base product (B1) or the customer (CM).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

The Target Entity UI in the Admin Tool allows users to add, edit and delete records. Records will be delivered out of the box for the star schemas tables delivered with the OUASA product. For these records, users will be able to edit only select fields. They cannot delete base product owned records. Create new records for any entity that customer wants to add to the ELT. For customer owned records, they will have full access to edit and delete them.

**Maintain Target Entity**

Main

Cancel

Save

Target Entity Id

9

Entity Name \*

CD\_SERVICE\_HIST\_TYPE

Entity Type \*

Dimension SCD1

Scheduling Parameters

Maximum Parallelism \*

1

Maximum Retries \*

2

Retry Interval \*

.0208333333

Scheduling Type \*

Incremental Update

Scheduling Time \*

00:00:00

Note : Specify in the following format 'HH24 MI:SS'

Scheduling Interval \*

0

day(s)

Slice Duration Type \*

Years

Slice Duration \*

13

ODI Package Name \*

B1\_PKG\_CD\_SERVICE\_HIST\_TYPE

Staging Retention Days

7

## Job Configuration

This table is used to hold configuration for the package (the executable logic) for populating the target entities for each instance. This table stores the current progress and an override package for use when the logic for one instance differs from the logic used for the other instances.

| Attribute           | Purpose                                                                                                                     |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------|
| Product Code        | A reference to the product code, identifying the product for this instance                                                  |
| Instance Number     | To be populated for instance specific deviations. Ideally most data should be reusable across instances of the same product |
| Entity              | A reference to the entity.                                                                                                  |
| Package             | An override option in case the common package cannot be reused across multiple instances.                                   |
| Slice Start         | The starting timestamp for next execution.                                                                                  |
| Active Indicator    | Indicates whether the schedule is active or not.                                                                            |
| User Exit Procedure | Procedure name which can be used to load the user defined columns in the entity.                                            |
| Last sync timestamp | Stores the timestamp of the most recent successful slice.                                                                   |

| Attribute          | Purpose                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------|
| Initialize Slices  | Controls whether the source data timestamps will be randomized for initial load. {For future use} |
| Execution Sequence | Controls the sequence in which jobs are submitted {For future use}                                |

The Job Configuration UI in the Admin Tool allows users to add, edit and delete records. As part of the OUASA installation process, job configuration records will be generated for the target entities available out of box. Users can then use this UI to change the configurations.

**Maintain Job Configuration**

Main

Job Configuration Id 14

Source Product \* Oracle Operational Device Management

Instance Number \* 1

Target Entity \* 14

Cancel

Delete

Save

Scheduling Parameters

Entity Active Flag \* Yes

Slice Start Date/Time \* 24-APR-2013 18:51:17

Initialize Flag \* Yes

Execution Sequence

Last Sync Date/Time 24-APR-13

Customization Attributes

Override ODI Package Name

User Exit Procedure

## Source Table

This table contains configurations controlling which source tables are replicated and the mode of replication.

| Attribute    | Purpose                                                                                  |
|--------------|------------------------------------------------------------------------------------------|
| Product Code | A reference to the product code, identifying the product for this instance               |
| Name         | Name of the source table. There will be a corresponding entry in the objects table also. |

| Attribute             | Purpose                                                                                                                                                                                                                                          |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mode                  | The following scenarios exist: <ul style="list-style-type: none"> <li>Source system tracks history using an effective date column</li> <li>No history in source but needs history for type II dimensions</li> <li>No history required</li> </ul> |
| Effective Date Column | Column name used for storing the effective dates in the source.                                                                                                                                                                                  |
| Base Replication      | Controls whether the table is required to be replicated for target entity load.                                                                                                                                                                  |
| Customize Replication | Extension for customizations. Additional tables to be marked for replication.                                                                                                                                                                    |
| Purge Enabled         | Controls whether the replicated table should be purged or not.                                                                                                                                                                                   |
| Retention Period      | Number of days the data should be retained in the replication layer.                                                                                                                                                                             |
| Owner Flag            | Indicates whether the record is owned by the base product (B1) or the customer (CM)                                                                                                                                                              |

The Source Table UI in the Admin Tool allows users to add, edit and delete records. For this table records will be delivered out of the box for the source tables that are configured to be part of the out of the box ELT. For these records users will be able to edit only select fields. They cannot delete base product owned records. However for customer owned records they will have full access to edit and delete them.

## Job Executions

This table is used for tracking the execution of the ELT processes. An entry will be created for each job execution.

| Attribute            | Purpose                               |
|----------------------|---------------------------------------|
| Job Configuration    | A reference to the job configuration. |
| Session Number       | Reference to the ODI session number.  |
| Scheduled Start Time | Start time as set in the schedule.    |
| Slice Start          | Starting timestamp of the slice.      |
| Slice End            | Ending timestamp of the slice         |

| Attribute         | Purpose                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Status            | <p>A composite status for the job. Primarily the status is derived from SNP_SESSION, however additional statuses are tracked in job executions:</p> <ul style="list-style-type: none"> <li>• Pending</li> <li>• Submitted</li> <li>• Running</li> <li>• Reprocessed</li> <li>• Error</li> <li>• Done</li> </ul> |
| Actual Start time | Timestamp when the job actually started.                                                                                                                                                                                                                                                                        |
| Actual End time   | Timestamp when the job actually ended.                                                                                                                                                                                                                                                                          |
| Duration          | Total execution time in seconds                                                                                                                                                                                                                                                                                 |
| Insert Count      | Number of records inserted                                                                                                                                                                                                                                                                                      |
| Update Count      | Number of records updated                                                                                                                                                                                                                                                                                       |
| Delete Count      | Number of deletions                                                                                                                                                                                                                                                                                             |
| Error Count       | Number of rows identified as error                                                                                                                                                                                                                                                                              |
| Total Count       | Number of rows processed. sum of the above                                                                                                                                                                                                                                                                      |

Use the Job Execution screen in the Admin Tool to monitor execution of the jobs. Users cannot edit or create records in this table.

**Job Execution Details**

Main Back

Job Execution Id 161

Job Configuration Id 7

Status Done

Scheduled Start Date/Time 24-APR-2013 00:00:00

Execution Details

Session 563602

Slice Start Date/Time 24-APR-2013 18:48:46

Slice End Date/Time 24-APR-2013 18:51:17

Session Start Date/Time 24-APR-2013 18:55:50

Session End Date/Time 24-APR-2013 18:56:08

Session Duration 18

Records Processed Details

Number of Inserts 4

Number of Updates 0

Number of Deletes 0

Number of Errors 0

Number of rows processed 7066



## Golden Gate Configuration

OUASA uses Golden Gate to capture the changed data. This table is used by the golden gate replication process to track its processing activities. This information is used by the scheduler process to ensure that the warehouse loading tasks only process data that has been synced.

| Attribute          | Purpose                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------|
| Configuration code | The code should either be the context code identifying the instance uniquely or B1 for the target data warehouse. |
| Manager Port       | The port that the Golden Gate Manager port listens at.                                                            |
| Dynamic Port Range | A port range that the golden gate process utilizes.                                                               |
| Algorithm          | The encryption algorithm to be used. Ex BLOWFISH.                                                                 |
| Encryption Key     | Name of the Key used for encryption/decryption.                                                                   |
| Database Host      | Host name of the database                                                                                         |
| Database Port      | Port Name for the database                                                                                        |
| Database ID        | System identifier for the database                                                                                |
| Database Home      | Database home folder where the database software is installed.                                                    |
| Golden Gate Home   | Home folder for the Golden gate software                                                                          |
| Shared Secret      | Shared Secret is the shared public key. The Encryption mechanism utilizes a Public Key private Key pair.          |

The Golden Gate Configuration UI in the Admin Tool allows users to add, edit and delete records. As part of the OUASA installation process, golden gate configuration records will be generated for the configured source application and the target BI application. Users can then use this UI to change the configurations.

Maintain Golden Gate Configuration

Main

Configuration Code \*

ODM1

Manager Port \*

7830

Dynamic Port Range Start \*

7830

Dynamic Port Range End \*

7880

Algorithm \*

1

Encryption Key \*

DEFAULT

Shared Secret \*

DEFAULT

Golden Gate Home \*

/bi\_oradata\_01/GoldenGateHome

Database Connection Details

Database Host \*

slc03rpv.us.oracle.com

Database Port \*

1521

Database SID \*

W2201XB2

Database Home \*

/scratch/gbuora/app/gbuora/product/11.2.0Std/dbhome\_2

# Appendix C

## Oracle Utilities Customer Care and Billing Extractor Details

This section includes the details for the Oracle Utilities Customer Care and Billing (CC&B) extractors:

| Fact / Dimension Table Name | Batch Control Name | Source Table Name | Trigger Name         | UDFs/UDMs being used |
|-----------------------------|--------------------|-------------------|----------------------|----------------------|
| CF_ARREARS                  | EXTSAARS           | CI_SA             | C1_BI_F_SAAC         |                      |
| CF_BILLED_USAGE             | EXTBLUSG           | CI_FT             | C1_BI_F_FTFZ         |                      |
| CF_CASE                     | EXTCASE            | CI_CASE           | C1_BI_F_CASE         |                      |
| CF_CASE_LOG                 | EXTCLOG            | CI_CASE_LOG       | C1_BI_F_CLOG         |                      |
| CF_CC                       | C1-CSCNT           | CI_CC             | C1_BI_F_CUSTCO<br>NT |                      |
| CF_COLL_EVT                 | C1-CUTEV           | CI_CUT_EVT        | C1_BI_F_CUTEV        |                      |
|                             | C1-ODEV            | CI_OD_EVT         | C1_BI_F_ODEV         |                      |
|                             | EXTCOLEV           | CI_COLL_EVT       | C1_BI_F_COLEV        |                      |
|                             | EXTSEVEV           | CI_SEV_EVT        | C1_BI_F_SEVEV        |                      |
| CF_COLL_PROC                | C1-ODPR            | CI_OD_PROC        | C1_BI_F_ODPR         |                      |
|                             | EXTCOLPR           | CI_COLL_PROC      | C1_BI_F_COLPR        |                      |
| CF_FT                       | EXTFIN             | CI_FT             | C1_BI_F_BUFZ         |                      |
| CF_FT_GL                    | C1-FTGL            | CI_FT_GL          | C1_BI_F_FTGL         |                      |
| CF_ORDER                    | C1-ORDER           | CI_ENRL           | C1_BI_F_ORDER        |                      |
| CF_PAY_TNDR                 | C1-PYTND           | CI_PAY_TNDR       | C1_BI_F_PAYTND<br>R  |                      |
| CF_RECENT<br>TD_ENTRY       | C1-RECTD           | CI_TD_ENTRY       | C1_BI_F_RECTD        |                      |
| CF_SA                       | EXTSAACC           | CI_SA             | C1_BI_F_SAAC         |                      |
| CF_TD_ENTRY                 | C1-TDENT           | CI_TD_ENTRY       | C1_BI_F_TDENT        |                      |

| Fact / Dimension Table Name | Batch Control Name | Source Table Name | Trigger Name  | UDFs/UDMs being used                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------|--------------------|-------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CF_UCOL_EVT                 | EXTUNCEV           | CI_WO_EVT         | C1_BI_F_UNCEV |                                                                                                                                                                                                                                                                                                                                                            |
| CF_UCOL_PROC                | EXTUNCPR           | CI_WO_PROC        | C1_BI_F_UNCPR |                                                                                                                                                                                                                                                                                                                                                            |
| CD_ACCT                     | EXTACCT            | CI_ACCT           | C1_BI_D_ACCT  | UDF1_CD -<br>CUST_CL_CD<br>UDF1_DESCR -<br>CI_CUST_CL_L.DESCR<br>UDF2_CD -<br>ACCT_MGMT_GRP_CD<br>UDF2_DESCR -<br>CI_ACCT_MGMT_GR_L.DESCR<br>R<br>UDF3_CD -<br>CIS_DIVISION<br>UDF3_DESCR -<br>CI_CIS_DIVISION_L.DESCR<br>UDF4_CD -<br>BILL_CYC_CD<br>UDF4_DESCR -<br>CI_BILL_CYC_L.DESCR<br>UDF5_CD -<br>COLL_CL_CD<br>UDF5_DESCR -<br>CI_COLL_CL_L.DESCR |
| CD_ADDR                     | EXTADDR            | CI_PREM           | C1_BI_D_ADDR  | UDF1_CD -<br>CITY_UPR<br>UDF1_DESC -<br>CITY_UPR<br>UDF2_CD -<br>COUNTY<br>UDF2_DESC -<br>COUNTY<br>UDF3_CD -<br>POSTAL<br>UDF3_DESC -<br>POSTAL<br>UDF4_CD -<br>STATE<br>UDF4_DESC -<br>CI_STATE_L.DESCR<br>UDF5_CD -<br>COUNTRY<br>UDF5_DESC -<br>CI_COUNTRY_L.DESCR<br>UDF6_CD -<br>GEO_CD<br>UDF6_DESC -<br>GEO_CD                                     |

| Fact / Dimension Table Name | Batch Control Name | Source Table Name   | Trigger Name                                                       | UDFs/UDMs being used                                                                                                               |
|-----------------------------|--------------------|---------------------|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| CD_ADJ_TYPE                 | EXTADJT            | CI_ADJ_TYPE         | C1_BI_D_ADJT,<br>C1_BI_D_ADJTD                                     | UDF1_CD -<br>AP_REQ_TYPE_CD<br>UDF1_DESCR -<br>CI_APREQ_TYPE_L.DESCR<br>UDF2_CD -<br>DST_ID<br>UDF2_DESCR -<br>CI_DST_CODE_L.DESCR |
| CD_CAMPAIGN                 | C1-CMPGN           | CI_CAMPAIGN         | C1_BI_D_CMPGN,<br>C1_BI_D_CMPAIG<br>ND                             | UDF1_CD -<br>CAMP_STATUS_FLG<br>UDF1_DESCR -<br>CI_LOOKUP_VAL_L.DESCR                                                              |
| CD_CASE_TYPE_STATUS         | EXTCTS             | CI_CASE_STATUS      | C1_BI_D_CTS                                                        |                                                                                                                                    |
| CD_CASE_COND                | EXTLKUP            | CI_LOOKUP_VAL       | C1_BI_D_LKUP<br>C1_BI_D_LKUPD                                      |                                                                                                                                    |
| CD_CC_TYPE                  | C1-CCTTY           | CI_CC_TYPE          | C1_BI_D_CCNCT<br>YPD                                               |                                                                                                                                    |
| CD_COLLEVT_TYPE             | C1-CUTET           | CI_CUT_EVT_TYP<br>E | C1_BI_D_CUTET<br>D                                                 |                                                                                                                                    |
|                             | C1-ODET            | CI_OD_EVT_TYP<br>E  | C1_BI_D_ODETD                                                      |                                                                                                                                    |
|                             | EXTSET             | CI_SEV_EVT_TYP<br>E | C1_BI_D_SET<br>C1_BI_D_SETD                                        |                                                                                                                                    |
|                             | C1-EXTCET          | CI_COLL_EVT_TY<br>P | C1_BI_D_CET<br>C1_BI_D_CETD                                        |                                                                                                                                    |
| CD_COLLPROC_STATUS          | EXTLKUP            | CI_LOOKUP_VAL       | C1_BI_D_LKUP,<br>C1_BI_D_LKUPD                                     |                                                                                                                                    |
| CD_COLLPROC_TMPL            | EXTCPT             | CI_COLL_PROC_T<br>M | C1_BI_D_CPTD                                                       |                                                                                                                                    |
|                             | C1-ODPT            | CI_OD_PROC_TM<br>P  | C1_BI_D_ODPTD                                                      |                                                                                                                                    |
| CD_FISCAL_CAL               | EXTFIPD            | CI_CAL_PERIOD       | C1_BI_D_FIPD<br>C1_BI_D_FIPDD                                      |                                                                                                                                    |
| CD_FT_TYPE                  | EXTLKUP            | CI_LOOKUP_VAL       | C1_BI_D_LKUP<br>C1_BI_D_LKUPD                                      |                                                                                                                                    |
| CD_GL_ACCT                  | C1-FTGL            | CI_FT_PROC          |                                                                    |                                                                                                                                    |
| CD_MSG                      | C1-MSG             | CI_MSG              | C1_BI_D_MSG<br>C1_BI_D_MSGD,<br>C1_BI_D_MSGCAT,<br>C1_BI_D_MSGCATD |                                                                                                                                    |
| CD_ORDER_STATUS             | EXTLKUP            | CI_LOOKUP_VAL       | C1_BI_D_LKU<br>C1_BI_D_LKUPD                                       |                                                                                                                                    |

| Fact / Dimension Table Name | Batch Control Name | Source Table Name | Trigger Name                                 | UDFs/UDMs being used                                                                                                                                                                                                                                                                                                                            |
|-----------------------------|--------------------|-------------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CD_ORDER_CAN_RSN            | C1-OCNRS           | CI_ENRL_CAN_RSN   | C1_BI_D_OCNRSND                              |                                                                                                                                                                                                                                                                                                                                                 |
| CD_ORDER_STATUS             | EXTLKUP            | CI_LOOKUP_VAL     | C1_BI_D_LKUP<br>C1_BI_D_LKUPD                |                                                                                                                                                                                                                                                                                                                                                 |
| CD_PAY_CAN_RSN              | C1-PCNRS           | CI_PAY_CAN_RSN    | C1_BI_D_PCNCRSND                             |                                                                                                                                                                                                                                                                                                                                                 |
| CD_PER                      | EXTPER             | CI_PER            | C1_BI_D_PER<br>C1_BI_D_PERN,<br>C1_BI_D_PERP |                                                                                                                                                                                                                                                                                                                                                 |
| CD_PKG                      | C1-PCKGE           | CI_PKG            | C1_BI_D_PKGD                                 |                                                                                                                                                                                                                                                                                                                                                 |
| CD_PREM                     | EXTPREM            | CI_PREM           | C1_BI_D_PREM                                 | UDF1_CD -<br>CIS_DIVISION<br>UDF1_DESCR -<br>CI_CIS_DIVISION_L.DESCR<br>UDF2_CD -<br>PREM_TYPE_CD<br>UDF2_DESCR -<br>CI_PREM_TYPE_L.DESCR<br>UDF3_CD - LS_SL_FLG<br>UDF3_DESCR -<br>CI_LOOKUP_VAL_L.DESCR<br>UDF4_CD -<br>TREND_AREA_CD<br>UDF4_DESCR -<br>CI_TREND_AREA_L.DESCR<br>UDF5_CD -<br>IN_CITY_LIMIT<br>UDF5_DESCR -<br>IN_CITY_LIMIT |
| CD_RATE                     | EXTRATE            | CI_RS             | C1_BI_D_RATE<br>C1_BI_D_RATED                | UDF1_CD -<br>SVC_TYPE_CD<br>UDF1_DESCR -<br>CI_SVC_TYPE_L.DESCR<br>UDF2_CD - FREQ_CD<br>UDF2_DESCR -<br>CI_FREQ_L.DESCR                                                                                                                                                                                                                         |

| Fact / Dimension Table Name | Batch Control Name | Source Table Name                                     | Trigger Name                                 | UDFs/UDMs being used                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------|--------------------|-------------------------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CD_SA                       | EXTSA              | CI_SA                                                 | C1_BI_D_SA                                   | UDF1_CD -<br>SVC_TYPE_CD<br>UDF1_DESCR -<br>CI_SVC_TYPE_L.DESCR<br>UDF2_CD -<br>CIS_DIVISION<br>UDF2_DESCR -<br>CI_CIS_DIVISION_L.DESCR<br>UDF3_CD - SA_TYPE_CD<br>UDF3_DESCR - CI_SA_TYPE_L<br>UDF4_CD - CI_SA_TYPE.<br>REV_CL_CD<br>UDF4_DESCR -<br>CI_REV_CL.DESCR<br>UDF5_CD - SIC_CD<br>UDF5_DESCR -<br>CI_SIC_L.DESCR<br>UDF6_CD - CI_SA_TYPE.<br>DEP_CL_CD<br>UDF6_DESCR -<br>CI_DEP_CL_L.DESCR<br>UDF7_CD -<br>CI_ENRL.CAMPAIGN_CD<br>UDF7_DESCR -<br>CI_CAMPAIGN_L.DESCR<br>UDF8_CD -<br>CI_SA_TYPE.DEBT_CL_CD<br>UDF8_DESCR -<br>CI_DEBT_CL_L.DESCR |
| CD_SA_STATUS                | EXTLKUP            | CI_LOOKUP_VAL                                         | C1_BI_D_LKUP,<br>C1_BI_D_LKUPD               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_SQI                      | EXTSQI             | CI_SQI                                                | C1_BI_D_SQID                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TD                       | C1-TD              | CI_TD_ENTRY                                           | C1_BI_D_TDENT                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TD_PRIORITY              | EXTLKUP            | CI_LOOKUP_VAL                                         | C1_BI_D_LKUP,<br>C1_BI_D_LKUPD               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TD_ROLE                  | C1-TDROL           | CI_ROLE                                               | C1_BI_D_TDROLE,<br>C1_BI_D_TDROLED           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TD_SKILL                 | C1-TDSKL           | Populated via<br>extracts from<br>Characteristic data |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TD_STATUS                | EXTLKUP            | CI_LOOKUP_VAL                                         | C1_BI_D_LKUP,<br>C1_BI_D_LKUPD               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TD_TYPE                  | C1-TDTYP           | CI_TD_TYPE                                            | C1_BI_D_TDTYP,<br>C1_BI_D_TDTYP<br>D         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TNDR_STATUS              | EXTLKUP            | CI_LOOKUP_VAL                                         | C1_BI_D_LKUP,<br>C1_BI_D_LKUPD               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CD_TNDR_SRCE                | C1-TNDCT           | CI_TNDR_SRCE                                          | C1_BI_D_TNDSRC<br>E,<br>C1_BI_D_TNDSRC<br>ED |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| <b>Fact / Dimension Table Name</b> | <b>Batch Control Name</b> | <b>Source Table Name</b> | <b>Trigger Name</b>            | <b>UDFs/UDMs being used</b> |
|------------------------------------|---------------------------|--------------------------|--------------------------------|-----------------------------|
| CD_TNDR_TYPE                       | C1-TNDTY                  | CI_TENDER_TYPE           | C1_BI_D_TNDTYPED               |                             |
| CD_TOU                             | EXTTOU                    | CI_TOU                   | C1_BI_D_TOUD                   |                             |
| CD_UCOLEVT_TYPE                    | EXTUET                    | CI_WO_EVT_TYPE           | C1_BI_D_UET,<br>C1_BI_D_UETD   |                             |
| CD_UCOLPROC_STATUS                 | EXTLKUP                   | CI_LOOKUP_VAL            | C1_BI_D_LKUP,<br>C1_BI_D_LKUPD |                             |
| CD_UCOLPROC_TMPL                   | EXTUCPT                   | CI_WO_PROC_TMPL          | C1_BI_D_WPTD                   |                             |
| CD_UOM                             | EXTUOM                    | CI_UOM                   | C1_BI_D_UOM,<br>C1_BI_D_UOMD   |                             |
| CD_USER                            | EXTUSER                   | SC_USER                  | C1_BI_D_USER                   |                             |



# Appendix D

## Oracle Utilities Work and Asset Management Extractor Details

This section includes the details for the Oracle Utilities Work and Asset Management extractors used in the Oracle Utilities Work and Asset Management (WAM) Edge Application:

| Fact/Dimension Table Name | Batch Control Name | Source Table Name | Trigger Name                | UDFs/UDMs being used                                                                                                                                                            |
|---------------------------|--------------------|-------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CD_ASSET                  | EXTDASSET          | SA_ASSET          | SDBT_BI_ADIU_ASSET          | UDF1: Asset Class<br>UDF2: Criticality<br>UDF3: Building<br>UDF4: Location<br>UDF5: Process<br>UDF6: Asset Record Type<br>UDF7: Facility<br>UDF8: Organization<br>UDF9: Company |
| CD_CREW                   | EXTDWRKC           | SA_CREW           | SDBT_BI_ADIU_CREW           |                                                                                                                                                                                 |
| CD_FAILURE                | EXTDFAIL           | SA_AUTHORITY      | SDBT_BI_ADIU_FAILURE        |                                                                                                                                                                                 |
| CD_OP_ACCT                | EXTDOPAC           | SA_ACCOUNT_DATA   | SDBT_BI_ADIU_ACCT_DATA      | UDF1: Area<br>UDF2 : Facility<br>UDF3 : Organization<br>UDF4 : Company<br>UDF5 : Level-1<br>Department<br>UDF6 : Level-2<br>Department<br>UDF7 : Level-3<br>Department          |
| CD_OP_ACTG_TY             | EXTDOATT           | SA_AUTHORITY      | SDBT_BI_ADIU_OP_ACCTG_TR_TY |                                                                                                                                                                                 |
| CD_OP_EXP                 | EXTDOPEX           |                   |                             | UDF1 : Expense Category<br>UDF2 : Facility                                                                                                                                      |
| CD_OP_UOM                 | EXTDOUOM           | SA_AUTHORITY      | SDBT_BI_ADIU_OUOM           |                                                                                                                                                                                 |

| Fact/Dimension Table Name | Batch Control Name | Source Table Name  | Trigger Name               | UDFs/UDMs being used                                                                                                                 |
|---------------------------|--------------------|--------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| CD_PLANNER                | EXTDWRKP           | SA_RULE_KEY        | SDBT_BI_ADIU_RULE_KEY_PLAN |                                                                                                                                      |
| CD_REPAIR                 | EXTDREPR           | SA_AUTHORITY       | SDBT_BI_ADIU_REPAIR        | UDF1: Facility                                                                                                                       |
| CD_ROOT_CAUSE             | EXTDROOT           | SA_AUTHORITY       | SDBT_BI_ADIU_ROOT_CAUSE    |                                                                                                                                      |
| CD_STOCK_ITMTY            | EXTDSITE           | SA_STOREROOM_LOG   | SDBT_BI_AI_STRM_LOG        | UDF1 : Stock Type<br>UDF2 : Stock Class<br>UDF3 : Commodity Category<br>UDF4 : Commodity Name<br>UDF5 : Commodity<br>UDF6 : Facility |
| CD_STRM                   | EXTDSTRM           | SA_STOREROOM_SETUP | SDBT_BI_ADIU_STRM_SETUP    | UDF1 :Storeroom Type<br>UDF2 : Facility<br>UDF3 : Organization<br>UDF4 : Company                                                     |
| CD_STRM_TR_TY             | EXTDSITT           | SA_STOREROOM_LOG   | SDBT_BI_AI_STRM_LOG        |                                                                                                                                      |
| CD_WRKORD_TY              | EXTDWOTY           | SA_AUTHORITY       | SDBT_BI_ADIU_WRK_ORD_TYPE  |                                                                                                                                      |
| CF_OP_ACTG                | EXTFOPAT           | SA_AUTHORITY       | SDBT_BI_ADIU_OP_ACTG_TR_TY |                                                                                                                                      |
| CF_STRM_INV               | EXTFSTOR           | SA_STOREROOM_SETUP | SDBT_BI_ADIU_STRM_SETUP    |                                                                                                                                      |
| CF_STRM_TR                | EXTFSTTR           | SA_STOREROOM_LOG   | SDBT_BI_AI_STRM_LOG        |                                                                                                                                      |
| CF_WRKORD_TK              | EXTFWRKT           | SA_WORK_ORDER_TASK | SDBT_BI_WORK_ORDER_TASK    | UDM1: Days Late<br>UDM2: Days to close<br>UDM3 : Scheduled Downtime Indicator                                                        |

# Appendix E

## Oracle Utilities Meter Data Management Extractor Details

This section contains details regarding each fact and dimension from the Oracle Utilities Meter Data Management (MDM) Edge Application for Synchronization BO-based extract and idiosyncratic Batch Extract as shown below:

### Synchronization BO-based Extract Details:

| Fact / Dimension Table Name | Initial Load Batch Control | Extract Batch Control | Sync BO                         | Extension                                                           |
|-----------------------------|----------------------------|-----------------------|---------------------------------|---------------------------------------------------------------------|
| CF_DEVICE_EVT               | D1-DEVIL                   | D1-DEVFX              | D1- DeviceEventFact             | Via Element Population Rules and Post Service Script on the Sync BO |
| CF_INSTALL_EVT              | D1-INEIL                   | D1-INEFX              | D1- InstallEventFact            | Via Element Population Rules and Post Service Script on the Sync BO |
| CF_SP                       | D1-SPIL                    | D1-SPAFX              | D1-SPAccumulationFact           | Via Element Population Rules and Post Service Script on the Sync BO |
| CF_DEVICE_ACTIVITY          | D1-ACTIL                   | D1-ACTFX              | D1-ActivityAccumulationFact     | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_CONS_TYPE                | D2-CSTIL                   | D2-CSTDY              | D2- ConsumSnapshotTypeDimension | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_MTR_DEVICE               | D1-DVCIL                   | D1-DVCDY              | D1-DeviceDimension              | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_MC                       | D1-MCIL                    | D1-MCDY               | D1- MeasuringComponentDimension | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_SPR                      | D1-SPRIL                   | D1-SPRDY              | D1-ServiceProviderDimension     | Via Element Population Rules and Post Service Script on the Sync BO |

| <b>Fact / Dimension Table Name</b> | <b>Initial Load Batch Control</b> | <b>Extract Batch Control</b> | <b>Sync BO</b>                  | <b>Extension</b>                                                    |
|------------------------------------|-----------------------------------|------------------------------|---------------------------------|---------------------------------------------------------------------|
| CD_SP                              | D1-SPIL                           | D1-SPDX                      | D1-SPDimension                  | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_ADDR                            | D1-SPIL                           | D1-ADRDY                     | D1-AddressDimension             | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_US                              | D2-USIL                           | D2-USDX                      | D2-USDimension                  | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_USAGE_GROUP                     | D2-UGIL                           | D2-UGDX                      | D2-UsageGroupDimension          | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_CONTACT                         | D2-CONIL                          | D2-CONDX                     | D2-ContactDimension             | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_MSRMT_COND                      | D2-MRCIL                          | D2-MRCDX                     | D2-MsrmtConditionDimension      | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_IE_STATUS                       | D1-IESIL                          | D1-IESDX                     | D1-IEBOSStatusAndReasnDimension | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_SP_STATUS                       | D1-SPSIL                          | D1-SPSDX                     | D1-SPBOSStatusAndReasnDimension | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_DAYS_LAST_MSRMT                 | D1-LNMIL                          | D1-LNMDX                     | D1-DaysSinceLastNormalMsrmtDim  | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_EXCP_TYPE                       | D2-EXTIL                          | D2-EXTDX                     | D2-ExceptionTypeDimension       | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_VEE_RULE                        | D2-VERIL                          | D2-VERDX                     | D2-VEERuleDimension             | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_DEVICE_ACTIVITY_STATUS          | D1-ACSIL                          | D1-ACSDX                     | D1-ActivityBOSStatusAndReasnDim | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_DEVICE_ACTIVITY_TYPE            | D1-ATYIL                          | D1-ATYDX                     | D1-ActivityTypeDimension        | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_DEVICE_EVT_STATUS               | D1-DESIL                          | D1-DESDX                     | D1-DEBOSStatusAndReasnDimension | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_DEVICE_EVT_TYPE                 | D1-DE'IL                          | D1-DE'TDX                    | D1-DeviceEventTypeDimension     | Via Element Population Rules and Post Service Script on the Sync BO |

| Fact / Dimension Table Name | Initial Load Batch Control | Extract Batch Control | Sync BO                     | Extension                                                           |
|-----------------------------|----------------------------|-----------------------|-----------------------------|---------------------------------------------------------------------|
| CD_SP_UT_AGE_TYPE           | D2-UTAIL                   | D2-UTADX              | D2-SPUTAggingTypeDimension  | Via Element Population Rules and Post Service Script on the Sync BO |
| CD_DAYS_LASTUT_TYPE         | D2-LUTIL                   | D2-LUTDX              | D2-DaysSinceLastUTDimension | Via Element Population Rules and Post Service Script on the Sync BO |

## Idiosyncratic Batch Extract Details:

| Fact / Dimension Table Name | Initial Load Batch Control | Extract Batch Control              | Sync BO      | Extension                                                     |
|-----------------------------|----------------------------|------------------------------------|--------------|---------------------------------------------------------------|
| CF_CONSUMPTION              | D2-SPCFX                   | Usage Snapshot                     | D2-SP-CA     | Via new snapshot algorithm                                    |
| CF_SP_SNAP                  | D1-SPSFX                   | Service Point Snapshot             | D1-SPSNAP-SE | Via Post Service Script on the snapshot algorithm's parameter |
| CF_SP_UT_AGE                | D2-SUAFX                   | Unreported Usage Analysis Snapshot | D2-SP-UT-AGE | Via new snapshot algorithm                                    |
| CF_VEE_EXCP                 | D2-SVEFX                   | SP VEE Exception Snapshot          | D2-SPVEEEXC  | Via new snapshot algorithm                                    |
| CD_UOM_TOU                  | D2-UTIL                    | n/a                                | n/a          | Via Post Service Script on the batch control parameter        |
| CD_UOM_TOU_SQI              | D2-UTSIL                   | n/a                                | n/a          | Via Post Service Script on the batch control parameter        |
| CD_IMD_TYPE                 | D2-ITLIL                   | n/a                                | n/a          | No extension                                                  |
| CD_EXCP_SEV                 | D2-EXLIL                   | n/a                                | n/a          | No extension                                                  |

# Appendix F

## Oracle Utilities Mobile Workforce Management Extractor Details

The chapter contains summary of the batch programs and sync Business Objects (BO) for each fact/dimension used in the Oracle Utilities Mobile Workforce Management (MWM) Edge Application:

| Fact /Dimension Table Name | Initial Load Batch Control | Extract Batch Control | Sync BO                        |
|----------------------------|----------------------------|-----------------------|--------------------------------|
| CD_CREW_SHIFT              | M1-SFTIL                   | M1-CRSDX              | M1-CrewShiftDimension          |
| CD_CREW_TM_USG             | M1-CTUIL                   | M1-CTUDX              | M1-CrewTimeUsageDimension      |
| CD_APPT_TM                 | M1-APTIL                   | M1-APTDX              | M1-AppointmentTimeDimension    |
| CD_APPT_TM_OF_DAY          | M1-ATDIL                   | M1-ATDDX              | M1-AppointmentTmOfDayDimension |
| CD_TRAVEL_DIST_DEV         | M1-TDDIL                   | M1-TADDX              | M1-TravelDurDeviationDimension |
| CD_SERVICE_AREA            | M1-SERIL                   | M1-SERDX              | M1-ServiceAreaDimension        |
| CD_CREW                    | M1-CREIL                   | M1-CREDX              | M1-CrewDimension               |
| CD_TASK_TYPE               | M1-TKTIL                   | M1-TKTDX              | M1-TaskTypeDimension           |
| CD_ADDR                    | M1-LOCIL,M1-CSAIL,M1-TKAIL | M1-ADRDX              | M1-AddressDimension            |
| CD_SHIFT_BO_STATUS         | M1-SBSIL                   | M1-SBSDX              | M1-ShiftBoStatusResDimension   |
| CD_TASK_BO_STATUS          | M1-TBSIL                   | M1-TBSDX              | M1-TaskBoStatusReasonDimension |
| CD_LATE_LOGON_TM           | M1-LLTIL                   | M1-LLTDX              | M1-LateLogonTimeDimension      |

| <b>Fact /Dimension Table Name</b> | <b>Initial Load Batch Control</b> | <b>Extract Batch Control</b> | <b>Sync BO</b>                |
|-----------------------------------|-----------------------------------|------------------------------|-------------------------------|
| CD_EARLY_LOGOFF                   | M1-ELTIL                          | M1-ELTDX                     | M1-EarlyLogoffTimeDimension   |
| CD_TRAVEL_DUR_DEV                 | M1-TADIL                          | M1-TDDDX                     | M1-TravelDistDevDimension     |
| CD_WORK_DUR_DEV                   | M1-WDDIL                          | M1-WDDDX                     | M1-WorkDurationDevtnDimension |
| CD_RESP_TM_DEV                    | M1-RTDIL                          | M1-RTDDX                     | M1-RespTimeDevDimension       |
| CF_FLD_ACTIVITY                   | M1-ACTIL                          | M1-ACTFX                     | M1-ActivityFact               |
| CF_CMP_SHIFT                      | M1-SFTIL                          | M1-CCSFX                     | M1-CompletedShiftFact         |
| CF_CREW_TASK                      | M1-SFTIL                          | M1-CRTFX                     | M1-CrewTaskFact               |

# Appendix G

## Oracle Utilities Network Management System Extractor Details

This section contains a summary of the extract scripts and the corresponding view names for each fact and dimension used in the Oracle Utilities Network Management System (NMS) Edge Application:

| Table Name      | Extract Program                         | Extract Procedure | Modify View       | Delete View       |
|-----------------|-----------------------------------------|-------------------|-------------------|-------------------|
| CD_ACCT         | bi_customer_extractor                   | PR_BI_EXTOACCT    | EXTOACCT_MODIFY_V | EXTOACCT_DELETE_V |
| CD_ADDR         | bi_customer_extractor                   | PR_BI_EXTOADDR    | EXTOADDR_MODIFY_V | EXTOADDR_DELETE_V |
| CD_CALL_INFO    | bi_event_extractor<br>AND nrt_extractor | PR_BI_EXTCINFO    | EXTCINFO_MODIFY_V | EXTCINFO_DELETE_V |
| CD_CREW         | bi_common_extractor                     | PR_BI_EXTOACCT    | EXTOCREW_MODIFY_V | EXTOCREW_DELETE_V |
| CD_CTRL_ZONE    | bi_common_extractor                     | PR_BI_EXTZONE     | EXTZONE_MODIFY_V  | EXTZONE_DELETE_V  |
| CD_DEVICE       | bi_common_extractor                     | PR_BI_EXTDEV      | EXTDEV_MODIFY_V   | EXTDEV_DELETE_V   |
| CD_EVENT        | bi_event_extractor<br>AND nrt_extractor | PR_BI_EXTJOB      | EXTJOB_MODIFY_V   | EXTJOB_DELETE_V   |
| CD_EVENT_STATUS | bi_common_extractor                     | PR_BI_EXTESTAT    | EXTESTAT_MODIFY_V | EXTESTAT_DELETE_V |
| CD_FEEDER       | bi_feeder_extractor                     | PR_BI_EXTFDR      | EXTFDR_MODIFY_V   | EXTFDR_DELETE_V   |
| CD_METER        | bi_customer_extractor                   | PR_BI_EXTOMTR     | EXTOMTR_MODIFY_V  | EXTOMTR_DELETE_V  |
| CD_PER          | bi_customer_extractor                   | PR_BI_EXTOPER     | EXTOPER_MODIFY_V  | EXTOPER_DELETE_V  |
| CD_PHASE        | bi_feeder_extractor                     | PR_BI_EXTPHASE    | EXTPHASE_MODIFY_V |                   |
| CD_PREM         | bi_customer_extractor                   | PR_BI_EXTOPREM    | EXTOPREM_MODIFY_V | EXTOPREM_DELETE_V |
| CD_SNL          | bi_customer_extractor                   | PR_BI_EXTCSP      | EXTCSP_MODIFY_V   | EXTCSP_DELETE_V   |
| CD_STORM        | bi_event_extractor and<br>nrt_extractor | PR_BI_EXTSTORM    | EXTSTORM_MODIFY_V | EXTSTORM_DELETE_V |



| Table Name               | Extract Program                               | Extract Procedure    | Modify View                  | Delete View          |
|--------------------------|-----------------------------------------------|----------------------|------------------------------|----------------------|
| CD_STORM_OUTAGE_TYPE     | bi_common_extractor<br>and bi_event_extractor | PR_BI_EXTSTORM<br>OT | EXTSTORMOT_MODIFY<br>_V      |                      |
| CD_SW_PLAN               | bi_switch_extractor                           | PR_BI_EXTSWSD        | EXTSWSD_MODIFY_V             | EXTSWSD_DELETE_V     |
| CD_SW_PLAN_STATE         | bi_switch_extractor                           | PR_BI_EXTVALST       | EXTVALID_STATES_MO<br>DIFY_V |                      |
| CD_USER                  | bi_common_extractor                           | PR_BI_EXTUSER        | EXTUSER_MODIFY_V             | EXTUSER_DELETE_<br>V |
| CF_CUST_RECENT_OUT<br>G  | nrt_extractor                                 | PR_BI_NRTCOF         | NRTSNL_MODIFY_V              | EXTSNL_DELETE_V      |
| CF_CUST_RST_OUTG         | bi_event_extractor                            | PR_BI_EXTCOF         | EXTSNL_MODIFY_V              | EXTSNL_DELETE_V      |
| CF_FEEDER_DLVRD_L<br>OAD | bi_feeder_extractor                           | PR_BI_EXTFDRD        | EXTFDRD_MODIFY_V             |                      |
| CF_RECENT_CALL           | nrt_extractor                                 | PR_BI_NRTINC         | EXTINC_MODIFY_V              | EXTINC_DELETE_V      |
| CF_RECENT_CREW           | nrt_extractor                                 | PR_BI_NRTCWA         | EXTCWA_MODIFY_V              |                      |
| CF_RST_CREW              | bi_event_extractor                            | PR_BI_EXTCWA         | EXTCWA_MODIFY_V              |                      |
| CF_RST_JOB               | bi_event_extractor                            | PR_BI_EXTJOB         | EXTJOB_MODIFY_V              | EXTJOB_DELETE_V      |
| CF_SW_PLAN               | bi_switch_extractor                           | PR_BI_EXTSWS         | EXTSWS_MODIFY_V              | EXTSWS_DELETE_V      |
| CF_SW_PLAN_STATE         | bi_switch_extractor                           | PR_BI_<br>EXTSWSLOG  | EXTSWSLOG_MODIFY_<br>V       |                      |

# Appendix H

## Oracle Utilities Validation Functions and Error Identification Procedure Names

This section lists the names for the Validation functions and Error Identification Procedure names for all Facts, along with the CM Error Procedure Name and Validation Function Name for those facts if a user wants to add more validations:

| Fact Name           | Error Procedure Name      | Validation Function Name  | Custom Error Procedure Name | UDFs/UDMs being used      |
|---------------------|---------------------------|---------------------------|-----------------------------|---------------------------|
| CF_ARREARS          | B1_ERR_F_ARREARS          | B1_VAL_F_ARREARS          | CM_ERR_F_ARREARS            | CM_VAL_F_ARREARS          |
| CF_BILLED_USAGE     | B1_ERR_F_BILLED_USAGE     | B1_VAL_F_BILLED_USAGE     | CM_ERR_F_BILLED_USAGE       | CM_VAL_F_BILLED_USAGE     |
| CF_CASE             | B1_ERR_F_CASE             | B1_VAL_F_CASE             | CM_ERR_F_CASE               | CM_VAL_F_CASE             |
| CF_CASE_LOG         | B1_ERR_F_CASE_LOG         | B1_VAL_F_CASE_LOG         | CM_ERR_F_CASE_LOG           | CM_VAL_F_CASE_LOG         |
| CF_CC               | B1_ERR_F_CC               | B1_VAL_F_CC               | CM_ERR_F_CC                 | CM_VAL_F_CC               |
| CF_CMP_SHIFT        | B1_ERR_F_CMP_SHIFT        | B1_VAL_F_CMP_SHIFT        | CM_ERR_F_CMP_SHIFT          | CM_VAL_F_CMP_SHIFT        |
| CF_COLL_EVT         | B1_ERR_F_CUTEV            | B1_VAL_F_CUTEV            | CM_ERR_F_CUTEV              | CM_VAL_F_CUTEV            |
| CF_COLL_EVT         | B1_ERR_F_SEV_EVT          | B1_VAL_F_SEV_EVT          | CM_ERR_F_SEV_EVT            | CM_VAL_F_SEV_EVT          |
| CF_COLL_EVT         | B1_ERR_F_ODEV             | B1_VAL_F_ODEV             | CM_ERR_F_ODEV               | CM_VAL_F_ODEV             |
| CF_COLL_EVT         | B1_ERR_F_COLL_EVT         | B1_VAL_F_COLL_EVT         | CM_ERR_F_COLL_EVT           | CM_VAL_F_COLL_EVT         |
| CF_COLL_PROC        | B1_ERR_F_COLL_PROC        | B1_VAL_F_COLL_PROC        | CM_ERR_F_COLL_PROC          | CM_VAL_F_COLL_PROC        |
| CF_COLL_PROC        | B1_ERR_F_ODPR             | B1_VAL_F_ODPR             | CM_ERR_F_ODPR               | CM_VAL_F_ODPR             |
| CF_CONSUMPTION      | B1_ERR_F_CONSUMPTION      | B1_VAL_F_CONSUMPTION      | CM_ERR_F_CONSUMPTION        | CM_VAL_F_CONSUMPTION      |
| CF_CREW_TASK        | B1_ERR_F_CREW_TASK        | B1_VAL_F_CREW_TASK        | CM_ERR_F_CREW_TASK          | CM_VAL_F_CREW_TASK        |
| CF_CUST_RECENT_OUTG | B1_ERR_F_CUST_RECENT_OUTG | B1_VAL_F_CUST_RECENT_OUTG | CM_ERR_F_CUST_RECENT_OUTG   | CM_VAL_F_CUST_RECENT_OUTG |
| CF_CUST_RST_OUTG    | B1_ERR_F_CUST_RST_OUTG    | B1_VAL_F_CUST_RST_OUTG    | CM_ERR_F_CUST_RST_OUTG      | CM_VAL_F_CUST_RST_OUTG    |
| CF_DEVICE_ACTIVITY  | B1_ERR_F_DEVICE_ACTIVITY  | B1_VAL_F_DEVICE_ACTIVITY  | CM_ERR_F_DEVICE_ACTIVITY    | CM_VAL_F_DEVICE_ACTIVITY  |
| CF_DEVICE_EVT       | B1_ERR_F_DEVICE_EVT       | B1_VAL_F_DEVICE_EVT       | CM_ERR_F_DEVICE_EVT         | CM_VAL_F_DEVICE_EVT       |

| <b>Fact Name</b>    | <b>Error Procedure Name</b> | <b>Validation Function Name</b> | <b>Custom Error Procedure Name</b> | <b>UDFs/UDMs being used</b> |
|---------------------|-----------------------------|---------------------------------|------------------------------------|-----------------------------|
| CF_FEEDER_DLVR_LOAD | B1_ERR_F_FEEDER_DLVRD_LOAD  | B1_VAL_F_FEEDER_DLVRD_LOAD      | CM_ERR_F_FEEDER_DLVRD_LOAD         | CM_VAL_F_FEEDER_DLVRD_LOAD  |
| CF_FLD_ACTIVITY     | B1_ERR_F_FLD_ACTIVITY       | B1_VAL_F_FLD_ACTIVITY           | CM_ERR_F_FLD_ACTIVITY              | CM_VAL_F_ARREARS            |
| CF_FT               | B1_ERR_F_FT                 | B1_VAL_F_FT                     | CM_ERR_F_FT                        | CM_VAL_F_FT                 |
| CF_FT_GL            | B1_ERR_F_FT_GL              | B1_VAL_F_FT_GL                  | CM_ERR_F_FT                        | CM_VAL_F_FT                 |
| CF_INSTALL_EVT      | B1_ERR_F_INSTALL_EVT        | B1_VAL_F_INSTALL_EVT            | CM_ERR_F_INSTALL_EVT               | CM_VAL_F_INSTALL_EVT        |
| CF_OP_ACTG          | B1_ERR_F_OP_ACTG            | B1_VAL_F_OP_ACTG                | CM_ERR_F_OP_ACTG                   | CM_VAL_F_OP_ACTG            |
| CF_ORDER            | B1_ERR_F_ORDER              | B1_VAL_F_ORDER                  | CM_ERR_F_ORDER                     | CM_VAL_F_ORDER              |
| CF_PAY_TNDR         | B1_ERR_F_PAY_TNDR           | B1_VAL_F_PAY_TNDR               | CM_ERR_F_PAY_TNDR                  | CM_VAL_F_PAY_TNDR           |
| CF_RECENT_CALL      | B1_ERR_F_RECENT_CALL        | B1_VAL_F_RECENT_CALL            | CM_ERR_F_RECENT_CALL               | CM_VAL_F_RECENT_CALL        |
| CF_RECENT_CREW      | B1_ERR_F_RECENT_CREW        | B1_VAL_F_RECENT_CREW            | CM_ERR_F_RECENT_CREW               | CM_VAL_F_RECENT_CREW        |
| CF_RECENT_JOB       | B1_ERR_F_RECENT_JOB         | B1_VAL_F_RECENT_JOB             | CM_ERR_F_RECENT_JOB                | CM_VAL_F_RECENT_JOB         |
| CF_RECENT_TD_ENTRY  | B1_ERR_F_RECENT_TD_ENTRY    | B1_VAL_F_RECENT_TD_ENTRY        | CM_ERR_F_RECENT_TD_ENTRY           | CM_VAL_F_RECENT_TD_ENTRY    |
| CF_RST_CALL         | B1_ERR_F_RST_CALL           | B1_VAL_F_RST_CALL               | CM_ERR_F_RST_CALL                  | CM_VAL_F_RST_CALL           |
| CF_RST_CREW         | B1_ERR_F_RST_CREW           | B1_VAL_F_RST_CREW               | CM_ERR_F_RST_CREW                  | CM_VAL_F_RST_CREW           |
| CF_RST_JOB          | B1_ERR_F_RST_JOB            | B1_VAL_F_RST_JOB                | CM_ERR_F_RST_JOB                   | CM_VAL_F_RST_JOB            |
| CF_SA               | B1_ERR_F_SA                 | B1_VAL_F_SA                     | CM_ERR_F_SA                        | CM_VAL_F_SA                 |
| CF_SP               | B1_ERR_F_SP                 | B1_VAL_F_SP                     | CM_ERR_F_SP                        | CM_VAL_F_SP                 |
| CF_SP_SNAP          | B1_ERR_F_SP_SNAP            | B1_VAL_F_SP_SNAP                | CM_ERR_F_SP_SNAP                   | CM_VAL_F_SP_SNAP            |
| CF_SP_UT_AGE        | B1_ERR_F_SP_UT_AGE          | B1_VAL_F_SP_UT_AGE              | CM_ERR_F_SP_UT_AGE                 | CM_VAL_F_SP_UT_AGE          |
| CF_STRM_INV         | B1_ERR_F_STRM_INV           | B1_VAL_F_STRM_INV               | CM_ERR_F_STRM_INV                  | CM_VAL_F_STRM_INV           |
| CF_STRM_TR          | B1_ERR_F_STRM_TR            | B1_VAL_F_STRM_TR                | CM_ERR_F_STRM_TR                   | CM_VAL_F_STRM_TR            |
| CF_SW_PLAN          | B1_ERR_F_SW_PLAN            | B1_VAL_F_SW_PLAN                | CM_ERR_F_SW_PLAN                   | CM_VAL_F_SW_PLAN            |
| CF_SW_PLAN_STATE    | B1_ERR_F_SW_PLAN_STATE      | B1_VAL_F_SW_PLAN_STATE          | CM_ERR_F_SW_PLAN_STATE             | CM_VAL_F_SW_PLAN_STATE      |
| CF_TD_ENTRY         | B1_ERR_F_TD_ENTRY           | B1_VAL_F_TD_ENTRY               | CM_ERR_F_TD_ENTRY                  | CM_VAL_F_TD_ENTRY           |
| CF_UCOL_EVT         | B1_ERR_F_UCOL_EVT           | B1_VAL_F_UCOL_EVT               | CM_ERR_F_UCOL_EVT                  | CM_VAL_F_UCOL_EVT           |
| CF_UCOL_PROC        | B1_ERR_F_UCOL_PROC          | B1_VAL_F_UCOL_PROC              | CM_ERR_F_UCOL_PROC                 | CM_VAL_F_UCOL_PROC          |
| CF_VEE_EXCP         | B1_ERR_F_VEE_EXCP           | B1_VAL_F_VEE_EXCP               | CM_ERR_F_VEE_EXCP                  | CM_VAL_F_VEE_EXCP           |
| CF_WRKORD_TK        | B1_ERR_F_WRKORD_TK          | B1_VAL_F_WRKORD_TK              | CM_ERR_F_WRKORD_TK                 | CM_VAL_F_WRKORD_TK          |

# Appendix I

---

## Oracle Warehouse Builder ETL Components

---

### External Table

The extract programs execute in the source application. They produce flat files that contain the data extracted from the source system. Each process creates:

- A single-record control file that contains information about the entire batch job
- Data files that contain the information to be loaded into the warehouse. These files are also referred to as the staging files.

Oracle external tables are defined in the warehouse for each type of control and data file. Specifically, two external tables are defined for each fact and dimension that is loaded from flat files. These external tables provide a SQL-based interface to the data in the flat files by the data mappings. A data mapping exists for each fact and dimension.

Within the Oracle database, the external tables have the following naming format:

- STG\_<table\_name>\_EXT for the data files
- STG\_<table\_name>\_CTL\_EXT for the control files that are used to load a specific table.

For example, the External Tables used to load the CD\_ACCT table are named STG\_ACCT\_EXT and STG\_ACCT\_CTL\_EXT.

The flat file names are different from the name of the external tables. The standard format for the file names are table\_name\_EXT.DAT and table\_name\_EXT.CTL. So for the CD\_ACCT table, the files will be named D\_ACCT\_EXT.DAT and D\_ACCT\_EXT.CTL.

Refer to the Oracle Utilities Advanced Spatial and Operational Analytics Data Mapping Guides for respective source application for the list of file names for each fact and dimension.

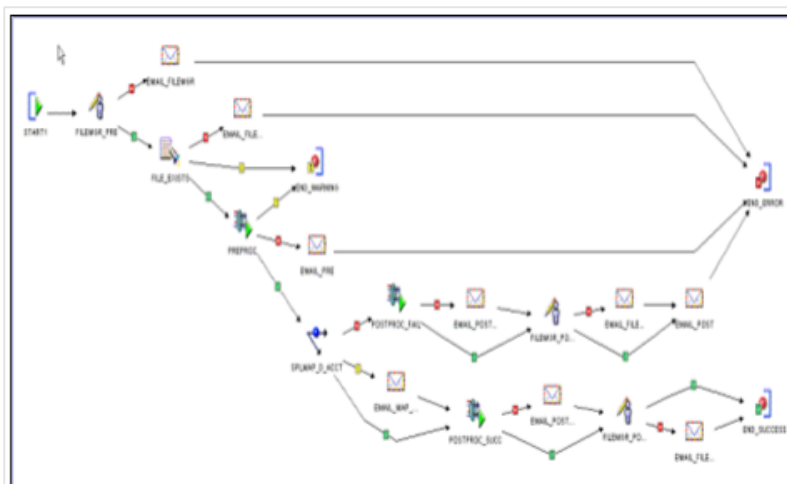
### Mapping

The data mappings load data from the external tables (produced by the extracts) into the facts and dimensions in the warehouse.

For a list of the facts and dimensions, their external tables, and the related data mappings, refer to the data mapping guide for your source application. This document describes the source application's facts and dimensions and how they are populated.

### Workflow

A separate process flow exists to execute each mapping along with the pre- and post-mapping processes. The following diagram shows a typical process flow:



Each data load process flow is designed to:

- Execute the file manager to perform housekeeping on the data and control files in pre- and post-mapping modes
- Execute the pre- and post-mapping functions to validate, load, and maintain batch information in the ETL job control transaction
- Execute the data mappings once the file is available and validated
- Send an email if an error occurs. Also, if an error occurs before the mapping executes, the process flow aborts the complete process. Otherwise, it sends an email and continues. Process flow modules allow you to group process flow packages.

## Packages

Process flow packages, allow you to group process flows. Together, the process flow modules and packages provide two levels to manage and deploy process flows. You can validate, generate, and deploy process flows at either the module or the package level. All process flows are presently grouped under the following packages for easier administration:

- **INIT\_PKG:** This package contains the process flows to load the default records into the dimensions. It also contains process flows to load the date & time dimensions and includes the purge workflow.
- **DIM:** This package contains process flows for dimensions delivered in Oracle Utilities Advanced Spatial and Operational Analytics.
- **DIM\_MDM:** This package contains process flows for dimensions delivered in Oracle Utilities Advanced Spatial and Operational Analytics for all Oracle Utilities Meter Data Management dimension tables.
- **DIM\_MWM:** This package contains process flows for dimensions delivered in Oracle Utilities Advanced Spatial and Operational Analytics for all Oracle Utilities Mobile Workforce Management dimension tables.
- **DIM\_UDD:** This contains the process flows for all user defined dimensions delivered in Oracle Utilities Advanced Spatial and Operational Analytics.
- **FACT:** This package contains process flows to load all of the fact tables in Oracle Utilities Advanced Spatial and Operational Analytics.

- **FACT\_MDM:** This package contains process flows for facts delivered in Oracle Utilities Advanced Spatial and Operational Analytics for all Oracle Utilities Meter Data Management fact tables.
- **FACT\_MWM:** This package contains process flows for facts delivered in Oracle Utilities Advanced Spatial and Operational Analytics for all Oracle Utilities Mobile Workforce Management fact tables.
- **MV\_RFSH:** This package contains process flows to refresh the default materialized views created for each fact table. If custom materialized views are created, then a copy of the fact table process flow should be created and the new materialized view refresh added to the copied process flow. Note that the refresh of the materialized views are done in parallel.
- **LOADRFSH:** This package contains process flows to load a fact table and then refresh the materialized views for that fact table. A load refresh process flow initiates the load for facts and subsequently executes the related materialized view refresh using the process flows under the package 'MV\_RFSH'.

Note the following about the various process flows:

- Process flows can be scheduled for execution using the file processor daemon. See Running and Monitoring Extract Loads for more information.
- Process flows for dimensions must be executed before the fact process flows.
- Each process flow executes its data mapping using parallel set-based processing with a commit frequency set to 0.

## File Manager

The file manager is a Perl program that resides on the database server. The program is responsible for performing housekeeping activities against the files holding the extracted information. It also ensures that the files are supplied in the correct order.

The program accepts the following parameters:

- The name of the file that the external table reads the data from. This name should match the value of the flat file name without the file extension. So for the load of the CD\_ACCT table, this would be D\_ACCT\_EXT.
- FILE-NAME parameter on the extract batch program.
- The location of the files.
- Mode of execution. The program can be executed in pre-mapping and post-mapping modes.
- Processing condition (success or failure).
- In the pre-mapping mode, the file manager performs the following actions
- Creates "error" and "processed" files inside the folder where the files are located.
- Sorts to get the name of the earliest control and data files that match the file name specified by the parameter passed.
- Copies the data file and the control file to the files that the external table reads. This is required because the external tables are defined to read data from one particular file and the extractor programs insert the data source indicator, batch number, and batch thread number in the data and control file names to avoid overwriting the generated files.
- Saves the name of the file being processed in a temporary file. This file is used later in the post-mapping stage to identify the name of the file that was processed. It is also used by the subsequent executions to know if a file is being processed.

In post-mapping mode, depending on the processing condition specified, the file manager moves the processed control and data file to either the error or the processed folder. It also removes the temporary file created in the pre-mapping mode.

# Appendix J

## Oracle Warehouse Builder Deployment

This section covers the details on how to generate and deploy the OWB code for the ETL of new star schemas.

### Oracle Warehouse Builder Code Generator

The Oracle Warehouse Builder (OWB) Code Generator is used to create OMBPlus TCL scripts. OMBPlus is an OWB scripting language that can create, alter, delete, and deploy OWB objects. The GenOWB.exe program is located in the Database Package in the scripts folder.

GenOWB.exe must be run on a Windows machine that can connect to the data warehouse database. Use the following syntax to run the OWB Code Generator:

The parameters and related values in the GenOWB.exe are described below:

```
GenOWB.exe -d <DBInfo> -t <TableName> -m <Mapping/Workflow
StagingTables/Facts/Dimension/All> -a <Dimensions/Facts/All> -h -g
```

The parameters and related values in the GenOWB.exe are described below:

| Parameter | Value                                                                                                                                                                                                                            |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -d        | Database information: Database User ID, password, database TNS name (for example, dwadm, dwadm, or bivmdv)                                                                                                                       |
| -t        | Name of dimension or fact tables                                                                                                                                                                                                 |
| -m        | Generate the following: <ul style="list-style-type: none"><li>• Mapping (M)</li><li>• WorkFlow (W)</li><li>• Staging Tables (S)</li><li>• Sequences (Q)</li><li>• Facts (F)</li><li>• Dimensions (D)</li><li>• All (A)</li></ul> |



| Parameter | Value                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a        | Generate mapping/Workflow/Staging Tables/Sequences/Cubes/Dimensions for all (D)imension Tables, all (F)act tables or (A)ll Dimension and Fact tables (D/F/A) |
| -x        | DROP statement? (Y)es or (N)o. Default is No.                                                                                                                |
| -h        | Help                                                                                                                                                         |
| -g        | generate debug info                                                                                                                                          |

When the Oracle Warehouse Builder (OWB) Code Generator is run for a table, the following files are generated:

- **seq\_nam.TCL:** A file that creates the sequence in Oracle Warehouse Builder (OWB) used for the primary key of the table to be loaded. For example, the SPL\_ACCT\_SEQ.TCL file creates the sequence used to populate the primary key of the CD\_ACCT table. Note that the sequence should also be created manually in the database, as it is not recommended to deploy sequences from Oracle Warehouse Builder (OWB) to the database.
- **tbl\_name.TCL:** A file that creates the table definition in the Oracle Warehouse Builder (OWB). For example, the CD\_ACCT.TCL script creates the CD\_ACCT table in the Oracle Warehouse Builder. Note that tables should also be created manually in the database, as it is not recommended to deploy tables from Oracle Warehouse Builder to the database.
- **stg\_file\_name.TCL:** A file that creates the data file definition in the Oracle Warehouse Builder (OWB). For example, the STG\_ACCT\_FF.TCL creates the definition of the CD\_ACCT extract file.
- **ctl\_file\_name.TCL:** A file that creates the control file definition in the Oracle Warehouse Builder (OWB). For example, the STG\_ACCT\_CTL\_FF.TCL creates the definition of the CD\_ACCT control file.
- **stg\_tbl\_name.TCL:** A file that creates the data file external table definition in the Oracle Warehouse Builder (OWB). For example, the STG\_ACCT\_EXT.TCL creates the definition of the CD\_ACCT external table STG\_ACCT\_EXT.
- **ctl\_tbl\_name.TCL:** A file that creates the control file external table definition in the Oracle Warehouse Builder (OWB). For example, the STG\_ACCT\_CTL\_EXT.TCL creates the definition of the CD\_ACCT control table STG\_ACCT\_CTL\_EXT.
- **owb\_map\_name.TCL:** A file that creates the Oracle Warehouse Builder (OWB) mapping, which loads the data from the external table into the data warehouse table. For example, the SPLMAP\_D\_ACCT.TCL script creates the SPLMAP\_D\_ACCT mapping, which reads records from the STG\_ACCT\_EXT and STG\_ACCT\_CTL\_EXT files and loads the extracted records into the CD\_ACCT table.
- **owb\_wf\_name.TCL:** A file that creates the process flow which takes an extract file and loads it into the fact or dimension table. For example, the SPLWF\_D\_ACCT.TCL creates the SPLWF\_D\_ACCT process flow, which checks to see if an account extract file exists in the data load directory, and if it exists, then loads it into the CD\_ACCT table.
- **OUBI\_LDRF\_wf\_name.TCL:** A file that is created only for fact loads. It creates a process flow, which calls the data file loading process flow and the materialized view refresh process flow sequentially. For example, the OUBI\_LDRF\_FT.TCL file creates the OUBI\_LDRF\_FT process flow that calls the SPLWF\_F\_FT and OUBI\_RFSH\_FT process flows.

**Note:** To create this process flow, the materialized view refresh process flow must exist.

Depending on which Oracle Warehouse Builder objects are changed, the parameters to the GenOWB.exe program is modified.

In the previous example for the CF\_CASE table change, the following two commands should be run:

The first command shown above creates the SPLMAP\_F\_CASE.TCL file and the second command creates the SPLWF\_F\_CASE.TCL and OUBI\_LDRF\_CASE.TCL files, with drop commands in each file since the objects should already exist in the Oracle Warehouse Builder (OWB) repository.

```
GenOWB.exe -d spluser,spluser_pw,BICONN -t CF_CASE -m M -x Y
GenOWB.exe -d spluser,spluser_pw,BICONN -t CF_CASE -m W -x Y
```

The first command shown above creates the SPLMAP\_F\_CASE.TCL file and the second command creates the SPLWF\_F\_CASE.TCL and OUBI\_LDRF\_CASE.TCL files, with drop commands in each file since the objects should already exist in the Oracle Warehouse Builder (OWB) repository.

## Deploying TCL Files

Once the TCL scripts are created, they need to be loaded into the Oracle Warehouse Builder (OWB) repository using OMBPlus. OMB Plus is a flexible, high-level command line metadata access tool for Oracle Warehouse Builder. Use OMB Plus to create, modify, delete, and retrieve object metadata in Warehouse Builder design and runtime repositories. For more information about OMBPlus refer to the Oracle Warehouse Builder API and Scripting Reference document.

To open an OMBPlus window, in the OWB Design Center select **View->OMB\*Plus**.

From within the OMBPlus window, there are many OMBPlus commands available, but the following commands are the two that are usually be used:

- `cd SOURCE_DIRECTORY`
- `source TCL_FILE`

Note that OMBPlus is case sensitive, so that the commands must be specified in lowercase. Also, the '\' is an escape character in OMBPlus, so within a directory name, two '\'s must be used if it is needed.

To load the files that would have been created by using the commands in the previous section, assume that the TCL files are in the c:\bi\tcl directory. The following OMBPlus commands can be used to load the files:

```
cd c:\\bi\\tcl
source SPLMAP_F_CASE.TCL
source SPLWF_F_CASE.TCL
source OUBI_LDRF_CASE.TCL
```

The order that the TCL files are loaded is important, as the objects are dependent on other objects. If an Object is deleted from OWB by running a TCL file, then references to that object are dropped from already existing OWB objects. So in all cases, the order that the TCL files are listed in the preceding section should always be used. Also, if an earlier object is recreated, all of the other objects that are listed afterwards also need to be created.

For example, if a change needs to be made to an OWB mapping, the owb\_map\_name.TCL, owb\_wf\_name.TCL and OUBI\_LDRF\_wf\_name.TCL (for facts) scripts will have to be regenerated and reloaded into OMBPlus.

After loading some of these TCL scripts, the customizations that were done prior to earlier deployments will be lost, so the preconditions to deployment must be redone. The following is a list of the scripts that you must rerun after changes are made:

- EditFFCS.tcl - this will need to be run if Flat File TCL scripts (stg\_file\_name.TCL and ctl\_file\_name.TCL) are loaded.

- EditFFLL.tcl - This will need to be run if External Table TCL scripts (stg\_tbl\_name.TCL and ctl\_tbl\_name.TCL) are loaded.
- EDITFP.tcl and editmail.tcl - These scripts will need to be run if Process Flow TCL scripts (owb\_wf\_name.TCL) are loaded.

## Scheduling New Workflows

When new extracts are set up on the source application side and new OWB process flows have been setup to load the new flat files, the File Processor Daemon needs to be extended to allow the processing of the new flat files.

More specifically the parameters file needs to be extended to include the new mappings. In the new CM parameters file these following two types of mapping need to be present:

- extract.file.mapping.override.count
- extract.file.mapping

For more details on the file processor daemon and the above mentioned parameters, please refer to Chapter 4 under the section "File Processor Daemon"

OWB object deploymentOwbdeployment.sh on Unix platform (Owbdeployment.cmd on Windows platform) performs following tasks:

- Imports the MDL files
- Configures Control Center
- Registers locations
- Runs TCL scripts to configure
- Deploys OWB objects in following order
- Connectors
- External Tables
- Sequences
- Dimensions
- Transformations
- Mappings
- Workflows

# Appendix K

---

## Terminology

Following Oracle Utilities Advanced Spatial and Operational Analytics terminologies are explained in detail in this chapter:

### Data Source Indicator

The data warehouse is an integrated database receiving data from multiple sources applications. The Data Source Indicator (DSI) in OUASA is used as an identifier for the originating source application or system-of-record for the data. Both fact and dimension tables have a DSI value for each record.

In the edge applications integrated with OUASA, namely CCB, MDM, MWM, NMS, and WAM, the DSI value is defaulted from environment ID. In case of multiple instances of same application loading data in data warehouse, e.g. two separate instances of Customer Care & Billing (CC&B) application, both the instances should have different DSI value.

The DSI value is also used for data lookup when loading the fact data. The confirmed dimension like address may have data populated from multiple source applications, e.g. CCB & MDM or may have data populated from multiple instances of same source application, e.g. two instances of MDM. It is also possible that two records, one from MDM and another from CCB (or second instance of MDM), may have the same ID in source application. Therefore, while loading the fact record, DSI value is also included in dimension record lookup columns along with the source ID column(s) so that the fact record from a given source application is correctly linked to the dimension record from the same instance of the source application.

### Late Arriving Dimension

Late arriving dimensions are the dimensions where the fact (measurable quantities) table records come early when compared to the dimension table records. We have implemented an auto correction mechanism into the fact load process to handle such scenarios.

To support auto correction of late arriving dimensions, there will be two default rows in each dimension.

| Primary Key | Default Values | Data Source Indicator | Remark                                                                                                                                                                                                 |
|-------------|----------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0           | ***            | NULL                  | This is used to refer NULL value references in the source. Possibly valid reasons for the references to be NULL.                                                                                       |
| -99         | -N/A           | NULL                  | This is be used to tag any unidentifiable references. The source data has a value for a reference, but corresponding record does not exist in the target. Possibly caused by late arriving dimensions. |

## Referencing Foreign Keys

- The Foreign Key value is set to 0 for any rows with NULL values for "Ref Natural Key".
- The Foreign Key value is set to -99 for any rows with some value for "Ref Natural Key", but no corresponding record in the dimension. Error reprocessing will automatically bind these rows with the correct rows in the dimension, assuming that the dimension value has arrived.

## Oracle Data Integrator

### Context

A context is a collection of logical schemas mapped to physical schemas. All logical schemas may not be mapped to physical schemas in a context. The source, journal & replication schemas are not associated for the default GLOBAL context as this context is not associated with any source instance.

A context identifies the physical schemas and connections that will be utilized for data access. Interfaces can be executed in any context as this allows reusing the same code for multiple instances.

Contexts are created while a source instance is configured for integration with BI. The context code is a composite string comprising of the product code and the instance number. This combination allows us to create uniquely identifiable context codes associated with independent source systems.

### Logical Schema

A logical schema is a logical representation of a storage location. Same physical schemas can be associated with different logical schemas.

In our implementation we use the following logical schemas

- Source: This is associated with the source instance.
- Journal: The journal schema is utilized for non-golden gate based replication. This represents the work schema where objects can be created for processing.

- Replication: This represents the data area where replicated data is stored. Each source system is associated with a replication schema on the target database.
- Target: All star schema objects are stored in this area.
- Metadata: The metadata configurations utilized to manage the solution are stored in this data area.
- Master: ODI master repository is maintained in a separate schema associated with the master logical schema.
- Repository: this schema is used to store the work repository of ODI
- Configuration: The configuration schema is associated with the file store location used for export or import of the metadata.

## Physical Schema

A physical schema is the actual schema or database user where data is retrieved or stored.

## Model

A model is an ODI entity which stores the list of objects, their structures, constraints and relationships. The model also allows creation of validation rules that can be applied on the data.

Model folders in the solution are organized based on the logical schemas which also form the different stages of data in the data processing pipeline.

The source models are not part of the base ODI work repository that is installed in a fresh install. As each source instance is added the associated source model(s) are created and actions executed on it. Each source instance may have one or more models associated. The model codes are by suffixing two character alphabets to the context code.

## Agent

An agent is a component that executes the ODI jobs. Multiple agents can be created and utilized for load balancing. The solution uses weblogic enterprise agents for stability and scalability.

## Reverse Engineer

Reverse engineering is the process of identifying the data structures, constraints and relations from an existing data store. The data store can be any DBMS, files or any other ODI supported data store.

## Journalization

Journalization is the process of keeping track of data changes enabling incremental data access, transport and transformation.

## Change Data Capture

Change data capture is the process of identifying and tracking data that has changed since the last known state. Change data capture mechanisms are implemented by the journalizing knowledge module.

## Scenario

A scenario is a self contained prepackaged executable version of a package, interface or procedure in ODI. The logic within the scenario cannot be viewed using the ODI client.

## Knowledge Module

Knowledge modules (KMs) in Oracle Data Integrator are components that implement reusable transformation and ELT (extract, load, and transform) strategies across different technologies.

Knowledge Modules (KMs) are code templates. Each KM is dedicated to an individual task in the overall data integration process.

A KM is reused across several interfaces or models. The benefit of Knowledge Modules is that you make a change once and it is instantly propagated to hundreds of transformations.

Six Types of Knowledge Modules:

- Reverse-engineering Knowledge Modules (RKM)
- Check Knowledge Modules (CKM)
- Loading Knowledge Modules (LKM)
- Integration Knowledge Modules (IKM)
- Journalizing Knowledge Modules (JKM)
- Service Knowledge Modules (SKM)

Refer to Oracle Data Integrator documents for more details.

# Appendix L

## Oracle Data Integrator ETL Components

### Packages

The following table lists out the entities populated using ODI, and the ODI package that is executed to perform the data load from replication to the target entity:

| Entity               | Package                     |
|----------------------|-----------------------------|
| CD_ADDR              | B1_PKG_CD_ADDR              |
| CD_ASSET_AGE         | B1_PKG_CD_ASSET_AGE         |
| CD_ASSET_DISP        | B1_PKG_CD_ASSET_DISP        |
| CD_ASSET_INSP_STATUS | B1_PKG_CD_ASSET_INSP_STATUS |
| CD_ASSET_INSTALL_AGE | B1_PKG_CD_ASSET_INSTALL_AGE |
| CD_ASSET_INSTORE_AGE | B1_PKG_CD_ASSET_INSTORE_AGE |
| CD_LOCATION          | B1_PKG_CD_LOCATION          |
| CD_OPR_DEVICE        | B1_PKG_CD_OPR_DEVICE        |
| CD_SERVICE_HIST_TYPE | B1_PKG_CD_SERVICE_HIST_TYPE |
| CD_UTIL_ASSET        | B1_PKG_CD_UTIL_ASSET        |
| CF_ASSET_LOC         | B1_PKG_CF_ASSET_LOC         |
| CF_OPR_DEVICE        | B1_PKG_CF_OPR_DEVICE        |
| CF_OPR_DEVICE_SNAP   | B1_PKG_CF_OPR_DEVICE_SNAP   |
| CF_SERVICE_HIST      | B1_PKG_CF_SERVICE_HIST      |
| CF_OPR_DEVICE        | B1_PKG_CF_OPR_DEVICE        |