Oracle Financial Services Analytical Applications

Data Model Extension Guidelines

Release 8.0.7.0.0

March 2019

ORACLE[•] Financial Services



Oracle Financial Services Analytical Applications Data Model Extension Guidelines

Copyright © 2019 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 errorfree. If you find any errors, please report them to us in writing.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 that 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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

For information on third party licenses, click here.

Document Control

Version Number	Revision Date	Change Log
2.0	March 2019	Added the chapter Oracle SQL Data Modeler

Table of Contents

1	Pre	face	5					
1	.1	Purpose of this Document	5					
1	.2	Audience	5					
1	.3	Related Documents	5					
2	Ove	erview of Data Model	6					
2	2.1	Overview of OFSAA Data Model	6					
3	Str	ucture of OFSAA Data Model	7					
3	5.1	Common Staging Area	7					
3	5.2	Application Specific Processing Area	7					
3	5.3	Common Results Area	7					
4	Ext	ension Process	8					
5	Des	sign Guidelines for Staging Area	9					
6	Des	sign Guidelines for Processing Area1	0					
7	Des	sign Guidelines for Reporting Area1	11					
8	Cor	nfiguring Entity UDPs for Application Data Interface of Data Integration Hub1	3					
9	San	nple Extensions to Model1	5					
10	Ora	acle SQL Data Modeler1	8					
	0.1 0.2	Deploying SQL Data Modeler Configuration Files1 Generating Customized Model						
11	Dat	a Model Merge Guidelines2	0					
12	Bui	ld Guidelines	2					
13	3 Appendix							

1 Preface

This Preface provides supporting information for the Oracle Financial Services Data Foundation Data Model Extension Guidelines and includes the following topics:

- Purpose of This Document
- <u>Audience</u>
- <u>Related Documents</u>

1.1 Purpose of this Document

The purpose of this document is to detail the extension process of the OFSAA data model. The objective is to ensure that implementation teams extend the OFSAA application as per guidelines, conformance, and standards established for content and design established.

1.2 Audience

This document is intended to be used by Oracle Financial Services Analytical Applications Data Foundation System Administrator.

1.3 Related Documents

This section identifies additional documents related to OFSDF Application 8.0.8.0.0. You can access Oracle documentation online from the Documentation Library (<u>OHC</u>).

- Oracle Financial Services Data Foundation Application Pack User Guide (<u>OHC</u>).
- Oracle Financial Services Data Foundation Application Pack Installation Guide (OHC).

2 Overview of Data Model

The OFSAA data model is organized into distinct areas based on usage and function performed by each area of the model.

2.1 Overview of OFSAA Data Model

The OFSAA data model is organized into distinct areas based on usage and function performed by each area of the model.

The data model conforms to the data architecture processes for each function and the modelling methodology employed in each of these functional areas representing the usage pattern.

3 Structure of OFSAA Data Model

This section details the structure of OFSAA Data Model.

3.1 Common Staging Area

The common staging area is the single source of data for all of OFSAA. Primary purpose of staging is data acquisition and integration. The common staging area is a consolidation of entities fact, dimensions and others. Completely de-normalized in structure and bears no relationship to one another in the physical model. Logical relationships exist and are established and validated through the staging area processes such as the Value code Mapping, Data Quality and GL Reconciliation processes.

3.2 Application Specific Processing Area

Application specific processing area is a collection of application specific entities used for incoming data, Application maintenance and Application metadata. The modelling methodology employed is based on the processing needs of each Application. The modelling methodology varies from Application to Application. Various degrees of normalization to de-normalization and dimensional modelling structures are employed. The modelling in the processing area is governed by the needs of specific processing engines. Processing engines have evolved at different points of time and have different data needs structurally.

3.3 Common Results Area

The result area is a rich combination of source data, value added metrics from each of the analytical applications and supports the derivation of higher order value added metrics across conformed dimensions spanning multiple functional domains regulatory, economic and more.

The common results area is a conformed set of facts and dimensions that combine results from various applications and create an integrated result environment for analytics across applications. Integrated reporting at transaction, account and customer granularities are managed through an integrated common result modelling approach.

4 Extension Process

The Data model life cycle process involves understanding of data requirements, data architecture and data flow requirements of the applications that are being implemented.

The governance process is broadly described as follows:

• Data Requirements

Identification of the additional data requirements for an implementation of the application is determined through customization/configuration documents. Use-cases have to be established for different areas in the OFSAA model.

• Data Flow

Data flow needs to be established for these additional data requirements.

Model Design

Model extensions to be designed based on the guidelines provided for each area. Extensions should also be raised as Enhancement Requests to respective applications through the standard support process. Enhancement Requests will be analyzed by the Product Engineering team and design approaches will be provided where applicable. Review the data model customization review template attached in the appendix.

Model Build

Model extensions to be built based on the guidelines provided for each area.

The supported tools are:

- Erwin
- Oracle SQL Data Modeler

For information about extension through Oracle SQL Data Modeler, see <u>Oracle SQL Data</u> <u>Modeler</u>.

5 Design Guidelines for Staging Area

The following are the design guidelines for staging area:

- Subject Area: Staging area is designed to hold data related to multiple applications. Entities are typically shared across subject areas.
- Entities: Typically, Staging area will have wide tables to accommodate data requirements related to multiple OFSA applications in a single row.
- Relationships: Staging area model should not contain any relationships amongst entities. Referential integrity checks are performed as part of Data Quality checks.
- No redundancy of information: Information required for multiple applications need to be captured only once, in a way suitable to all the applications.
- Multiple periods: Staging is designed to capture and store data for multiple periods. This will be useful to support operational requirements such-as re-run of processes for earlier periods.
- No update: Staging data, as received from source systems should not be modified. In case of exceptions that require data updates, following are the options:
 - Data needs to be corrected in source systems and re-loaded into staging.
 - Data quality framework should provide the ability to identify data quality issues and correct the data with audit trail.
- Alpha-numeric Codes:
 - Master data should be designed to hold alpha-numeric codes.
 - Stage tables corresponding to the dimension column should also be explicitly designed as alpha-numeric.
- Static Information: Data created and maintained within the application such as setup, static master information and so on should not be represented in staging.
- Impact Analysis: Staging area data model is shared across multiple OFSA applications. Hence, when a property of any entity or attribute is modified, impact analysis needs to be performed with application owners before performing the change. Metadata browser can be used for performing the impact analysis.
- Data Origin: Each staging entity needs to include a column for storing the source of data. This information is useful to understand the lineage of data.
- Download Specification: Staging area is designed to receive data from source systems. Download specification for each application can be generated from staging area model. Userdefined properties for table and column are referenced in the generation of download specs. Mention UDP values for any attributes added as part of the extension process.
- Naming Standards: Staging data model should follow naming standards as detailed in the Appendix.
- Customization related User Defined Properties: Column level UDP's named 'Custom' (NO/YES), 'Customization Reason' (Pending ER, Specific to Customer, Modified data element) and 'Type of Change' (Length, Datatype, Logical Name, Description, Addition) have to be specified for attributes added/modified as part of the extension process.

6 Design Guidelines for Processing Area

Data model for the processing area is designed as per the requirements for the applications. The following are the guidelines for processing area:

- Design: Processing area for each application is designed for optimal performance of the calculation engines respective to the applications.
- Performance: Data will be moved from staging into processing area. Typically, processing involves multiple SELECT and UPDATE SQL statements and hence data model should be designed to enable high performance SELECT and UPDATE operations on the table.
- Relationships: Processing area data model should enforce non-identifying and identifying relationships amongst entities as applicable.
- Naming Standards: Processing data model should follow naming standards as detailed in the Appendix.
- Diagram Representation: Processing area entities should be grouped logically and represented in the erwin data model. In erwin, default display view should have primary key designators.
- Limited History: Processing is designed to support storage of limited history. This historical data is useful for verification of process outputs for earlier dates, if required.

7 Design Guidelines for Reporting Area

The purpose of the Reporting data model is to perform the following:

- Address application specific reporting requirements
- Address integrated reporting across applications
- Drill through and drill-across in reports.

The following are the guidelines for reporting area:

- 1. Application-specific Reporting: Application specific reporting area can have specific fact tables to store measures related to that application. However, dimensions that are referred in any of the application specific reporting need to be conformed across all the applications.
- **2.** Integration of Results: Integration of results is achieved through common structures for account and customer as described as follows:
 - Common Account Summary
 - Consolidates measures at the granularity of the account that are output from different applications.
 - Any account level reporting should be driven out of integrated Account Summary set of entities. Account attributes captured from source system and those which do not vary between runs needs to be part of the Common Account Summary entity. Any application specific attributes and measures that are computed by application needs to be part of application specific account summary entities. Such a data model design enables integrated reporting of account information.
 - Common Customer Summary
 - Consolidates measures at the granularity of the customer that are output from different applications.
 - Any customer level reporting should be driven out of integrated Customer Summary set of entities. Customer attributes captured from source system and those, which do not vary between runs, need to be part of the Common Customer Summary entity. Any application specific attributes and measures that are computed by application needs to be part of application specific customer summary entities. Such a data model design enables integrated reporting of customer information.
- Data Modeling Technique: Reporting data model is structured as typical star schemas' corresponding to analytical reporting requirements.
- Aggregate Entities: Depending on performance requirements for each application, information can be reported out of aggregate entities. However, a drill through to the base entity from the aggregate entity is mandatory.
- Reporting and Local Currency: Include additional attributes in the fact tables to store reporting and local currency equivalent of base measures. These attributes need to be computed by looking into the exchange rates.
- Dimensional Conformance: Same dimensional information should not be represented in different forms within reporting area.

- Slowly Changing Dimensions: Dimensions that need to track changes in attributes need to be handled through slowly changing dimensions (SCD) process. Such dimension entities need to have a standard set of attributes required for SCD process.
- Measures: Same fact information should not be represented in multiple fact tables.
- Intermediate Results: Reporting data model should hold both final and intermediate computed outputs of each application.
- Performance: Data is going to be loaded into reporting area from processing. Since reporting will involve multiple SELECT queries, data model should be designed to enable high performance SELECT operations on the table. Also, since reporting model provides integrated information across applications, entities will be wide compared to entities modelled in the processing area.
- Relationships: Reporting area data model should enforce non-identifying and identifying relationships between fact and dimension entities as applicable.
- Naming Standards Reporting data model should follow naming standards as detailed in the Appendix.
- Subject Area: Subject areas should be created for a logical set of reports. Ideally, each subject area should have a single fact and multiple dimension entities.
- Diagram Representation: Within the subject area, fact entities should be represented in the centre surrounded by dimension entities. In erwin, default display view should have primary key designators.
- Full History: Reporting area model is designed to support storage of full history.

8 Configuring Entity UDPs for Application Data Interface of Data Integration Hub

> In OFSAA DIH Application, the uploaded data model is visible as Application Data Interface (ADI). These ADI names are given through erwin data model entity UDP for each table. For any new table addition which is added as part of customization should have valid value in following entity UDPs to get the proper ADI names in DIH.

- OFSAA Data Interface Name: This Entity UDP expects the logical name of the table or logical name for a group of tables which will be displayed as an ADI in DIH. This UDP is mandatory for all Stage and Results tables to get a valid ADI in DIH which corresponds to each table in model.
- OFSAA Data Interface Subtype: This Entity UDP expects the logical name of a table where the table is part of a group of tables where we have a group name in OFSAA Data Interface Name UDP. This UDP is optional and applicable only if you have a one OFSAA Data Interface Name against a group of tables.

For Example:

 STG_CASA, STG_LOAN_CONTRACTS, STG_TD_CONTRACTS, FCT_COMMON_ACCOUNT_SUMMARY are defined as group of tables and the UDPs for the tables are given in the following table:

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_CASA	Customer Account	CASA Contracts
STG_LOAN_CONTRACTS	Customer Account	Loan Contracts
STG_TD_CONTRACTS	Customer Account	Term Deposits
FCT_COMMON_ACCOUNT_ SUMMARY	Customer Account	Common Account Summary

• If you have Stage and Fact tables for the same purpose, you need to define it as same name given for OFSAA Data Interface Name, OFSAA Data Interface Subtype.

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_ACCT_FEATURE_MAP	Account Feature Map	Account Feature Map
FCT_ACCOUNT_FEATURE_ MAP	Account Feature Map	Account Feature Map

• If you have independent tables like only one either stage or dim or fact, then Subtype can be left blank.

TABLE_NAME	OFSAA Data Interface Name	OFSAA Data Interface Subtype
STG_VEHICLE_MASTER	Vehicle Master	
FCT_TXN_CHANNEL	Transaction Channel	
DIM_ACCOUNT	Account Dimension	

9 Sample Extensions to Model

The following table summarizes some of the common types of extensions to model:

Extension type	Staging	Processing/Reporting
Add new dimension	 Create entity named STG_<dimension NAME>_MASTER.</dimension Primary key for the master entity should be the alpha- numeric dimension key identifier and MIS Date. Include attributes for the dimension as columns in the master table. 	 Create a set of 4 tables if the dimension/hierarchy needs to be maintained within OFSAA – DIM_<dimension name="">_B – stores members; DIM_<dimension name="">_TL – stores translated members; DIM_<dimension name="">_ATTR - stores values of attributes of dimension members; DIM_<dimension name="">_HIER - stores hierarchy.</dimension></dimension></dimension></dimension> Create the dimension table if the dimension needs to be used in reporting. Design the structure of the dimension table depending on the requirement to have level based or parent-child hierarchies. Dimension table should have support for slowly changing dimension. Configure data movement processes to load dimension tables in processing and reporting.
Add new Attributes to dimension	Enhance staging master table to include this attribute.	 In case the dimension is maintained within OFSAA, configure this attribute for the dimension. Model need not be changed. In case the dimension is used for reporting and attribute is required for reporting, then include this attribute in the dimension table as a column. Modify data movement processes to load this additional attribute into dimension tables in processing and reporting.
Modify data type/data length for any attributes of the dimension	Check if this data type/data length change does not impact any applications.	 Modify the data type/data length in the corresponding dimension table in reporting. Some of the dimensional attributes might be de-normalized in fact entities for performance reasons. Datatype/Data length needs to be modified in all such cases.

Extension type	Staging	Processing/Reporting
Add new entities for receiving hierarchy data from any MDM system	Create a set of four entities named STG_ <dimension name="">_B – holds the dimension members. STG_<dimension name="">_TL – holds the translated dimension members. STG_<dimension NAME>_ATTR – holds the values of attributes of dimension. STG_<dimension name="">_HIER - holds the hierarchy</dimension></dimension </dimension></dimension>	 Create a set of 4 tables if the dimension/hierarchy needs to be maintained within OFSAA – DIM_<dimension name="">_Bstores members</dimension> DIM_<dimension name="">_TL stores translated members</dimension> DIM_<dimension name="">_ATTR stores values of attributes of dimension members</dimension> DIM_<dimension name="">_HIER stores hierarchy.</dimension> Create the dimension table if the dimension needs to be used in reporting. Design the structure of the dimension table depending on the requirement to have level based or parent-child hierarchies. Dimension table should have support for slowly changing dimension. Configure data movement processes to load dimension tables in processing and reporting.
Add attributes to product processor	 Make sure that attributes that are being added do not exist in any form within the entity. Extend product processor to hold these attributes. 	 Extend processing fact tables to hold this information. Modify T2Ts to move this information from staging. Extend common account summary entity to hold this information if the attribute is required for reporting. Modify T2Ts to move this information from staging to reporting.
Add attributes to any fact entity	 Make sure that attributes that are being added do not exist in any form within the entity. Extend the entity with this additional attribute. 	 Extend the processing/reporting entity to include this attribute. Modify the data movement processes to move the information from staging.

Extension type	Staging	Processing/Reporting
Make changes to primary key of the fact entity	 Make sure that the impact of changing the granularity is carried for all the OFSA applications that are being implemented. Extend the fact entity in staging to include attribute as part of the PK. Modify the data movement processes to move data as per processing/reporting model. 	 Make sure that the impact of changing the granularity is carried for all the OFSA applications that are being implemented. Any primary key change in processing/reporting will impact OFSA applications. Hence, raise service requests in product support with the use-cases.
Add a new entity	 Make sure that the information to be captured in the new entity does not exist elsewhere in staging model. Modify the data movement processes to move data as per processing/reporting model. 	 In cases where data captured in the new entity in staging needs to be moved to an existing processing/reporting entity, modify the data movement processes. In cases where there is a need to represent fact information in a separate entity, extend processing/reporting model to create this entity. Build data movement processes to move data from staging.

10 Oracle SQL Data Modeler

Deploy the patch **29467329** from <u>https://support.oracle.com/</u> as per the steps detailed in the readme file of the patch. Follow the below mentioned steps to configure and extend the model. For more information, see <u>OFSAA Datamodel extensions through SQL Data Modeler (Doc ID 2521335.1)</u> at <u>My</u> <u>Oracle Support</u>.

NOTE Before proceeding with the model customization procedure, take a backup of the DB.xml file, and config and atomic schemas.

10.1 Deploying SQL Data Modeler Configuration Files

Follow these steps to deploy SQL Data Modeler configuration files:

- 2. Rename the file defaultdomains.xml (from previous step) to OFSAA_DOMAINS_MASTER.xml.
- 3. Repeat the steps 1 and 2 each time in case of changes, if any, to the file defaultdomains.xml.
- 4. Perform model changes as required.

10.2 Generating Customized Model

These are the generic tasks to generate customized model:

Adding Custom Column

Cracle SQL Developer Data Modeler : Relational_1 (FSD	F_Template)			and the second			
<u>File Edit View Team Tools Window Help</u>							
	33 @ @ 19 19 19						
Browser =	Welcome Page	Logical (FSDF_Template)	Relational_1 (FSDF_Template)			Navigator	×1.
Control (1) Contro (1) Control (1) Control (1) Control (1) Control (svn: 216501 2019-03-21 17:4 2019-03-21 17:4 2019-03-21 17:4 2019-03-21 17:4 2019-03-21 17:4 2019-03-21 17:1 2019-03-21 18:1 2019-03-21 18:1	17:27 - Open Design: "25	ctitities\RAGHU_ASSIGNMENTS\ "Df_Template g_account_testing	02DF\.DE7_6071.5QL_M00ELE3_FILE\subree\.920F_Teeple	a) (5) ta_Spdated()FD07_Ta		

• Setting User defined property (UDP) for Column to YES.

• Setting Version UDP value to current version.

g: Oracle SQL Developer Data Modeler : Relational_1 (SDF_Template)							
File Edit View Team Tools Window Help							
▶ 目 12 13 15 15 15 14 1 1 1 1 1 1 1 1 1 1 1 1 1 1							
Browser × E	Welcome Page 💉 🌇 Logical (P	SDF_Template) × Relational 1 (FSDF_Template) ×					
	Welcome Page E Logical (P	SDF_Template) [] Relational_1 (FSDF_Template) []		Navigator			
Designs [1]	(×			
Green Company Com	😽 Table Properties - stg_trust	5					
	٩	1					
B- B Relational Models [1]	General	User	Defined Properties				
⊕ - 219 Domains [99]	Columns						
🖶 - 🎯 Data Types Model	Primary Key	Properties					
🗱 Distinct Types 🛛	Unique Constraints	Property	Value				
🛃 Structured Types 🛛	Indexes	Staging - EC		÷			
🧮 Collection Types 🛛	Table Level Constraints	OREC					
Predefined Structured Types [2]	Existence Dependencies	Forex Swaps Instrument					
E Predefined Collection Types []	Nested Columns	CI - Institutional ESTCP					
🖧 SubViews []	Valid Time Dimensions	IRM-EBA-DA					
Displays [] Process Model	Materialized Ouery	BASEL CRD					
	Volume Properties	Staging - HM					
- E Information Structures I	Spatial Properties	Staging - BSP					
🖼 Events 🛙	Column Groups	Staging - LLFP					
🐻 All Information Stores 🛛	Comments	Staging - ALM Analytics					
🖏 Roles 🛛	Comments in RDBMS	GLRECON Profitability - Other Class					
BE Record Structures []	Notes	Oracle BD FR					
🚼 Transformation Packages []	Impact Analysis	Staging - Retail Pooling					
Business Information	Measurements	Instrument Profitability					
📰 Change Requests []	Change Requests Responsible Parties	Staging - Market Risk					
Sensitive Types []	Documents	Ledger Stat					
C- 10 ISOP Policies []	Scripts	Oracle BD AMLTF BASEL III - Credit Risk - Non Securitization - STD					
	Dynamic Properties	TP Non-Cash Flow					
		Staging - Basel III					
	-Redaction Policy	Processing - Basel					
	Classification Types	VERSION	80710				
	Rest Services	OIPA					
	20 -Summary	Staging - GL RECON					
	20	Reporting-HM					
	20	"					
	20	QK Apply	Naming Rules Gancel Help				
	2019-03-21 17:57:24 - Crea	- T-bl-					
	2019-03-21 17:57:24 - Crea 2019-03-21 18:00:30 - Crea						
2013-03-21 18:11:13 - Create Table							
				*			
	•)	- F			
	Messages SVN Console × SVN Co	onsole × Logging Page ×					

11 Data Model Merge Guidelines

This section details the instructions to merge two data models. To merge the two data models:

- **1.** Open the erwin model.
- 2. Navigate to the Actions menu and click the Complete Compare option.
- 3. The Complete Compare dialog opens, which consists of seven input forms.
- **4.** To access any of the seven input forms, in the left panel of the Complete Compare dialog, click the required form title. The seven input forms are:
 - Overview
 - Left Model
 - Right Model
 - Left Object Selection
 - Right Object Selection
 - Type Selection
 - Advanced Options
- 5. Overview: This form provides a brief description of the Complete Compare function
- **6.** Left Model: Using the Left Model input form, you can select data model for the left side of the comparison in the Complete Compare dialog.

To select the required model for the left side of comparison, click Load. The file Open dialog appears. Search for the required data model file, select it and click Open. The selected data model loads into the Complete Compare dialog. You can see the selected data model in the Open Models in Memory list box.

Alternatively, you can locate the required data model file on your computer and double-click to open it. The selected data model opens by default in the erwin model. Alternatively, if available, you can select required data model in the Open Models in Memory list box.

If the default selection of the data model for the left side of the comparison is not correct, follow step 4 to select the required data model. Else, click Next to go to the Right Model input form.

NOTE When you launch the Complete Compare function with one data model, that data model loads into the Right Model form of the Complete Compare dialog. When you open the second data model for comparison, existing data model in the Right Model form moves to the Left Model form, and the second data model loads into the Right Model form.

Left Model Selection Select the Left Model			Compare Level:
Please select a model for the lef	t side of complete compare.		Logical and Physical
	Load From		
	File	🔘 Database / Script	Mart
<u>Overview</u>			Load
Left Model	Open Models in Memory:		
Right Model	Open Models	Location	
	b OFSAA	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin	
Type Selection	b OFSAA (2)	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin	
Left Object Selection			
Right Object Selection			
Advanced Options			
	Set selected model as	s read only	
Load Session Save Session		< Back Next > Comp	are Close Help

7. Right Model: Using the Right Model input form, you can select the data model for the right side of the comparison in the Complete Compare dialog. The Right Model input form is identical to the Left Model input form.

To select the required model for the right side of comparison, click Load. The file Open dialog appears. Search for the required data model file, select it and click Open. The selected data model loads into the Complete Compare dialog. You can see the selected data model in the Open Models in Memory list box. Click Next to go to the Type Selection input form.

NOTE	When you launch the Complete Compare function with one data model, that data model loads into the Right Model form of the Complete Compare dialog. When you open the second data model for comparison, existing data model in the Right Model form moves to the Left Model form, and the second data model loads into the Right Model form.
------	--

Right Model Selection			
Select the Right Model Please select a model for the rig	ght side of complete compare.		Compare Level: Logical and Physical
	Load From		
	 File 	🔘 Database / Script	○ Mart
<u>Overview</u>			Load
Left Model	Open Models in Memory:		
Right Model	Open Models	Location	
-	CFSAA	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\BSP_Datamodel.erwin	
Type Selection	🎒 OF5AA (2)	D:\Ranveer_Universal_Shared_Folder\Mdel_Merge_Issues\Merge\OFS_ALM_Datamodel.erwin	
Left Object Selection			
Right Object Selection			
Advanced Options			
	Set selected model as	read only	
	1		
Load Session Save Session	J	< <u>Back</u> <u>N</u> ext > Compar	e Close Help

8. Type Selection: This form is critical in the entire Complete Compare process. Using this form, you can customize the layout for the merged output data model based on the existing layout of the input data models. This form restricts the comparison to specific object types and properties of the data models.

a. In the Type Selection input form, expand Model. In the list box, select the required options using checkbox.

ΝΟΤΕ	 If the Type Resolution dialog does not appear, go to the n form. Therefore, click Next. 	ext
	 When you select Type Selection form, it may prompt the display of the Type Resolution dialog to equalize the UDP across the left and right models. This is GDM-46 error. 	S

- **b.** You can select the following type of multiple options in the Type Selection form:
 - Subject Areas
 - UDP
 - Physical Table and Physical Column
 - Logical Entity and Logical Attribute
 - Subtype and Supertype
- **c.** Below are the prompted values for Subtype (child table) set and Subtype items (child table elements) selection. To select all values from the Subtype Symbol, select the Subtype Symbol checkbox. Or to select individual Subtype items, expand Subtype Symbol, and select checkboxes against required Subtype items.

NOTE	When the EAC-3033 error occurs, upload the merged model using the
	slice model and ignore the error.

Left Model	Complete Compare Type Sele Please select the types of obje	ction cts to be compared in complete compare.		Compare Level: Logical and Physical
Left Model Right Model Type Selection Left Check Selection Right Model Right Model Left Check Selection Right Model Right Mod			Physical Level	Database Level
Bitt Model Image: Properties Type Selection Image: Properties Bitt Object Selection Image: Properties Advanced Options Image: Properties Advanced Options Image: Properties Image: Properties Image: Properties Image: Properimate Properimate Properties Image: Prop	Overview	Option Set: Standard Default Option Set		- 🔁 🗋 🎽 🗍
Bidt Model Image: Cluber Type Selection Data Source Object Left Object Selection Image: Cluber Right Object Selection Image: Cluber Advanced Options Image: Cluber Advanced Options Image: Cluber Image: Cluber Image: Cluber <t< td=""><td>Left Model</td><td>Model</td><td></td><td>A</td></t<>	Left Model	Model		A
Type Selection Image: Control of Control o	Right Model	😥 🔚 🔃 Annotation		E
Left Object Selection Bubit Object Selection Advanced Options Default Trigger Template Default Trig	Type Selection	🖶 🔲 🚍 Data Movement Rule		
Built Object Selection Advanced Options Advanced Options Image: Control of the second of the	Left Object Selection			
Advanced Options Advanced Options Default Trigger Template Default		👳 💷 🞯 Database Link		
Image: Source Image: Source <td< td=""><td></td><td>🐵 🖳 🚳 Default Trigger Template</td><td></td><td></td></td<>		🐵 🖳 🚳 Default Trigger Template		
Image: Constraint of the source Image: Constraint of the source <td>Advanced Options</td> <td></td> <td></td> <td></td>	Advanced Options			
		🗄 🖳 📋 Disk Group		
Constaint		😥 🔲 🚍 Entity/Table		
Function				
Image: Standard View Image: Standard Image: Standard Standard Standard Image: Standard Standard Standard Image: Standard Standard Standard Image: Standard Standar		🗼 💷 💿 Function		
Image: Standard Image: Standard S		🐵 🔲 🋄 Library		
Anning Standard Anni				
Package P		😥 🗐 🚳 Naming Standard		
Comment C		🖶 🔲 🤣 Package		
Script Template				
Subtype Symbol Goment		⊕		
Subtype Symbol Comment Comment Constrai		🐵 💷 🎦 Stored Procedure		
Comporties Comment Comment Constraint Costraint Costraint Costraint Costrainte				
Constraint C				
Osciminator Atributes Gild Extended Notes Order				
- Extended Notes Order		Discriminator		
				•
		·		

- d. Then click Next to go to the Left Object Selection form.
- **9.** Left Object Selection: Using the Left Object Selection input form, you can select which objects (table properties) can be included from the Left Model form in the comparison process.

As shown in the image below, to choose the required Object Sets in the Choose Objects Using Sets list box, select the checkboxes against the Object Set names. The selected Object Sets appear in the Selected Objects list box on the right side.

NOTE You can select individual item in each Object set. For example, expand Subject Areas object set, then select the individual checkboxes against required Subject Area item.

GENERATING CUSTOMIZED MODEL

Left Model Object Selection Left Model Object Selection Please select the objects you wa	ant to compare from left model.	Compare Level: Logical and Physical
Overview Left Model Right Model Type Selection Left Object Selection Right Object Selection Advanced Options	Choose Objects Using Sets Choose Objects Using Sets Subject Areas Tablespaces Tablespaces Triggers Owned By Owned By	Selected Objects:
Load Session	< Back Next >	Compare Close Help

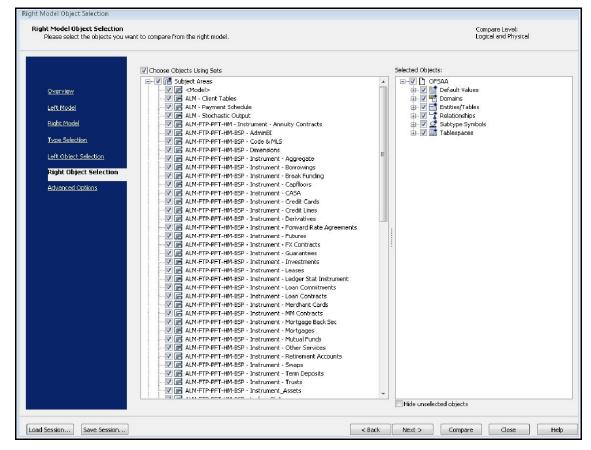
To include tables in the comparison process, you must select table Properties in the Type Selection form. Then the tables will appear in the Left Object Selection input form.

Type Selection			
Complete Compare Typ Please select the types	e Selection of objects to be compared in complete compare.		Compare Level: Logical and Physical
	Compare Level	Physical Level	Database Level
<u>Overview</u>	Option Set: Standard Default Option Set		- 2 1
Left Model	Model		<u> </u>
<u>Right Model</u>			E
Type Selection	🗄		

10. Right Object Selection: Using the Right Object Selection input form, you can select which objects (table properties) can be included from the Right Model form in the comparison process.

To choose the required Object Sets from the Choose Objects Using Sets list box, select the checkboxes against the Object Set names (you can select Object Sets like Subject Areas, Tablespaces, Indexes, Triggers). The selected Object Sets appear in the Selected Objects list box on the right side. Click Next to go to the Advanced Options form.

NOTE	• You can select individual item in each Object set. For example, if you are performing Subject Areas-wise Complete Compare, expand Subject Areas object set, then select the individual checkboxes against required Subject Area item. In the image below, based on apps requirement, we have selected all the Subject Areas.
	• If there are very less number of tables to merge with a very huge model, then you can create a Dummy Subject Area in a small model and move the tables into Dummy Subject Area. To Complete Compare under the Left Object Selection form or the Right Object Selection form, select the Dummy Subject Area and perform Complete Compare.



11. Advanced Options: This section consists of the functional Options. In the image below, the selected options are mandatory. As per your requirement, select further options.

GENERATING CUSTOMIZED MODEL

Overview Verview Left Model Verview compare Babk Added Depay Hidden Objects in the left model Tyres Selection Depay to Hidden Objects in the left model Babk Added Depay to Hidden Objects in the left model Babk Added Depay to Hidden Objects in the left model Babk Added Depay to Hidden Objects in the left model Babk Added Verview Subject Area & Babk Advanced Options Using Name Matching

- **12.** To initiate the comparison process, click Next and then click Compare. The Resolve Differences window opens.
- **13.** In the Resolve Differences window, there are the three methods to merge data models:
 - In the first method, you can merge two data models at the data model level. To merge a data model (Left Model) with another data model (Right Model), expand Model, then click arrow symbol. Or, to merge Right Model with Left Model, click arrow symbol.

Resolve Differencer		a (a) a
00000000000		
(A) (A) (B) (B)		
D 4 4 @ [75]		
	90 OFSAA - D. Raweer, Universal Shared Folder/Mdel Merce Issues/Merce185P. Datamodel.evvin	Rs OFSAA (2) - Di/Ranveer Universal Shared Folder/Hidel Merce Issues/Merce/OFS AUM Datamodel.even ***
B D Model	OPSAA	CESALE OF SAAL
Cobert Meer Model Model Solitype Symbols Solitype Symbols Solitype Symbols		000 07544
(*)		· · ·
Property Vew	OFSAA - D. (Ranveer_Liniversal_Shared_Folder/Med_Merge_Issues/Merge/ISSP_Datamodel.envin	OFSAA (2) - D: (Ranveer_Universal_Shared_Folder)/MdeLMerge_Issues)Merge(OFS_ALM_Datamodel.envin
		Frish Canol
		Finish Cancel Details

In the second method, you can merge two data models at the table level. To merge two data

models at the table level, expand Model, expand Tables, select from the menu bar. Then to merge the Right Model with the Left Model at table level, click the required table and click the arrow symbol, and repeat the same for each table. Or, to merge the Left Model with the Right Model at the table level, click the required table and click the same for each table.

Resolve Differences		0 0 0
000000000		
08 18 19 02		
Object View	OPSAA - D: (Ranveer_Universal_Shared_Folder)Mdel_Merge_Jssues(Merge)85P_Datamodel.envin	OPSAA (2) - D: (Ranveer_Universal_Shared_Folder)/Mdel_Merge_Issues(Merge(OPS_AUM_Datamodel.envin
E D Model	OFSAA	60 004 60 60 60 60 60 60 60 60 60 60
E Subtype Symbols E Tables		20
E Entty/Table	2 Bsp. Leaf. Columns	
E Entty/Table	t≱ [Bog_Load_Columns ✿ Cm_Basel_Acord_Columns ✿ Cm_Basel_ColumnRole ₽ Φ Cm_Basel_ColumnRole Φ ColumnBasel_ColumnRole Φ ColumnBasel_ColumnBasel_ColumnBasel_ColumnBasel Φ ColumnBasel_ColumnBase	
E Entry/Table	≠ Dim_Basel_Bank_Role	(c)
B Erekty/Table B Erekty/Table	Dim_Basel_Guarantor_Type Dim_Basel_Methodology	00
B Entby/Table	The Bard Bookst Tune	And
B Entry/Table	<i>≠</i>	Din Broker Deposit, Type
🕑 🧮 Entity/Table	Din_Capital_Comp_Group	400 400
E Entty/Table	Den Credit Score	
E Enkty/Table	**	010 Din Financia Viear 018 Din di Joné
E Entry/Table	Z Den Large Excessive Category	dan Salah Sa
E Erkty/Table	≠ Dm Large Exposure_Category ≠ Dm Log Band	(1)() (1)() (1)()
E Erkty/Table	≠	Other Link
E Entry/Table	≠ ≠ Dim_Criginal_Maturity_Band	Die Der Gerd Indexter
B) Entity/Table Entity/Table	Chi_Criginal_Maturity_Band Dim_Pd_Band_Listr	
E tritty/Table	≠ Dm_Risk_Type	
E fotty/Table	≠ Dim_Standard_Acct_Head	
R Posturalia	* A a state to the	
Property New	CFSAA - D: (Ranveer_Universal_Shared_Folder)/Mdel_Merge_Jssues(Merge)BSP_Datamodel.envin	CF54A (2) - D:(Rarveer_Universal_Shared_Folder)/Mail_Merge_Issues/Mergel;OFS_AUM_Datamodel.evvin
Property Vew	OrbAA - UtiRativeer_Universal_prozer (noter (noter (noter (note) (net get (power)))). University of the control of the cont	(A244 (5) - 0.189/wearTowaraeTowaraeTowar0.com/undefined@Tomarbac@iffu:27mu/Paramonerauxu
		Finish Cancel
		Details

 In the third method, you can merge two data models at the Subtype and Supertype table level To merge two data models at Subtype and Supertype level:

During an ongoing Supertype Subtype model complete comparison process, you cannot move the Supertype table directly from the Left Model or the Right Model. You must find all

the relationships between the Supertype table (parent table) and Subtype tables (child tables). Then you can move the Subtype table and after that, move the Supertype table.

How to find the Supertype or Subtype table relationship?

- i. Open the model where you are unable to move the table from left to right or from right to left. For illustration, we are considering BSP and ALM data models along with Supertype and Subtype.
- ii. In Subject Areas, create a Dummy Subject Area with any name. Right-click Dummy Subject Area name, click Properties. The Dummy Subject Area '[name]' Editor window opens.

CA [Rwin DM - (BSP_DatamodeLerwin / assassass : ER_Diagram_47797 * (Read-Only)]		
File Edit View Diagram Model Actions Tools Window Help		_ # ×
	(1) ····································	
Model Explorer	9 ×	-
Orland Orland		
Nodel [2] Subject Area	ER. Diegum, 47797 /	
Action Log		0 x
白馬吉田岡山山市の		
Constant Section Provider With PA Classifier Constant Section Provider With PA Classifier Constant Section Provider View Constant Section Prov Constant Section Prov		
Cotals A Summery	Si Non-Mart Model Oracle 30	a 100% - 4
For Help, press F1	Bal New-Mart Model Oracle 10	9 1075 - +

iii. In the Members tab, select the required Table Name and click symbol. This action moves selected object and the objects mapped to it from Available Objects list box to the Included Objects list box. In this example, in the Included Objects list box, BSP_LEAF_COLUMNS is the Supertype (parent table) and the remaining are Subtype (child tables).

🕑 🖄 🖿 🖉	Enter filter text						
ame		Auto-p	oopulate				
aaaaaaa							
M - Client Tables							
M - Payment Schedule							
MBI							
M-FTP-PFT-HM-BSP - AdminBI							
M-FTP-PFT-HM-BSP - Code & MLS							
LM-FTP-PFT-HM-BSP - Dimensions							
LM-FTP-PFT-HM-BSP - Instrument - Borrowings							
LM-FTP-PFT-HM-BSP - Instrument - Break Funding							
M-FTP-PFT-HM-RSP - Instrument - Canfloors			-				
	d objects an Table		1000	作 《 本 同 同)		Type Physical Only	
App_Filter_Dim_Map Asset_Pft_Bi_Columns	Table				There are no item	s to show.	
the second se	Table						
Basic_Instrument_Req	Table						
Basic_Ledger_Class_Req Bsp_Leaf_Columns	Table						
Captioor_Cash_Flow_Proc_Req	Table						
Cash_Flow_Edit_Req	Table						
Cash_Flow_Proc_Req	Table						
Commitment_Contracts	Table						
Currency_Master	Table						
Dim_Account	Table						
Dim_Account_Classification	Table						
Dim_Account_Purpose	Table						
Dim_Accrual_Status	Table						
Dim_Acct_Portfolio	Table						
Dim_Acct_Status	Table						
	Table						
Dim_Accumulation_Type	Table						
	Table						
Dim_Accumulation_Type	Table						
Dim_Accumulation_Type Dim_Agency_Type							
Dim_Accumulation_Type Dim_Agency_Type Dim_Amortization_Type	Table						
Dim_Accumulation_Type Dim_Agency_Type Dim_Amortization_Type Dim_Attrition	Table Table						
Dim_Accumulation_Type Dim_Agency_Type Dim_Amortization_Type Dim_Attrition Dim_Balance_Type	Table Table Table						

- iv. The Spanning Neighborhood window opens. In the Ancestors and Descendants sections, in Level, select value 1 to search for one-to-one parent-child table relationship. The list of the Subtype tables mapped to the Supertype table appears. In this example, the list of Subtype tables mapped to the Supertype table BSP_LEAF_COLUMNS appears.
- v. To include the required Subtype tables (for this Dummy Subject Area) in the comparison process, select them and click Add. Close the Spanning Neighborhood window.
- vi. In the Subject Area '[name]' Editor window, in the Included Objects list box, you can see the Supertype table (in this example, BSP_LEAF_COLUMNS) and its Subtype tables.

▶ 🔮 🏝 🥫 🛃 🦠				Enter filter text					
		Auto	-popul	ate					
666666									
- Client Tables									
- Payment Schedule									
BI									
FTP-PFT-HM-BSP - AdminBI									
1-FTP-PFT-HM-BSP - Code & MLS									
M-FTP-PFT-HM-BSP - Dimensions									
M-FTP-PFT-HM-BSP - Instrument - Borrowings									
FTP-PFT-HM-BSP - Instrument - Brea	k Funding								
ETP-PET-HM-RSP - Instrument - Canf	loors								
eral Members Definition UDP									
vailable Objects				nduded Objects 注 🔃 🗇 🥏 4幅 🛃 📷					
lame	Type Pi		1 -	Vame		Physical Only			
App_Filter_Dim_Map	Table			Bsp_Leaf_Columns	Table				
Asset_Pft_Bi_Columns	Table		E	Fsi_D_Annuity_Contracts	Table				
Basic_Instrument_Req	Table			Fsi_D_Annuity_Txns	Table				
Basic_Ledger_Class_Req	Table			Fsi_D_Borrowings	Table				
Capfloor_Cash_Flow_Proc_Req	Table		E	Fsi_D_Borrowings_Txns	Table				
Cash_Flow_Edit_Req	Table			Fsi_D_Break_Funding_Charges	Table		:		
Cash_Flow_Proc_Req	Table			Fsi_D_Capfloors	Table				
Commitment_Contracts	Table			Fsi_D_Casa	Table				
Currency_Master	Table		. 6	Fsi_D_Casa_Txns	Table				
Dim_Account	Table			Fsi_D_Credit_Cards	Table				
Dim_Account_Classification	Table		1	Fsi_D_Credit_Cards_Txns	Table		-		
Dim_Account_Purpose	Table		E	Fsi_D_Credit_Lines	Table				
Dim_Accrual_Status	Table			Fsi_D_Credit_Lines_Txns	Table				
Dim_Acct_Portfolio	Table			Fsi_D_Forward_Rate_Agmts	Table				
Dim_Acct_Status	Table			Fsi_D_Futures	Table				
Dim_Accumulation_Type	Table			Fsi_D_Fx_Contracts	Table				
Dim_Agency_Type	Table			Fsi_D_Guarantees	Table				
Dim_Amortization_Type	Table			Fsi_D_Guarantees_Txns	Table				
Dim_Attrition	Table			Fsi_D_Investments	Table				
Dim_Balance_Type	Table			Fsi_D_Investments_Txns	Table				
Dim_Bands	Table			Fsi_D_Leases	Table				
Dim_Bank_Base_Role	Table			Fsi_D_Leases_Txns	Table				
Dim_Bank_Instrument_Type	Table			Fsi_D_Ledger_Stat_Instrument	Table				
		•		Fsi D Loan Commitments	Table	Π			

- vii. Close the Subject Area '[name]' Editor window. In the data model window, click Save
- **viii.** to save the changes made in the model.
- ix. Repeat all the steps for data model merge until the Left Object Selection form (that is, repeat steps from 1 to 8).
- In the Left Objection Selection form, in the Choose Objects Using Sets list box, select all the Subject Areas along with the newly created Dummy Subject Area. Into this Dummy Subject Area, you moved the Subtype and Supertype tables in previous steps. Click Next.
- **xi.** Repeat steps 10 and 11.
- **xii.** To initiate the comparison process, click Next and then click Compare. The Resolve Differences window opens.
- xiii. In the Resolve Differences window, to merge two data models at table level, expand

Model, expand Tables, select from the menu bar. Select the Supertype table (BSP_LEAF_COLUMN in this example), and move it to left (to merge Right Model with

Left Model, click \diamondsuit arrow symbol) or right (to merge Left Model with Right Model, click \diamondsuit arrow symbol) side of the comparison as required.

In the considered illustration, BSP_LEAF_COLUMN is the parent table (Supertype) from the source table and its respective child tables (Subtype) are named as FSI_ACCID_HEALTH_INS_CO_CD, FSI_ACCID_HEALTH_INS_CO_MLS,

DIM_ACCOUNT_CLASSIFICATION and so forth. First, we moved the child table names (Subtype table names) from the source model to the target model, and then we moved the Supertype table name BSP_LEAF_COLUMN to the target model.

14. Click Finish to complete the data model merging, and then click Save to save the outputmerged model with Merged data model as a part of the output model name.

12 Build Guidelines

Changes identified and approved needs to be included in the model per the naming standards.

- Table addition/modification
 - Data model standards to be followed for each change.
 - Make sure Entity name and description are provided.
 - Make sure values are set for relevant application usage UDP's like ALM, FTP, PFT, Basel II, CREC and so on.
- Column addition/modification
 - Data model standards to be followed for each change.
 - Make sure Attribute name, description are provided.
 - Make sure values are set for relevant application usage UDP's like ALM, FTP, PFT, Basel II, CREC etc.
 - Column level UDPs named 'Custom' (NO/YES), 'Customization Reason' (Pending ER, Specific to Customer, Modified data element) and 'Type of Change' (Length, Datatype, Logical Name, Description, Addition) have to be specified for attributes added/modified as part of the extension process.

NOTE

Make sure that the Model Properties, Model Level UDP – VERSION is not changed while extending the model as per the requirement.

13 Appendix

For more information related to the usage of data models, refer to the following documents:

- Data Model Naming Standards Guide.
- Data Model Customization Review Template.

OFSAAI Support Contact Details

Raise an SR in <u>My Oracle Support</u> (MOS) if you have any queries related to EPM, ERM, and FCCM applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.

