Oracle® Retail Data Extractor for Merchandising

Implementation Guide

Release 16.0.105

E90933-01

October 2017

Oracle Retail Data Extractor for Merchandising Implementation Guide, Release 16.0.105

E90933-01

Primary Author: Nathan Young

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments

Oracle Retail Data Extractor for Merchandising Implementation Guide, Release 16.0.105

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at `http://www.oracle.com`.

Preface

The Oracle Retail Data Extractor Implementation Guide provides detailed information useful for implementing the application. It helps you to view and understand the behind-the-scenes processing of the application.

Audience

The Implementation Guide is intended for Oracle Retail Data Extractor application integrators and implementation staff.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at **http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc**.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit **http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info** or visit **http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs** if you are hearing impaired.

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 16.0) or a later patch release (for example, 16.0.1). If you are installing the base release and additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following web site:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

(Data Model documents are not available through Oracle Technology Network. You can obtain these documents through My Oracle Support.)

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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

The Oracle Retail Data Extractor (ORDE) application enables customers to extract data from Oracle Retail Merchandising System (RMS) for Oracle Retail Insights (RI) consumption.

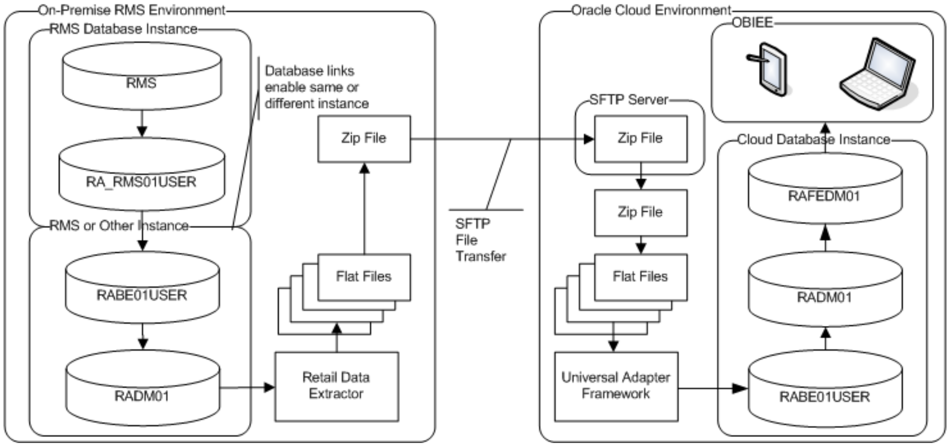
Oracle Retail Analytics (RA) was converted from being an on-premise application to a cloud based service offering named Oracle Retail Insights Cloud Services (RI). Oracle Data Integrator (ODI) mappings for extracting data from Oracle Retail Merchandising System (RMS) called as SDE Mappings (Source Dependent Extracts) were removed because RMS is an on-premise product and could not be sourced directly from the cloud.

RMS is the primary source of information for the RI application, so there was requirement to be able to extract data from RMS into flat files for uploading and importing into RI staging tables for loading the RI data warehouse.

Oracle Retail Data Extractor (ORDE) 16.0 provides this functionality to integrate with RMS and extract data in ORDE staging tables and then unload them in to respective flat files for RI consumption. The main characteristics of the Retail Data Extractor product are:

- Source Integration Solution: Retail Data Extractor provides data integration with source applications as RMS and RPM.
- Performant ETL Code: Retail Data Extractor data processing tool, ODI, offers high performance for the database batch processes on Oracle database.
- Extensibility: Retail Data Extractor ETL code can be customized and extended for client specific needs.

Figure 1–1 Overall Data Flow Diagram



2

Setup and Configuration

The Setup and Configuration chapter provides parameters for setting up Retail Data Extractor. The following sections are included:

- "Sizing Information"
- "Data Initial Load from RMS"

Sizing Information

This section provides a list of factors that should be taken into account when making sizing plans.

There are two major hardware components that make up the Retail Data Extractor physical environment:

- Middle Tier Application Server - The middle tier application server hosts software components such as Oracle WebLogic Server.
- Database - The Oracle Database stores large amounts of data that are queried in generating Oracle BI reports. The daily data loading process and report query processing process are both heavily dependent on the hardware sizing decision.

Sizing is customer-specific. The sizing of the Retail Data Extractor application is sensitive to a wide variety of factors. Therefore, sizing must be determined on an individual installation basis.

Testing is essential. As with any large application, extensive testing is essential for determining the best configuration of hardware.

Database tuning is essential, just like any other database. The Oracle database is the most critical performance and sizing component of Retail Data Extractor. As with any database installation, regularly monitoring database performance and activity levels and regularly tuning the database operation are essential for optimal performance.

Factors to Consider

- Back End Database

  Functions Used Determine Tables to be populated - Retail Data Extractor is designed to be a functional system so that some functions (such as supplier compliance or order processing) that are available do not have to be used. To the extent that some functions are not used, the amount of resources may be reduced correspondingly.

  Database Backup and Recovery - The importance of the data and the urgency with which a recovery must be made will drive the backup and recovery plan. The backup and recovery plan may have a significant impact on disk space requirements.

- Data Storage Requirements

  Transaction Volume - Sales - the higher the number of sales records, the higher the disk storage requirements and the higher the resource requirements to process.

  Positional Data - Inventory, Price, Cost - Positional data (data that is a snapshot at a specific point in time, such as inventory data "as of 9:00AM this morning") can result in very large tables. The Retail Data Extractor concept of data compression (not to be confused with database table compression) is important in controlling the disk space requirement.

  Extract, Transform, Load - Daily Processing - The daily Extraction process is a batch process of loading data from external systems into various stage tables. Disk space and other resources are necessary to support the ETL process.

- Configuration issues

  Archivelog mode - If the database is being operated in archivelog mode, additional disk space is required to store archived redo logs.

  SGA and PGA sizing - the sizing of these memory structures is critical to efficient processing of report queries, particularly when there are multiple queries running simultaneously.

  Initialization Parameters - The initialization parameter settings enable you to optimize the daily data loading and report query processing.

- Hardware Architecture

  Number and Speed of Processors - More and faster processors speed both daily data loading and report query processing in the database. The application server needs fewer resources than the database.

  Memory - More memory enables a larger SGA in the database and will reduce the processing time for report queries.

  Network - Since the data from the report queries needs to go from the back-end database to the application server, a faster network is better than a slower network, but this is a relatively low priority.

  Disk - RPMs, spindles, cache, cabling, JFS - I/O considerations are very critical to optimal performance. Selection of disk drives in the disk array should focus on speed. For example, faster RPMs, more spindles, larger cache, fiber optic cabling, JFS2 or equivalent.

  RAID - The selection of a RAID configuration is also a critical decision. In particular, RAID5 involves computations that slows Disk I/O. The key is to select the RAID configuration that maximizes I/O while meeting redundancy requirements for data protection.

  Backup and Recovery - The backup and recovery strategy drives disk configuration, disk size, and possibly the number of servers, if Dataguard or Real Application Clusters are used.

Data Seeding of Positional Facts

To report positional data correctly in the Retail Data Extractor user interface, data seeding is required if clients launch Retail Data Extractor later than source system (such as RMS). For performance reasons, it is recommended that all date range partitioned positional fact tables must seed data on the first date or first week of each partition. This avoids searching the data across partitions. Data used for seeding can come from RMS or from client legacy systems. The following are some recommendations to seed data:

- If seeding data is for new tables, you may need to provide snapshots of your positional fact data. See "Data Initial Load from RMS" on page 2-3 for how to provide initial snapshots of positional fact data.

Data Initial Load from RMS

In order to report Retail Data Extractor positional data correctly, all Retail Data Extractor positional compressed tables need to be seeded with source data (RMS) correctly before they can be loaded using Retail Data Extractor batch ETL with daily data. This seeding process is to load positional fact data for each item location combination available from RMS to Retail Data Extractor as initial data. This can be done by using following recommended approach. This approach assumes that user uses RMS as Retail Data Extractor source system and the required data are available from RMS.

Inventory Position Initial Loading

This initial inventory position data loading includes loading seeding date from RMS to RDE W_RTL_INV_IT_LC_DY_FS table. Perform the following steps:

1. In RMS, set the RMS vdate to the date that the seeding data will be used for.
2. Execute the Retail Data Extractor SDE script etlrefreshgensde.ksh to load RMS system data (including vdate) to the Retail Data Extractor temporary table RA_SRC_CURR_PARAM_G which is under Retail Data Extractor RMS batch user schema.
3. Populate the RMS IF_TRAN_DATA table with all combinations of item and location that have SOH. This can be done by using the data from the RMS ITEM_LOC_SOH table. Only columns ITEM, LOCATION, and LOC_TYPE on the IF_TRAN_DATE table will be used by the Retail Data Extractor SDE program and other columns on this table can be given any dummy value for this seeding purpose.
4. Execute the Retail Data Extractor SDE script invildsde.ksh to populate the Retail Data Extractor inventory staging table W_RTL_INV_IT_LC_DY_FS.

Pricing Initial Loading

This initial Pricing data loading includes loading seeding date from RMS to Retail Data Extractor W_RTL_PRICE_IT_LC_DY_FS table. Perform the following steps:

1. In RMS, set the RMS vdate to the date that the seeding data will be used for.
2. Execute the Retail Data Extractor SDE script etlrefreshgensde.ksh to load the RMS system data (including vdate) to the Retail Data Extractor temporary table RA_SRC_CURR_PARAM_G. This is under the Retail Data Extractor RMS batch user schema.
3. Modify the Retail Data Extractor SDE ODI program as follows:
   1. In the ODI SDE interface SDE_RetailPriceTempLoad, modify the filter on PRICE_HIST table to change the filter condition from PRICE_HIST.ACTION_DATE = TO_DATE('#RA_SRC_BUSINESS_CURRENT_DT','YYYY-MM-DD') to

PRICE_HIST.ACTION_DATE <= TO_DATE('#RA_SRC_BUSINESS_CURRENT_DT','YYYY-MM-DD').

   b. Regenerate the SDE_RetailPriceFact and MASTER_ SDE_RetailPriceFact ODI scenarios.

4. Execute the Retail Data Extractor SDE script prcildsde.ksh to populate the Retail Data Extractor Price staging table W_RTL_PRICE_IT_LC_DY_FS.
5. In the Price staging table, only keep the records with the same PROD_IT_NUM, ORG_NUM, MULTI_UNIT_QTY, but maximum DAY_DT. This ensures that the staging table only contains the latest price for each combination of item, location, and multi unit quantity.
6. Update the Pricing staging table to replace DAY_DT with the current business date that will be used for seeding data.
7. Change the filter condition back to the original condition and regenerate the two scenarios as in 3.b.

Net Cost Initial Loading

This initial Net Cost data loading includes loading seeding date from RMS to the Retail Data Extractor W_RTL_NCOST_IT_LC_DY_FS table. Perform the following steps:

1. In RMS, set the RMS vdate to the date that the seeding data will be used for.
2. Execute the Retail Data Extractor SDE script etlrefreshgensde.ksh to load the RMS system data (including vdate) to the Retail Data Extractor temporary table RA_SRC_CURR_PARAM_G. This is under the Retail Data Extractor RMS batch user schema.
3. Modify the Retail Data Extractor SDE ODI program as follows:
   a. In the ODI SDE interface SDE_RetailNetCostTempLoad, modify the filter on FUTURE_COST table to change the filter condition from FUTURE_COST.ACTIVE_DATE = to_date('#RA_SRC_BUSINESS_CURRENT_DT','YYYY-MM-DD') to FUTURE_COST.ACTIVE_DATE <= to_date('#RA_SRC_BUSINESS_CURRENT_DT','YYYY-MM-DD').
   b. Regenerate the SDE_RETAILNETCOSTFACT and MASTER_SDE_RETAILNETCOSTFACT ODI scenarios.
4. Execute the Retail Data Extractor SDE script ncstildsde.ksh to populate the Retail Data Extractor Net Cost staging table W_RTL_NCOST_IT_LC_DY_FS.
5. In the Net Cost staging table, only keep the records with the same PROD_IT_NUM, ORG_NUM, SUPPLIER_NUM, but maximum DAY_DT. This ensures that the staging table only contains the latest net cost for each combination of item, location and supplier.
6. Update the Net Cost staging table to replace DAY_DT with the current business date that will be used for seeding data.
7. Change the filter condition back to the original condition and regenerate the two scenarios as in 3.b.

Base Cost Initial Loading

This initial Base Cost data loading includes loading seeding date from RMS to Retail Data Extractor W_RTL_BCOST_IT_LC_DY_FS table. Perform the following steps:

1. In RMS, set the RMS vdate to the date that the seeding data will be used for.
2. Execute the Retail Data Extractor SDE script etlrefreshgensde.ksh to load the RMS system data (including vdate) to the Retail Data Extractor temporary table RA_SRC_CURR_PARAM_G. This is under the Retail Data Extractor RMS batch user schema.
3. Modify the Retail Data Extractor SDE ODI program as follows:
   a. In the ODI SDE interface SDE_RetailBaseCostTempLoad, modify the filter on the PRICE_HIST table to change the filter condition from PRICE_HIST.POST_DATE = TO_DATE('#RA_SRC_BUSINESS_CURRENT_DT','YYYY-MM-DD') to PRICE_HIST.POST_DATE <= TO_DATE('#RA_SRC_BUSINESS_CURRENT_DT','YYYY-MM-DD').
   b. Regenerate the SDE_RETAILBASECOSTFACT and MASTER_SDE_RETAILBASECOSTFACT ODI scenarios.
4. Execute the Retail Data Extractor SDE script cstisldsde.ksh to populate the Retail Data Extractor Base Cost staging table W_RTL_BCOST_IT_LC_DY_FS.
5. In the Base Cost staging table, only keep the records with the same PROD_IT_NUM, ORG_NUM but maximum DAY_DT. This ensures that the staging table only contains the latest base cost for each combination of item and location.
6. Update the Base Cost staging table to replace DAY_DT with the current business date that will be used for seeding data.
7. Change the filter condition back to the original condition and regenerate the two scenarios as in 3.b.

Competitor Pricing Initial Loading

This initial competitor pricing data loading includes loading seeding date from RMS to Retail

Data Extractor W_RTL_COMP_PRICE_IT_LC_DY_FS table. Perform the following steps:

1. In RMS, set the RMS vdate to the date that the seeding data will be used for.
2. Execute the Retail Data Extractor SDE script etlrefreshgensde.ksh to load the RMS system data (including vdate) to the Retail Data Extractor temporary table RA_SRC_CURR_PARAM_G. This is under the Retail Data Extractor RMS batch user schema.
3. Modify the Retail Data Extractor SDE ODI program as follows:
   a. In the ODI SDE interface SDE_RetailCompetitorPriceTempLoad, modify the filter on the COMP_PRICE_HIST table to change the filter condition from COMP_PRICE_HIST.REC_DATE = TO_DATE('#RA_SRC_BUSINESS_CURRENT_DT','YYYY-MM-DD') to COMP_PRICE_HIST.REC_DATE <= TO_DATE('#RA_SRC_BUSINESS_CURRENT_ DT','YYYY-MM-DD').
   b. Regenerate the SDE_RETAILCOMPETITORPRICEFACT and MASTER_SDE_RETAILCOMPETITORPRICEFACT ODI scenarios.
4. Execute the Retail Data Extractor SDE script comppricingsde.ksh to populate the Retail Data Extractor competitor pricing staging table W_RTL_COMP_PRICE_IT_LC_DY_FS.
5. In the competitor pricing staging table, only keep the records with the same PROD_IT_ NUM, ORG_NUM but maximum DAY_DT. This ensures that the

staging table only contains the latest competitor price for each combination of item and location.

6. Update the competitor pricing staging table to replace DAY_DT with the current business date that will be used for seeding data.
7. Change the filter condition back to the original condition and regenerate the two scenarios as in 3.b.

Unavailable Inventory Initial Loading

This initial unavailable inventory data loading includes loading seeding date from RMS to RDE W_RTL_INVU_IT_LC_DY_FS table. Perform the following steps:

1. In RMS, set the RMS vdate to the date that the seeding data will be used for.
2. Execute the Retail Data Extractor SDE script etlrefreshgensde.ksh to load RMS system data (including vdate) to the Retail Data Extractor temporary table RA_SRC_CURR_PARAM_G which is under Retail Data Extractor RMS batch user schema.
3. Populate the RMS IF_TRAN_DATA table with all combinations of item and location that have SOH. This can be done by using the data from the RMS ITEM_LOC_SOH table. Only columns ITEM, LOCATION, REF_NO_2 and TRAN_CODE on the IF_ TRAN_DATE table will be used by the Retail Data Extractor SDE program and other columns on this table can be given any dummy value for this seeding purpose. REF_NO_2 column should be the INV_STATUS column from INV_STATUS_QTY table and TRAN_CODE should be 25.
4. Execute the Retail Data Extractor SDE script unavinvildsde.ksh to populate the Retail Data Extractor unavailable inventory staging table W_RTL_INVU_IT_LC_DY_FS.

3 Internationalization

Internationalization is the process of creating software that is able to be translated more easily. Changes to the code are not specific to any particular market. Retail Data Extractor has been internationalized to support multiple languages. Retail Data Extractor serves as the source to Retail Insights Internationalization, extracting RMS languages for translation.

> **Note:** Retail Data Extractor uses DB language code and not ISO codes for all the supported languages. Retail Data Extractor will look up RMS language codes and Retail Insights will convert them to DB language codes. If, in the case a language supported by Retail Insights is not available in the source system, then the language under SRC_LANGUAGE_CODE will be used as the local language.

This section describes configuration settings and features of the software that ensure that the base application can handle multiple languages.

Translation

Translation is the process of interpreting and adapting text from one language into another. Although the code itself is not translated, components of the application that are translated may include the following:

- Graphical user interface (GUI)
- Error messages
- Reports

The following components are not translated:

- Documentation (online help, release notes, installation guide, user guide, operations guide)
- Batch programs and messages
- Log files
- Configuration tools
- Demonstration data
- Training materials

RMS is setup to translate data into 17 languages other then English. Retail Data extractor extracts this language information from RMS and is passes the same to Retail Insights for translation at user interface.

- Chinese (simplified)
- Chinese (traditional)
- Croatian
- Dutch
- French
- German
- Greek
- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Portuguese (Brazilian)
- Russian
- Spanish
- Swedish
- Turkish

Multi-Language Setup

Retail Insights data is supported in 18 languages. This section provides details of various scenarios that may come across during implementation. See "Translation" on page 3-1 for a list of supported languages.

Since multi-language data in Retail Data Extractor is dependent on the availability of the multi-language data in the source system, it is important to understand various scenarios the user may encounter. Before proceeding review the following facts about multi-language support:

- Retail Data Extractor programs extracts multi-language data from source systems.
- Retail Data Extractor securely transfers this data to Retail Insights in the form of flat files.
- A list of languages for multi-language data support can be chosen during the installation process. Please refer to the *Oracle Retail Insights Installation Guide* for more details.
- Depending on the implementation, the source system may or may not have data for particular supported language(s). For example, RMS supports Item Descriptions in multiple languages but the item's description may not be available in the translated languages.
- For source system released languages, please refer to source system Operations Guides.
- A primary language will be set up in Retail Insights for data purposes to be supported within the source system.

4 Performance

Retail Data Extractor is a high performance data Extraction utility, capable of storing and moving the daily transaction data. For any BI solution, including Retail Insights, smart decisions on how to extract bulk data will ensure that you are getting the most out of it. This chapter contains information that will help you get the best performance out of Retail Data Extractor and identifies common contributors that can weaken performance, as well as best practices that will ensure Retail Data Extractor is running in the most optimal manner.

All implementations are unique and the factors that are beneficial for one implementation may not have the same effect for all the implementations. It is a good practice to test several settings/approaches for the factors and recommendations listed below and use the ones that work best for your environment. The factors listed in this chapter are the key factors that impact performance but no absolute values or settings can be provided for implementation purposes due to the uniqueness of each environment.

Oracle Retail Data Extractor includes ODI for extraction of data from source transaction systems, and transfers this data in a secure and efficient way to the Retail Insights Data warehousing system.

ETL Programs Performance

Based on the complexity of the report, Oracle BI EE sometimes generates complex SQL, causing the Oracle Database to pick a less than optimized execution plan. In order to avoid this scenario, it is recommended that the "SQL Plan Baseline" functionality of the Oracle 12c be enabled (it is disabled by default). For more details refer to the *Oracle 12c Performance Tuning Guide*.

ETL Programs Performance

Setting ETL Program Multi-threading

Retail Data Extractor base fact stage extract programs can be configured to run using multiple threads. The default number of threads for these programs is set to one and can be configured based on requirements. For additional information on how multi-threading works, see the Program Overview chapter of the *Oracle Retail Data Extractor Operations Guide*.

1. Finalize the multi-threading strategy for the base fact stage extract programs.
2. Number of threads for each program may vary based on the data volume that program handles and resource availability. Different thread numbers should be

tested by clients during implementation to achieve optimal results from multi-threading.

3. In the C_ODI_PARAM table, update the value of the PARAM_VALUE column to the desired number of threads. This applies to all records with the value 'LOC_NUM_OF_THREAD' in the PARAM_NAME column and the name of the program that requires multi-threading set in the SCENARIO_NAME column. See an example below for scenario named SIL_Test, where the desired number of threads needs to be set to 2 from 1 (default).

   ```
   UPDATE C_ODI_PARAM
   SET PARAM_VALUE = 2
   WHERE PARAM_NAME = 'LOC_NUM_OF_THREAD'
   AND SCENARIO_NAME = 'SIL_Test'
   ```

4. If the number of thread required is more than 10, you need to modify the DDL for intermediate temp tables used by the ODI scenario. DDL changes require adding extra partitions to hold the data. The number of partitions on the intermediate temp table must be the same or higher than the required number of threads (which is the value for LOC_NUM_OF_THREADS set in the previous step).

5. The value setup in the C_ODI_PARAM (in step 3) should be bigger or equal than the max value of column ETL_THREAD_VAL in the staging tables. Otherwise, some records could get missing.

ODI Configuration

ODI must be configured prior to implementing Retail Data Extractor. See the *Oracle Retail Data Extractor Administration Guide* for details on configuring ODI.

ETL Batch Scheduling

- Set up the proper dependencies between the applications to ensure resources are fully utilized, which helps the nightly batch finish earlier.
- Retail Data Extractor extraction programs (SDE programs) must not wait for all the extraction programs (sde) to finish before starting. Some of them can start executing in parallel if they are not dependent on other staging tables. For more information on setting up dependencies, refer to the *Oracle Retail Data Extractor Operations Guide*.
- Ensure that your source applications batch is optimized. Retail Data Extractor runs towards the end of the nightly batch. Retail Data Extractor jobs are often the last jobs to start due to the dependencies on the source system jobs, so Retail Data Extractor is often the last to finish. Optimizing the source applications batch helps Retail Data Extractor jobs to start earlier.

Data Base Configuration

Retail Data Extractor is built on Oracle Database 12c and must be optimized and configured for a retailers' needs. Refer to the Setting up your Data Warehouse System chapter of the *Oracle 12c Data Warehouse Guide*.

Adequate Hardware Resources

ETL programs performance is dependent on the hardware resources. For more information, see Chapter 2, "Setup and Configuration".

Leading Practices

Customizations

Changes and modifications to the Retail Data Extractor delivered code or development of new code is considered customization. Retail Data Extractor does not support custom code developed by clients unless the issue related to customization can be recreated using Retail Data Extractor delivered objects. Listed below are recommendations that will help you in maintaining Retail Data Extractor code:

- Naming convention: it is recommended that you use a good and consistent naming convention when customizing Retail Data Extractor delivered code or building new code in the Retail Data Extractor environment.

  This strategy is helpful in identifying custom code and also helps when merging a retailer's Retail Data Extractor repository with future releases of the Retail Data Extractor repository. There is a possibility of losing customizations to Retail Data Extractor provided ODI scripts repository, if the customized code uses the same object/script names that are used by Retail Data Extractor.

- As a best practice, keep all the documentation up-to-date for capturing any changes or new code that has been developed at a site. For example, if table structure has been customized, create or update the custom Data Model Guide with these changes.

ODI Best Practices

For customizations to existing ODI code or while creating new ODI code, refer to the *ODI Best Practices Guide* included with your product code.

Batch Schedule Best Practices

The following best practices are recommended for Retail Data Extractor:

Automation

The batch schedule should be automated as per the *Oracle Retail Data Extractor Operations Guide*. Any manual intervention should be avoided.

Recoverability

Set up the batch schedule in such a manner that the batch can resume from the point where it failed.

Batch Efficiency

Keep revisiting the batch timings on a periodic basis to identify the candidates for performance improvements.

5

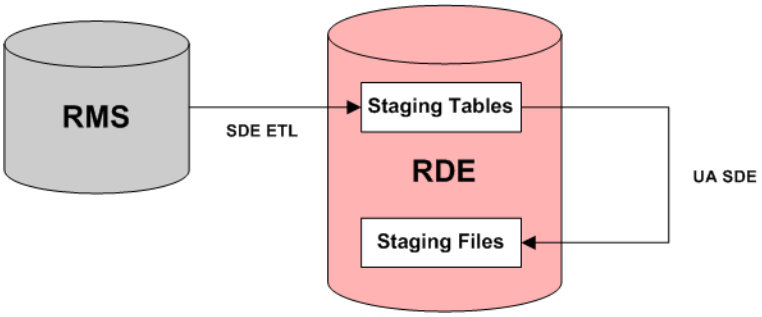
Retail Data Extractor Universal Adapter

This chapter describes the process of implementing the Retail Insights Universal Adapter Framework.

Overview of Retail Insights Universal Adapter Framework

The Retail Data Extractor BI product offering is intended to work closely with Oracle Retail's transactional schema, RMS. As such, the earlier version of Retail Insights shipped with source dependent extraction (SDE) routines designed to move data from RMS tables into RI staging tables. The new version of Universal Adapter for RDE loads extracted data from RMS to RDE Stage tables which is then passed to Retail Insights in the form of text files. (see Figure 5–1, "RDE to Retail Insights Staging Data Flow").

Figure 5–1 RDE to Retail Insights Staging Data Flow



The goal of the RDE Universal Adapter Framework (UAF) is to simplify the process of moving source dependent extracts into Retail Insights staging tables for customers . In terms of implementation, the UAF first requires RMS extracted data in RDE staging tables and the Universal Adapter unloader program generates pipe ('|') separated value (DAT file) text file extracts to be provided as inputs to RI. All date column should use a format of "YYYY-MM-DD;HH24:MI:SS". Once the DAT files are in place, these can be transferred to the Retail Insights system by zipping them into a single directory and FTP the zip file to Retail Insights server mount point.

- The control files(*.ctx) can be created by the Universal Adapter unloading program in case there is a difference in the datatype of a stage table between RDE environment and RI environment.
- Customers can use this utility to generate dat files and FTP the data in a more secured way. The earlier approach of DBlink is not required anymore, thereby providing higher security

Universal Adapter Execution

Execute the Retail Data Extractor SDE script rtluasde.ksh and rabeuasde.ksh to run the Universal Adapter for generating the staging DAT files. Script rabeuasde.ksh is for the target tables owned by RI batch user and script rtluasde.ksh is for the target tables owned by RI data owner.

Syntax:

```
rtluasde.ksh <Target table> <RA_GEN_CTXFILE>[Optional]
rabeuasde.ksh <Target table> <RA_GEN_CTXFILE>[Optional]
```

The parameters used are

- Target Table: Table for which you want to generate the DAT File.
- RA_GEN_CTXFILE: This parameter if passed as TRUE will generate CTX file in addition to .dat file.

  Possible Value FALSE Default: TRUE

By Default Universal adapter generates only the dat files (with data) and generation of ctx control file is disabled. In order to generate the ctx file for adjusting datatypes between RDE and RI environments please pass the second parameter as explained above.

> **Note:** The generation of a ctx file is required only once to adjust the ctl file in Retail Insights.

Batch Logging:

The batch for Universal Adapter will have the same logging logic as other RI batch programs. The execution status can be found in RI batch maintenance table C_LOAD_DATES and ODI Operator.

File Transfer to Retail Insights

Universal adapter generates the data and control files (*.dat and *.ctx) which needs to be transferred to Retail Insights for loading Facts and Dimensions.

Automate the below steps in order to transfer the files to Retail Insights.

1. Navigate the to data/staging directory where Universal Adapter files are generated.

   ```
   Path:  cd <$MMIHOME>/data/staging
   ```

2. Generate the tar file with all the files in data/staging directory.

   Command: tar cf $MMHOME/data/staging/< Merch_Extract_date>.tar $MMHOME/data/staging/*.dat $MMHOME/data/staging/*.dat.ctx

3. Connect to <server> port 22.
4. Log in with the SFTP User credentials.
5. Transfer all data files to the directory /<SFTP User>/EXPORT.
6. Create a directory called COMMAND under /<SFTP User>if it does not already exist.
7. Change to the /<SFTP User>/COMMAND directory.

8. Transfer an empty file called COMPLETE.

6

Frequently Asked Questions

The following issues may be encountered while implementing Retail Data Extractor. The accompanying solutions will help you work through the issues.

Issue:
Why the extract program performance is not improving even after using ODI multi-threading?

Solution:
This can occur because of several reasons. Check the following settings:

- The number of threads must be appropriate for the hardware and data volume.
- The number of partitions on the intermediate temp table must be equal to or higher than the number of threads.

Issue:
Does Retail Data Extractor only expect one RDWT file for sales from Oracle Retail Sales Audit (ReSA)?

Solution:
The ReSA batch generates the RDWT file based on store day level. It always generates one RDWT file per store/day. The ReSA batch module saexpdw.pc is used to fetch all corrected sale and return transactions that do not have Retail Data Extractor errors from the ReSA database tables for transmission to Oracle Retail Data Extractor. The data will be sent at the store day level.

However, Retail Data Extractor expects only one RDWT file per day. Therefore, the customer has to consolidate all the stores RDWT files for a day into one RDWT file before loading to Retail Data Extractor.

Issue:
How do you execute the failed threads for multi-threading programs?

Solution:
This can be done by using the batch log in the C_LOAD_DATES table. Table C_LOAD_DATES has a record for the execution status of each batch at thread number level. Same thread of a same batch cannot be executed twice unless the log record is deleted manually. This provides a possibility to re-execute only one thread for a case when only one thread fails and other threads complete successfully.

To re-execute failed threads, the user can manually delete the threads that need to be executed and keep all other threads untouched in the C_LOAD_DATES table. Then the user can start the batch again. When the re-execution is done, the program will show errors in the UNIX console, but the threads that need to be re-executed should complete successfully. The error in the UNIX console is for the re-execution of the threads that completed successfully in the first execution, so it can be ignored.

Issue:
While running the packages it is possible that there could be a scenario failure with error "Variable has no value" due to ODI out of memory.

Solution:
If this error occurs, verify the values of the following two parameters are set as below (for more details refer to the *Oracle Retail Data Extractor Administration Guide*) and regenerate the scenario that is failing.

- ODI_INIT_HEAP=256M
- ODI_MAX_HEAP=1024M