

**Oracle[®] Retail Data Warehouse
Operations Guide Addendum
Release 11.0.1
December 2005**

Copyright © 2005, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, ProfitLogic and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

- Preface..... v**
 - Audiencev
 - Related Documentsv
 - Customer Supportvi
- 1 Introduction..... 1**
- 2 Process to Update Records in RDW Change 3**
 - Process to Update Records in RDW3
- 3 General Fact Processing Change 5**
 - General Fact Processing.....5
- 4 rdw_config.env Settings Change 7**
 - rdw_config.env Settings7
- 5 Program Reference Lists Change 9**
 - Fact Programs9
 - Program Type and Operation Type Descriptions.....10
 - Fact Types.....10

Oracle Retail Operations Guides are designed so that you can view and understand the application's 'behind-the-scenes' processing, including such information as the following:

- Key system administration configuration settings
- Technical architecture
- Functional integration dataflow across the enterprise

Audience

Anyone with an interest in developing a deeper understanding of the underlying processes and architecture supporting Oracle Retail Data Warehouse functionality will find valuable information in this guide. There are three audiences in general for whom this guide is written:

- Business analysts looking for information about processes and interfaces to validate the support for business scenarios within RDW and other systems across the enterprise.
- System analysts and system operations personnel:
 - Who are looking for information about RDW's processes internally or in relation to the systems across the enterprise.
 - Who operate RDW regularly.
- Integrators and implementation staff with overall responsibility for implementing RDW.

Related Documents

You can find more information about RDW in these resources:

- RDW Data Model
- RDW Installation Guides
- The latest RETL Programmer's Guide
- RDW User Guide
- RDW online help

Customer Support

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Retek customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method Contact Information

E-mail support@rettek.com

Internet (ROCS) rocs.retek.com

Retek's secure client Web site to update and view issues

Phone +1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia +1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)

France 0800 90 91 66

Hong Kong 800 96 4262

Korea 00 308 13 1342

United Kingdom 0800 917 2863

United States +1 800 61 RETEK or 800 617 3835

Mail Oracle
Customer Support
950 Nicollet Mall
Minneapolis, MN 55403

When contacting Customer Support, please provide:

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

Introduction

The information in this document reflects modifications and updates to the latest Oracle Retail Data Warehouse (RDW) Operations Guide. Each section title in this document corresponds to a section title in the RDW Operations Guide.

Process to Update Records in RDW Change

Process to Update Records in RDW

Because RETL does not currently support a database update operator, the actual updates into the database are accomplished through one of two processes, depending upon whether a normal update or an incremental update is occurring. An incremental update (applicable to fact processing only) sums the incoming records with the old records in the target table and replaces those old records with the new summed records. A normal update uses incoming records to replace old records in the target table. There are two ways to perform normal updates. The first way is 'delete, insert' logic. The second way is to the Oracle 'merge' statement. Oracle 'merge' is used on larger volume tables to improve performance.

Note: The temporary tables that are mentioned throughout this operations guide are always dropped by the batch code every day, after the various batch processes that use the temporary tables complete.

Normal Update Description

Update using 'delete, insert' logic:

1. The dataset (containing the new records) is written into a temporary table.
2. This temporary table is used to determine which of the old update records in the target table should be deleted.
3. The old records are deleted from the target table.
4. The new records are inserted into the target table.

Update using 'merge' logic:

1. The dataset (containing the new records) is written into a temporary table.
2. This temporary table is used to determine which of the old update records in the target table should be merged (updated).
3. The old records are updated and new records are inserted into the target table by using merge statement.

Incremental Update Description

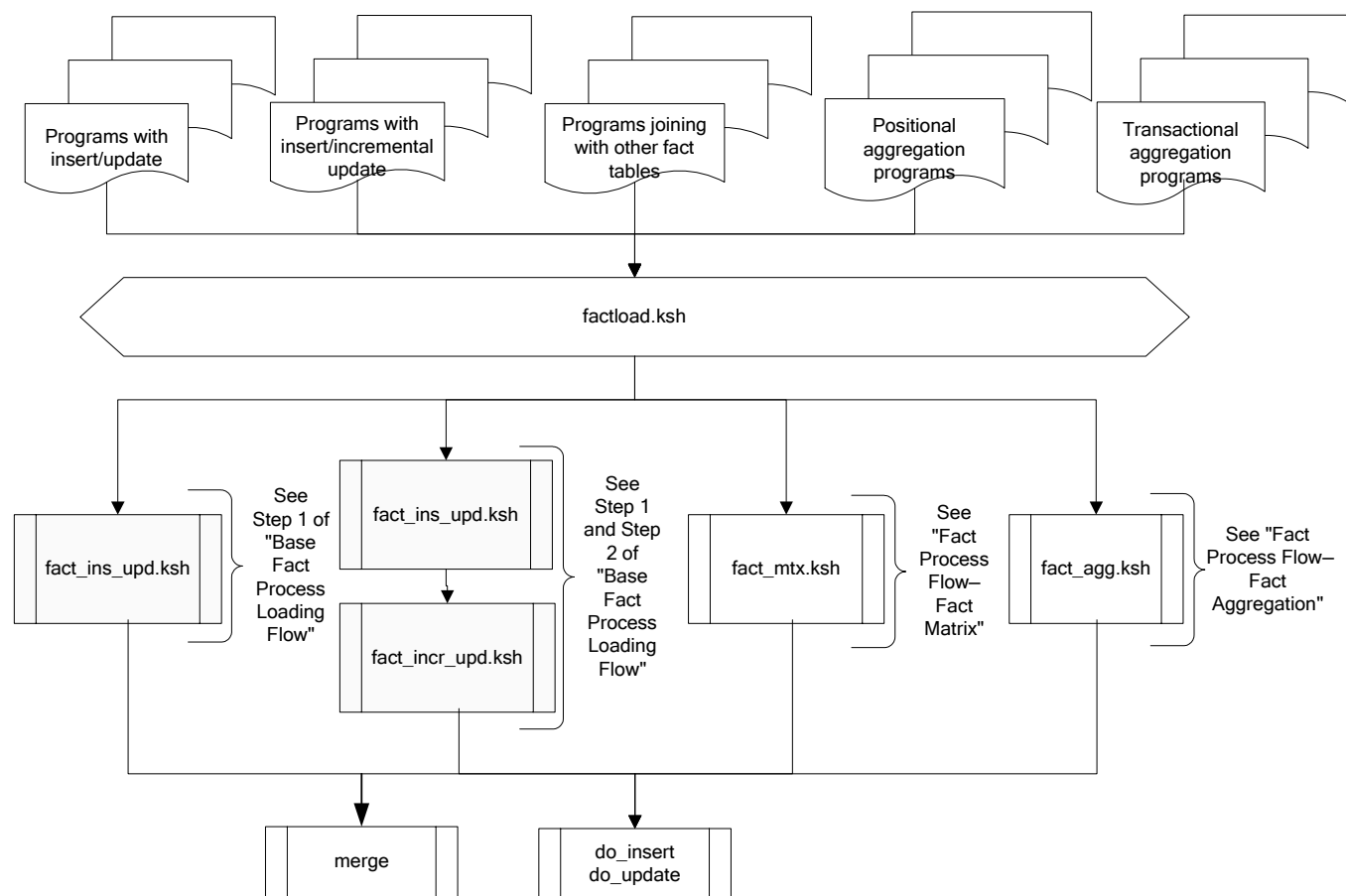
The 'incremental' process has not changed with RDW 11.0.1 patch.

General Fact Processing Change

General Fact Processing

The following diagram illustrates the fact process flow in RDW. The flow proceeds from the fact programs (that require the use of sub-libraries) to the factload.ksh. This library interprets the needs of the programs in order to direct them to call the correct sub-library or sub-libraries. Essentially, Factload.ksh serves as the regulatory fact program for this library. Note that almost every fact program that uses sub-libraries must call factload.ksh so that it can be properly directed. Once the applicable sub-library has processed the program, the system can make the correct changes to RDW's fact tables. The very few standalone modules that do not use factload.ksh are omitted from the diagram.

The flow diagrams described below illustrate specifically how, and in what context, data is processed within each applicable Korn shell sub-library.



Fact Process Flow—General

rdw_config.env Settings Change

rdw_config.env Settings

- LOOK_TYPE refers to the method that RETL uses to look up item key. This setup depends on the number of items that clients have, the hardware capacity, as well as a number of other variables. Clients should test the two methods in their environment to see which method performs best. As a rule, 'lookup' typically performs better for clients with a smaller number of items (less than 1M) and 'innerjoin' typically performs better for clients with a larger number of items (more than 1M). The default value for this setting is 'lookup'.
 - **LOOK_TYPE=innerjoin** looks up item key using RETL 'innerjoin' operator.
 - **LOOK_TYPE=lookup** looks up item key using RETL 'lookup' operator.
- PARA_NUM refers to the parallel degree used in SQL statement. The parallel degree can affect the performance of a module. Currently it is only used for inventory aggregation module invblddm.ksh. The setup of this value depends upon client system capacity. Clients should perform testing on different values to select the best. It is suggested this value be half the number of CPUs available on the server if this server is only used for RDW. To disable parallelism, set this setting to a value of '0'. The initial value is set to '0'.

Program Reference Lists Change

Fact Programs

Program	Functional Area	Module Type	External Data Source	Source Table or File	Schema File	Target File or Table	Program_Control DM.program_type	Program_Control DM.operation_type	Arguments	Notes
invblddm.ksh	Inventory Position	Positional Aggregation				INV_SBC_LD_DM	FACT_AGG_POS	UPDATE_ME		
stlblwdm.ksh	Stock Ledger	Base fact with update		stlblwdm.txt	stlblwdm.schema	INV_VAL_SBC_LW_DM	BASEFACT_UPD	UPDATE_ME	\$MMHOME/data/stlblwdm.txt	This module runs weekly.
prdisldm.ksh	Item-Supplier-Location Cross Dimension	Dimension Load	RMS	prdisldm.txt	prdisldm.schema	PROD_ITEM_SUPP_LOC_DM	DIM_MTX	UPDATE		

Program Type and Operation Type Descriptions

Fact Types

Program type	Program type description	Operation type	Operation type description
BASEFACT_UPD	<ul style="list-style-type: none"> Used for modules that insert new records, and/or update the current records. A temporary table is used to hold the current day's data to be used in the inserts and updates. 	UPDATE	<ul style="list-style-type: none"> Records are updated from the temporary table to the target table. The temporary table is dropped.
		UPDATE_L	<ul style="list-style-type: none"> Records are inserted into a temporary table. The temporary table is kept around for use by the module itself and another module later in the scheduling flow. The module itself performs updates and inserts based on the temporary table created by the library. It needs to update its program status to 'completed' and drops the temporary table if no aggregation is needed later. All compressed day level tables use this operation type.
		UPDATE_A	<ul style="list-style-type: none"> Records are updated/inserted from the temporary table to the target table. The temporary table is retained for use by another module later in the scheduling flow.

Program type	Program type description	Operation type	Operation type description
		UPDATE_ME	<ul style="list-style-type: none">• Records are update/inserted from the temporary table to the target table by using Oracle merge statement.• The temporary table is dropped.

Program type	Program type description	Operation type	Operation type description
FACT_AGG_POS	<ul style="list-style-type: none"> • Used for modules that hold positional data for time and aggregates from a lower level to a higher level in the product hierarchy only. • A temporary table from the previous module in the aggregation flow is used to hold the current day's data. 	INSERT	<ul style="list-style-type: none"> • Records are updated /inserted on the target table by calling merge function based on the temporary table created by the previous module in the aggregation flow. • The temporary table is dropped.
		UPDATE_F	<ul style="list-style-type: none"> • A temporary table is created by parameters specified by the module. • Records are updated on the target table based on the temporary table. • The temporary table is dropped. • Any existing temporary tables from previous modules are dropped.
		UPDATE_G	<ul style="list-style-type: none"> • A temporary table is created by parameters specified by the module, including the standard aggregation for product hierarchy. • Records are updated on the target table based on the temporary table. • The temporary table is retained for another module in the flow. • Any existing temporary table from previous modules is dropped.

		UPDATE_GF	<ul style="list-style-type: none"> • A temporary table is created by parameters specified by the module, including the standard aggregation for product hierarchy. • Records are updated on the target table based on the temporary table. • The temporary table is dropped. • Any existing temporary table from previous modules is dropped.
		UPDATE_KO	<ul style="list-style-type: none"> • A temporary table is created by parameters specified by the program. • Records are updated on the target table based on the temporary table. • The temporary table is dropped. • Any existing temporary table from previous programs will be retained for another program in the flow.
		UPDATE_ME	<ul style="list-style-type: none"> • Records are update/inserted from the temporary table to the target table by calling merge function. • The temporary table is dropped.