

Retek® Data Warehouse™ 10.2.5

Operations Guide Addendum

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA
888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000
Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom
Switchboard:
+44 (0)20 7563 4600
Sales Enquiries:
+44 (0)20 7563 46 46
Fax:
+44 (0)20 7563 46 10

The software described in this documentation is furnished under a license agreement, is the confidential information of Retek Inc., and may be used only in accordance with the terms of the agreement.

No part of this documentation may be reproduced or transmitted in any form or by any means without the express written permission of Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403, and the copyright notice may not be removed without the consent of Retek Inc.

Information in this documentation is subject to change without notice.

Retek provides product documentation in a read-only-format to ensure content integrity. Retek Customer Support cannot support documentation that has been changed without Retek authorization.

The functionality described herein applies to this version, as reflected on the title page of this document, and to no other versions of software, including without limitation subsequent releases of the same software component. The functionality described herein will change from time to time with the release of new versions of software and Retek reserves the right to make such modifications at its absolute discretion.

Retek® Data Warehouse™ is a trademark of Retek Inc.

Retek and the Retek logo are registered trademarks of Retek Inc.

This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws. In the event of publication, the following notice shall apply:

©2005 Retek Inc. All rights reserved.

All other product names mentioned are trademarks or registered trademarks of their respective owners and should be treated as such.

Printed in the United States of America.

Customer Support

Customer Support hours

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@retек.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	Retek Customer Support Retek on the Mall 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.

Contents

Chapter 1 – Fact processing changes	1
Process to update records in RDW	1
Normal update description	1
Chapter 2 – Fact data concepts change	3
General fact processing	3
Chapter 3 – Program flow diagrams change	5
rdw_config.env settings	5
Chapter 4 – Program reference lists	7
Fact programs	7
Program type and operation type descriptions	8
Fact types	8

Chapter 1 – Fact processing changes

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 use the Oracle ‘merge’ statement. Oracle ‘merge’ is used on larger volume tables to improve performance.

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:

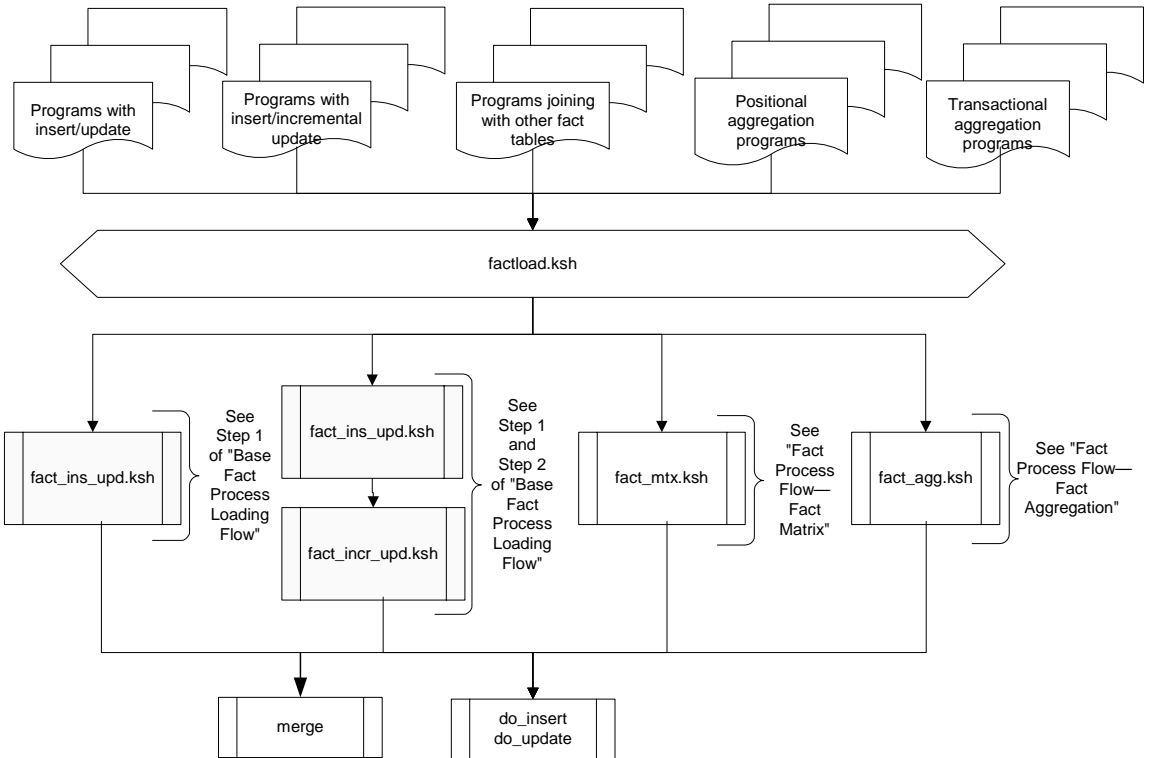
1. The dataset (containing the new records) is written into a temporary table.
2. This temporary table is used to determine which of the old update records in the target table should be merged (updated)
3. The old records are updated and new records are inserted into the target table by using merge statement.

Chapter 2 – Fact data concepts change

General fact processing

The following diagram illustrates the fact process flow in RDW 10.2. 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. 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 not shown in the diagram.

The flow diagrams described below illustrate specifically how, and in what context data is processed within each applicable Korn shell sub-library. Thus, adjacent to each sub-library in the diagram below is a callout that refers to the specific process flow diagrams (and any applicable steps therein).



Chapter 3 – Program flow diagrams 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 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 on client system capacity. Clients should test it with different values to select the best degree of parallelism for them. 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 to a value of '0'. The default value is set to '0'.

Chapter 4 – Program reference lists

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 Aggregati on				INV_SBC_LD_DM	FACT_AGG_POS	UPDATE_ME		
stlblwdm.ksh	Stock Ledger	Base fact with update		stlblwdm.txt	stlblwdm.s chema	INV_VAL_SBC_LW_DM	BASEFACT_UPD	UPDATE_ME	\$MMHOME/d ata/stlblwdm.txt	This module runs weekly.

Program type and operation type descriptions

Fact types

BASEFACT_UPD

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 has 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 kept around for use by another module later in the scheduling flow.
		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.

FACT_AGG_POS

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 will be 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 kept around for another module in the flow.Any existing temporary table from previous modules is dropped.

Program type	Program type description	Operation type	Operation type description
		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_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.