

Retek[®] Invoice Matching[™] 11.0.3

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[®] Invoice Matching[™] 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This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws.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@retex.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 – Batch processes changes.....	1
Batch names and Java packages.....	1
Functional descriptions and dependencies	2
Batch designs.....	8


Chapter 1 – Batch processes changes

In “Chapter 7 – Batch processes”, all references to `ReceiverAdjustmentService` have been removed. They do not belong in the ReIM Operations Guide. The `ReceiverAdjustmentService` is called by `ReasonCodeActionRollupService` batch. For purposes of clarification, the references to `ReceiverAdjustmentService` within Chapter 7 are shown below but include a strikethrough line through them.

The functional description and the design of `ReasonCodeActionRollupService` has been modified. The modified functional description of the `ReasonCodeActionRollupService` is in the table within the ‘Functional descriptions and dependencies’ section below. The modified design of `ReasonCodeActionRollupService` is in the ‘Batch designs’ section below.

Batch names and Java packages


The following table describes ReIM’s batch processes and their associated Java packages. The table’s order reflects the dependencies that exist among the ReIM batch processes but does not include any dependencies that exist between ReIM and the merchandising system it interacts with.

Batch name	Batch process	Package
 Note: This batch process is applicable <i>only</i> for those retailers using RMS 10.1 and earlier.	TermsRankingService	com.retek.reim.services
Batch purge	BatchPurge	com.retek.reim.purge
Discrepancy purge	DiscrepancyPurge	com.retek.reim.purge
EDI invoice upload	Ediupinv	com.retek.reim.batch.ediupinv.threading
Receiver adjustment	ReceiverAdjustmentService	com.retek.reim.services
Auto-match	AutoMatchService	com.retek.reim.services.matching
Receipt write-off	ReceiptWriteOff	com.retek.reim.services
Reason code action rollup	ReasonCodeActionRollupService	com.retek.reim.services
Disputed credit memo action rollup	DisputedCreditMemoResolutionRollupService	com.retek.reim.services
Resolution posting	ResolutionPostingService	com.retek.reim.services
EDI invoice download	EdiDownload	com.retek.reim.batch.ediupinv

Batch name	Batch process	Package
Complex deal upload	ComplexDealUpload	com.retek.reim.batch.deal
Fixed deal upload	FixedDealUpload	com.retek.reim.batch.deal

Functional descriptions and dependencies

The following table summarizes ReIM's batch processes and includes both a description of each batch process's business functionality and its batch dependencies:

Batch processes	Details	Batch dependencies
Terms ranking (TermsRankingService)  Note: This batch process is applicable <i>only</i> for those retailers using RMS 10.1 and earlier.	Retailers send terms ranking files to ReIM on a periodic (usually monthly) basis. ReIM has built an API to read this file and populate the terms ranking table.	
Batch purge (BatchPurge)	This process deletes data from database tables while maintaining database integrity. This process deletes records from the ReIM application that meet certain business criteria (for example, records that are marked for deletion by the application user, records that linger in the system beyond certain number of days, and so on).	
Discrepancy purge (DiscrepancyPurge)	The discrepancy purging program deletes data from database tables while maintaining database integrity. This program deletes records from ReIM that have discrepancies of zero.	
EDI invoice upload (ediupinv)	This batch process uploads merchandise, non-merchandise invoices, credit notes, debit memos, and credit note requests from the EDI into the invoice-matching tables.	

Batch processes	Details	Batch dependencies
Receiver adjustment (ReceiverAdjustmentService)	The process compares the unit cost and/or quantity received for the item on the shipment with the expected unit cost and/or quantity on the IM_RECEIVER_COST_ADJUST and/or IM_RECEIVER_UNIT_ADJUST tables. If a match exists, the receiver cost and/or unit adjustment has occurred in RMS (or the equivalent merchandising system). As a result, the process sets the 'pending adjustment' flag on IM_INVOICE_DETAIL table to false for the invoice line. The reason code actions are only rolled up for an invoice if no invoice lines on the invoice have any pending adjustments.	The receiver adjustment needs to be run after a retailer written check for Receiver Cost Adjustments (RCAs) and Receiver Unit Adjustments (RUAs).
Auto-match (AutoMatchService)	Auto-match is a system batch process that attempts to match invoices to receipts without manual intervention. Invoices that are in ready for match, unresolved, or multi-unresolved status are retrieved from the database to be run through the auto-match algorithm. The processing consists of two levels – summary and detail.	EDI upload (Invoice Matching) Receipt upload (Merchandising system, such as RMS)

Retek Invoice Matching

Batch processes	Details	Batch dependencies
Receipt write-off (ReceiptWriteOff)	In order for retailers to track received goods not invoiced, they must have the ability to 'write-off' these goods for financial tracking. ReIM has a system parameter (which can be overwritten at the supplier level) defining the maximum amount of time an open, non-fully matched receipt will be available for matching. Every time the Receipt write-off process is run, each non-fully matched open receipt received date is compared with the current date minus the system parameter. If the received date is before this difference, the receipt is 'written-off' and the invoice match status is closed.	Auto-match and any associated processing must run prior to this batch processing

Batch processes	Details	Batch dependencies
Reason code action rollup (ReasonCodeActionRollupService)	<p>This batch process sweeps the action staging table and creates debit and credit memos as needed. Only a single debit or credit memo is created per invoice, with line details from all related actions. This process deletes these records when completed; they are deleted after posting. The process compares the unit cost and/or quantity received for the item on the shipment with the expected unit cost and/or quantity on the IM_RECEIVER_COST_ADJUST and/or IM_RECEIVER_UNIT_ADJUST tables. If a match exists, the receiver cost and/or unit adjustment has occurred in RMS (or the equivalent merchandising system). As a result, the process sets the 'pending adjustment' flag on IM_INVOICE_DETAIL table to false for the invoice line. The reason code actions are only rolled up for an invoice if no invoice lines on the invoice have any pending adjustments.</p> <p>The action staging table is used during posting to post the reason code actions to the financial staging table.</p>	Receiver adjustment must occur prior to this batch process.

Batch processes	Details	Batch dependencies
Disputed credit memo action rollup (DisputedCreditMemoResolutionRollupService)	<p>The disputed credit memo action rollup process checks the records on the IM_REVERSAL_RESOLUTION_ACTION table and rolls up the credit memo detail lines by document/item/reason code. The rollup occurs only if all lines on a disputed credit memo have been completely resolved (that is, no cost or quantity discrepancy records remain for the credit memo).</p> <p>After the rollup, a new set of detail lines associated with the resolution reason codes replace the original set of detail lines associated with the debit reason codes on the IM_DOC_DETAIL_REASON_CODES table.</p>	The disputed credit memo action rollup must occur before resolution posting and after receiver adjustment.
Resolution posting (ResolutionPostingService)	<p>A recurring resolution posting process retrieves all matched invoices and approved documents.</p> <p>For each invoice, the batch process engages in the following high-level steps:</p> <ol style="list-style-type: none"> 1. Performs any resolution actions (for example, instigates the creation of payment documents). 2. Calls the posting process to write applicable financial accounting transactions to the financials staging table, IM_FINANCIALS_STAGE. 	

Batch processes	Details	Batch dependencies
EDI invoice download (EdiDownload)	<p>The EdiDownload module creates a flat file to match the EDI invoice download file format. The module retrieves all header, detail and non-merchandise information and formats the data as needed.</p> <p>In other words, the EDI invoice download process retrieves debit memos, credit note requests, and credit memos in 'approved' status from the resolution posting process and creates a flat file. The client converts the flat file into an EDI format by the client and sends it via the EDI invoice download transaction set.</p>	Auto-match must run prior to the EDI invoice download.
Complex deal upload (ComplexDealUpload)	This module reads data from RMS staging tables, creates credit memos, debit memos, and credit note requests out of the data, and stores the supporting deal data on an ReIM table for later use during posting.	The RMS staged data must be purged after the upload.
Fixed deal upload (FixedDealUpload)	This module reads data from RMS staging tables, creates credit memos, debit memos, and credit note requests out of those, and stores the supporting deal data on an ReIM table for later use during posting.	The RMS staged data must be purged after the upload.

Batch designs

Receiver adjustment batch design

Overview

To resolve a cost discrepancy, the user can select a 'Receiver Cost Adjustment' action from the cost resolution screen. Similarly, to resolve a quantity discrepancy, the user can select a 'Receiver Unit Adjustment' action from the quantity resolution screen. The actions are written to the IM_RESOLUTION_ACTION table in an unrolled status with the amount of adjustment. The IM_INVOICE_DETAIL table also receives a flag that signifies 'pending adjustment' for the invoice line.

At the same time, the actions are written to the IM_RECEIVER_COST_ADJUST and IM_RECEIVER_QTY_ADJUST tables to indicate the expected receiver adjustment amount on the RMS (or equivalent merchandising system) side. In sum, these two tables serve as the staging tables for the RMS (or equivalent merchandising system) process to actually perform the adjustment. For a receiver cost adjustment, IM_RECEIVER_COST_ADJUST holds the order unit cost for the item after the adjustment. For a receiver unit adjustment, IM_RECEIVER_UNIT_ADJUST holds the received quantity for the item on the shipment after the adjustment.

The process compares the unit cost and/or quantity received for the item on the shipment with the expected unit cost and/or quantity on the IM_RECEIVER_COST_ADJUST and/or IM_RECEIVER_UNIT_ADJUST tables. If a match exists, the receiver cost and/or unit adjustment has occurred in RMS (or the equivalent merchandising system). As a result, the process sets the 'pending adjustment' flag on IM_INVOICE_DETAIL table to false for the invoice line. The reason code actions are only rolled up for an invoice if no invoice lines on the invoice have any pending adjustments.

Because ReIM cannot control when and how the receiver adjustments are happening on the RMS side (or the equivalent merchandising system), records written to the IM_RECEIVER_COST_ADJUST and IM_RECEIVER_UNIT_ADJUST tables are considered final. As a result, when the user resolves a cost or quantity discrepancy, the receiver adjustment must fully resolve a discrepancy before the user leaves the screen, and there should be no re-route actions involved. On the RMS side, the amount of adjustment must be exactly the same as expected.

The IM_PARTIALLY_MATCHED_RECEIPTS table holds the amount of a receipt item that has been matched during invoice matching. The quantity received on the SHIPSKU table subtracts the quantity matched on the IM_PARTIALLY_MATCHED_RECEIPT table, giving the available to match quantity for the receipt item. Auto-match, summary matching, detail matching and quantity discrepancy resolution processes all keep track of the matched quantity bucket to determine how much of the receipt item has already been matched and how much of the receipt item remains available to be matched. In the case of a Receiver Unit Adjustment, the IM_PARTIALLY_MATCHED_RECEIPTS table is updated to reserve the entire remaining unmatched bucket for the receipt item. This logic prevents the adjusted receipt quantity from being used for any other matching or quantity resolutions.

Assumptions and scheduling notes

The receiver adjustment process must be run after a client written check for Receiver Cost Adjustments (RCAs) and Receiver Unit Adjustments (RUAs).

Primary tables involved

- ~~IM_COST_DISCREPANCY~~
- ~~IM_QTY_DISCREPANCY~~
- ~~IM_RECEIVER_COST_ADJUST~~
- ~~IM_RECEIVER_UNIT_ADJUST~~
- ~~IM_RESOLUTION_ACTION~~
- ~~IM_INVOICE_DETAIL~~
- ~~IM_DOC_HEAD~~
- ~~IM_PARTIALLY_MATCHED_RECEIPTS~~

Reason code action rollup batch design

Overview

Reason code actions are resolutions assigned at the discrepancy line level. A number of fixed actions are available to resolve a line item discrepancy; the specific results depend on the action.

The resolution posting process sweeps the IM_RESOLUTION_ACTION table and creates debit and credit memos as needed. Only a single debit or credit memo is created per invoice, with line details from all related actions.

This process does not delete these records when completed; rather, they are deleted after posting.

The action staging table is used during posting to post the reason code actions to the financial staging table.

To resolve a cost discrepancy, the user can select a 'Receiver Cost Adjustment' action from the cost resolution screen. Similarly, to resolve a quantity discrepancy, the user can select a 'Receiver Unit Adjustment' action from the quantity resolution screen. The actions are written to the IM_RESOLUTION_ACTION table in an unrolled status with the amount of adjustment. The IM_INVOICE_DETAIL table also receives a flag that signifies 'pending adjustment' for the invoice line.

At the same time, the actions are written to the IM_RECEIVER_COST_ADJUST and IM_RECEIVER_QTY_ADJUST tables to indicate the expected receiver adjustment amount on the RMS (or equivalent merchandising system) side. In sum, these two tables serve as the staging tables for the RMS (or equivalent merchandising system) process to actually perform the adjustment. For a receiver cost adjustment, IM_RECEIVER_COST_ADJUST holds the order unit cost for the item after the adjustment. For a receiver unit adjustment, IM_RECEIVER_UNIT_ADJUST holds the received quantity for the item on the shipment after the adjustment.

The process compares the unit cost and/or quantity received for the item on the shipment with the expected unit cost and/or quantity on the IM_RECEIVER_COST_ADJUST and/or IM_RECEIVER_UNIT_ADJUST tables. If a match exists, the receiver cost and/or unit adjustment has occurred in RMS (or the equivalent merchandising system). As a result, the process sets the 'pending adjustment' flag on IM_INVOICE_DETAIL table to false for the invoice line. The reason code actions are only rolled up for an invoice if no invoice lines on the invoice have any pending adjustments.

Because ReIM cannot control when and how the receiver adjustments are happening on the RMS side (or the equivalent merchandising system), records written to the IM_RECEIVER_COST_ADJUST and IM_RECEIVER_UNIT_ADJUST tables are considered final. As a result, when the user resolves a cost or quantity discrepancy, the receiver adjustment must fully resolve a discrepancy before the user leaves the screen, and there should be no re-route actions involved. On the RMS side, the amount of adjustment must be exactly the same as expected.

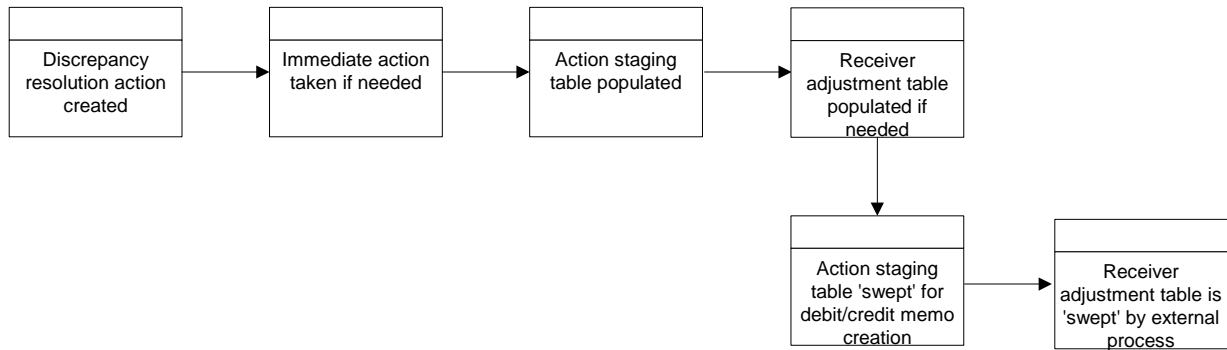
The IM_PARTIALLY_MATCHED_RECEIPTS table holds the amount of a receipt item that has been matched during invoice matching. The quantity received on the SHIPSKU table subtracts the quantity matched on the IM_PARTIALLY_MATCHED_RECEIPT table, giving the available to match quantity for the receipt item. Auto-match, summary matching, detail matching and quantity discrepancy resolution processes all keep track of the matched quantity bucket to determine how much of the receipt item has already been matched and how much of the receipt item remains available to be matched. In the case of a Receiver Unit Adjustment, the IM_PARTIALLY_MATCHED_RECEIPTS table is updated to reserve the entire remaining unmatched bucket for the receipt item. This logic prevents the adjusted receipt quantity from being used for any other matching or quantity resolutions.

Assumptions and scheduling notes

- If the receiver adjustment table sweep does not occur before the auto-match process, there will be a delay of one day before the detail matching work can be performed for the applicable invoice.
- The memo staging table sweep must occur before the posting batch process, or a delay of one day results before posting can occur.

High-level flow diagram

The following diagram offers a high-level view of the processing logic utilized within the reason code action rollup batch process.



ReIM's reason code action rollup flow

Primary tables involved

- IM_DOC_HEAD
- IM_INVOICE_DETAIL
- IM_PARTIALLY_MATCHED_RECEIPTS
- IM_RESOLUTION_ACTION
- IM_RECEIVER_COST_ADJUST
- IM_RECEIVER_UNIT_ADJUST
- IM_COST_DISCREPANCY
- IM_QTY_DISCREPANCY