

Retek® Invoice Matching™ 11.0.5

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[®] 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This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws.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 – Introduction	1
Chapter 2 – Backend system administration and configuration..	3
System assumptions	3
reim.properties file.....	4
Connection information for the database	4
Authentication section.....	4
Minimum and maximum pool size.....	5
Security ports.....	5
Standard formats.....	5
Language to be loaded.....	6
Size of batch updates	6
Date formats for specific locales	6
Set the end of week day for the system	7
Batch log and error file paths	7
Exception handling and error logging threshold	8
Locking timeout variable.....	9
Auto-match threading options	10
Generic threading options	10
Number of new documents that EDI invoice upload should insert at a time	11
Cache sizes for translation service	11
Invoice characters.....	11
Deal detail purge parameter	12
Chapter 3 – Functional design	13
Invoice and credit note matching process flow.....	13
The auto-match process	16
VAT on header level only invoices	16
Chapter 4 – Interfaces and file layouts.....	17
The EDI interface and layouts	17
Overview	17
The EDI reject table	17
The EDI reject file.....	18

Chapter 5 – Batch processes 19

Functional descriptions and dependencies..... 19

Auto-match batch design 24

 Overview 24

 The four algorithms 24

 Assumptions and scheduling notes..... 25

 Post processing 26

 High-level flow diagram 26

 Primary tables involved..... 26

Chapter 1 – Introduction

The information in this document reflects modifications and updates to the latest ReIM Operations Guide. Each chapter title and section title in this document corresponds to a chapter title and a section title in the ReIM Operations Guide.

Please note that entire sections have been included from the original ReIM Operations Guide for your reference, and the changes that have been made to those sections are in bold.

Chapter 2 – Backend system administration and configuration

System assumptions

- ReIM expects all invoices to be in eaches or the standard unit of measure (SUOM) converted to eaches. No other units of measure can be invoiced using ReIM.
- ReIM uses non-merchandise codes defined on the RMS table NON_MERCH_CODE_HEAD. The form that allows users to enter non-merchandise codes in RMS is not available when the RMS invoice match ind is set to N. Instead, non-merchandise codes should be added to the NON_MERCH_CODE_HEAD table using the database.
- A record must be inserted into the IM_SYSTEM_OPTIONS table in order to allow successful login to the application.
- Supplier options
All suppliers must have options defined in order for their invoices to be processed by the system, and the terms defined for those suppliers have to be completely updated in RMS. In order to support the use of suppliers in ReIM, the Enabled_Flag (set to 'Y'), Start_Date_Active and End_Date_Active are the required entries in the TERMS table in RMS 10.1 and in the TERMS_DETAIL table in RMS 10.2 and RMS 11.0.
- GL account maintenance
All reason codes, non-merchandise codes, and basic transactions must be mapped through GL account maintenance to support posting to the retailer's financial solution. Transactions are posted to a staging table in ReIM, the extract to update the accounts payable/financial solution is the retailer's responsibility..
- Multiview
The Document Find, Group Entry List, and Group Entry pages allow the retailer to define how certain fields display in these screens. The Multiview functionality allows the user to move fields around on the pages and save those views for future use. In order for Multiview to work and for these screens to populate correctly, IM_GLOBAL_PREFERENCES must be populated.
- VAT
If VAT is turned on, the retailer must have VAT regions, VAT items, and VAT codes set up in the merchandising system (such as RMS) to support validation of invoiced VAT charges. Verify the following values on the IM_SYSTEM_OPTIONS table:



Note: The values below should not be changed after initial setup. Changing them can cause errors in the system.

- VAT_IND is set to 'Y'.
- VAT_VALIDATION_TYPE is set to 'R'econcile Vat, Always Use 'I'nvoice VAT, or Always use 'S'ytem VAT.
- The DEFAULT_VAT_HEADER is set to 'Y' or 'N'.
- VAT_DOCUMENT_CREATION_LVL is set to 'ITEM' or 'FULL_INVOICE'.

reim.properties file

Retailer-defined configurations for ReIM are located in the reim.properties file. The key system parameters contained in this file are described in this section.

In the properties file, certain values are preceded by a “#” sign. This indicates the line is a comment and is not used as a setting

Every setting in the reim.properties file is configurable. When retailers implement code in their environment, they must update these values to their specific settings, taking system performance, for example, into consideration.

Connection information for the database

This portion of the file identifies what JDBC driver the system is utilizing. (For more information about JDBC, see “Chapter 3 – Technical Architecture”.) This data also includes what datasource (merchandising system) that ReIM is utilizing for its foundation data and the environment information associated to that datasource. The following settings apply:

- JDBC driver
- URL
- Datasource
- Username
- Password
- SchemaOwner
- BeanDriver

Authentication section

Authentication within ReIM addresses the legitimacy and the security privileges of users, an important aspect of the system’s security handling process.

Within the authentication section of this file, the retailer selects either LDAP or DATABASE, depending upon which is applicable.

For retailers selecting LDAP, see “Chapter 5 – Interfaces and file layouts” later in this document and see the ReIM Installation Guide for more settings and information.

Minimum and maximum pool size

The pool size pertains to the number of available database connections that the retailer intends to keep available in the pool. A system administrator is encouraged to adjust these values per configuration to match the retailer's anticipated number of users. The default values are intended to be a mere starting point.

In the example below, a minimum of 5 connections are available, and no more than 10 are available.

```
# Minimum and maximum pool size to maintain
pool.min=5
pool.max=10
```

Pool SQL trace

When this setting is 'true', SQL statements are shown. This setting might be used during development, build time, performance tuning, debugging, troubleshooting, and so on.

In the example below, the retailer has decided *not* to log the SQL statements.

```
pool.sqltrace=FALSE
```

Pool implicit cache

This parameter specifies whether or not the application needs to implicitly inform the connection pool cache that a connection is being used.

Security ports

Within the enterprise, a port is an endpoint to a logical connection and the way a client program specifies a specific server program on a computer in the network. A security port is analogous to an address for a given machine. The security port 'listens' at this address, and if the system needs to process security-related data, it must 'talk' only to that address to do so. The security-related processing can only occur at the place where the 'listening' is occurring.

```
# Security Ports
security.ssl_mode=2
security.port_non_ssl=8080
security.port_ssl=8443
```

Standard formats

Batch date format

With regard to incoming EDI sent data, retailers can define (through their vendors) the batch date format that they would prefer the system to receive. The system uses the batch date format that the operator enters in this section of the file. For example:

```
batch_date_format=yyyymmddhhmmss
```

Quantity decimals allowed

The database tables within the system allow for quantities to be held in decimals (for example, 12.5). Quantity decimals allowed means how many decimals the system displays for a quantity field. In general, quantity decimals are utilized by grocery retailers.

This is an integer value. ReIM is set to 0 because the system assumes eights. If a retailer wished to show 4 decimals, the value would be 4.

Language to be loaded

This setting instructs the system as to where to go to get an applicable locale's properties file. This file includes the localized labels, select list boxes, and so on. For example, the system's default English file is ReIMResources.properties. Another example is ReIMResources_fr_FR.properties for French.

The applicable values follow Java standards for languages and countries. Languages are always two letters and lower case. Countries are always two letters and upper case.

For example:

```
# Which language should be loaded?
language=en
country=US
```

Size of batch updates

This property establishes the size of the batch updates to the database. An array in this context is a collection of data. The value is in records.



Note: The EDI invoice upload batch process does *not* use this property. For a description of the property that the EDI invoice upload batch process uses, see the section, 'Number of new documents that EDI invoice upload should insert at a time', later in this chapter.

For example:

```
# Bulk insert and update array size
ARRAY_PROCESS_SIZE=30
```

Date formats for specific locales

To provide a user-friendly localized date format that is understood by users, the retailer may select one of four date formats that are available. For any entry of dates through the GUI, the user can enter a date in the format defined. The applicable values include the following:

- dd/MM/yyyy
- dd-MM-yyyy
- MM-dd-yyyy
- MM/dd/yyyy

For example:

```
date_format=MM/dd/yyyy
```

Set the end of week day for the system

The system administrator establishes this value to inform the system what that end of the weekday is. Sunday is equal to 1, and Saturday is equal to 7.

Batch log and error file paths

Batch log file path

The name and directory of the batch log files are established through this setting.

For example:

```
batcherrorlogpath=/files0/ReIM11/dev/error
```

Batch error file path

The name and directory of the batch error files are established through this setting. All errors and all routine processing messages for a given program on a given day go into this error file.

For example:

```
batcherrorlogpath=/files0/ReIM11/dev/error
```

Exception handling and error logging threshold

The ReIMException class automatically logs itself to the application log file. The level of logging may be raised or lowered in the setting below. The operator's choice instructs the system to log that level of error and errors above that level.

In the example below, an operator has configured the system to only display ERROR messages and above:

```
ReIMLoggerLogLevel=ERROR
```

Because the logger reveals message priority levels using numbers, the word equivalents are shown below:

- UNKNOWN = -999
- FATAL = 2
- ERROR = 3
- WARN = 4
- VALIDATION = 5
- INFO = 6
- DEBUG = 7
- PERFORMANCE = 8

Locking timeout variable

When a user tries to commit information to the database, the system checks to determine that he or she continues to have a lock because locks can time out. The number of seconds until the time out occurs is the locking timeout variable, defined in this file. If the user no longer has a lock, the user receives a message saying that changes cannot be saved.

The retailer should set the timeout period that makes the most sense for its business needs. Ideally, the timeout period should be long enough so that users can finish working on one record, but short enough so that unintentional locks (during lunch, and so on) do not delay other users an inordinate amount.

During a session, when the system has been idle for longer than the locking timeout variable, the system does not release the lock. Rather, if a second user attempts to lock the same table, the system's locking service determines whether the locking timeout variable has been exceeded. If the locking timeout variable has been exceeded, the locking service continues locking the table but for the second user. For the first user, the lock on the table has expired.

ReIM has a lock table defined for the bulleted areas below. For example, the table IM_DOC_HEAD locks the corresponding values from IM_DOC_HEAD_LOCK. To establish the duration of the timeout, the retailer can enter a mathematical expression using the variables. In the list below, the retailer has set the tables, with one exception, to be released after one hour of session inactivity. The table is only released at the end of the user's session or, in the case of a system crash, at the beginning of the next user's session.

- `business_roles_lock_timeout=1*hour`
- `doc_group_list_lock_timeout=1*hour`
- `doc_head_lock_timeout=no_expire`
- `edi_reject_doc_lock_timeout=1*hour`
- `supplier_options_lock_timeout=1*hour`
- `system_options_lock_timeout=1*hour`
- `tolerance_dept_lock_timeout=1*hour`
- `tolerance_supp_lock_timeout=1*hour`
- `tolerance_supp_trait_lock_timeout=1*hour`
- `tolerance_system_lock_timeout=1*hour`

Locking timeout variables are in milliseconds. Conversion data is provided below for the retailer's convenience.

- `millisecond=1`
- `second=1000`
- `hour=3600000`
- `day=86400000`
- `month=2592000000`
- `no_expire=-1`

Auto-match threading options

These parameters are used to configure auto-match threading. Auto-match can either be run as a single thread, or it can be threaded by the location hierarchy. Currently, this parameter is defaulted to thread auto-match by district. Changing the thread parameter is as simple as commenting out one parameter and uncommenting another.

For example:

```
auto_match_thread_by=ThreadByDistrict
```

Generic threading options

For more information about the batch processes mentioned below, see “Chapter 7 – Batch processes”.

Parameter used by EdiUpload only

The `thread.backgroundThreadTimeout` parameter is currently only used by EdiUpload for rejection files. The value represents how long the log writing thread polls an empty work queue before shutting down. Units are expressed in milliseconds.

For example:

```
thread.backgroundThreadTimeout=1800000
```

Parameters used by EdiUpload, AutoMatch, ComplexDealUpload, and FixedDealUpload

The `thread.consumerThreadTimeout` parameter represents how long the consumer pool threads. The value is used for executing the transactions for both EdiUpload and AutoMatch. Units are expressed in milliseconds.

For example:

```
thread.consumerThreadTimeout=60000
```

The `thread.consumerThreadKeepAlive` parameter represents how long the consumer/worker stays alive. Units are expressed in milliseconds.

For example:

```
thread.consumerThreadKeepAlive=60000
```

The `thread.consumerThreadPoolMin` and `thread.consumerThreadPoolMax` parameters represent the range of consumer/worker threads that can be created for the pool.

For example:

```
thread.consumerThreadPoolMin=10
```

```
thread.consumerThreadPoolMax=100
```


Number of new documents that EDI invoice upload should insert at a time

The EDI invoice upload (ediupinv) uploads merchandise and non-merchandise invoices and credit notes from the EDI into the invoice-matching tables. This parameter, which is related to bulk processing, establishes the number of documents that the system inserts at one time into one or more invoice-matching tables. In the example below, 1000 documents has been established as the value.

For example:

```
#How many new documents should EdiUpload insert at a time  
NBR_OF_EDI_DOC_BULK=1000
```

Cache sizes for translation service

To enhance the system's performance speed, the system utilizes a cache when performing data translations into another language.

For example, suppose the system has been configured to offer French translations. When a French user encounters a location name, the system retrieves the translated location name from the database and then stores it in a cache. If the system needs to retrieve the same translated location name at a later time (for another user, for example), the system would retrieve it from the cache rather than from the database. This parameter represents the number of entries within the cache that the system is allowed to use for such processing.

For example:

```
translation.items_desc_cache_size=100000
```

Invoice characters

Allowable invoice characters

This validation-related parameter describes what characters are allowed on an invoice. Note that '\ ' represents 'escape' characters.

For example:

```
#Invoice number validation regular expression.  
#Allowed Characers are :0-9, A-Z, space, minus sign, plus sign and  
underscore.  
#If this property is omitted(commented out) the system will default  
to =[0-9A-Z]+$ (only Alpha-Numeric).  
INVOICE_NUMBER_VALIDATION_REGULAR_EXPRESSION=^[0-9A-Za-z\ \+\-\_\_]+$
```

Invoices beginning with zero

This parameter either allows or disallows invoices to begin with the number zero. in the example below invoice numbers are *not* allowed to begin with the number zero.

For example:

```
INVOICE_NUMBER_VALIDATION_ALLOW_ZERO=FALSE
```

Deal detail purge parameter

This parameter is related to ReIM's process of extracting deal-related data from RMS. Once a document is posted, this value determines how many days ReIM waits before deleting the detail for the document.

For example:

```
purge_deals_after_days=2
```

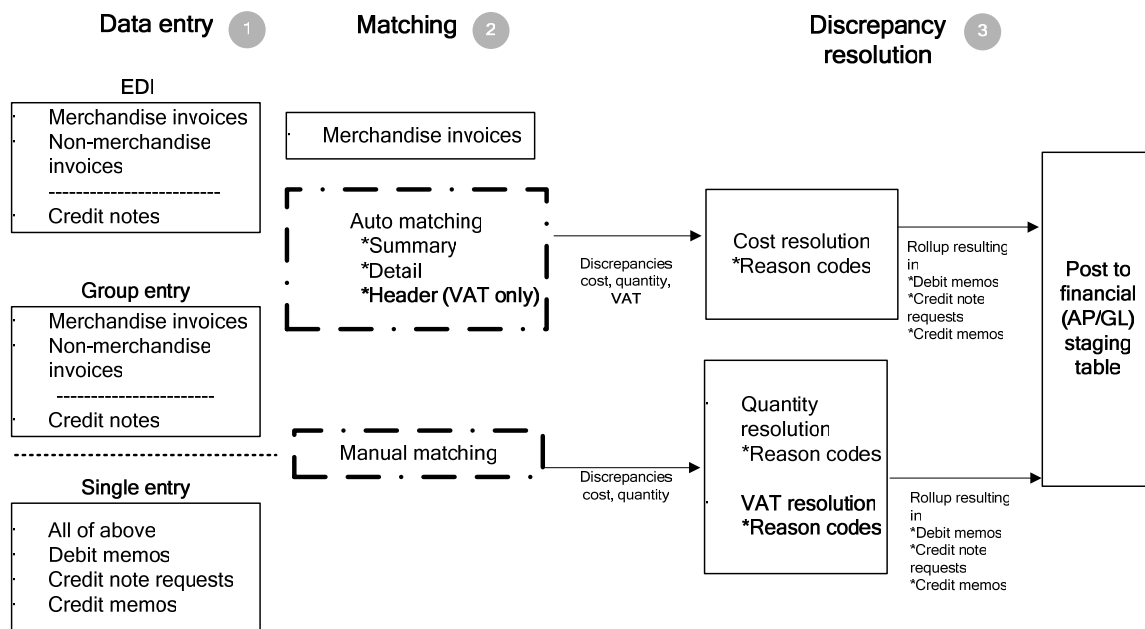
Chapter 3 – Functional design

Invoice and credit note matching process flow

This section provides a high-level explanation of the process flow in ReIM for each of the following areas:

- Data entry
- Matching
- Discrepancy resolution

Explanations of each numbered item on the diagram follow it.



High-level view of the invoice and credit note matching process



Note: Documents ‘drop out’ of the flow when they need no further processing. For example if an invoice is matched in step 2, ‘Matching’, the document would not continue to step 3, ‘Discrepancy resolution’. The document would be posted directly to the financial (AP/GL) staging table after step 2.

1. Data entry

There are three ways in which invoices and other documents enter the ReIM system:

▪ EDI

EDI allows ReIM to upload all of the following:

- Merchandise invoices
The bill for goods or services received from a supplier or partner. Merchandise invoices may have both of the following:
- Merchandise costs
Costs that are associated with items on documents. Any other costs on an invoice are non-merchandise costs. The sum of the merchandise costs and non-merchandise costs is the total document cost.
- Non-merchandise costs
Costs that are indirectly associated with invoice items, such as freight or handling charges.
- Non-merchandise invoices
Bills for non-merchandise costs only (a snow plowing service, for example). Non-merchandise invoices cannot contain items. Either suppliers or partners can create non-merchandise invoices.
- Credit notes
A document received from the supplier, often issued in response to a credit note request from the retailer, which results in a reduction of the retailer's balance owing to a supplier. A credit note request, may be raised in lieu of a deduction from invoice (that is, a debit memo) resulting from invoice over-charges, RTV's, rebate bill backs, and so on. Note that, compared to invoices, credit notes represent a separate functional process flow, where credit notes are matched against credit note requests.

▪ Group entry

Group entry facilitates summarized, on-line entry of paper documents. The group entry process accommodates the same types of documents as supported through the EDI process.

▪ Single entry

Single entry is designed as an exception handling tool made for invoices and documents not entered (for whatever reason) within a group. Single entry accommodates the same types of documents supported in the EDI and group entry processes, and in addition (if not created automatically through other processes):

- Debit memos
A document created to support a deduction from the invoice being paid. Deductions may result from a price or quantity discrepancy. A debit memo also refers supplier billing for rebates, RTV's, and so on. Debit memos can also be created as 'stand-alone' documents (that is, created on-line, but not supported by any processes in ReIM or the merchandising system).
- Credit note requests
A document sent from the retailer to the supplier, requesting a credit note for an over-invoiced amount or in support of various billing activities (for example, rebates, RTV's). If a credit note request is not satisfied by the supplier in a timely manner, ReIM provides the ability to convert it into a debit memo. Credit note requests may also be created as 'stand-alone' documents.

- Credit memos
A document created to refund a supplier for an under-invoiced or over-billed (for example, for rebates not meeting threshold performance levels) amount. Credit memos may also be created as 'stand-alone' documents.

2. Matching



Note: Credit notes must be matched online against credit note requests. Credit note matching is **not** supported by the automatic matching process.

- Auto-matching
Merchandise invoices are grouped by common PO/location; ReIM requires these attributes in all merchandise invoices. ReIM accesses the merchandising system to determine what shipments (receipts) were created for the PO/location. The auto-matching process attempts to support invoice cost and quantities against receipt quantities at PO cost, within user defined tolerances.

If the auto-matching identifies cost or quantity differences outside of the pre-established tolerance range, the system creates corresponding discrepancies (cost or quantity). Otherwise, matched invoices are posted to the financial staging table.

For header level only invoices, VAT validation is performed as a final validation step, after cost and quantity matching has been performed.

For more functional information about summary and detail-level auto-matching, see 'The auto-match process' section later in this chapter.

- On-line matching
The on-line matching dialog provides users with the ability match invoices with even greater flexibility than the auto-match process. Invoices are initially grouped by their PO/location, but the groups can be modified beyond the common PO/location relationship based on available (that is, 'unmatched') invoices and receipts, to support matches.

On-line matching either matches a document, which is posted to the financial staging table, or supports creation and resolution of a cost and/or quantity discrepancy.

3. Discrepancy resolution



Note: The user group for VAT discrepancies is only accessed to determine whether the specific user has view or edit privileges. The VAT list is not populated by user group.

Cost, quantity, and VAT discrepancies are routed to online lists by a user group (pre-established user groups, and routing rules determine which discrepancies populate which user group list). For example, in many companies the merchant/buyer is responsible for verification of invoice cost against the PO. To support this functionality, a user group of buyers by department or class might be a logical association to assign to an on-line Cost Discrepancy Review List (each user group would only see discrepancies assigned to them). Each user group is empowered to resolve discrepancies according to their authorization. Similarly, it may be logical to assign users groups to Quantity Discrepancy Review Lists based on receiving location. ReIM does not require the resolution of discrepancies through the routing process; the application will support a more centralized business process for resolving discrepancies using only the on-line matching dialog.

Users assign pre-defined reason codes against cost, quantity, and VAT discrepancies to support resolutions. The reason codes direct the system to take a specific action (for example, create a debit memo, receiver adjustment, and so on). Once all discrepancies are resolved for the document, it is posted to the financial staging table along with any corresponding debit memos, and so on, for posting to the retailer's accounts payable solution. Documents supporting discrepancy resolution (that is, debit memos, credit note requests and credit memos) are available for EDI download to the supplier (or the retailer may develop reporting from these values stored in the ReIM tables). These document records (except credit note requests) are also posted to the financial staging table.

The auto-match process

VAT on header level only invoices

The auto-matching process determines whether the VAT values on header level-only invoices are correct. The system only processes invoices that do not have any unresolved VAT discrepancies.

The invoice status determines whether an invoice can be processed by the Auto-match batch process (AutoMatchService). Only invoices in a status of 'Ready for Match' are processed. Those with a status of VAT discrepancy are not processed by the batch. See "Chapter 5 – Batch processes" for more information.

Invoices created without details are not able to have their VAT information validated at invoice creation. All header level-only invoices are created with a status of 'Ready for Match'. These invoices must have a VAT validation executed as part of the invoice matching process. This validation determines whether a header level-only invoice that has been matched to a receipt should continue in the matching and posting process or whether it should be marked as having a VAT discrepancy and removed from the matching process.

Chapter 4 – Interfaces and file layouts

The EDI interface and layouts

Overview

Electronic Data Interchange (EDI) facilitates the computer-to-computer transmission of business information and transactions, such as invoices and purchase orders. EDI represents a convenient method by which a retailer and its suppliers can transfer information back and forth. The Voluntary Interindustry Commerce Standard (VICS) EDI is used by the general merchandise retail industry.

ReIM has two file-based EDI interfaces. Note that neither follows the VICS EDI standard. The ReIM EDI interfaces have been customized, and the retailer must translate them.

The interfaces represent the upload of invoices or other documents from a supplier or another application and the download of documents to suppliers. These two common types of EDI are described below:

- EDI invoice upload is the standard description for an EDI process that uploads documents.
- EDI invoice download is the standard description for an EDI process that downloads Debit Memo, Credit Note Request, and Credit Memo data from ReIM to suppliers.

For information about ReIM's batch processes related to both of these types of EDI, see "Chapter 7 – Batch processes".

Note that although the vast majority of invoices are created through either EDI upload or batch entry, users can also create invoices online and add details, or use the online dialog to add details to an invoice that was EDI uploaded.

The EDI reject table

The EDI invoice upload (ediupinv) batch process uploads invoices and credit notes from the EDI into the invoice-matching tables. This process validates the information in the file against itself and against the RMS (or equivalent merchandising system)/ReIM database. A limited set of data validation errors cause the invalid transaction to be written to error tables (IM_EDI_REJECT_DOC_xxx) where the data can be corrected through an online process.

The following errors are written to the EDI reject table for the user to manually correct through the front end:

- Supplier number (or Partner ID)
This value must be a valid supplier (SUPS table) or partner (PARTNER table) in RMS (or the equivalent merchandising system).
- Order number(s)
Order number(s) must be approved and created for the supplier or linked suppliers in RMS (or the equivalent merchandising system) on the ORDHEAD table. Non-merchandise invoices may not have any order numbers associated, so this validation should be skipped for this type of invoice.

- Order/location combination
The system validates that all order number/location combinations in the file are valid within RMS or the equivalent merchandising system (meaning that the relationship must exist on the ORDLOC table).
- Terms code
All terms must exist within RMS or the equivalent merchandising system on the TERMS table.
- Invoice date
A document cannot be older than the v-date minus the post-dated document days' system level parameter value or newer than the v-date.
- Item number
Item numbers must exist within RMS on the ITEM_MASTER table (version 10.x or higher). If a UPC or reference number is passed, this number should also be validated. The item number should also exist for the supplier.



Note: With regard to the interface with RMS 10.x and higher, items must be transaction level.

- Duplicate invoice number for the supplier
The supplier's invoice number must be unique for the supplier.
- Merchandise invoices cannot be associated with a partner; they must only be associated with a supplier.
- Credit notes from a partner cannot have item records attached unless the partner type is a manufacturer, distributor, or wholesaler (type S1, S2, or S3).
- The system determines whether the invoice ID is valid if given.
- If the total quantity is given, the system determines whether the individual item quantities sum to total (the system only needs to check this if the supplier -level 'Match Total Qty' indicator is 'Yes').
- The system determines whether the total merchandise cost on the THEAD line matches the sum of costs from the TDETL lines (the sum of unit cost *qty).
- Either an item or a reference item must be specified on all documents except non-merchandise invoices or credit notes from a partner.
- The paid indicator must be either 'Y'es or 'N'o.


The EDI reject file

A limited set of data validation errors (identified in the file layout 'Validation' column) cause the invalid transaction to be written to the reject file (named by the retailer). When VAT processing is active within ReIM, all failed validations result in EDI uploads' being rejected to a file. There are no reject-to-table cases, and the EDI Maintenance screens are not accessible to the retailer.

Chapter 5 – Batch processes

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 three levels – summary, detail and header (VAT only).	EDI upload (Invoice Matching) Receipt upload (Merchandising system, such as RMS)

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
Reason code action rollup (ReasonCodeActionRollupService)	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. Note that a separate, retailer-created batch process sweeps the receiver adjustment table. The action staging table is used during posting to post the reason code actions to the financial staging table.	Receiver adjustment must occur prior to this batch process.

Batch processes	Details	Batch dependencies
<p>Disputed credit memo action rollup (DisputedCreditMemoResolutionRollupService)</p>	<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>	<p>The disputed credit memo action rollup must occur before resolution posting and after receiver adjustment.</p>
<p>Resolution posting (ResolutionPostingService)</p>	<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)  Note: This batch process is applicable <i>only</i> for those retailers using RMS 11.0.	<p>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.</p>	The RMS staged data must be purged after the upload.
Fixed deal upload (FixedDealUpload)  Note: This batch process is applicable <i>only</i> for those retailers using RMS 11.0.	<p>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.</p>	The RMS staged data must be purged after the upload.

Auto-match batch design

Overview

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 three inputs into the auto-match process include the following:

1. Invoices
2. Receipts
3. Purchase orders

ReIM 'owns' invoices, while receipts and purchase orders are 'owned' by a merchandising system, such as RMS.

The processing consists of three levels: summary, detail, and header. Summary-level matching attempts to match all invoices to receipts at a summary level. Detail-level matching attempts to match all invoices (that do not match at a summary level) to receipts at a line item level. Header level matching attempts to validate VAT before continuing to attempt to match all invoices.

The auto-match process attempts to match the invoices to receipts to the best of its abilities. The process assign different statuses according to the level of matching achieved.

If an invoice arrives prior to a receipt (for a particular PO), the auto-match process attempts only to match invoice unit cost to PO unit cost.

When a complete match cannot be made, manual intervention is required through online processes.

The four algorithms

The following four algorithms comprise the auto-match process:

1. Cost pre-matching
This process identifies any cost discrepancies prior to the arrival of receipts. If no receipts exist for the PO location, the invoices are sent to the cost pre-matching algorithm. Cost pre-matching is where unit costs on the invoice are compared with unit costs on the purchase order at a line level. If a match can be obtained, the invoice remains in ready-for-match status and is retrieved again for matching once the receipt comes in. If no match can be obtained, a cost discrepancy is created and routed immediately.
2. Summary matching
Invoices are grouped with receipts based upon purchase order location. A match is attempted for all invoices and receipts for the PO location. The invoices' total extended costs are summed and compared with the receipts' total extended costs. Based on a supplier option, the invoices' total quantity is summed and compared with the receipts' summed total quantity. If a match is achieved, all invoices and receipts are set to matched status. Otherwise, one-to-one matching is attempted for the PO location.

3. One-to-one invoice matching

This processing attempts to match a single invoice to a single receipt for the applicable PO location. If all invoices and receipts are set to matched status, the next PO location is processed.

If a multi-unresolved scenario exists (where more than one invoice can be matched with one or more receipts), all un-matched invoices are given the multi-unresolved status and no further processing occurs for this PO location.

4. Detail matching

During detail matching processing, an attempt is made to match each line on the invoice to an unmatched receipt line for the same item. Both the unit cost and quantity are always compared at the line level. If both the cost and quantity match, the invoice line and receipt line are placed into matched status. If the cost fails or the quantity fails, the cost or quantity discrepancies are generated and routed.

5. Header matching – (VAT only)

Invoices created without details are not able to have their VAT information validated at invoice creation. All header level only invoices are created with a status of 'Ready for Match'. For VAT validation, this processing determines whether a header level only invoice that has been matched to a receipt should continue in the matching and posting process or whether it should be marked as having a VAT discrepancy and removed from the matching process.

Assumptions and scheduling notes

- Although not recommended, auto-match can be run during the day when there are users online interacting with the system.
- Both the invoice unit cost and the PO's unit cost must be expressed in the same currency. In order to compare the invoice unit costs with the PO's unit costs, auto-match does not engage in currency conversion.

The system assumes that tolerance costs are always in the system's primary currency. If RMS is the applicable merchandising system, auto-match performs currency conversion if the currency on the order is different from the primary currency. RMS's existing currency conversion engine is used to perform this conversion. If RMS is not being utilized, another currency conversion engine must be provided to support this functionality.

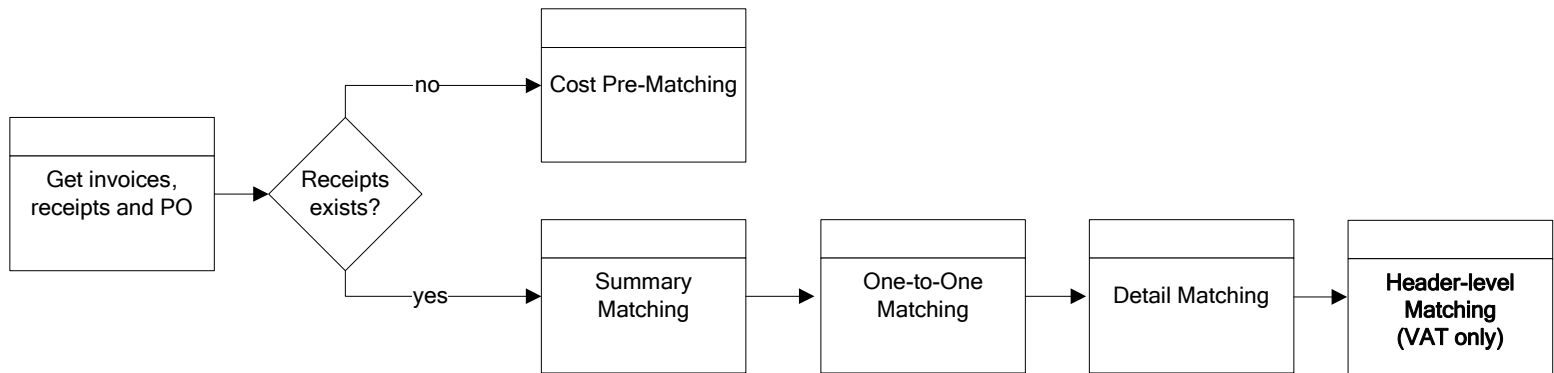
- The quantities on the invoice must be expressed in the same unit of measure as the quantities on the receipt. Auto-match performs no unit of measure conversion.
- The batch process runs after EDI upload (Invoice Matching) and Receipt upload (Merchandising system, such as RMS).
- Supplier options
All suppliers must have options defined in order for their invoices to be processed by the system, and the terms defined for those suppliers have to be completely updated in RMS. In order to support the use of suppliers in ReIM, the Enabled_Flag (set to 'Y'), Start_Date_Active and End_Date_Active are the required entries in the TERMS table in RMS 10.1 and in the TERMS_DETAIL table in RMS 10.2 and RMS 11.0.

Post processing

- Auto-match automatically invokes the ‘best terms calculation’ for invoices that it matches.
- Auto-match automatically posts invoices that it matches.

High-level flow diagram

The following diagram offers a high-level view of the processing logic utilized within the auto-match batch process.



ReIM's auto-match flow

Primary tables involved

- IM_DOC_HEAD
- IM_INVOICE_DETAIL
- SHIPMENT (RMS)
- SHIPSKU (RMS)
- IM_PARTIALLY_MATCHED_RECEIPTS
- ORDHEAD (RMS)
- ORDSKU (RMS)
- ORDLOC (RMS)
- IM_TOLERANCE_DEPT
- IM_TOLERANCE_SUPP
- IM_TOLERANCE_SYSTEM
- IM_COST_DISCREPANCY
- IM_QTY_DISCREPANCY
- IM_QTY_DISCREPANCY_RECEIPT
- IM_QTY_DISCREPANCY_ROLE
- IM_SUPPLIER_OPTIONS
- IM_SYSTEM_OPTIONS