

Reconciliation & Exception Intelligence: Full Analysis

Halbrook Industrial Supply

Which records actually require attention, and why?

Records supplied	Linked automatically	Ground-truth exceptions	Link precision
66,860	89.8%	747	99.34%

Demonstration project using synthetic data designed to simulate a realistic multi-system finance reconciliation problem.

1. The business problem

Halbrook Industrial Supply raises roughly 24,831 AR records a year and receives payment through a bank lockbox. Two clerks apply those receipts to open invoices by hand.

At the year end \$1,425,588 of cash the company was holding had arrived and been swept against the revolver but had not been posted into the ERP cash-receipts process. That is what the aged receivables report — the one collections works from through July — was overstated by on the date it mattered most.

Not all of it was a backlog, and the distinction is the point. \$99,145 was keyed in the first few days of July in the ordinary course; every finance function runs a day or two behind its bank. \$1,326,443 across 764 receipts was still unposted at the end of the observation window, the oldest of them nineteen banking days old at the cutoff.

2. Why standard reporting is not enough

The ERP can tell you which invoices are open. It cannot tell you which of them are open only because nobody posted the cash.

Answering that needs five sources, not two: the invoices open at the start of the period, the invoices raised during it, the bank file, the cash receipts journal, and the aged receivables report the business is actually reading. The backlog is the gap between what the bank received and what finance recorded, and no single system holds both sides.

3. Analytical approach

Five passes, in order of decreasing certainty, each seeing only what the one before it could not resolve. The cheapest and most certain evidence is spent first, so what reaches a person is genuinely the part that needs judgment.

What each pass resolved

Tier is a fixed score per pass, not a calibrated probability. Anything below 0.95 is proposed for confirmation rather than posted. The grouped pass carries two: an ambiguous grouping, where more than one set of invoices sums to the check, drops to 0.60.

Pass	Invoices settled	Tier	Ambiguous
Exact	16,855	1.00	—
Tolerance	310	0.95	—
Identifier	181	0.85	—
Grouped	4,410	0.60–0.80	441
Fuzzy	87	0.70	—

142 consolidated payments have more than one set of invoices that sums to the check, covering 441 invoice links. Those are flagged rather than presented as an answer, because a reviewer shown a single option will confirm it — and grouped links are markedly less reliable when the combination is not unique.

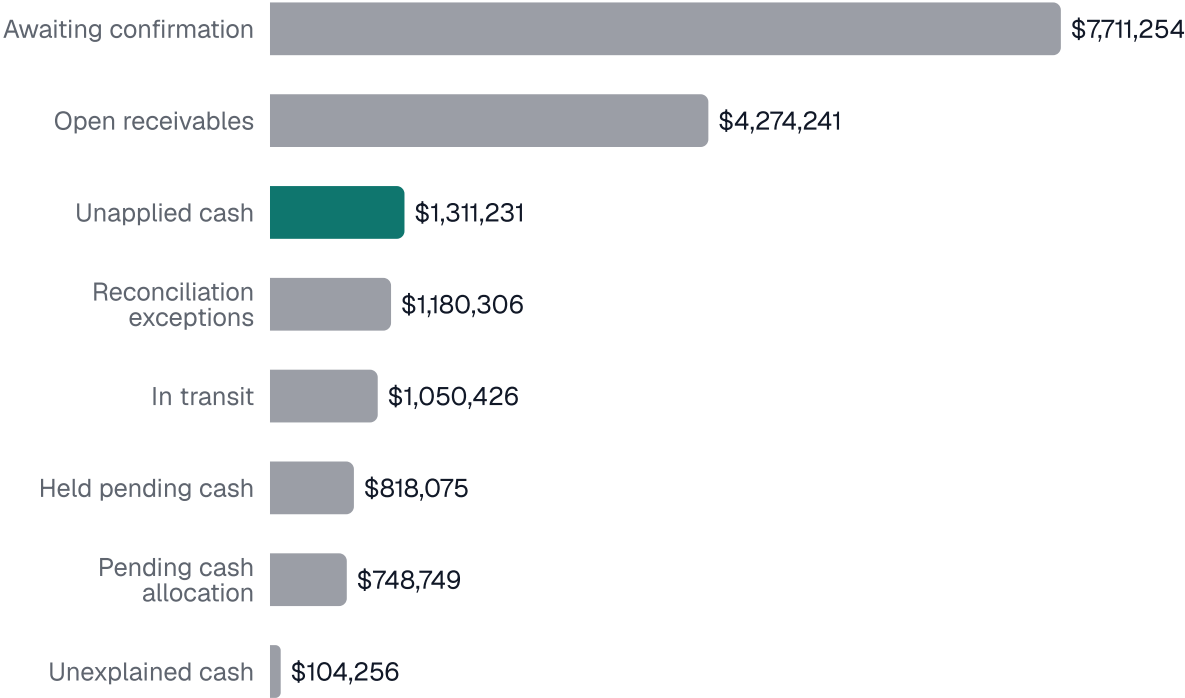
Thresholds are derived from the data rather than assumed. short pay <= 24.3% of the invoice; fuzzy similarity >= 0.900; grouped payments look back 135 days (calibrated)

4. What the solution found

The output is 8 populations, not one queue. They are shown separately and their values are never added together: an open invoice, a payment awaiting sign-off and a short-pay variance are different quantities. The cash the engine could not allocate and the invoices it holds back are two of them for the same reason — adding a check to the invoices it may have settled counts the same money twice.

Where the exception value sits

Face value by population, ignoring sign. These are not added together: an open invoice, a payment awaiting sign-off and a short-pay variance are different quantities. One population holds both debits and credits, so its netted total would mean nothing and the value on the desk is shown instead. **Teal** marks cash the engine matched that nobody had posted.



What each population is

Population	Action	Items
Awaiting confirmation Links the engine is confident enough to propose, not to post.	Confirm	1,547
Open receivables Invoices simply not yet paid. Normal receivables, not an exception.	Collect	2,447
Unapplied cash Cash the engine matched that finance never posted.	Apply	758
Reconciliation exceptions Genuine anomalies. Someone has to find out what happened.	Investigate	747
In transit Cash in transit across the cutoff. Visible, no action.	Monitor	569
Held pending cash Invoices a customer's unallocated cash may already have settled. Held back until it is resolved, not chased.	Hold	448
Pending cash allocation Cash the engine could not allocate to specific invoices. Allocate it before anyone calls the customer.	Allocate first	140
Unexplained cash Cash with no invoice behind it in either register. Somebody has to find out what it belongs to.	Investigate	44

\$1,425,588 of cash was unapplied at the year end, and \$1,326,443 of it was still unposted weeks later.

Posting is not matching — this was never a failure to identify the payments. The difference was a reconciling item the ERP held; what was missing was a report that reconciled the two. What separates one figure from the other is below, in the order the arithmetic actually runs.

From what the bank sent to what a person can post

Every sentence in this document about the backlog is written from these rows. Stated as one ordered bridge because the populations are easy to subtract in an order that is not true.

Stage	Receipts	Value
Positive receipts with no ERP cash-receipt posting at the period end Received on or before 2026-06-30.	812	\$1,438,124
Less: returned by the bank before the period end The company was not holding this money on the day.	(6)	(\$12,536)
Valid cash unapplied at the period end What the aged receivables report was overstated by.	806	\$1,425,588
Less: posted in ordinary lag afterwards Routine latency, not a backlog. Every finance function runs behind its bank.	(42)	(\$99,145)
Still not posted into the cash-receipts process Posting activity observed to 2026-07-15. Nothing beyond that date has been seen, so this is not 'never posted'.	764	\$1,326,443
Less: application would overpay an invoice a credit note has reduced The reference is exact and the application is not. Routed for investigation.	(6)	(\$15,212)
Less: the engine could not place it Unposted and unmatched is a different, smaller problem.	(0)	(\$0)
Directly applicable, exactly as it stands The apply queue.	758	\$1,311,231

Where the two systems disagree

Matched pairs only, so both sides are present. The reference is shown exactly as the customer wrote it.

Invoice	Billed	Customer wrote	Paid	Difference	Reason
—	—	no advice	\$4,519.17	-\$4,519.17	—
—	—	no advice	\$6,409.86	-\$6,409.86	—
—	—	P0736027	\$1,872.36	-\$1,872.36	—
—	—	P0965363	\$891.79	-\$891.79	—
—	—	P0651606	\$1,942.33	-\$1,942.33	—
—	—	INV-129346	\$1,197.82	-\$1,197.82	—
—	—	P0931963	\$5,580.39	-\$5,580.39	—
—	—	INV-137139	-\$279.55	\$279.55	—

5. Management use

Separating the populations is what makes the queue workable. Only 747 items are genuine anomalies carrying \$54,969 of variance. The open receivables belong in an aged debt report and cash in transit needs nothing at all.

One population earns its place by what it stops. Where a customer has cash the engine could not allocate, that customer's open invoices are held rather than chased, because the check may already have settled them. The effect is measured against the true position rather than asserted, in both directions: 95.2% of the collections queue is genuinely unpaid, and 93.4% of the invoices that should be chased are on it. Without the hold precision was 76%, and a collections team would have called customers about \$1.29M they had already paid — the exact failure this project exists to prevent, recreated by the tool meant to fix it. Recall is the figure that flatters nobody: \$268,548 of genuinely unpaid invoices are not on the list, most of them because a grouped match wrongly closed them.

The hold itself is the least accurate thing here, and it is published that way: 73.4% precision and 76.9% recall, \$215,851 held that did not need holding against \$184,999 that did. Guessing which invoices a check settled is exactly what the engine has said it cannot do confidently, so the control a supervisor should work from is the account, not the invoice.

Resolve the cash before the call

One row per customer holding cash the engine could not allocate. Net exposure is what the account is worth chasing once the cash is placed, whichever invoices it turns out to have settled — a number that does not depend on the guess.

Customer	Unresolved cash	Open invoices	Net exposure
C0001	\$72,307	\$412,009	\$331,143
C0002	\$54,660	\$260,895	\$197,204
C0005	\$52,835	\$236,480	\$175,926
C0003	\$45,953	\$272,235	\$223,777
C0062	\$24,969	\$75,871	\$50,282
C0008	\$22,156	\$134,676	\$111,138
C0006	\$21,661	\$159,984	\$135,062
C0027	\$20,575	\$44,837	\$23,023

For the CFO the output is narrower: the receivables report can be reconciled to the underlying cash position again, \$1,311,231 is posted rather than chased, and the two clerks spend their week on the items that need a person.

6. Methods and technical note

The evidence package ships a verifier that rebuilds every published figure from the data in the package, using nothing but the standard library. **95 of 95 checks pass on this build** — across both demonstrations, and nothing in the exception headline is declared rather than derived. Run it yourself: `python verify.py`.

Python, pandas, and rapidfuzz. The pipeline starts at a file, not a dataframe: extracts are written as CSV with the mess a real export carries — a byte order mark, amounts as text with thousands separators, two date formats in one column — and read back through an intake stage that coerces them, checks control totals against what was declared, validates the schema, and reports everything it had to change. Nothing is cleaned silently.

What was wrong with the files before any analysis ran

Reported, not silently fixed. A value that becomes zero unnoticed is how a confident wrong answer happens.

File	Severity	Finding
erp opening ar open items	note	amount: 609 of 2,701 values needed cleaning before use.
erp opening ar open items	note	invoice_date: 608 of 2,701 values needed cleaning before use.
erp opening ar open items	note	due_date: 583 of 2,701 values needed cleaning before use.
erp opening ar open items	note	customer_name: 413 of 2,701 values needed cleaning before use.

Measured against known answers

Every problem was deliberately introduced and recorded. Two rates, because a record can carry more than one true condition: led is whether the engine gave this exact reason, handled is whether it surfaced the record under any reason genuinely true of it. A short payment that also arrived after the cutoff is handled correctly when the engine leads with the deduction.

Problem	Introduced	Correctly classified	Led	Handled
One to many	1,600	1,460	91.3%	91.3%
Duplicate record	45	45	100.0%	100.0%
Amount difference	310	310	100.0%	100.0%
Identifier mismatch	180	180	100.0%	100.0%
Formatting difference	420	398 resolved	—	94.8% silent
Near match unconfirmed	95	87	91.6%	91.6%
Credit note	130	130	100.0%	100.0%
Payment reversal	76	75	98.7%	100.0%
Missing in source	40	40	100.0%	100.0%
Timing difference	658	569	86.5%	99.5%
Missing in target	2,464	2,310	93.8%	93.8%
Application backlog	770	758	98.4%	99.2%

The precision figure is checkable rather than asserted. `link_audit.json` ships 400 randomly sampled links with the pairing that actually happened beside each one; 398 are correct, or 99.50% against the

99.34% claimed across all links. `ground_truth_links.json` and `ground_truth_issues.json` ship alongside, so the published accuracy figures can be independently checked against the demonstration ground truth.