

Every line is matched or explained, and the books tie to the cent

The scenario

An allied health group with four clinics banks through four accounts: operating, payroll clearing, card terminal settlements and a USD account. Plan managers pay many invoices in one transfer, Stripe pays out net of fees, card terminals settle daily, payroll goes out line by line, and one day's statement was imported into Odoo twice. Odoo has 4,327 bank journal items for the quarter and 479 open invoices and bills.

What the build does

- A Python engine (match.py) applies 14 rules in priority order and puts every bank line and Odoo item into one of 1,322 match groups.
- 1,618 of 1,623 statement lines (99.7%) are matched by a rule; the 5 left over are escalated with a note.
- The workbook rebuilds 12 standard bank reconciliation statements with 48,995 formulas. The unexplained difference is 0.00 on every one.
- 204 exceptions get a reason code, an owner and a proposed action; 189 balanced journals are linked to them.

Position at 30 September 2026

ACCOUNT	STATEMENT	IN TRANSIT	UNPRESENTED	ODOO	DIFFERENCE
OPS	549,048.87	8,420.00	(286.00)	465,755.94	0.00
PAY	45,511.30	0.00	(20,541.30)	25,000.00	0.00
MER	319,969.33	9,173.55	0.00	336,243.02	0.00
USD	75,006.00	5,000.00	0.00	79,040.00	0.00

USD account in USD. The gap between statement and Odoo is explained by the items on the statements page; after the proposed journals, Odoo plus in-transit items equals the bank.

Decisions needed

- **5 escalated items:** a branch cash deposit, a card debit with no receipt, an unknown USD wire, a terminal adjustment and a receipt that quotes another customer's invoice.
- **12 decisions:** 3 short payments booked as paid in full, 2 receipts registered twice, 2 direct debits with no bill, a chargeback, a returned supplier payment, 2 terminal refunds missing from the POS export and a stale cheque.
- **187 routine items** (fees, Stripe fees with GST, receipts to register) are ready to post.