
WOOCOMMERCE • STRIPE • PAYPAL

Payoutproof

*Turning payouts into journal entries that balance
to the cent against the bank deposit*

THE RECONCILIATION IDENTITY

`gross - refunds - disputes - fees ± fx ± reserves = deposit`

What Payoutproof does

A payout arriving in your bank is a single net number. Payoutproof takes it apart, matches each piece to the order that produced it, and gives you a journal entry you can post without checking it first.

That payout is gross sales, less processing fees, less refunds, less chargebacks, adjusted for currency and reserve holds, covering orders placed across several days. Nothing in WooCommerce maps it back to the orders behind it, which is why most stores rebuild it by hand in a spreadsheet every month.

It reads the keys you already have

You will not be asked for an API key. Payoutproof finds the credentials your payment gateway already stores and confirms they work before claiming anything.

This is why WooPayments works. WooPayments never exposes Stripe API keys to the merchant, so external accounting services cannot reach it at all. Payoutproof runs inside WordPress and reads its data directly.

What it refuses to do

The output is worth something only because it can be trusted without inspection, so Payoutproof declines in three situations rather than guess:

A payout does not balance	Parked, never posted, with the failed check shown.
A transaction type is unrecognised	Parked until you say where that money belongs.
Your chart is not mapped	Export stays disabled. The defaults are a common chart, not yours.

Getting started

Install, activate, and open **WooCommerce** → **Payoutproof**. If a supported gateway is configured, your payouts are already there.

Requirements

Platform	WordPress 6.4+, WooCommerce 8.0+ with HPOS, PHP 8.1+
A gateway	Stripe or PayPal, through any of seven plugins — or a key you enter

Confirming detection

The screen names every gateway it can read, the account it found, and whether that account is in test or live mode. From the command line:

```
$ wp payoutproof detect
gateway          account          mode
WooCommerce Stripe Payment Gateway  acct_1Qi5nj2Ns42bcMb1  test
```

Nothing was pasted. Payoutproof walked the active plugins, read each gateway's own stored credential, and called the payment API to prove it worked before listing it.

It knows seven: WooPayments, WooCommerce Stripe, Payment Plugins for Stripe, FunnelKit Stripe, ThemeHigh Stripe, WooCommerce PayPal Payments and Payment Plugins for PayPal. Anything else goes in by hand on the **Connection** tab.

More than one gateway is normal and fully supported. Each is reconciled separately and never merged, because each settles into its own bank line.

About your credentials

Payoutproof reads the key when it needs it and never keeps a copy.

Rotating the key in your gateway is therefore enough: there is no second place to update, and no second copy of a live secret sitting in your database.

Narrowing what it can see

If you would rather the plugin could not do everything your gateway key can, create a Stripe **restricted key** with read access to payouts, balance transactions and charges. Nothing more is needed, and such a key cannot move money even if it leaked.

Payouts	Read — the deposits themselves
Balance transactions	Read — what each deposit is made of
Charges	Read — to match a transaction to its order

If you take PayPal

PayPal settles a *period* rather than a payout, and reading one needs the **Transaction Search** permission on the REST app the credentials belong to. Checkout credentials rarely carry it, so Payoutproof names the missing permission rather than reporting an empty month. Tick it in PayPal's dashboard, then allow nine hours for their token cache.

WooPayments needs no key at all. It never exposes one; the data is read through WooPayments' own interface inside your site.

Payoutproof never writes a credential to a log, an error message or a support export.

The Payouts tab

The summary answers the only question that matters on arrival: is anything wrong?

Payoutproof

Payouts

Period activity

Chart of accounts

Connection

DEPOSITED

AUD 884.56

PAYOUTS

2 last 90 days

PARKED

1 needs attention

WooCommerce Stripe Payment Gateway

TEST

acct_1Qi5nj2Ns42bcMb1

Total AUD 884.56

STATUS	PAYOUT	ARRIVES	AMOUNT	JOURNAL
Parked	po_1UIsuD2Ns42bcMb1SnXJ4LME	2026-09-24 2 days ago	AUD 167.18	4 lines
Transfer	po_1UIIoUF2Ns42bcMb14qMeGBN0	2026-09-24 2 days ago	AUD 717.38	2 lines

Refresh payouts

Unlock CSV export

Each gateway gets its own card and its own total. One payout here is parked, and the summary says so before you have read a single row.

Deposited is the total that reached your bank across every gateway in the window. **Parked** is the count that failed a check — if it reads zero, every payout listed is safe to post.

Reading a row

Each row shows its state, processor ID, arrival date, amount, and how many journal lines sit behind it. Arrival dates are frequently in the future — a payout on a two-day settlement delay is dated two days out — so the row says *in 2 days* or *2 days ago* beside the date rather than leaving you to count.

The three states

Balanced

The payout settled a known set of transactions, the debits equal the credits, and the bank line matches. Post it.

Transfer

A manual payout. Complete and correct, but it carries no sales — the revenue behind it is on the Period activity tab. Explained overleaf.

Parked

A check failed. The entry is not posted and the panel names which check, so you can see what to fix rather than what went wrong.

Opening the journal

Any row expands into the entry behind it — the accounts, where each posts in your chart, and the totals.

WooCommerce Stripe Payment Gateway TEST acct_1Qi5nj2Ns42bcMb1			Total AUD 884.56	
STATUS	PAYOUT	ARRIVES	AMOUNT	JOURNAL
Parked	po_1UIsuD2Ns42bcMb1SnXJ4LME	2026-09-24 2 days ago	AUD 167.18	· 4 lines
Transfer	po_1UIoUF2Ns42bcMb14qMeGBN0	2026-09-24 2 days ago	AUD 717.38	· 2 lines
A manual payout: money you moved out of your balance, rather than the processor settling a set of orders. It posts as a transfer from your clearing account to your bank. The sales, fees and refunds behind it are on the Period activity tab.				
ACCOUNT	FROM	DEBIT	CREDIT	
Bank Account 091	payout.amount	717.38		
Processor Clearing 090	payout.amount		717.38	
Totals (AUD)		717.38	717.38	

The account code beside each name is where that line will post, taken from your chart of accounts.

Debits and credits must agree on every entry Payoutproof is willing to show as posted. If they ever disagree, the row is Parked instead, with the difference stated to the cent.

Automatic and manual payouts

These are different kinds of event, and they need different journals. Getting this wrong is the most expensive mistake available in reconciliation.

An **automatic** payout settles a specific set of transactions. The processor knows exactly which ones, so the payout can be decomposed into the sales, fees and refunds that make it up.

A **manual** payout is you moving an arbitrary amount out of your balance. It settles nothing in particular, and Stripe will not tell you what is "in" it — not from the balance transaction list, not from the payout reconciliation report. That is not a gap in their API. There is genuinely nothing to list.

Why attributing activity to a manual payout is wrong, not merely imprecise. Withdraw 500 from a balance of 1,000 and the period's activity cannot equal the payout, by construction. Any tool that claims to itemise a manual payout is showing you a number it invented.

The two-journal model

So the money moves in two independent entries, which is also how an accountant would record it by hand:

ENTRY	DEBIT	CREDIT
Activity <i>per month</i>	Clearing Fees, Refunds, Disputes	Sales Tax
Payout <i>per transfer</i>	Bank	Clearing

Each balances on its own. What remains in **Clearing** is money still held by your processor — check it against their reported balance, which is a stronger control than payout attribution could ever have given you.

Period activity

One month of sales, fees and refunds, booked against your clearing account. If your payouts are manual, this is where your revenue lives.

WooCommerce Stripe Payment Gateway acct_1Qi5nj2Ns42bcMb1			69 transactions
ACCOUNT	FROM	DEBIT	CREDIT
Processor Clearing 090	period net		4006.54
Chargebacks & Fees 404	7 txn		221.30
Chargebacks & Fees 404	bt. fee		150.00
Refunds 200	12 txn		1606.92
Sales Revenue 200	48 txn		6216.90
Processing Fees 404	bt. fee		232.14
Totals (AUD)		6216.90	6216.90

Sixty-four balance transactions as one balanced entry.

Pick a month and Payoutproof aggregates every balance transaction that settled in it. Payout rows are deliberately excluded — they move money *out* of the balance and are booked by the transfer entries on the Payouts tab. Counting them here would record the same money twice.

If your payouts are automatic

You can ignore this tab. Your payouts already carry their own components. It remains useful as a way to see a whole month in a single entry, but nothing here is missing from the Payouts tab.

The control worth running

Clearing is debited by the month's activity and credited by each payout. Whatever is left is money still at your processor. Compare it with the balance your processor reports: if they agree, your books and your processor agree, and you have verified that without trusting either one alone.

Your chart of accounts

Payoutproof ships with a widely used chart as a starting point. It almost certainly does not match yours, so export stays disabled until you have said so yourself.

MONEY OF THIS KIND	POSTS TO CODE	NORMAL BALANCE
Processor Clearing	090	Debit
Bank Account	091	Debit
Sales Revenue	200	Credit
Sales Tax Payable	820	Credit
Refunds	200	Debit
Chargebacks & Fees	404	Debit
Processing Fees	404	Debit
FX Gain / Loss	499	Debit
Processor Reserve	090	Debit
Internal Transfer	090	Debit
Processor Advance	900	Debit

All eleven accounts, the code each posts to, and its normal balance.

Check every code against your own chart. A wrong code imports perfectly cleanly and puts the money in the wrong account. Posting processing fees to 404 is right on one chart and wrong on another where 404 is Motor Vehicle Expenses — and nothing will warn you.

The **normal balance** column is a check on yourself: revenue carries credits, expenses and assets carry debits. Map Sales Revenue to an expense code and that column is where you will notice.

Exporting

Choose a month and take either format. Export is scoped to that month, because a journal labelled with a period it only partly covers is worse than none.

Manual journal

The layout most accounting packages accept. Each payout becomes one multi-line journal: the first line carries the narration and date, continuation lines leave both blank, and that is how the importer knows they belong together. Amount is a single signed column where a positive figure is a debit.

```
Narration,Date,Description,AccountCode,TaxRate,Amount
"Settlement activity 2026-09",2026-09-26,"Processor Clearing - period net",090,"Tax Exempt",4017.62
,,,"Sales Revenue - 46 txn",200,"Tax Exempt",-6131.90
"Payout po_1UIoUF...",2026-09-24,"Bank - payout.amount",091,"Tax Exempt",717.38
,,,"Processor Clearing - payout.amount",090,"Tax Exempt",-717.38
```

Import it wherever your package takes manual journals, and do not rename the column headers — the import fails if you do.

Audit CSV

Explicit Debit and Credit columns with the payout named on every row — the one to open when you want to check the work, and the easier to map onto an importer that expects separate columns.

What will not be in the file

Parked payouts are excluded rather than exported with a warning: everything in the download is something you can post without reading it first.

Most imports cap a file at 300 lines. Payoutproof says so rather than produce one the importer will reject.

Unrecognised transaction types

Processors add transaction types, and platform accounts carry types an ordinary account never sees. When one appears that Payoutproof does not know, every payout containing it is parked.

That is deliberate. The alternative is filing money somewhere plausible and letting you discover it at year end. But parking alone is a dead end, so the Payouts tab offers the fix directly: each unknown type, and the buckets you can assign it to.

Revenue	Money earned
Contra-revenue	Refunds and reversals
Chargebacks	Disputes and their fees
Processing fees	Processor and platform charges
Currency	FX gain or loss
Reserve	Funds the processor is holding
Internal movement	Transfers between your own balances
Exclude	Not part of the journal at all

Code it once and the parked payouts recalculate immediately.

Your decision outlives our updates. Codings are stored separately from the map Payoutproof ships and layered on top of it. If a later release maps the same type differently, yours still wins — otherwise an update could quietly move your money to another account, which is precisely the failure this plugin exists to prevent.

Command line

Every screen has a WP-CLI equivalent, for cron, for checking a store whose admin you cannot open, and for running across many stores at once.

```
# which gateways can be read
$ wp payoutproof detect

# reconcile and print the journals
$ wp payoutproof reconcile --days=30
$ wp payoutproof reconcile --days=30 --period
$ wp payoutproof reconcile --format=json

# repair orders whose processor link was never written
$ wp payoutproof repair
$ wp payoutproof repair --limit=200 --commit
```

`reconcile` exits non-zero when anything fails to balance, so it works as a monitoring check and not only as a report.

Repairing broken order links

Orders placed through pay-for-order links, created in the admin, or taken through some express checkout paths never receive their payment intent ID. Those orders cannot be refunded from the admin and webhooks cannot match them.

The link survives on the processor's side, so Payoutproof can restore it.

`repair` is a dry run unless you pass `--commit`, it verifies the amount and currency before writing anything, and it will not change an order's status unless you explicitly ask — that fires customer emails and reduces stock, which is a business event rather than a data repair.

If something looks wrong

No gateway detected	Payoutproof reads credentials from a configured gateway. Check it is active and has its keys saved, then reload — it need not be <i>enabled at checkout</i> , only configured. For a gateway it has no adapter for, enter a key on the Connection tab.
A payout is Parked	Expand it. The panel names the check that failed. An unrecognised type is fixed by coding it; an imbalance usually means the processor returned a partial page, and Refresh payouts will clear it.
Figures look stale	Results are cached for fifteen minutes so the screen does not call the API on every load. Refresh payouts discards the cache. Only clean results are ever cached, so a failure is never held onto.
A payout is dated in the future	Correct. Arrival dates reflect the settlement delay on your account — often two days. The row says how far out it is.
Export button is missing	Either the chart of accounts has not been saved, or the site is on the free plan. The chart must be saved even if you keep every default.
The import is rejected	Column headers must be exactly as exported. A spreadsheet editor that reformats the date column or strips the leading zero from an account code will break it — import the file as downloaded.

Support. info@allanninal.dev — include the payout ID and what the panel says. If a payout is parked, the failed check is the single most useful thing you can send.