indition QuickBooks

QuickBooks

Feature List

Capability overview · Version 5.5.2 · 2026-07-02

At a glance

QuickBooks (found under **Orders → QuickBooks**) gets your store's sales into QuickBooks two ways: **download a file** you import yourself (IIF for QuickBooks Desktop, CSV for QuickBooks Online), or **automatic sync** that connects a QuickBooks Online company and posts each order as a balanced journal entry on a schedule. It keeps a full history of every run, maps your sales to your real chart of accounts, and can route multiple stores to separate QuickBooks companies.

Delivery methods

You choose one of two methods on the main screen (until you pick, only the two method cards show):

Method	What it does	Connection needed?
Download a QuickBooks file	Generates an IIF or CSV file that you import into QuickBooks manually.	No
Automatic sync (Send Via API)	Posts each order straight into QuickBooks Online on a schedule, no manual import.	Yes — connect a QuickBooks Online company over OAuth2.

File formats

- **IIF** — for QuickBooks **Desktop**.
- **CSV** — for QuickBooks **Online** and most importers.
- You pick the format at download time. Either format can be produced from any history row.
- Filename: `quickbooks-sales-YYYYMMDD-His.iif|csv` (prefix overridable via `ORDER:EXPORT_FILE_PREFIX`).
- File bytes are never stored — they are regenerated on demand, so a re-download always yields identical data.

Detail levels

Three levels of granularity control how orders are grouped:

Level	Journal entries	CSV columns
`per_order`	One journal entry per order (IIF DOCNUM = order #).	One row per order.
`daily_summary`	One condensed entry per day (cash / net sales / tax).	Date, Orders, Net Sales, Sales Tax, Order Total.
`daily_split`	One entry per day, split across every account.	Date, Orders, Gross Sales, Discount, Sales Tax, Shipping, Card Fee, Order Total.

Scope & date selection

- **Since last export** (`since_last`) — everything newer than the last recorded export, tracked automatically by order-id high-water mark.
- **Date range** (`date_range`) — an explicit From/To window filtering on `date_added`, whole days (00:00:00–23:59:59).
- **Presets** — This month, Last month, This quarter, Last quarter, This year, Last year, All orders. Presets are computed from the browser clock; orders themselves filter by the database timezone.

Accounting

Sales are read from the canonical order financial columns and posted as balanced double-entry General Journal entries across six posting categories:

Category	Default account	Debit / Credit	Source
Cash received	Undeposited Funds	Debit	`order_total` + card surcharge
Discounts	Sales Discounts	Debit	`discount`
Item sales	Sales	Credit	`sub_total`
Sales tax	Sales Tax Payable	Credit	`sale_tax` + `order_shipping_tax`
Shipping	Shipping Income	Credit	`shipping_cost`
Card fees	Card Processing Fees	Credit	card surcharge (from `payment_transaction.card_fee`)

- The **card surcharge is charged on top of the order total**, so Cash received (Undeposited Funds) = `order_total + card_fee`.
- Every entry is forced to balance to exactly 0; any rounding residual is absorbed by the Sales line.
- `daily_summary` condenses to just cash / net sales / tax.
- On sites without the `payment_transaction.card_fee` column, card fee exports as 0.

Account mapping

- Map each of the six posting categories to a real account in your chart of accounts (per connection).
- For automatic sync, on-screen dropdowns are populated by fetching the company's chart of accounts **live**.
- **Auto-match** picks the closest-named existing account for each category (exact match wins, else substring, else token overlap); you can override.
- **Find-or-create**: if a mapped API account name doesn't exist, it is created by name with a sensible account type/subtype.

- For file downloads the six default account names are used in the file — rename them on import in QuickBooks.

QuickBooks Online sync

Capability	Detail
OAuth2 connect	Authorize with Intuit, then choose the company; Realm (Company) ID captured automatically by the callback.
Posting	One balanced JournalEntry per order. PrivateNote = "Order <#> (<date>)".
Idempotency	DocNumber = order #; a run checks whether that DocNumber already exists and skips it — safe re-runs, no duplicates.
Schedule	Runs as a platform Scheduled Job (Admin → Scheduled Jobs), daily or monthly, default 02:30 — not a CLI cron.
Run sync now	Posts immediately, per group or for all groups from the grid.
Test / Disconnect	Test verifies the connection and returns the company name; Disconnect forgets tokens/realm and turns off the API.
Auto token refresh	Access token refreshes automatically (401 → refresh → retry once).
Stop and resume	Sync stops on the first hard error; the next run resumes after the last successfully-posted order.
Encrypted secrets	Client secret, refresh token and access token are stored encrypted; the client secret is write-only in the UI.

Multi-store (groups)

- With more than one store, the screen becomes a **multi-store home** — a card per group plus the shared history grid.
- A **group** = a connection + the stores mapped to it, with its own delivery method, account mapping and (when API) its own QuickBooks company.
- Every store belongs to exactly one group; the **default group** ("All stores") catches any unmapped store so no order is orphaned.

- **Separation mode** — how several stores stay separate inside one QuickBooks company: `class` (tag each entry with a Class = store name), `memo` (prefix the memo with the store name), or `none`.
- **Presets** scaffold groups fast: "All together" (one group for all stores) or "One per store".
- Each group has its own "Run sync now" and its own default download timeframe & granularity.

Export history

- **Grid columns:** Starting / Ending Order #, Orders count, Total, Exported By, Export Date, Status.
- **Statuses:** Downloaded, Sent, Error.
- **Actions:** re-download as IIF or CSV, or delete the record (delete never touches order data).
- Every manual download and API sync is recorded, including errors — errored runs auto-retry next run. A clean "nothing new" API run is not recorded (avoids noise).
- **First run** shows an empty state telling you how many orders are waiting to export.

Under the hood

For technical readers. QuickBooks lives in the Order submodule of Shop, runs on PHP 7.4 with PostgreSQL, and is driven by the XService engine `Order.QuickbooksExport.*`. Migrations apply via `/admin/Admin/patches` or a clean install.

Table	Purpose
`shop_quickbooks_export`	One row per export/sync run (history).
`shop_quickbooks_setting`	One row per connection (a QuickBooks company/app), with account map.
`shop_quickbooks_store_connection`	Store → connection map with per-store high-water mark.

Service action	Role
`pendingInfo` / `resolveScope`	Count what is waiting; resolve the order-id range for a scope.
`generate`	Builds IIF/CSV and journal-line math.
`runScheduledSync`	Scheduled Job entry point that posts to QuickBooks Online.

Component	Value
Order module branch	5.5.2 (menu in Shop 5.17.2)
API version	Accounting API v3, minorversion 73; sandbox / production toggle
Per-run caps	5000 orders, 480 seconds