# Accounting Handbook

A full accrual accounting engine — chart of accounts, journals, receivables, payables, banking, tax, fixed assets, trust and period-end — all posting to one shared ledger the rest of the platform feeds.

Version 1.0 · ixlcore.com

## Reference

Accounting is the finance engine at the centre of IXL CORE. It holds your chart of accounts, records every journal, runs your receivables and payables, reconciles your bank, computes your tax returns, tracks your fixed assets and controls how each accounting period is opened, closed and locked. Because the whole platform shares this ledger, invoices, bills, payments and credit notes raised in Accounting — or events raised elsewhere on the platform — post here through one posting engine, so there is nothing to re-key and nothing to reconcile between two systems.

This guide is a reference for what the module does and how the pieces fit together. It describes IXL CORE **version 1.0**, which is aligned to IFRS (International Financial Reporting Standards, the accounting rules most businesses in Kenya and South Africa report under). The module is set up per entity for a chosen jurisdiction — Kenya (KE) or South Africa (ZA) — with its own base currency and financial-year start month.

### Overview

At a glance, Accounting covers the following connected areas:

- **Chart of accounts** — the structured list of accounts every transaction posts to, seeded from a template or imported.
- **Journals & recurring journals** — manual and automatic entries, with maker-checker control.
- **General ledger, trial balance & financial statements** — the running history and the reports drawn straight from it.
- **The posting engine** — rules that turn platform events into journals, with a retry queue for failures.
- **Tax** — VAT and withholding-tax returns computed from the ledger, plus jurisdiction tax packs.
- **Receivables** — customers, invoices, receipts, credit notes, ageing and online payments.
- **eTIMS** — Kenya invoice fiscalisation.
- **Payables** — suppliers, bills, supplier payments and batch payment runs.
- **Banking** — bank accounts, statement import, reconciliation, banking rules, transfers and expenses.

- **Exchange rates, revenue recognition & deferrals** — FX governance and accrual timing.
- **Trust accounting** — client money held in trust, for law firms and similar.
- **Cost centres & budgets** — segment reporting and plan-versus-actual.
- **Fixed assets** — the asset register, depreciation and disposal.
- **Period-end & consolidation** — FX revaluation, year-end close and a group view.

Everything is scoped to your organisation and entity, and every action is governed by permissions (see Access & permissions).

## Chart of accounts

The **chart of accounts** is the structured list of accounts your transactions are recorded against. Each account has a code, a name and a type — asset, liability, equity, income or expense — and accounts can be nested under a parent to give you a proper hierarchy. Rather than building the chart by hand, you can **instantiate** a chart from a **template** chosen for your industry, then adapt it. A CSV **import** is also available: download a template, validate it to a staging **preview** with no commit, then commit the validated rows into the entity chart.

Certain accounts play a **system role** — debtors control, creditors control, bank, VAT, retained earnings and so on — so the platform knows which account to post to automatically. You set these roles per entity, which means a business can keep its own account codes while the engine still knows where each posting belongs.

**Typical steps**

1. Go to **Chart of accounts** and **instantiate** a template, or **import** your own from CSV.
2. Add or edit accounts, setting each one's code, name, type and parent.
3. Assign the **system roles** (debtors, creditors, bank, tax, retained earnings) to the right accounts.

## Journals

A **journal** is a balanced entry — debits equal credits — that moves value between accounts. You can raise journals by hand for accruals, corrections and adjustments, record **opening balances**, and the module checks that every entry balances before it can be posted. Journals follow a maker-checker lifecycle: a draft can be edited, but once **posted** it becomes part of the permanent record and can only be corrected by a **reversal**, which writes an equal-and-opposite entry rather than deleting history.

**Recurring journals** automate entries that repeat — a monthly accrual, an amortisation — on a schedule. They generate as drafts by default so someone reviews them, or can be set to post automatically, and you can **pause**, **resume** or **end** a schedule at any time.

**Typical steps**

1. Go to **Journals** and click **New journal**.
2. Add lines with debits and credits until the entry balances.
3. **Post** it, or route it for approval where a rule requires sign-off.
4. To correct a posted journal, use **Reverse** and enter the reason.

## General ledger, trial balance & financial statements

Behind every account sits the **general ledger** — the full, ordered history of postings that make up its balance. The **trial balance** lists every account with its debit or credit balance, and a **journal register** lists the journals themselves. From the same postings the module produces the core financial statements: **profit and loss**, **balance sheet**, **cash flow** and **changes in equity**. Because all of these are drawn from one set of postings, they are internally consistent by construction. A management **dashboard snapshot** summarises cash, receivables, payables, a P&L summary and budget-versus-actual for the overview panel, and every report can be exported to **CSV** or a branded **PDF**, including a bundled **audit pack**.

## The posting engine

Most journals are never typed at all. A central **posting engine** turns platform **financial events** — an invoice issued, a credit note raised, a bill posted — into the matching journal using configurable **posting rules**, and tags each one with its source so you can trace a posting back to the document that caused it. **Posting requests** record each attempt; if one fails it lands in a queue you can inspect and **retry**, so an integration problem is visible and recoverable rather than silently lost.

## Tax

Accounting computes tax returns straight from posted journals. You maintain per-entity **tax codes** and can **install** a jurisdiction **tax pack** to seed the standard codes for Kenya or South Africa in one step. From the ledger it computes a **VAT return** and a **withholding-tax return** for a period as read-only reports; **filing** a return posts the reconciliation journal and records it in the filed-returns list. IXL CORE never hard-codes tax rates — the rates you apply are the ones in the tax codes and packs you configure.

## Receivables

The receivables area runs the sell-side. You keep **customers** — created directly or materialised from a CRM account — and raise **invoices** that move draft !' issued !' void, with a branded **PDF** and the option to **send** them. Issuing an invoice posts the receivable through the engine. **Recurring invoices** generate on a schedule, **receipts** record money received and allocate it against open invoices, and **credit notes** post the reversal. An **AR ageing** report shows what is outstanding by age.

**Online payments** let a customer pay an issued invoice through a tokenised public page: they view the invoice, see the available **gateways** and pay, with a webhook confirming the result. Internally you can request an online payment for an invoice ("Pay now"), track each **gateway payment** attempt and cancel

a still-pending one.

**Typical steps**

1. Create a **customer**, or pick one from CRM.
2. Raise an **invoice**, then **issue** and optionally **send** it.
3. Record a **receipt** against open invoices, or let the customer pay online.
4. Raise a **credit note** to reverse, and watch the **AR ageing** report.

## eTIMS

For Kenya, **eTIMS** handles KRA invoice fiscalisation. You register your **eTIMS items** and submit an invoice or credit note for fiscalisation, tracking the status per document and retrying where needed. A deterministic stub drives the flow until a platform operator enters real KRA credentials, so the workflow is ready before go-live.

## Payables

The payables area mirrors receivables on the buy-side. You keep **suppliers** — created directly or materialised from a Supply Chain supplier — and enter **bills** that move draft !' submitted !' posted !' void, with maker-checker so the person entering a bill need not be the one who posts it. Posting a bill raises the payable and any input VAT through the engine. **Supplier payments** settle open bills, handling withholding tax and realised FX, and a batch **payment run** gathers due bills, routes them for approval so the preparer is not the approver, then executes — posting one payment per supplier. An **AP ageing** report shows what is owed by age.

## Banking

You register your **bank accounts** and **import** statements against them from a CSV template. Statement lines can be **categorised** to an account by hand or automatically via **banking rules** that match description patterns, and a line can be **uncategorised** if it was matched wrongly. **Reconciliation** then agrees the statement to the books: you start a reconciliation, group statement lines against the matching ledger lines into **match groups**, work through what is left, and **complete** it once everything ties out. **Bank transfers** move money between two of your own accounts, and **expenses** let you spend straight from a bank or cash account without going through payables.

**Typical steps**

1. Register a **bank account** and **import** the statement.
2. Let **banking rules** categorise what they can, then finish the rest by hand.
3. Open a **reconciliation**, match lines to the ledger and **complete** it.

## Exchange rates, revenue recognition & deferrals

A governed **exchange-rate** store holds one rate per currency pair and date; documents look it up automatically so foreign-currency entries use a consistent, recorded rate. **Revenue recognition** schedules defer revenue to a deferred-revenue account when an invoice is issued, then release it to revenue over time or against milestones — run for what is due, or recognise a single milestone on demand. **Deferral schedules** do the same for prepayments and accruals, amortising a balance over a set number of monthly periods, and any schedule can be cancelled.

## Trust accounting

For law firms and similar, **trust accounting** keeps client money held in trust — always a liability, never income. You open a **matter** with its own ledger, record movements against it (a receipt into trust, a transfer to the firm, a refund, or a void) with an over-draw guard that stops a matter going negative, and close a matter once its balance is zero. A statutory **three-way trust reconciliation** confirms the trust bank balance, the sum of matter balances and the trust-control account all agree, with the per-matter breakdown a firm files.

## Cost centres & budgets

**Cost centres** let you tag journal lines to a part of the business — a department, branch or project — so you can report performance below the company total. **Budgets** capture your plan per account and cost centre by month; you can **import** budget lines from a CSV template into a draft, put the budget through maker-checker **approval**, and read a **budget-versus-actual** report that draws actuals from posted journals so overspend is visible while there is still time to act.

## Fixed assets

The **fixed-asset register** tracks capital items. You define **asset categories** carrying depreciation defaults — useful life, method and the relevant GL accounts — and register **assets** under them. Depreciation runs on a **straight-line** or **reducing-balance** basis; a periodic **depreciation run** posts the charge, and an asset can be **disposed** of when it leaves the business. **Register** and per-asset **movement-schedule** reports are available on screen and as CSV.

## Period-end & consolidation

The books are organised into **accounting periods** you can **close** and **reopen** under control, so figures cannot quietly shift after they have been reported. At period-end, **FX revaluation** restates foreign-currency balances at a chosen rate — run a **preview** first, then commit, idempotent per period and reversing into the next. The **year-end close** rolls the year forward: preview it, prepare it, route it through maker-checker approval, then commit to move the result into retained earnings, with a guarded, approval-routed **reopen** if it must be revisited.

Where you run more than one entity, **consolidation** combines them into a group. You define a group with its members and produce a combined **trial balance** and combined **statements** across them, with manual **elimination journals** the combined report nets out so intercompany balances are not double-counted. (Members are combined on a same-currency basis; FX translation across differing currencies

is deferred.)

## Access & permissions {#access-and-permissions}

Every Accounting action is governed by a capability — viewing the chart, creating a journal, posting or approving one, issuing an invoice, posting a bill, reconciling a bank account, filing a tax return, running depreciation, closing the year, and so on are each a separate permission (for example `accounting.journals.view`, `accounting.ar.view`, `accounting.period_end.manage`). Sections split cleanly into a view capability and a manage capability, so the people who prepare work need not be the people who approve or post it. This is enforced on every request, not merely hidden in the interface — which is what makes the separation of duties around posting, approval and payment real rather than cosmetic. Access is also gated by your commercial entitlement to the module.

## How Accounting connects

Accounting is the foundation the rest of the platform stands on:

- **CRM** feeds receivables — you can materialise an AR customer straight from a CRM account.
- **Supply Chain** feeds payables — you can materialise a supplier and its bills from a Supply Chain supplier.
- **The posting engine** turns finalised documents from across the platform into journals automatically, tagged with their source, and **reversals stay honest** — voiding a document reverses its ledger entry rather than erasing it.
- **Cost centres and periods** are shared, so the controls you set here apply to postings that originate anywhere on the platform.

That is the point of one connected ledger: a document is raised once, in the module where the work happens, and the books stay in step without a second system to reconcile.

# How-to guides

## Set up your chart of accounts

*Instantiate a chart of accounts from a template, add or edit accounts, and map system roles so postings land correctly.*

Your chart of accounts (CoA) is the backbone of the Accounting suite. In IXL CORE you can instantiate a ready-made chart from a template, then add or refine individual accounts and map the system roles that automated postings rely on.

### Before you start

- Decide the **scope** at which the chart lives. Every account carries a `scope_level` — one of Organisation, Entity, Branch, Department or Position. Organisation-level accounts are shared masters that any entity can reference.
- Confirm you hold the CoA capability for that scope.

### Steps

1. Open **Accounting !' Chart of Accounts**[screenshot: Chart of Accounts landing page]
2. **Instantiate from a template.** Choose a template and confirm. The only field is **Template code** (`template_code`, required) — it must match a known CoA template. Instantiation is validated by scope, so set the scope first. [screenshot: Instantiate chart of accounts dialog]
3. **Add or edit an account.** Provide the required fields: **Code** (`code`, up to 60 characters), **Name** (`name`, up to 200 characters) and **Account type** (`account_type`, one of *asset*, *liability*, *equity*, *income* or *expense*).
4. Optionally set **Subtype** (up to 120 characters), a **Parent account** (`parent_account_id`, to nest the account), **Postable** (`is_postable`), an account **Currency** (three-letter code) and **Status** (*active* or *inactive*). An **Owner user** may also be assigned. [screenshot: Account create form]
5. **Map system roles.** Open the roles panel and, for each role, pick the account it maps to. Both **Role** (`role`) and **Account** (`account_id`, required) must be supplied. Roles include AR control, AP control, VAT output/input, retained earnings, opening-balance equity, suspense and many more — these tell the engine where automated entries post.
[screenshot: System role mapping panel]

### Result

Your chart is live at the chosen scope, accounts are ready to receive postings, and system roles are mapped so invoices, bills, receipts and opening balances can post to the correct accounts automatically.

## Related

- Accounting reference
- Enter opening balances
- Post a journal

# Enter opening balances

*Post a controlled opening-balance journal so each account starts with the correct carried-forward figure.*

When you move onto IXL CORE mid-life, opening balances carry each account's starting position forward. Accounting posts these as a single controlled opening-balance journal, with the contra routed to the suspense / opening-equity role account so the entry balances.

## Before you start

- Instantiate your chart of accounts first, and map a **suspense / opening-equity role account** — the opening balance cannot post without it.
- Confirm the accounts you will use are all in the same scope you are posting at.
- Note that only one opening-balance journal is allowed per entity.

## Steps

1. Open **Accounting !' Journals** and choose **Opening balance**. [screenshot: Opening balance entry screen]
2. Set the **Accounting date** (`accounting_date`, required) — the date the balances are carried in at.
3. Choose the **Currency** (`currency`, required, three-letter code).
4. Optionally add a **Description** (`description`, up to 2000 characters).
5. Add each account's balance under **Balances** (`balances`, at least one row required). For every row select the **Account** (`acc_account_id`, required) and enter either a **Debit** (`debit`) or a **Credit** (`credit`) amount — both are optional per row and must not be negative, but each row needs one side. [screenshot: Opening balance line grid]
6. Review, then post. The engine adds the balancing contra to the suspense / opening-equity role account automatically.

## Result

A posted opening-balance journal sets every account's carried-forward figure, with the suspense / opening-equity account absorbing the contra so the books balance from day one. If a suspense role account is not mapped, or an opening balance already exists for the entity, posting is refused.

## Related

- Accounting reference

- Set up your chart of accounts
- Post a journal

# Post a journal

*Capture a balanced double-entry journal, or set up a recurring journal template, with maker-checker approval on large postings.*

Journals are the direct double-entry into your ledger. Every journal must balance, and large manual journals route through a maker-checker approval before they post. You can also define recurring journal templates that generate entries on a schedule.

## Before you start

- Have your chart of accounts and system roles in place.
- Remember the golden rule: total debits must equal total credits, and a journal needs at least two lines.
- Confirm the period you are posting into is open.

## Steps

### Post a one-off journal

1. Open **Accounting !' Journals** and start a new journal. [screenshot: New journal form]
2. Set the **Accounting date** (`accounting_date`, required) and **Currency** (`currency`, required, three-letter code). Optionally add a **Description** (up to 2000 characters) and an **FX rate** (`fx_rate`).
3. Add at least two **Lines** (`lines`, minimum two). For each line select the **Account** (`acc_account_id`, required) and enter a **Debit** or **Credit** amount (both optional per line, never negative). Each line may also carry a line **Description**, **Cost centre** (`acc_cost_centre_id`), **Tax code**, and entity/branch/department dimensions.
   [screenshot: Journal line grid]
4. **Post.** Balance is checked up front — an unbalanced journal is rejected before anything else. If the journal exceeds the maker-checker threshold, it routes to an eligible approver and posts only on approval; otherwise it posts immediately after balance and period checks. If no eligible approver exists, posting is refused. [screenshot: Journal posted confirmation]

### Set up a recurring journal

1. Open **Accounting !' Recurring Journals** and create a template.
2. Provide the **Name** (`name`, required), **Currency** (required), **Frequency** (`frequency` — *weekly*, *monthly* or *yearly*) and **Start date** (required). Optionally set an **Interval**, **End date**, **Max occurrences** and **Auto-post** (`auto_post`).
3.

Add **Template lines** (at least two). Each line requires an **Account**, a **Debit** and a **Credit** — a line must be one or the other, not both, and cannot be zero on both sides.
[screenshot: Recurring journal template]

## Result

Your journal posts to the ledger once balanced (and approved, where required), or your recurring template generates balanced journals automatically on schedule.

## Related

- Accounting reference
- Enter opening balances
- Set up your chart of accounts

# Bill a customer (invoices)

*Create an AR customer, raise a draft invoice, and optionally set up a recurring invoice template.*

Accounting owns Receivables. You bill a customer by first registering them as an AR customer, then raising an invoice. Tax is computed on the server from each line's tax code — you never enter rates yourself. Repeat billing can run from a recurring invoice template.

## Before you start

- Set your **scope** and confirm your chart of accounts is live.
- If the customer already exists in CRM, you can link the CRM master; otherwise the AR customer is finance-only.

## Steps

### Register the customer

1. Open **Accounting !' Receivables !' Customers** and add a customer.
2. Enter the **Name** (`name`, required). Optionally add **KRA PIN**, **Email**, **Mobile**, **Address**, **Default currency** (three-letter code), **Payment terms (days)** (`payment_terms_days`), an **AR account code**, a linked **CRM account** (`crm_account_id`) and **Status** (*active* or *inactive*). [screenshot: AR customer form]

### Raise an invoice

1. Open **Accounting !' Receivables !' Invoices** and start a new invoice.
2. Select the **Customer** (`acc_customer_id`, required), set the **Invoice date** (required) and **Currency** (required). Optionally add a **Due date**, **FX rate**, **Trader invoice no.**, **Branch code** and a **Remark**. [screenshot: New invoice header]
3. Add at least one **Line** (`lines`, minimum one). Each line needs a **Quantity** and **Unit price** (both required), plus either a **Product** (`supply_product_id`) or a **Description** — one of the two is required. Optionally set a **Tax code** and, to spread revenue over time, a **Recognition method** (*straight_line*, *milestone* or *point_in_time*); *straight_line* also requires **Recognition periods**. [screenshot: Invoice line grid]
4. Save and issue. Tax is calculated server-side from each line's tax code.

### Recurring invoices

Under **Recurring Invoices**, create a template with a **Name**, **Customer**, **Currency**, **Frequency** (*weekly*/*monthly*/*yearly*) and **Start date**. Add template lines (quantity + unit price + product or description), and optionally enable **Auto-issue**.

*[ Screenshot: Recurring invoice template ]*

## Result

Your customer is on file and the invoice is raised with tax computed automatically, ready to be issued and later receipted.

## Related

- Accounting reference
- Record a customer receipt
- Issue a credit note

# Record a customer receipt

*Receive a customer payment and allocate it across their outstanding invoices.*

When a customer pays, you record a receipt against the bank or cash account and, optionally, allocate it across their open invoices. Allocation totals are checked server-side against the live ledger, so you can never over-allocate.

## Before you start

- The customer must already exist as an AR customer.
- Know which **bank or cash account** the funds landed in — it must be a real account in your chart.
- Have the invoices you intend to settle to hand.

## Steps

1. Open **Accounting !' Receivables !' Receipts** and start a new receipt.
   [screenshot: New receipt form]
2. Select the **Customer** (`acc_customer_id`, required).
3. Set the **Receipt date** (`receipt_date`, required) and **Currency** (required, three-letter code). Optionally add an **FX rate**.
4. Enter the **Amount** (`amount`, required, greater than zero).
5. Choose the **Bank / cash account** (`bank_account_id`, required) the money was received into.
6. Optionally set the **Payment method** (`payment_method`) — cash, credit, cheque, card, mobile money or other — and a **Reference** (up to 120 characters). [screenshot: Receipt payment details]
7. Optionally add **Allocations** to spread the receipt across invoices. For each allocation select the **Invoice** (`acc_invoice_id`) and the **Amount** (greater than zero) to apply. The server checks that total allocations do not exceed the receipt amount, and that no invoice is over-paid against its live outstanding balance. [screenshot: Receipt allocation grid]
8. Save. Unallocated amounts sit as a credit on the customer's account.

## Result

The receipt posts to the bank/cash account, the customer's balance reduces, and any allocated invoices are settled up to the amounts applied.

## Related

- Accounting reference
- Bill a customer (invoices)
- Issue a credit note

# Issue a credit note

*Credit a customer against a source invoice or with explicit lines, letting the system compute the tax.*

A credit note reverses value back to a customer — for returns, corrections or goodwill. You can credit a specific invoice (the note prefills from it) or enter explicit lines. As with invoices, tax is computed on the server from each line's tax code.

## Before you start

- The customer must exist as an AR customer.
- Decide whether you are crediting a **source invoice** or entering **lines** from scratch — you need at least one of the two.

## Steps

1. Open **Accounting !' Receivables !' Credit Notes** and start a new credit note.
   [screenshot: New credit note form]
2. Select the **Customer** (`acc_customer_id`, required).
3. Optionally choose a **Source invoice** (`acc_invoice_id`) to credit — the note prefills its lines from that invoice.
4. Set the **Credit note date** (`credit_note_date`, required) and **Currency** (required, three-letter code). Optionally add an **FX rate**, a **Reason** (up to 500 characters) and a **Branch code**.
   [screenshot: Credit note header]
5. Add **Lines** if you are not crediting from a source invoice. The lines block is optional overall, but if you add lines each row needs a **Quantity** and **Unit price**, plus either a **Product** (`supply_product_id`) or a **Description**. A **Tax code** is optional per line. You must supply either at least one line or a source invoice — otherwise the note is rejected.
   [screenshot: Credit note line grid]
6. Save and issue. Tax is calculated server-side from each line's tax code.

## Result

The credit note is issued against the customer, reducing what they owe. Where a source invoice was selected, the note ties back to it for a clean audit trail.

## Related

- Accounting reference
-

- Bill a customer (invoices)
- Record a customer receipt

# Enter a supplier bill

*Register an AP supplier and capture a supplier bill, letting the system compute input VAT from each line's tax code.*

Accounting owns Payables. To record what you owe, first register the supplier, then capture their bill. Each line targets an expense account, and input VAT is computed on the server from the line's tax code — rates are never entered by hand.

## Before you start

- Set your **scope** and confirm your chart of accounts is live.
- If the supplier already exists in Supply Chain, you can link that master; otherwise the AP supplier is finance-only.

## Steps

### Register the supplier

1. Open **Accounting !' Payables !' Suppliers** and add a supplier.
2. Enter the **Name** (`name`, required). Optionally add **KRA PIN**, **Email**, **Phone**, **Address**, **Default currency** (three-letter code), **Payment terms (days)** (`payment_terms_days`), an **AP account code** and **Status** (*active* or *inactive*).
3. For withholding tax, optionally set **WHT applicable** (`wht_applicable`) and a **Default WHT tax code**. You may also link a Supply Chain supplier via **Supplier** (`supply_supplier_id`). [screenshot: AP supplier form]

### Enter the bill

1. Open **Accounting !' Payables !' Bills** and start a new bill.
2. Select the **Supplier** (`acc_supplier_id`, required), set the **Bill date** (required) and **Currency** (required). Optionally add the **Supplier invoice no.**, a **Due date**, **FX rate**, **Branch code** and a **Remark**. [screenshot: New bill header]
3. Add at least one **Line** (`lines`, minimum one). Each line needs a **Quantity** and **Unit price** (both required), plus either a **Product** (`supply_product_id`) or a **Description** — one is required. Optionally set a **Tax code** and target an **Expense account** (`expense_account_id`, or `expense_account_code`); if omitted, the product's default expense account is used. [screenshot: Bill line grid]
4. Save. Input VAT is calculated server-side from each line's tax code.

## Result

The supplier is on file and the bill is recorded against Payables, with each line posted to its expense account and input VAT computed automatically — ready for ageing and payment.

## Related

- Accounting reference
- Post a journal
- Set up your chart of accounts

# Pay suppliers and run a payment run

*Pay a single supplier against their outstanding bills, or select several due bills and settle them in one batch payment run.*

Settle what you owe suppliers — either pay one supplier against their bills, or gather several due bills and clear them together in a payment run.

## Before you start

- Pick the company you are working in and the **Scope level** for the payment.
- Make sure the supplier's bills are posted with an outstanding balance — you can only pay against live, outstanding bills.
- Know which bank account the money leaves from.

## Steps

### Pay a single supplier

1. Open **Accounting !’ Payables** and start a supplier payment.
2. Set the **Scope level** (required) plus any parent scope such as entity or branch.
3. Choose the **Supplier** (required) and the **Payment date** (required).
4. Enter the three-letter **Currency** code (required) and, for a foreign-currency payment, an **FX rate**.
5. Optionally choose a **Payment method** — Cash, Credit, Cheque, Card, Mobile money or Other.
6. Select the **Bank account** the payment is drawn from (required).
7. Optionally add a **Reference** (up to 120 characters).
8. Add one or more **Allocations** (at least one required): for each, pick the **Bill** and enter an **Amount** greater than zero.[screenshot: Supplier payment allocations]
9. Save. The system checks each bill's outstanding balance and computes any withholding tax against the live ledger.

### Run a batch payment run

1. In **Accounting !’ Payables**, start a payment run.
2. Set the **Scope level** (required) and any parent scope.
3.

Enter the **Run date** (required) and the three-letter **Currency** (required).

4. Choose the **Bank account** the run pays from (required).
5. Select the due **Bills** to include (at least one required).[screenshot: Payment run bill selection]
6. Save. The run builds one item per selected bill at that bill's outstanding amount.

## Result

A single payment records against the chosen bills, reducing their outstanding balances (with withholding tax handled automatically). A payment run settles every selected bill at its outstanding amount in one batch, posted from the nominated bank account.

## Related

- Record an expense
- Set up bank accounts and reconcile
- Accounting reference

# Record an expense

*Record a spend straight from a bank or cash account without raising a supplier bill, letting the system compute the input VAT.*

An expense records money spent directly from a bank or cash account — no supplier bill needed — and lets the system work out the input VAT for you.

## Before you start

- Pick the company you are working in and the **Scope level** for the expense.
- Know which account the money was paid from and which category (expense) account it belongs to.
- Have any tax code ready if the spend is taxable.

## Steps

1. Open **Accounting !' Expenses** and start a new expense.
2. Set the **Scope level** (required) plus any parent scope.
3. Enter the **Expense date** (required).
4. Choose the **Paid from account** — the bank or cash account the money left (required).
5. Choose the **Category account** — the expense account the spend posts to (required).
6. Enter a **Description** (required, up to 255 characters).
7. Optionally add a **Supplier name** as free text (up to 255 characters) — an expense needs no full supplier record.
8. Enter the three-letter **Currency** code (required) and, for a foreign-currency spend, an **FX rate**.
9. Enter the **Net amount** (required, greater than zero).[screenshot: Expense form]
10. Optionally set a **Tax code** (up to 60 characters); the system computes the input VAT and total against the resolved tax code.
11. Optionally add a **Reference** (up to 120 characters) and any **Document references**.
12. Save.

## Result

The expense posts directly against the paid-from account and the chosen category account, with input

VAT and the total computed from the tax code you supplied. No accounts payable entry is created — the money is treated as already spent.

## Related

- Pay suppliers and run a payment run
- Set up bank accounts and reconcile
- Set up tax codes and tax packs
- Accounting reference

# Set up bank accounts and reconcile

*Create bank accounts, import statements, set banking rules, transfer between accounts, and reconcile statement lines against the general ledger.*

Set up your bank accounts, bring in statements, and match every statement line to the ledger so your books agree with the bank.

## Before you start

- Pick the company you are working in and the **Scope level** for each record.
- Have the cash or bank GL account from your chart ready to link to each bank account.
- Export or prepare the bank statement you want to import (the file is parsed in your browser).

## Steps

### Create a bank account

1. Open **Accounting !' Banking !' Bank accounts** and add a bank account.
2. Set the **Scope level** (required) and any parent scope.
3. Enter the **Name** (required) and, optionally, the **Bank name** and **Account number**.
4. Enter the three-letter **Currency** code (required).
5. Choose the **GL account** — the cash/bank account from the chart this bank feeds (required).
6. Optionally set whether the account **is active**. Save.

### Import a statement

1. In **Banking**, choose the **Bank account** (required) and import a statement.
2. Optionally set the **Statement date**, **Opening balance**, **Closing balance** and imported file name.
3. The parsed **rows** are posted: each row needs a **date**, **description** and **amount**, with optional **reference** and **running balance**. Duplicate lines are detected and skipped.[screenshot: Bank statement import]

### Add banking rules (optional)

1. Under **Banking rules**, add a rule with a **Name** (required).
2.

Set the **Match field** (description or reference), **Match operator** (contains, equals or starts with) and **Match value** (all required).

3. Choose the **Contra account** to suggest (required). Optionally set a specific bank account (blank applies to all), whether it is active, and a **Priority**.

### Transfer between accounts

1. Under **Transfers**, pick the **From bank account** and **To bank account** (must differ) and a **Transfer date**.
2. Enter the **Amount** (greater than zero). For a cross-currency move, add the **To amount** and **FX rate** . Optionally add a **Reference**.

### Reconcile

1. Start a reconciliation for a **Bank account** (required), entering the **Statement balance** (required) and an optional **GL as of** date and **Period label**.[screenshot: Reconciliation workspace]
2. **Match** a statement line to a general-ledger line, or use a **grouped match** for several statement lines to several GL lines, with an optional small-difference **adjustment** (account plus memo).
3. **Categorise** any unmatched line by choosing a **Contra account** to post it to.

## Result

Each bank account links to its GL account, statements import without duplicates, and every statement line is matched, grouped or categorised so the bank balance reconciles to the ledger.

## Related

- Record an expense
- Pay suppliers and run a payment run
- Accounting reference

# Set up cost centres

*Create cost centres as a financial dimension so spend and income can be tracked and budgeted by area of the business.*

Cost centres are a financial dimension you can tag against transactions and budgets, letting you track income and spend by area of the business.

## Before you start

- Pick the company you are working in and the **Scope level** for the cost centre.
- Decide on a short, unique code and a clear name for each cost centre.

## Steps

1. Open **Accounting !’ Cost centres** and add a new cost centre.
2. Set the **Scope level** (required) plus any parent scope such as entity or branch.
3. Enter the **Code** (required, up to 64 characters) — this must be unique per organisation and entity.
4. Enter the **Name** (required, up to 160 characters).[screenshot: Cost centre form]
5. Optionally add a **Description** (up to 1,000 characters).
6. Optionally set the **Status** to **Active** or **Inactive** (new cost centres are active by default).
7. Save.

## Result

The cost centre is available as a financial dimension across Accounting — you can tag it on budget lines and analyse income and spend by cost centre. Inactive cost centres remain on record but are no longer offered for new entries.

## Related

- Create and track a budget
- Record an expense
- Accounting reference

# Create and track a budget

*Build a budget for a fiscal year with lines per profit-and-loss account, entered monthly or as an annual figure, and compare against actuals.*

A budget sets your planned figures for a fiscal year, account by account, so you can compare plan against actual as the year runs.

## Before you start

- Pick the company you are working in and the **Scope level** for the budget.
- Know the fiscal year you are budgeting and which profit-and-loss accounts to include.
- Set up any cost centres first if you want to budget at that level.

## Steps

1. Open **Accounting !’ Budgets** and add a new budget.
2. Set the **Scope level** (required) plus any parent scope.
3. Enter the **Name** (required, up to 160 characters).
4. Enter the **Fiscal year** (required, a four-digit year).
5. Add **Budget lines** (at least one required). For each line:
   [screenshot: Budget lines editor]
6. Save.

## Result

The budget holds a line per account (optionally split by cost centre), captured either as a single annual figure or spread across the twelve months. Once saved, you can compare planned figures against actuals in **Budget vs actual** to see where you are over or under.

## Related

- Set up cost centres
- Record an expense
- Accounting reference

# Set up tax codes and tax packs

*Install a jurisdiction tax pack, create individual tax codes, and register sellable items with eTIMS for Kenyan tax compliance.*

Get your tax set up by installing a ready-made jurisdiction pack, adding your own tax codes, and registering items with eTIMS where required.

## Before you start

- Pick the company you are working in and the **Scope level** for each record.
- Decide which jurisdiction applies — Kenya (KE) or South Africa (ZA).
- Have your chart of accounts in place so tax codes can map to the right roles.

## Steps

### Install a tax pack

1. Open **Accounting !' Tax** and choose to install a tax pack.
2. Set the **Scope level** (required) plus any parent scope.
3. Choose the **Jurisdiction** — **KE** or **ZA** (required).
4. Install. The pack seeds the standard VAT and withholding codes for that jurisdiction. [screenshot: Install tax pack]

### Create a tax code

1. Under **Tax codes**, add a new code.
2. Set the **Scope level** (required) and any parent scope.
3. Enter the **Code** (required, up to 60 characters) and **Name** (required, up to 200 characters).
4. Choose the **Jurisdiction** — **KE** or **ZA** (required).
5. Set the **Tax type** — **VAT** or **WHT** (required).
6. Enter the **Rate** as a percentage (required).
7. Optionally set whether the tax **is recoverable**, an **Output role** and **Input role** (the GL system roles the tax posts to), and the **Status** (active or inactive).

### Register an item with eTIMS

1. Under **eTIMS**, register a sellable item.
2. Identify the item by either a **Supply product** or a free-text **Item code** (one is required).
3. Optionally set the **Name**, **Class code**, **Item type**, **Unit of quantity** and **Packaging unit**.
4. Optionally choose the **Taxation type** — A (exempt), B (standard), C (zero-rated) or D (non-VAT). [screenshot: eTIMS item registration]

## Result

Your jurisdiction's standard tax codes are installed, any custom codes carry their rate and GL roles, and sellable items are registered with eTIMS ready for compliant invoicing.

## Related

- Record an expense
- Pay suppliers and run a payment run
- Accounting reference

# Manage exchange rates

*Record and maintain the exchange rates used to translate foreign-currency amounts, keyed by currency pair and date.*

Keep a table of exchange rates so foreign-currency transactions and revaluations translate at the right figure for each date.

## Before you start

- Pick the company you are working in and the **Scope level** for the rate.
- Know the currency pair you are quoting and the rate that applies on a given date.
- Rates are entered by hand; the recorded source lets you tell a manual override from a fetched feed later.

*[ Screenshot: Exchange Rates workspace with the rate table and the "Add rate" form ]*

## Steps

### Record an exchange rate

1. Open **Accounting !' Exchange Rates** and start a new rate.
2. Set **From currency** (required) — a three-letter currency code.
3. Set **To currency** (required) — a three-letter code that must be **different** from the from-currency.
4. Choose the **Rate date** (required) — the date the rate applies from.
5. Enter the **Rate** (required) — a number greater than zero (units of the to-currency per one from-currency).
6. Optionally set the **Source** of the rate. When omitted the store defaults it; the source records where the rate came from.
7. Save the rate.

*[ Screenshot: Exchange rate form showing From currency, To currency, Rate date and Rate ]*

### Edit or remove a rate

1. Select an existing rate to edit any of the same fields, then save.
2. Delete a rate you no longer need from the rate table.

## Result

The rate is stored against its currency pair and date and becomes available to the FX lookup used across Accounting — including period-end foreign-currency revaluation and any transaction captured in a foreign currency.

## Related

- Accounting reference
- Run period and year-end close
- Consolidate group entities

# Manage fixed assets and depreciation

*Set up asset categories with depreciation defaults, register individual assets, and run monthly depreciation to post the periodic charge.*

Register your capital assets, group them into categories that carry the depreciation defaults, and let the depreciation run post each period's charge.

## Before you start

- Pick the company you are working in and the **Scope level** for each record.
- Set up your asset **categories** first — an asset takes its depreciation defaults from its category.
- Depreciation is calculated by **straight line** ((cost " salvage) over the useful life) o**reducing balance** (net book value times a monthly rate).

*[ Screenshot: Fixed Assets workspace showing categories, the asset register and the depreciation run ]*

## Steps

### Create an asset category

1. Open **Accounting !' Fixed Assets** and add a category.
2. Enter the **Name** (required) and an optional **Description**.
3. Set the **Default useful life (months)** (required, between 1 and 1200).
4. Choose the **Default depreciation method** (required) — Straight line or Reducing balance.
5. Optionally set the **default salvage rate**, the **default reducing-balance rate**, and per-category GL overrides for the cost, accumulated-depreciation and depreciation-expense accounts.
6. Optionally set the category **status** (active or inactive) and save.

*[ Screenshot: Asset category form with useful life and depreciation method defaults ]*

### Register an asset

1. Add a new asset and enter its **Name** (required) and optional **Description**.
2. Link it to an asset category (optional — it supplies the depreciation defaults).
3. Enter the **Acquisition cost** (required, at least 0.01) and the **Acquisition date** (required).
4. Optionally set the **In-service date**, **Useful life (months)**, **depreciation method**, **Salvage value** and **reducing-balance rate**; when left blank these default from the category.

5. Optionally set the **status** (draft or active) and a cost centre, then save.

*[ Screenshot: Fixed asset form with acquisition cost, dates and depreciation fields ]*

### Run depreciation

1. Run depreciation for a period to post the periodic charge for active assets.
2. Review each asset's **depreciation schedule** and the **movement schedule** (cost and accumulated depreciation opening, additions, disposals and closing), which you can export to CSV.

## Result

Categories hold the defaults, each asset carries its own basis, and the depreciation run posts the charge — building the fixed-asset register and its movement schedule.

## Related

- Accounting reference
- Run period and year-end close
- Set up revenue recognition and deferrals

# Set up revenue recognition and deferrals

*Build recognition schedules that release deferred revenue to income and deferral schedules that amortise prepayments or accruals over time.*

Spread income and costs over the periods they belong to: recognition schedules release deferred revenue to income, and deferral schedules amortise a prepayment or accrual.

## Before you start

- Pick the company you are working in and the **Scope level** for each schedule.
- Have the GL accounts ready — the amount per period is derived for you (total divided by the periods, with the last period absorbing the rounding).
- Decide the method: straight line spreads evenly, milestone recognises discrete amounts on demand, and point-in-time releases the whole amount on a date.

*[ Screenshot: Revenue Recognition workspace showing the schedule list and form ]*

## Steps

### Create a recognition schedule

1. Open **Accounting !' Revenue Recognition** and add a schedule.
2. Choose the **Recognition method** (required) — Straight line, Milestone or Point in time.
3. Enter the **Description** (required) and the **Total amount** (required, greater than zero); currency is optional.
4. Set the **Start date** (required).
5. Choose the **Revenue account** (required) — it must differ from the deferred account. The **Deferred revenue account** is optional and defaults to the deferred-revenue role account.
6. For straight line, set **Periods (months)** (required for this method, 1–600). For milestone, add **milestones** — each with an optional label and a required amount and due date.
7. Save the schedule.

*[ Screenshot: Recognition schedule form with method, accounts and periods ]*

### Edit and run

1. Editing is a partial update — you may change just the description, or the full basis when nothing has been recognised. When part-recognised, the service limits you to the description and the future (unposted) **entries**.

2. Use **Recognise due now** to release straight-line and point-in-time periods that have fallen due; mark a **milestone** entry done to recognise it.

### Create a deferral schedule

1. From the deferrals area, add a schedule and choose the **Deferral type** (required) — Prepayment or Accrual.
2. Enter the **Description** and **Total amount** (both required), set the **Balance account** and the **Expense account** (both required, and different), the **Start date** and the **Periods (months)** (1–600).
3. Save, then use **Post due now** to post each period's journal as it falls due.

## Result

Deferred revenue is released to income on schedule and prepayments or accruals amortise over their term, each posting its periodic journal automatically.

## Related

- Accounting reference
- Run period and year-end close
- Manage fixed assets and depreciation

# Run period and year-end close

*Open accounting periods, run foreign-currency revaluation, then close and reopen periods and complete the year-end close.*

Define your accounting periods, run any foreign-currency revaluation, then lock a period once it is complete and run the year-end close at the end of the financial year.

## Before you start

- Pick the company you are working in and the **Scope level** for the period.
- Reopening a period and the year-end close are approval-controlled actions.
- Have your exchange rates in place if you rely on foreign-currency revaluation at period-end.

*[ Screenshot: Periods list with the "Open period" form ]*

## Steps

### Open a period

1. Open **Accounting !’ Periods** and open a new period.
2. Enter the **Period code** (required, up to 20 characters).
3. Set **Starts on** (required) and **Ends on** (required — must be on or after the start date).
4. Save the period.

*[ Screenshot: Open period form with period code and start/end dates ]*

### Revalue foreign currency

1. Go to **Accounting !’ Period-End** and choose the period.
2. Use **Preview reval** to see the revaluation by account (carrying, revalued and adjustment amounts), then run it to post the adjustment.

*[ Screenshot: Period-End tab showing the FX revaluation preview ]*

### Close and reopen a period

1. From the Periods list, **Close** a period once it is complete to lock it against further posting.
2. To make changes again, use **Reopen (approval)** — this raises an approval request.

### Run the year-end close

1. On the Period-End tab, pick the **Financial year** and use **Preview** to review the year-end result and equity movement.
2. Run the close to complete the year-end (approval-controlled). If needed, **Reopen** a completed year-end, which is also approval-controlled.

## Result

Periods are opened, revalued and locked in order, and the year-end close finalises the financial year with the appropriate approvals recorded.

## Related

- Accounting reference
- Manage exchange rates
- Consolidate group entities

# Consolidate group entities

*Define a consolidation group of member entities, produce a combined trial balance and statements, and post manual elimination journals.*

Bring several entities together into a consolidation group, translate them to a single presentation currency, and post the eliminations a group set of accounts needs.

## Before you start

- Pick the company you are working in and the **Scope level** for the group.
- Know which entities are members and, optionally, which is the parent.
- Elimination journals are balanced double-entry postings; the journal service enforces balance and period control.

*[ Screenshot: Consolidation groups list with the group form ]*

## Steps

### Create a consolidation group

1. Open **Accounting !’ Consolidation** and add a group.
2. Enter the **Name** (required, up to 160 characters).
3. Set the **Presentation currency** (required) — a three-letter currency code.
4. Optionally choose the **parent entity**.
5. Add the **member entities** (required — at least one).
6. Optionally set the group **status** (active or inactive) and save.

*[ Screenshot: Consolidation group form with presentation currency and member entities ]*

### Review the combined figures

1. Open a group to view its **Combined trial balance** across all members.
2. Review the combined statements the group produces.

*[ Screenshot: Combined trial balance for a consolidation group ]*

### Post an elimination journal

1. From the group, post a manual **elimination journal** tagged to that group.
2. Set the **Accounting date** (required); currency and a description are optional.
3. Add at least two **lines**, each with an **account** (required) and a **debit** or **credit** amount, plus an optional line description.
4. Save — the lines must balance, and the journal is subject to period control.

## Result

The group draws its members into a combined trial balance and statements in the presentation currency, with your elimination journals removing intra-group balances.

## Related

- Accounting reference
- Manage exchange rates
- Run period and year-end close

# Manage trust accounting

*Open trust matters, record receipts, transfers to firm and refunds against a trust bank account, and check the statutory three-way reconciliation.*

Hold client money separately from firm money: open a matter per engagement, record every movement against a trust bank account, and confirm the statutory three-way reconciliation agrees.

## Before you start

- Pick the company you are working in and the **Scope level** for each record.
- Every movement amount is positive; the action you choose (receipt, transfer to firm, or refund) sets the direction.
- Per-matter balance and over-draw guards run against the live matter balance, so you cannot draw a matter below zero.

*[ Screenshot: Trust workspace with the Matters, Transactions and Reconciliation tabs ]*

## Steps

### Open a trust matter

1. Open **Accounting !' Trust** and add a matter.
2. Enter the **Matter name** (required, up to 200 characters).
3. Optionally link a **Matter client** (an existing customer) and add a **Matter description**.
4. Save the matter.

*[ Screenshot: Trust matter form with name and client ]*

### Record a trust movement

1. Choose **Record a trust movement** and pick the action — **Receipt into trust**, **Transfer to firm** or **Refund**.
2. Select the **Matter** (required) and the **Trust bank account** (required).
3. Set the movement date (required) and the **Movement amount** (required, greater than zero); currency is optional.
4. Add a **Movement reference** (optional). On a **Transfer to firm** you may link the invoice the transfer settles.
5. Save the movement. A movement can later be voided.

*[ Screenshot: Record a trust movement form with matter and trust bank account ]*

### Check the reconciliation

1. Open the **Reconciliation** tab to view the statutory three-way check: the **Trust bank balance**, the **matter balance total** and the **trust control balance** must all agree for the reconciliation to be marked reconciled.
2. Review the **per-matter breakdown** showing each matter's status and trust balance.

## Result

Client money is tracked matter by matter, every movement is audited, and the three-way reconciliation confirms the trust bank, matter balances and trust control account all agree.

## Related

- Accounting reference
- Run period and year-end close
- Manage exchange rates