One payroll, many countries

Compliant cross-border pay — without the spreadsheet sprawl

Version 1.0 · ixlcore.com

Executive summary

Payroll is the one process a business cannot get wrong. Pay people late and you lose trust. File late and you attract penalties. Miscalculate a statutory contribution and the exposure sits quietly on the books until an audit finds it. For a single country, most businesses eventually manage — a spreadsheet, a template, a person who knows the rules.

Then the business crosses a border. It opens an entity in a second country, or hires a team there, and payroll quietly doubles in complexity. Each country has its own statutory rules, its own contributions, its own filing deadlines and its own forms. The spreadsheet that held together for one country now has to hold together for two — and it doesn't. This paper argues that multi-country payroll shouldn't mean multiplying a fragile process. It should mean one connected payroll that is **compliant by default** — where each jurisdiction's rules are encoded in the system, not carried in someone's head.

The problem: payroll multiplies with every country and entity

Cross-border payroll rarely arrives as a decision. It accumulates. A business wins work in a neighbouring country, registers an entity to invoice locally, hires two people there — and now it runs payroll in two places. Win in a third, and it's three.

The trouble is that these payrolls don't stack neatly. Each one is different:

- **Different statutory rules.** Kenya and South Africa calculate income tax, social contributions and levies in entirely different ways, on different bases, with different thresholds. Nothing carries over.
- **Different filing calendars.** Each country files on its own schedule, to its own authority, in its own format. Miss one and the penalty is real.
- **Different edge cases.** An employee paid partly in foreign currency, a mid-month joiner, a back-dated increase — each country handles these differently, and each has to be handled correctly.

So the business ends up running a separate process per country, per entity — and every one of them is fragile. Worse, they usually depend on a few people who "know the rules" for their jurisdiction. When that person is on leave, or leaves, the knowledge goes with them. Multi-country payroll is precisely where errors, late filings and compliance risk concentrate — because it is the most complex, least connected part of the business, run on the least robust tools.

Why the usual answers fall short

Four options present themselves, and each solves a different problem than the one you have.

Spreadsheets are where almost everyone starts, and they work — until they don't. A spreadsheet has no memory of the rules and no guard against a mistake. It calculates whatever the formula says, even when the formula is a year out of date or the statutory rate has changed. Run it across two countries and you now maintain two sets of formulas by hand, hoping both stay current. Spreadsheets don't fail loudly; they fail quietly, and you find out at filing time.

Single-country payroll apps are built well for one jurisdiction — and only one. To pay staff in a second country, you buy a second app, learn a second interface and keep two systems that share nothing. The employee who moves between entities exists twice. There is no single view of what the business spends on people, and no single place the rules live. You've solved one country by acquiring the multi-tool problem.

Outsourced bureaus move the work to a specialist, which can help — but it moves the knowledge outside the business too. You send data out, you get numbers back, and you're dependent on a third party's timelines, accuracy and responsiveness. Coordinating one bureau per country, each with its own process, recreates the fragmentation you were trying to escape — now at arm's length, where you can see it least.

Heavy ERP can, in principle, do multi-country payroll. But traditional ERP is a long, expensive implementation priced for large corporations, and its payroll is often a generic module bent to fit each country after the fact. For a growing business, it's a great deal of weight to carry for one process.

So the business is caught: tools too narrow to cross borders, or systems too heavy to adopt — with the compliance risk sitting in the gap between them.

A different idea: payroll that is compliant by default

The premise is simple. If a jurisdiction's statutory logic is known, it should live in the system — not in a formula someone maintains, not in one employee's memory, not in a bureau's back office. Payroll should be **compliant by default**: you run it, and the rules apply themselves, correctly, every cycle.

That means one payroll system, not one per country. The countries you operate in are configuration, not separate products. Each jurisdiction's rules are encoded once and applied automatically — so adding a country is a matter of switching on a rule-pack, not standing up a parallel process. The knowledge stops being a person and becomes part of the platform.

And because that payroll sits on the same shared foundation as the rest of the business, it is connected by default. A new hire flows into payroll, permissions and the org chart at once — one record, seen from every angle — rather than being re-entered into a separate system that keeps its own private copy of who works where.

How it works

Country statutory rule-packs. Each country's income tax, statutory contributions, levies, thresholds and filing outputs are encoded as a rule-pack. IXL CORE ships with rule-packs for Kenya and South Africa, and the model is extensible — additional countries are added as plugins, so the same one

system grows across borders instead of forcing a second one. When a rule-pack is maintained centrally, every entity that uses it stays current at once.

Per-entity runs. Each legal entity runs its own payroll against the right country's rules, on its own calendar — but from a single platform, with a single view across all of them. You see each entity's run in isolation for compliance, and the whole picture together for control.

A monthly-first cycle. The engine is built monthly-first, matching how salaried teams across the market are actually paid, with a clear run cockpit that walks each cycle from draft to approval to pay — so nothing is filed on a formula nobody checked.

FX, gross-up and arrears in the engine. The hard cases are handled where they belong. An employee paid in a foreign currency, a net-pay guarantee that has to be grossed up before tax, a back-dated increase that owes arrears across prior periods — these are calculated by the engine, consistently, not improvised in a spreadsheet cell. The logic that used to live in someone's head lives in the system, the same way every time.

Components and analytics. Pay is assembled from defined components — earnings, deductions, benefits, statutory items — so every payslip is built the same way and every line is explainable. Analytics then read across entities and countries: total cost of employment, statutory liabilities, movements over time — one lens on people spend, wherever your people are.

What it looks like in practice

Consider a business paying staff in both Kenya and South Africa.

Before. Two payrolls, run two different ways. A spreadsheet for the Kenyan entity, maintained by the person who knows Kenyan rules. A separate app — or a bureau — for the South African entity, with its own login, its own calendar, its own point of contact. Each month, two processes run in parallel, two sets of numbers are produced, and someone stitches them together to see what the business actually spent on people. When a statutory rate changes, one process gets updated and the other, sometimes, doesn't. When the person who knows the Kenyan rules is away, that payroll waits.

After. One system. The Kenyan entity runs against the Kenya rule-pack, the South African entity against the South Africa rule-pack — each correct for its jurisdiction, both from the same cockpit. A new hire in either country enters once and appears in payroll, permissions and the org chart together. Foreign-currency pay, gross-ups and arrears are computed by the engine. Analytics show the combined cost of employment across both entities in a single view. There is nothing to stitch together, because there was only ever one payroll — configured for two countries.

The outcome

When multi-country payroll moves onto one compliant-by-default system, the change isn't that a task gets faster. It's that a risk gets smaller. Filings go out on time because the calendar and the rules are in the system, not in a person's diary. Calculations hold because the statutory logic is encoded once and applied everywhere. The business stops depending on a handful of people who carry the rules in their heads — and adding the next country becomes a configuration, not a crisis.

The question for any business crossing borders is no longer *"who runs payroll in the new country?"* It's *"why would payroll get riskier every time we grow?"* On one connected payroll, it doesn't.

About IXL CORE

IXL CORE is the Business Operating System for growing businesses — one connected platform for customers, money, stock, people and work, shaped to your industry and built for how businesses across Africa actually operate. Its payroll runs many countries as one, compliant by default, on the same foundation as HR and accounting — so the business grows across borders without multiplying the risk.