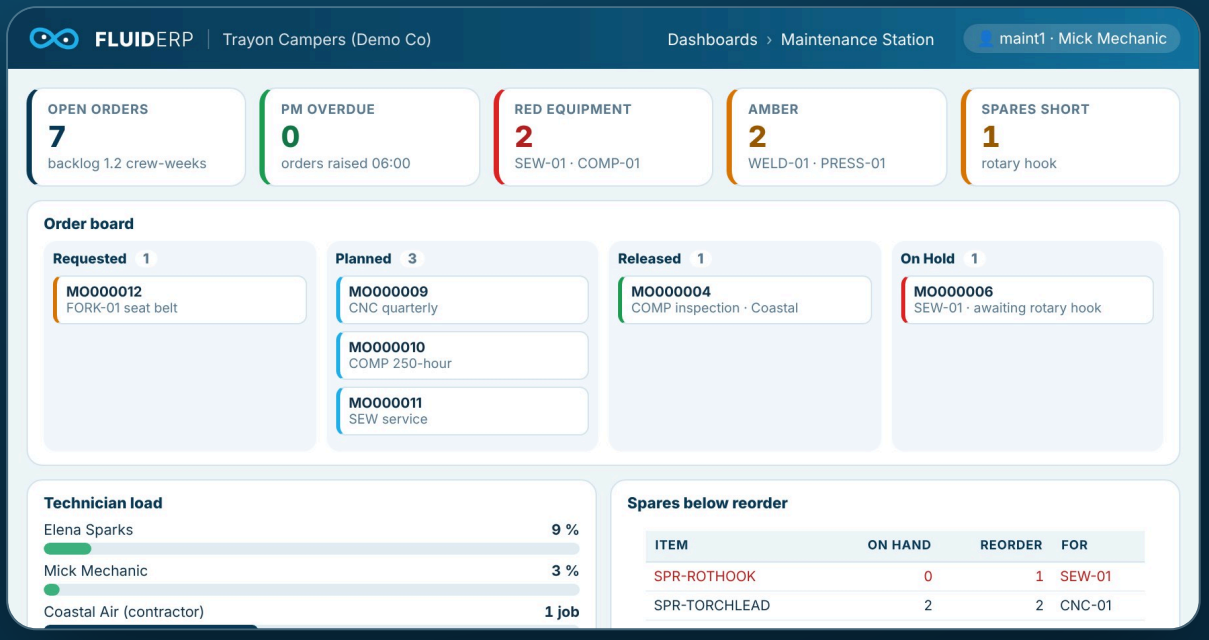
FLUID ERP · USER MANUAL

FLUID Plant Maintenance

Keep the line running.

Requests, work orders, checklists, spares, labour, downtime, preventive and condition-based plans, contractors and capitalisation — captured once and posted to stores, the ledger and fixed assets.

Requests & work orders · Preventive · meter · condition · Spares & stores · Downtime & availability · Contractor portal · Capitalise to fixed assets



Contents

1 About this manual 3

Who it is for 3

Conventions 3

2 How FLUID works: capture once, post everywhere 4

Three kinds of action 4

Rules you can rely on 5

3 Getting started 6

Signing in 6

The launcher 6

Anatomy of a screen 7

Everyday actions 8

4 Requests and work orders 9

Raise a maintenance request 9

Acknowledge the request 10

Create the work order 11

Release the order and load the checklist 12

5 Spares, hours, downtime and closing 13

Issue spares 13

Book hours, log downtime and close 14

6 Preventive and condition-based maintenance 16

Review the plans 16

Raise due PM orders 17

Record meter readings 18

7 Contractors 19

The contractor updates the job (portal) 19

Raise the contractor invoice 20

Approve and post the purchase invoice 21

8 Capitalising an improvement 22

Capitalise an order 22

9 Critical dashboards 23

Maintenance Station 23

Reliability 24

Schedule 25

Maintenance cost 26

10 Reports 27

Running and printing a report 27
Report catalogue 27
Job card 28
Failure Pareto 29
PM due and overdue 30
Work centre availability 31
11 Matching 32
How every matching screen works 32
Allocate the contractor's payment 33
Count the spares cage 34
12 Messages, refusals and what to do 35
13 Data reference and glossary 36
Tables you will meet 36
Glossary 36
A Walk it once on the demo package 38
For the administrator 38

1 About this manual

This manual explains how to run maintenance on FLUID Plant Maintenance: how an operator reports a problem, how the order is planned and worked, how spares, hours and downtime are recorded, how preventive work is raised, how contractors are used and paid, and how improvements are capitalised. The worked examples follow the Trayon Campers factory on 20 August 2026.

Who it is for

READER	READ THESE CHAPTERS
Operators	3, 4 (raising a request)
Technicians	3, 4, 5
Maintenance planners and supervisors	3 – 11
Contractors	7 (the contractor portal)
Accounts	5 (spares), 7 (contractor invoice), 8 (capitalise), 9, 11

Conventions

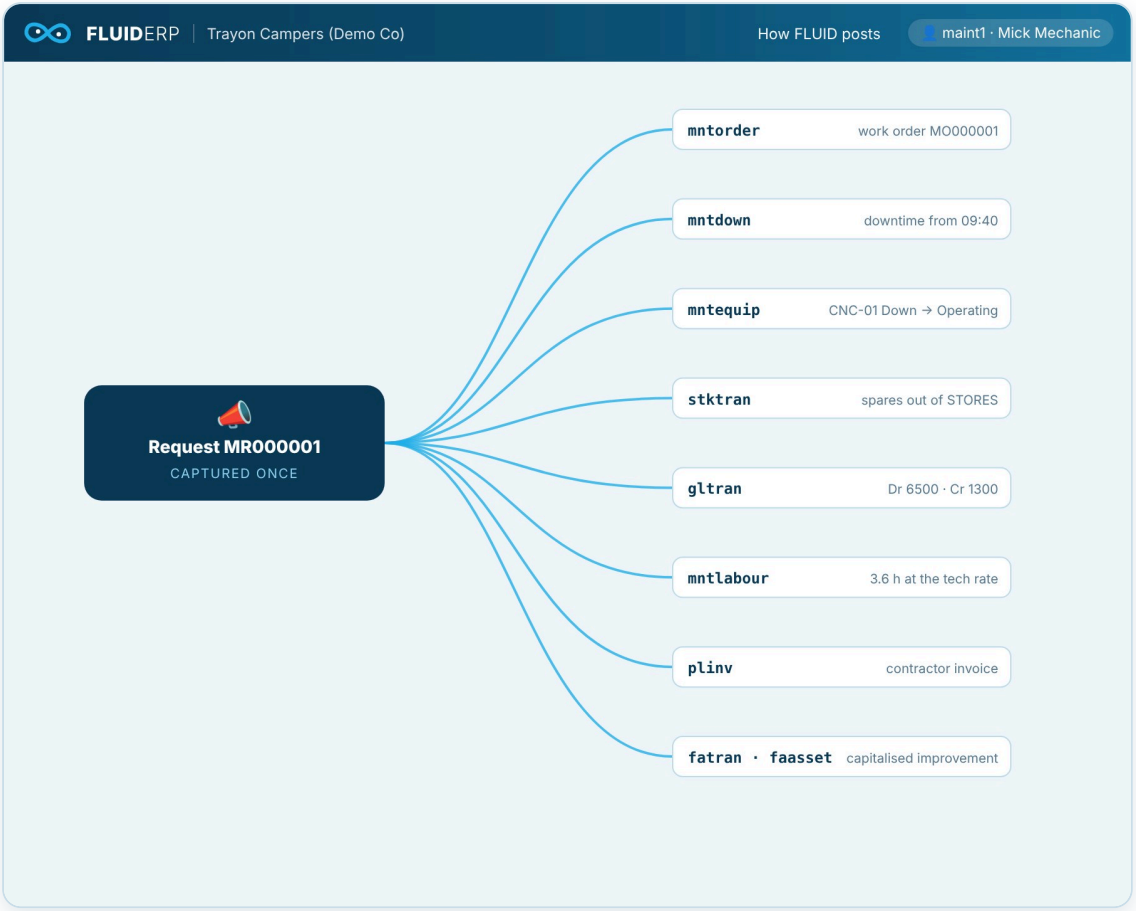
YOU SEE	MEANS
Bold names such as **Shifts**	A tile on the launcher, a screen, a tab or a field label, exactly as it appears in FLUID.
A button with an emoji, e.g. 🚚 **Post the Shift**	A posting button. Posting buttons are aqua; they write to other tables. 💾 **Save** is navy and saves only the record in front of you.
A field marked ƒ	A calculated field. You never type it — the calculation engine fills it from other fields or other tables, and keeps it right when they change.
`mneshift`	The name of the underlying table (collection). Useful when talking to your administrator or reading a report definition.
SH000006, MO000001 …	Record numbers. They are issued automatically by counters and carried on every row a button writes, so you can trace any figure back to the record that caused it.

ABOUT THE EXAMPLES

The equipment, plans, requests, orders and the main figures (MR000001, MO000001 $1,184, the 97 °C compressor reading, the $1,850 contractor quote, $683 capitalised) follow the Stage 56 Trayon story. Some invoice, payment and dashboard figures are illustrative. Screenshots come from the FLUID Plant Maintenance demonstration film.

2 How FLUID works: capture once, post everywhere

The operator reports the breakdown once. From that request the buttons make the order, open the downtime, set the machine down, issue the spares, cost the labour, pay the contractor and capitalise the improvement — each row carrying the order number.



One request fans out to the order, downtime, equipment status, stock, the GL, labour, the contractor's invoice and fixed assets.

Three kinds of action

ACTION	WHAT HAPPENS	TYPICAL BUTTON
Capture	A person (or a feed, a file, a phone app) records something that happened — once. It is saved on its own table and gets a number.	💾 Save · a form on the phone · a file import or feed
Post	A posting button reads the captured record and writes every row that follows from it — ledgers, stock, other documents, stamps — each carrying the source number. Most posting buttons are one-way: the record is marked posted and cannot be posted again.	🛠️ Create Work Order · 📦 Issue Spares · 🏗️ Capitalise

ACTION	WHAT HAPPENS	TYPICAL BUTTON
Derive	Calculations keep totals, balances, KPIs, flags and due dates right. Dashboards and reports read those rows directly; nothing is copied into a spreadsheet.	none — fields marked ƒ, dashboards, reports

Rules you can rely on

1. **Nothing is typed twice.** If a number exists in one record, every other table that needs it receives it from the button, not from a person.
2. **A posted record is sealed.** Pressing a posting button a second time writes nothing and says so. To change a posted figure you raise a correcting document (a credit note, a reversal, a correction) — which is itself captured once and posted.
3. **Every row is traceable.** Each row a button writes carries the reference of the record that caused it (for example `ref = MO000001`). Drill from any dashboard figure to the rows behind it and from there to the source record.
4. **Rules are enforced at the server.** Approvals, segregation of duties, embargoes, seals and signatures are checked by the gateway for every user, not just on screen.
5. **Imports are idempotent.** Files and feeds are keyed so that sending the same data again updates rows instead of duplicating them.

READ A POSTING PANEL

When you press a posting button, a panel lists every table it is writing to, one tick at a time, and ends with a green line such as “Posted — 5 writes to 5 tables from one capture”. That list is exactly what the button changed; the receiving tables show the same reference number.

3 Getting started

Signing in

Open the FLUID app (Windows, Android) or the browser client and sign in with the user name and password your administrator gave you. Each sign-in belongs to a group; the group decides which tiles you see and which fields and buttons you may use.

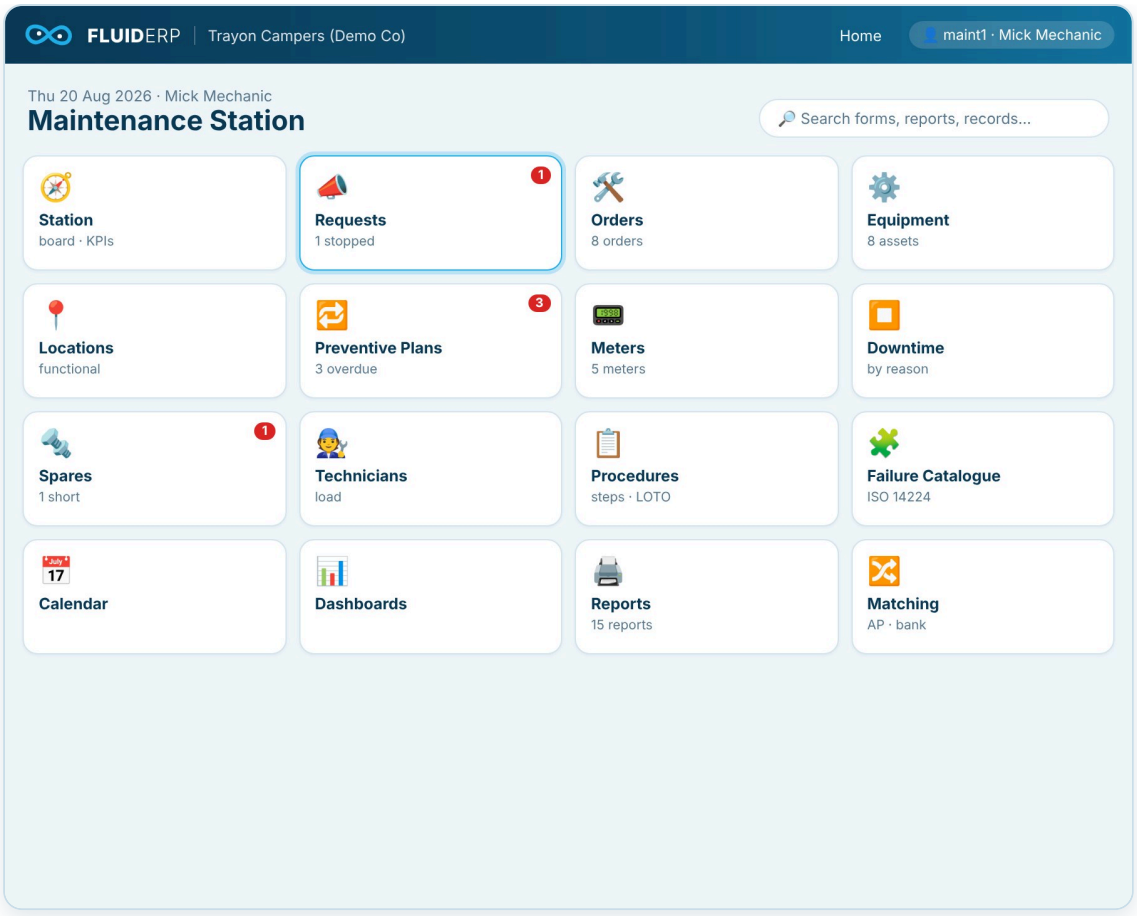
SIGN-IN	GROUP / DESK	USES FLUID TO ...
maint1	tm_maintenance (Mick Mechanic)	Requests, orders, plans, meters, spares, the station
maint2	tm_maintenance (Elena Sparks)	As above
prod1	tm_production (operator)	Raise requests, see equipment, downtime, readings and the board
acct1	tm_accounts	Approve and post contractor invoices; cost reports
service@coastalair.example.com	contractor (portal)	🛠️ Maintenance Jobs for Us — their own jobs only

DEMO SIGN-INS

Staff passwords are the same as the user name in the demo package; the contractor uses customer123. Change every password at go-live.

The launcher

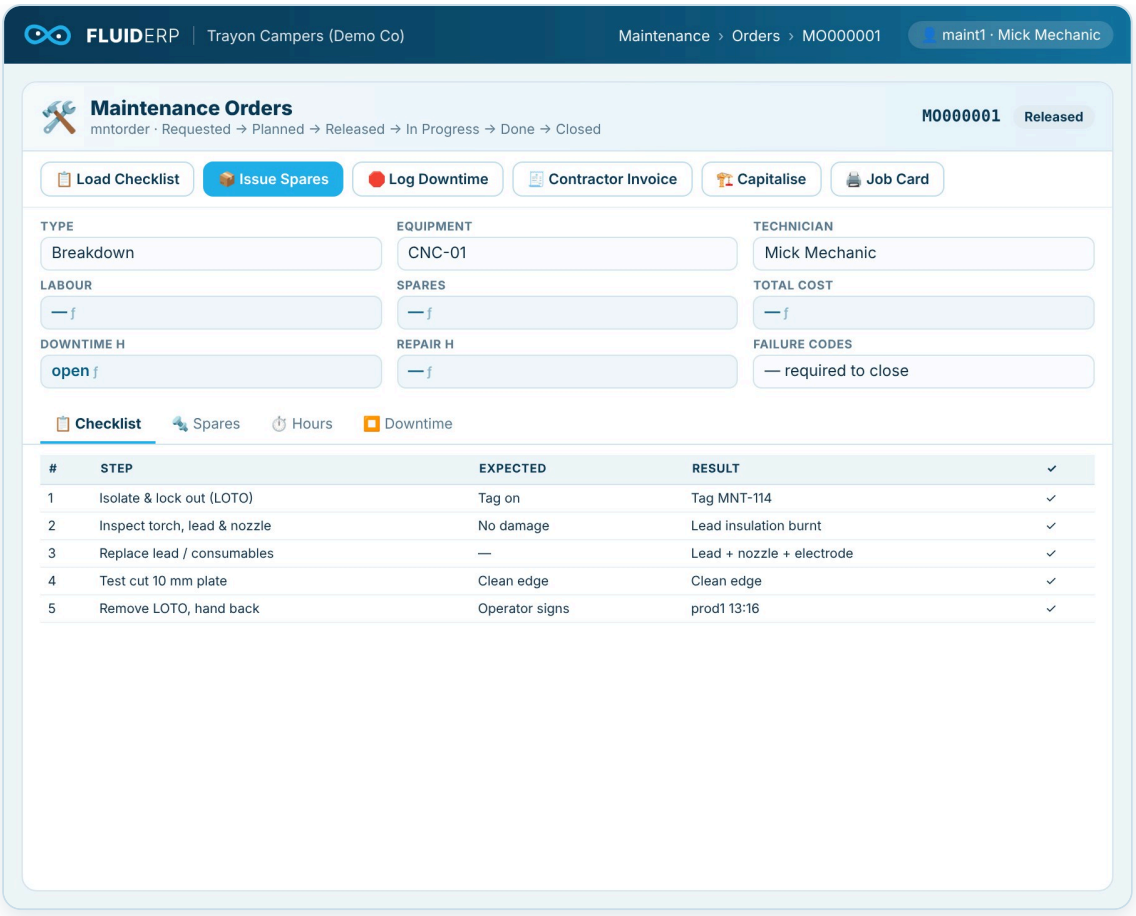
After sign-in you see the launcher: a grid of tiles for the screens, dashboards, reports and matching screens your group may open. A red badge on a tile counts records waiting for you. Use the search box to jump straight to a form, a report or a record number.



The Maintenance Station launcher: a stopped machine, overdue plans and a spare short are badged.

Anatomy of a screen

1. **Header** — the form name, the table it belongs to, the record number and a coloured **status** pill (for example Closed, Posted · sealed, Approved).
2. **Toolbar** — 💾 **Save** (navy) saves this record. Aqua buttons are **posting buttons**: they write to other tables. Outline buttons recalculate (🧮), print (🖨️) or open related screens.
3. **Fields** — typed fields are white; calculated fields (ƒ) are tinted and read-only. Red, amber and green fields are flags raised by a calculation (expired, due soon, OK).
4. **Tabs** — the record's lines and related records (lines, movements, attachments, signatures). Child rows are saved with the parent.
5. **Banners** — an amber or red banner explains why a record is locked, sealed, embargoed or refused. Read it before calling support; it names the rule.



A maintenance order: the toolbar carries the posting buttons; labour, spares, total cost, downtime and repair hours are calculated (ƒ).

Everyday actions

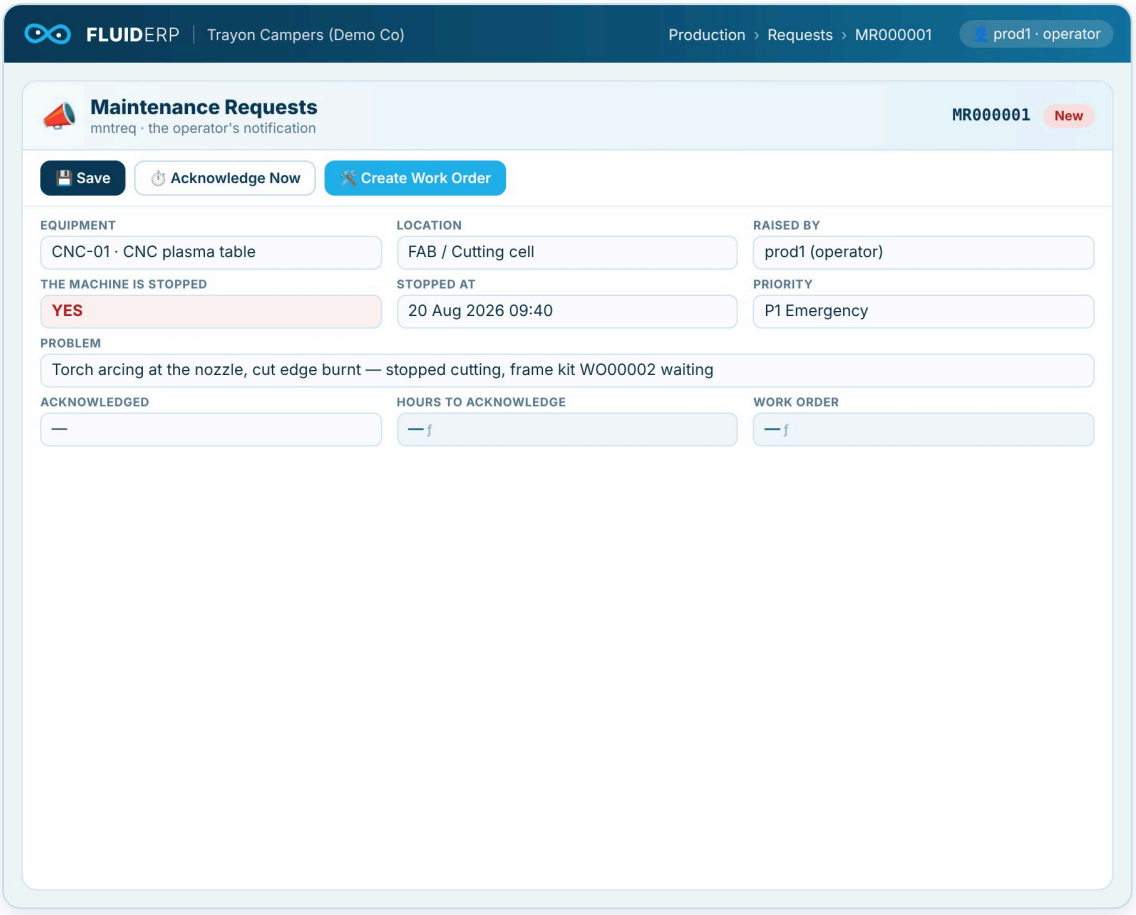
TO ...	DO THIS
Create a record	Open the tile, press ➕ **New**, fill the white fields, press 💾 **Save**. The number is issued on first save.
Find a record	Use the search box on the launcher or the filter row above any list. Record numbers, names and (where enabled) full text are searchable.
Refresh calculated fields	They refresh on save. 🧮 **Recalculate** re-runs them on demand (for example after a related record changed).
Print or save as PDF	🖨️ **Print** opens the printed copy in the document viewer; choose 🖨️ **Print / PDF** for paper or a PDF, or 💾 **Save** for the file.
Work offline (phone apps)	Forms marked as captures keep working with no signal; they wait in the **outbox** and send when you are back in range.

4 Requests and work orders

Anyone on the floor can raise a request. Maintenance acknowledges it and turns it into an order with one press.

Raise a maintenance request

Who **prod1 (operator)** | Where **Production › Requests** | Table **mntreq**



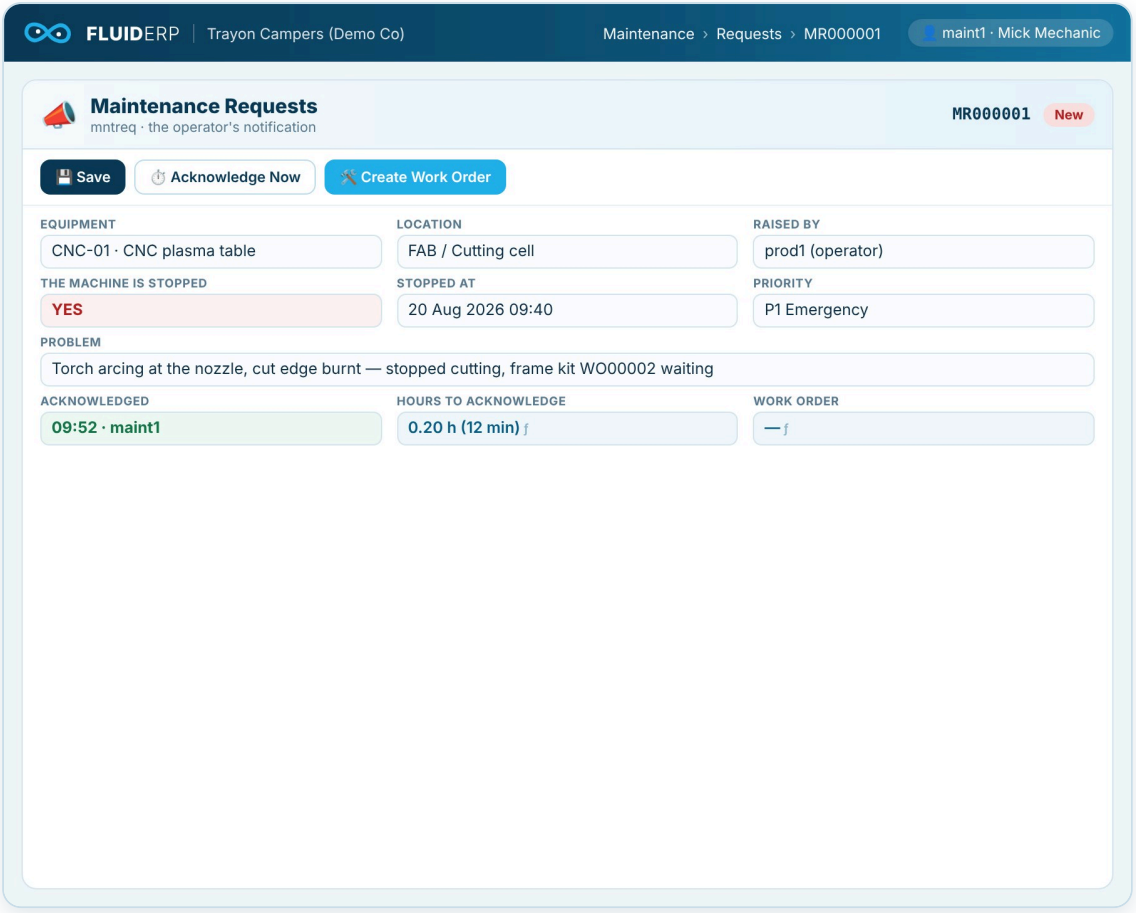
MR000001 — the plasma table stopped at 09:40; the machine-is-stopped switch is on.

1. Open 📣 **Requests** and press ➕ **New**.
2. Pick the **Equipment** (or scan its QR tag). Location comes from the equipment.
3. Describe the problem, add a photo, and switch on **The machine is stopped** if it is. Enter **Stopped at**.
4. Set the priority and press 💾 **Save**.

> **TIP**
> The stopped-at time is what downtime runs from. Nobody re-enters it later.

Acknowledge the request

Who **maint1** Where **Maintenance › Requests**

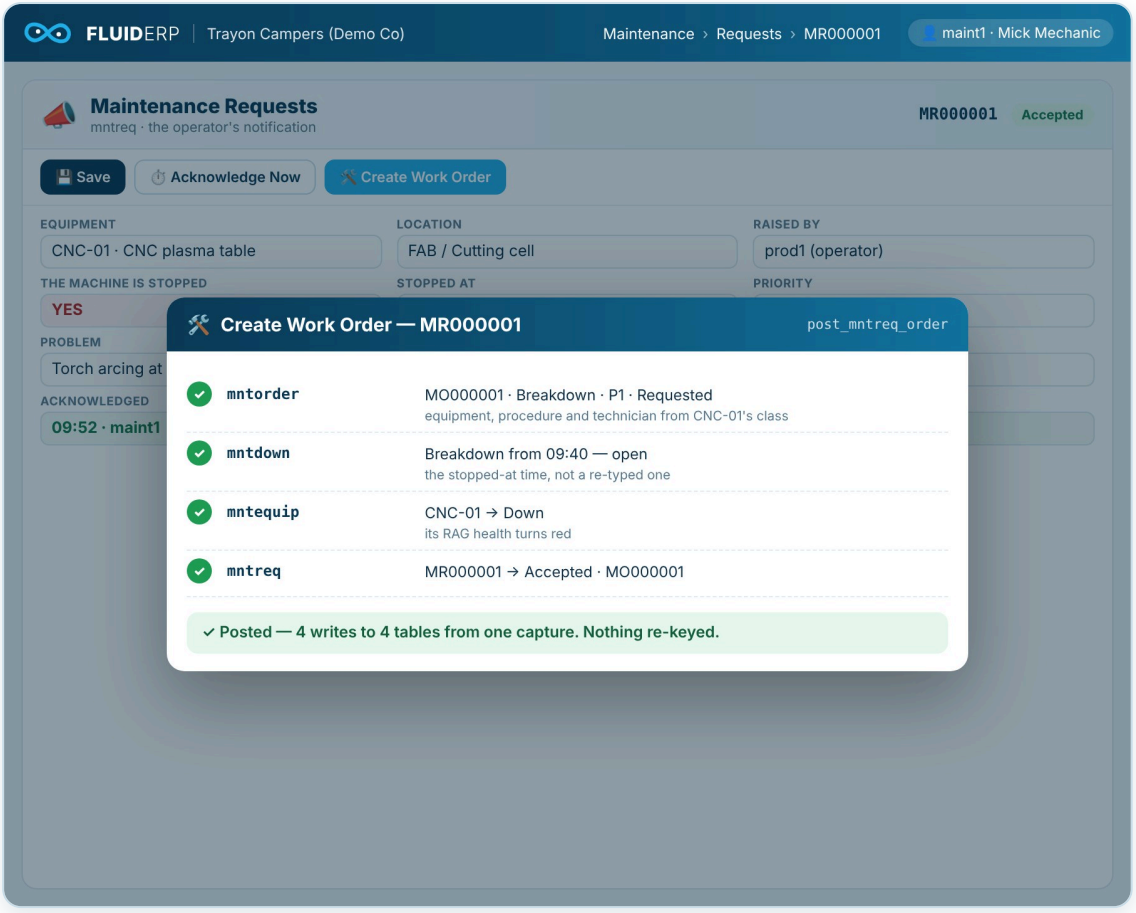


Acknowledged at 09:52 — hours-to-acknowledge 0.20 h (12 minutes) is calculated.

1. Open the new request and press ⏱️ **Acknowledge Now**.

Create the work order

Who **maint1** Where **Maintenance › Requests › MR000001**



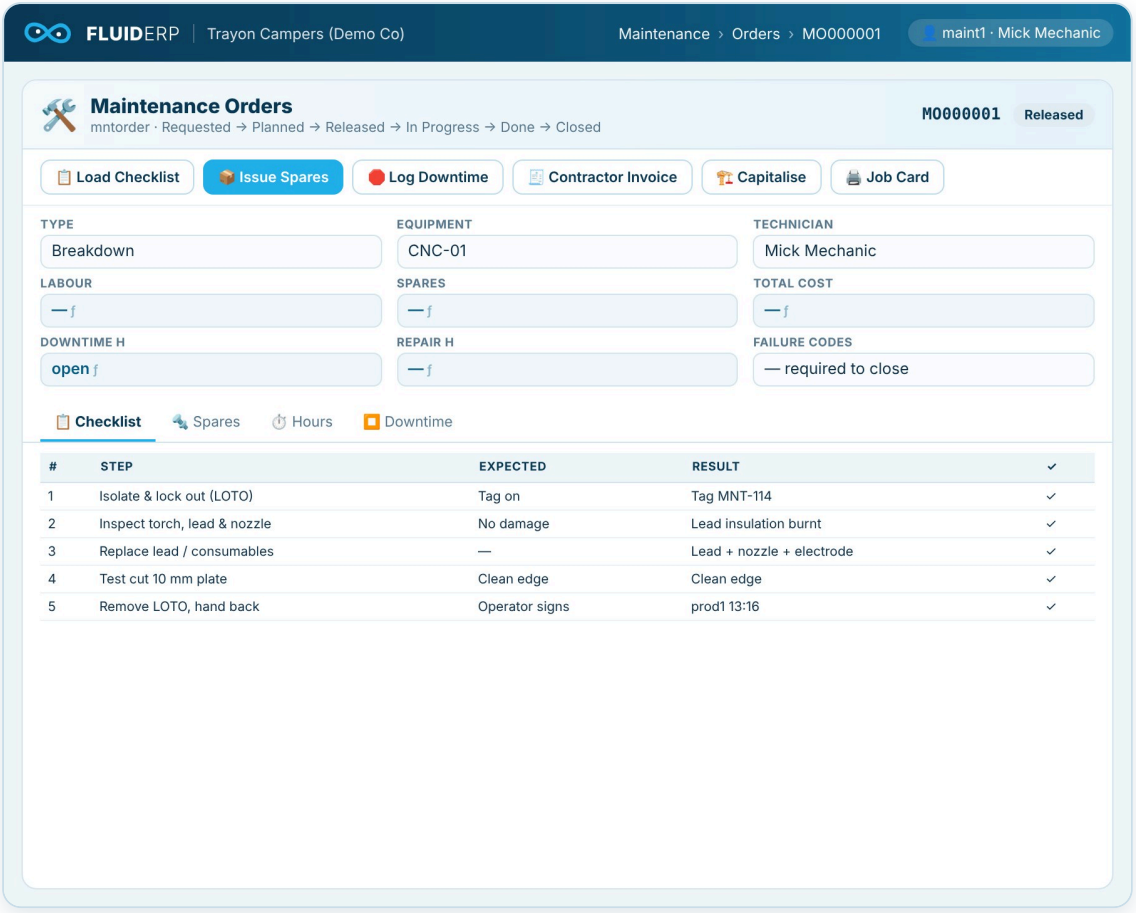
🛠️ Create Work Order — a Breakdown order, the open downtime, the machine Down and the request Accepted.

1 Press 🛠️ **Create Work Order**.

What 🛠️ Create Work Order writes	post_mntreq_order
✓ **mntorder**	MO000001 — Breakdown (because the machine is stopped), P1, procedure and technician from the equipment class
✓ **mntdown**	a Breakdown downtime row open from 09:40
✓ **mntequip**	CNC-01 → Down (its health turns red)
✓ **mntreq**	MR000001 → Accepted with the order number

Release the order and load the checklist

Who **maint1** | Where **Maintenance › Orders › MO0000001** | Table **mntorder · mntorderlines**



MO0000001 released to Mick; 📋 Load Checklist copied the procedure's five steps, lock-out first.

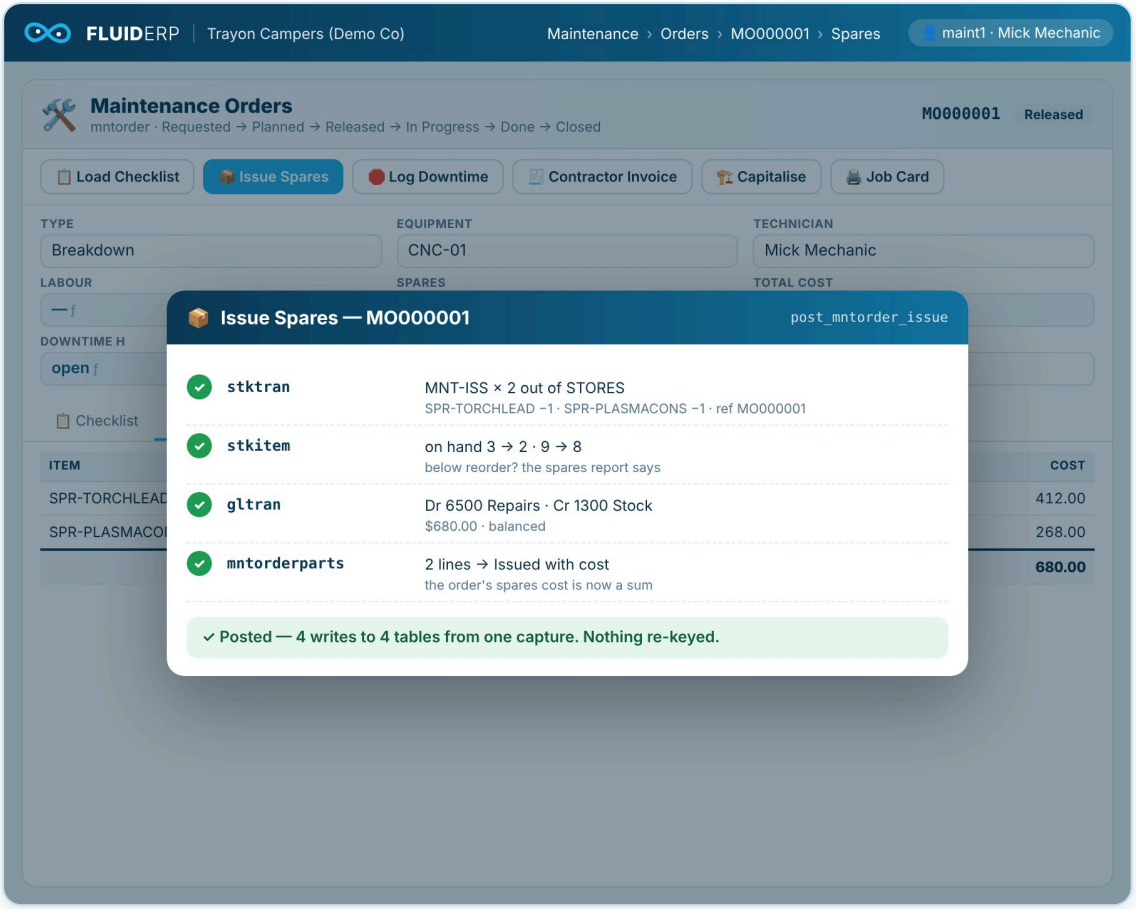
1. Set the technician and scheduled date; move the order to **Released**.
2. Press 📋 **Load Checklist**. The procedure's steps arrive with their expected results; photo-required steps need a photo to tick.
3. Work the steps, recording each result.

5 Spares, hours, downtime and closing

What the job used is recorded once, and costs itself.

Issue spares

Who **maint1** | Where **Maintenance › Orders › MO000001 › Spares** | Table **mntorderparts**



📦 Issue Spares — two lines out of STORES with the GL entry.

1. On 🔩 **Spares** add the parts (they are Reserved; a short line is flagged).
2. Press 📦 **Issue Spares**.

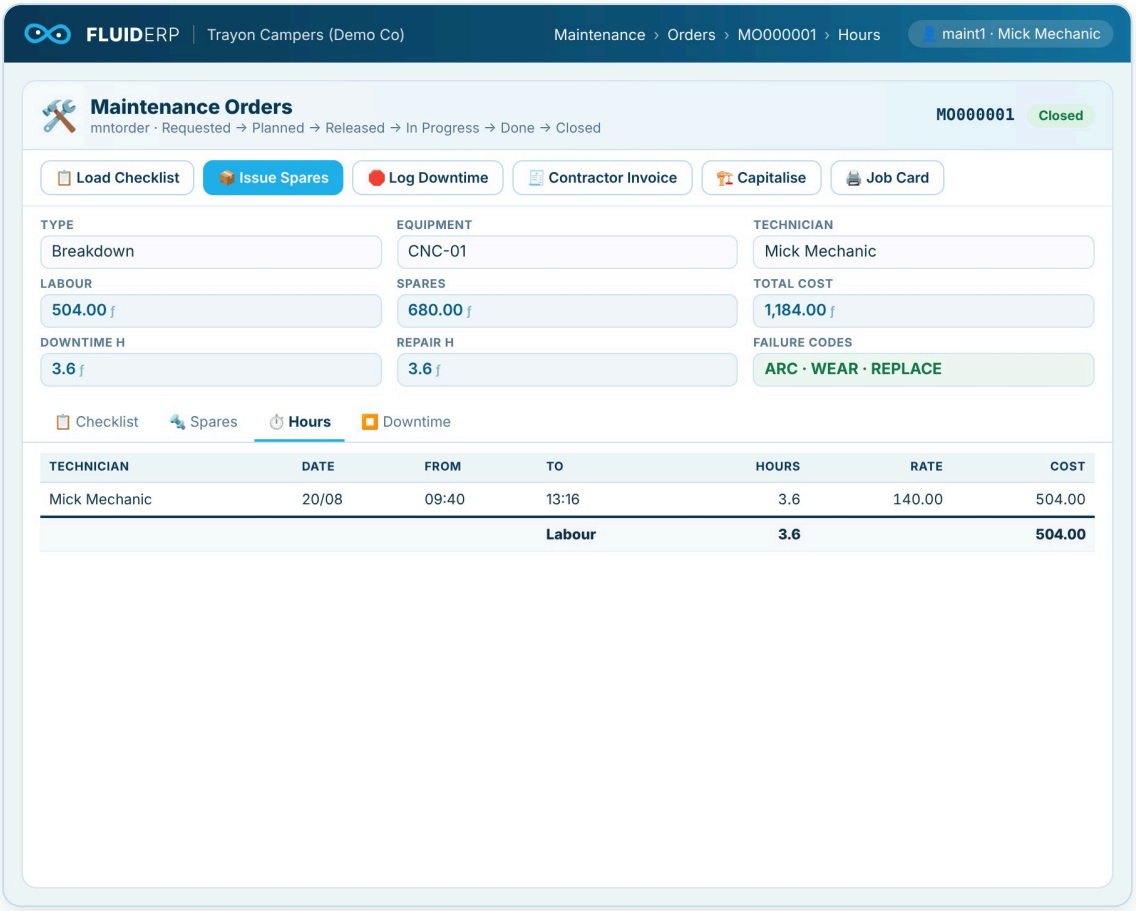
What 📦 Issue Spares writes	`post_mntorder_issue`
✓ **`stktran`**	MNT-ISS out of STORES per line, ref MO000001
✓ **`stkitem`**	on hand reduced (SPR-TORCHLEAD 3 → 2, SPR-PLASMACONS 9 → 8)
✓ **`gltran`**	Dr 6500 Repairs · Cr 1300 Stock, $680.00 at unit cost
✓ **`mntorderparts`**	lines → Issued with their cost

CHECK IT

Finance › Ledgers shows the stock and GL rows with ref MO000001; the order's Spares field reads 680.00.

Book hours, log downtime and close

Who **maint1** | Where **Maintenance › Orders › MO000001 › Hours**



3.6 h at Mick's rate = $504.00; downtime 09:40–13:16 logged; closed with ISO 14224 codes.

1. On ⏱️ **Hours** add the technician, date, from and to. The line costs itself at the technician's rate.
2. Press 🛑 **Log Downtime**: the downtime row closes at the order's finish and the equipment goes back to **Operating**.
3. Move the order to **Done** with findings, then **Closed** with the failure **mode**, **cause** and **activity**.

What 🛑 Log Downtime writes	`post_mntorder_downtime`
✓ **`mntdown`**	the window 09:40–13:16 as Breakdown (or Planned maintenance)
✓ **`mntequip`**	CNC-01 → Operating

If the system refuses

YOU SEE	WHAT IT MEANS / WHAT TO DO
Failure codes required to close	Pick mode, cause and activity from the failure catalogue.

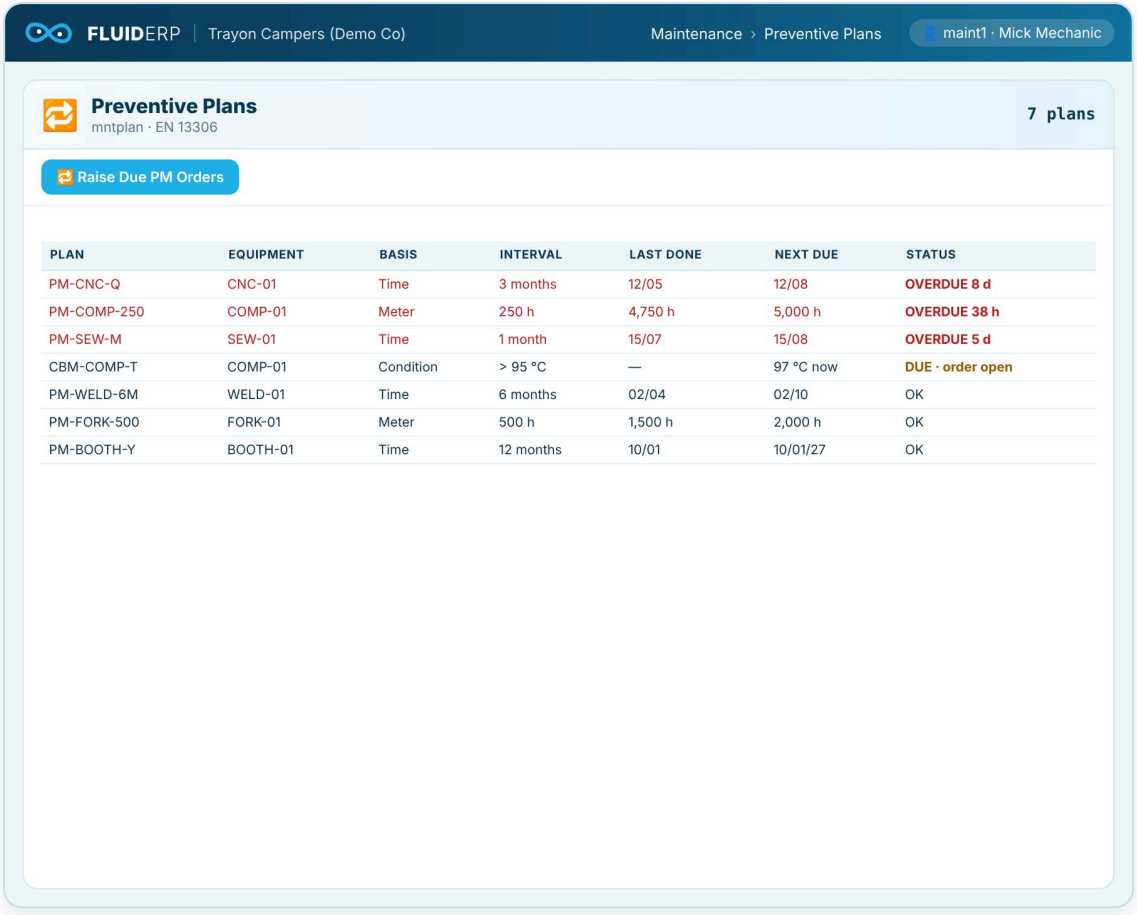
6 Preventive and condition-based maintenance

Plans are time-based, meter-based or condition-based. Their due status is calculated; the orders are raised by a button or every morning at 06:00.

Review the plans

Who **maint1** | Where **Maintenance › Preventive Plans** | Table **mntplan**

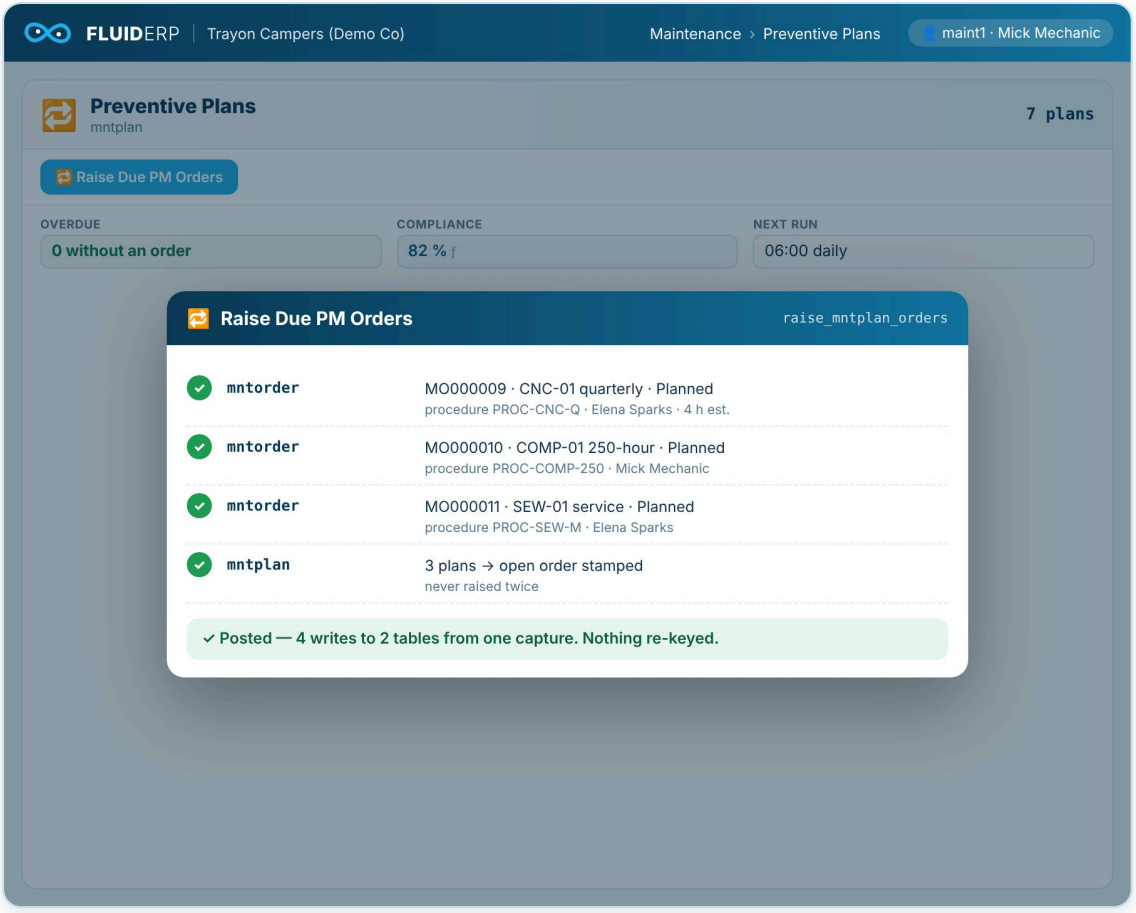
Each plan shows last done, next due (date or meter), whether it is due or overdue, its open order and compliance %.



Seven plans; three overdue (CNC quarterly, compressor 250-hour, sewing service) and the condition plan due.

Raise due PM orders

Who **maint1 (or the 06:00 posting service)** Where **Maintenance › Preventive Plans**



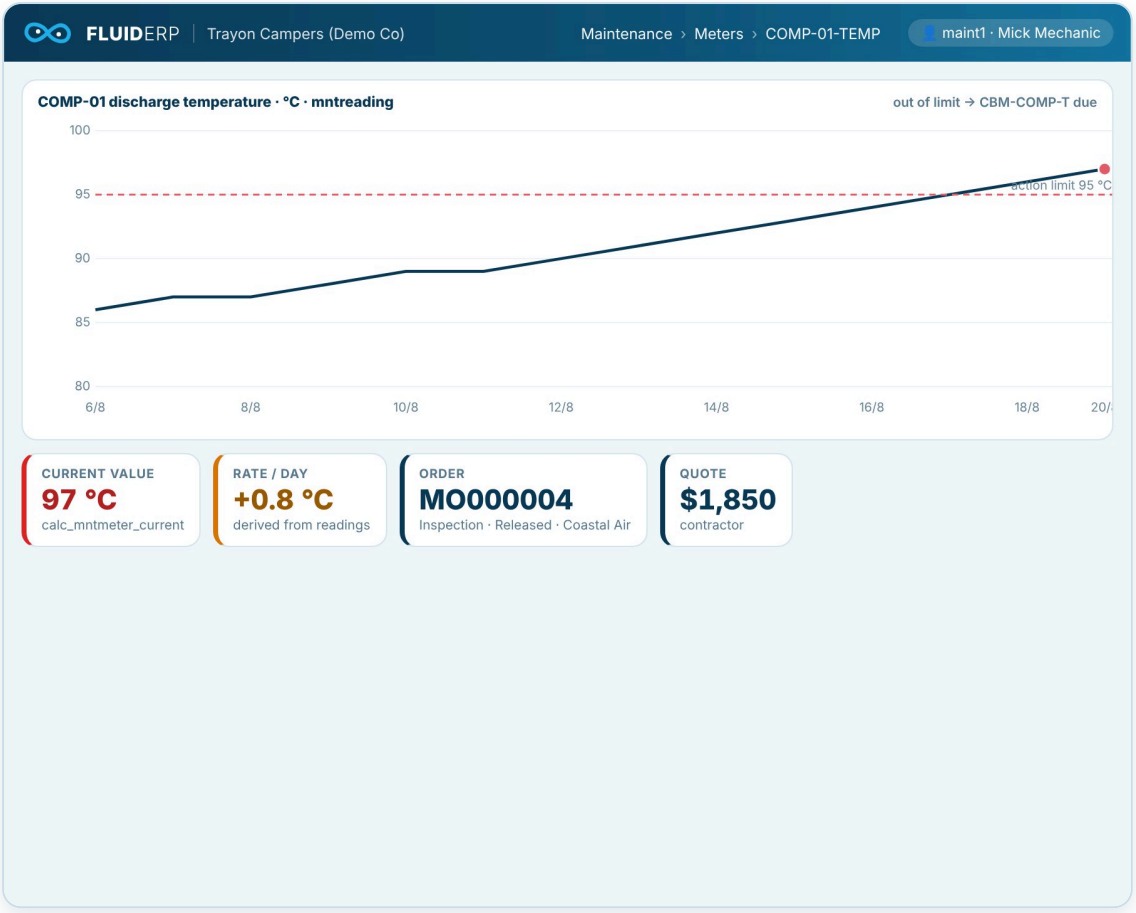
🔁 Raise Due PM Orders — one Planned order per due plan without an open order.

1. Press 🔁 **Raise Due PM Orders** (the posting service also runs it daily at 06:00).

What 🔁 Raise Due PM Orders writes	raise_mntplan_orders
✓ **mntorder**	MO0000009 CNC quarterly, MO0000010 compressor 250-hour, MO0000011 sewing service — Planned, with procedure, technician and estimate
✓ **mntplan**	each plan stamped with its open order — never raised twice

Record meter readings

Who **operators, technicians** | Where **Maintenance › Meters** | Table **mntmeter · mntreading**



COMP-01 discharge temperature reached 97 °C against a 95 °C limit — the condition plan fell due.

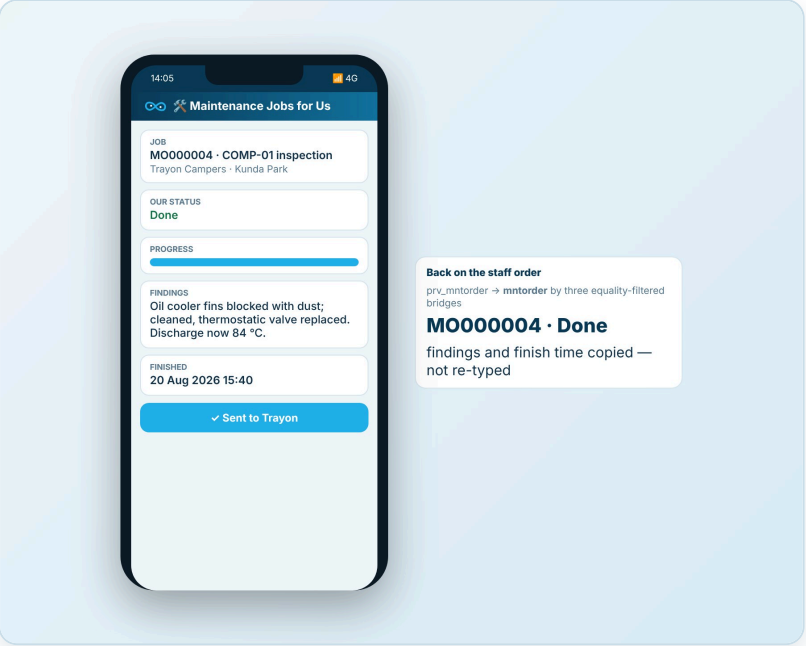
1. Open the meter and add a reading (value, date, source) — or let a feed or another posting write it.
2. Current value, rate per day and out-of-limit are calculated; condition plans fall due by themselves.

7 Contractors

Contractors see only their own jobs on the portal. What they type comes back to the staff order; their invoice is made from the order.

The contractor updates the job (portal)

Who **contractor** | Where **Portal › 🛠️ Maintenance Jobs for Us** | Table **prv_mntorder → mntorder**

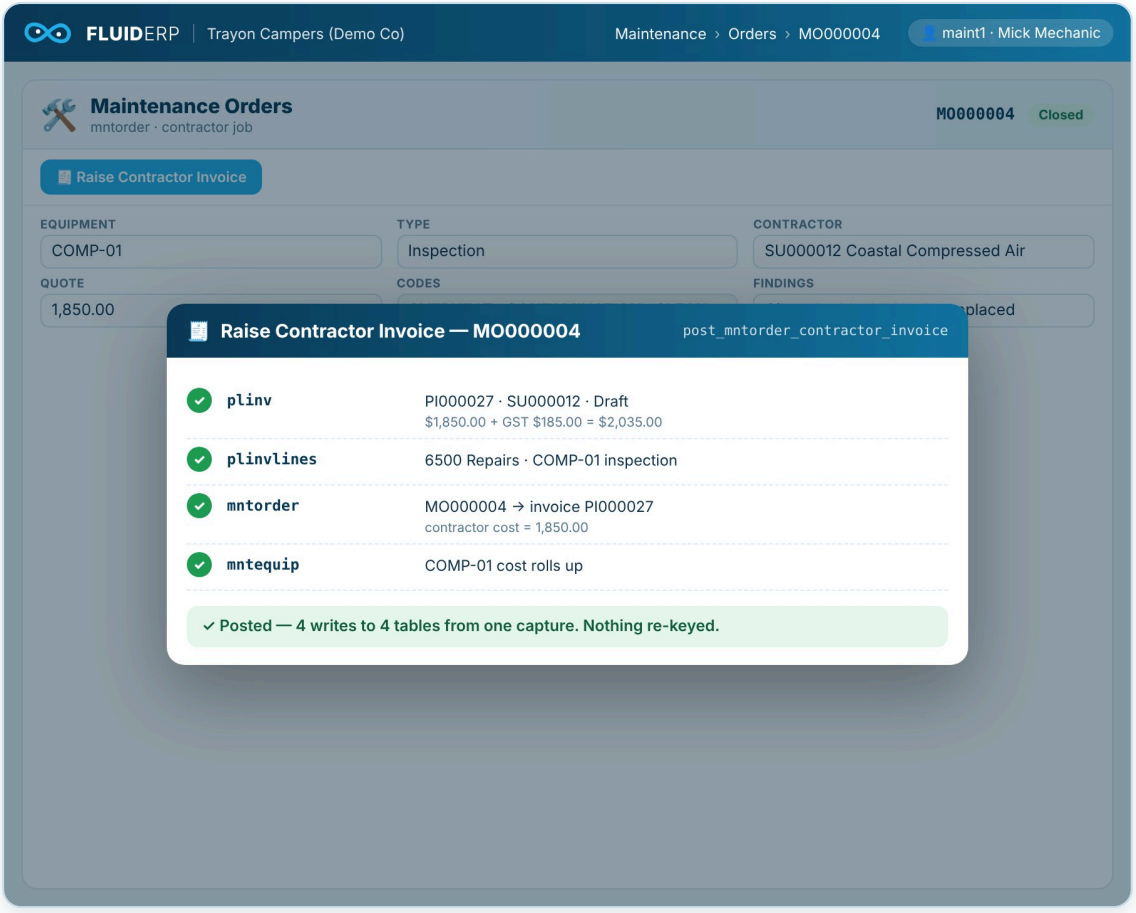


Coastal Air moves MO000004 Accepted → In Progress → Done with findings and the finish time.

1. Sign in to the portal; open the job.
2. Set **Our status** (Accepted, In Progress, Done), the progress, findings and finish time.
3. Press **Update**. Bridges copy it to the staff order — no re-typing.

Raise the contractor invoice

Who **maint1** Where **Maintenance › Orders › MO000004**



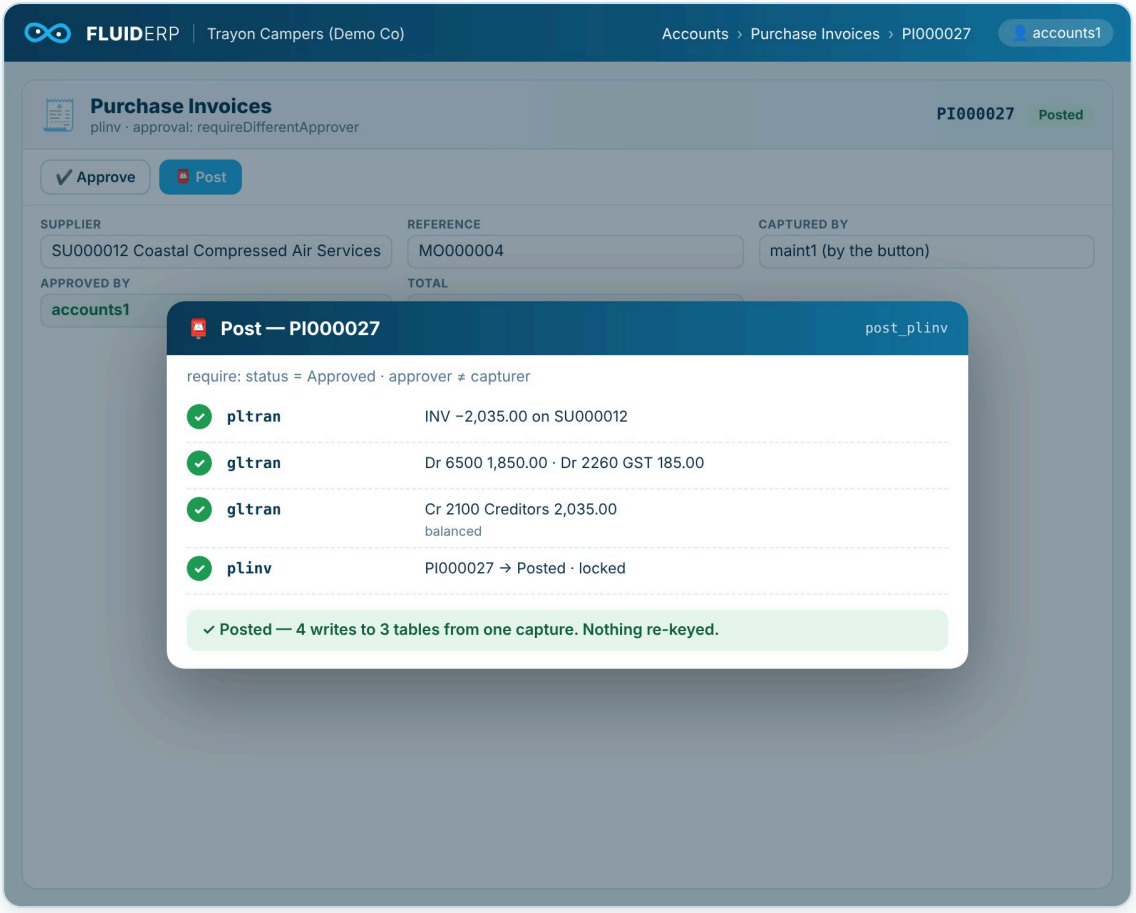
🧾 Raise Contractor Invoice — a purchase invoice for the quote, against repairs, with GST.

1. Close the order with its failure codes.
2. Press 🧾 **Raise Contractor Invoice**.

What 🧾 Raise Contractor Invoice writes	post_mntorder_contractor_invoice
✓ **plinv**	PI000027 to SU000012 Coastal Compressed Air — $1,850.00 + GST $185.00, Draft
✓ **plinvlines**	6500 Repairs · COMP-01 inspection
✓ **mntorder**	invoice number and contractor cost stamped
✓ **mntequip**	the equipment's cost rolls up

Approve and post the purchase invoice

Who **acct1** | Where **Accounts › Purchase Invoices › PI000027**



A different person approves; posting writes the supplier ledger and the GL.

1. Open the invoice and press ✔️ **Approve** (you cannot approve an invoice you captured).
2. Press 📮 **Post**.

What 📮 Post writes	post_plinv
✓ **pltran**	INV −2,035.00 on SU000012
✓ **gltran**	Dr 6500 1,850.00 · Dr 2260 GST 185.00 · Cr 2100 Creditors 2,035.00
✓ **plinv**	PI000027 → Posted · locked

If the system refuses

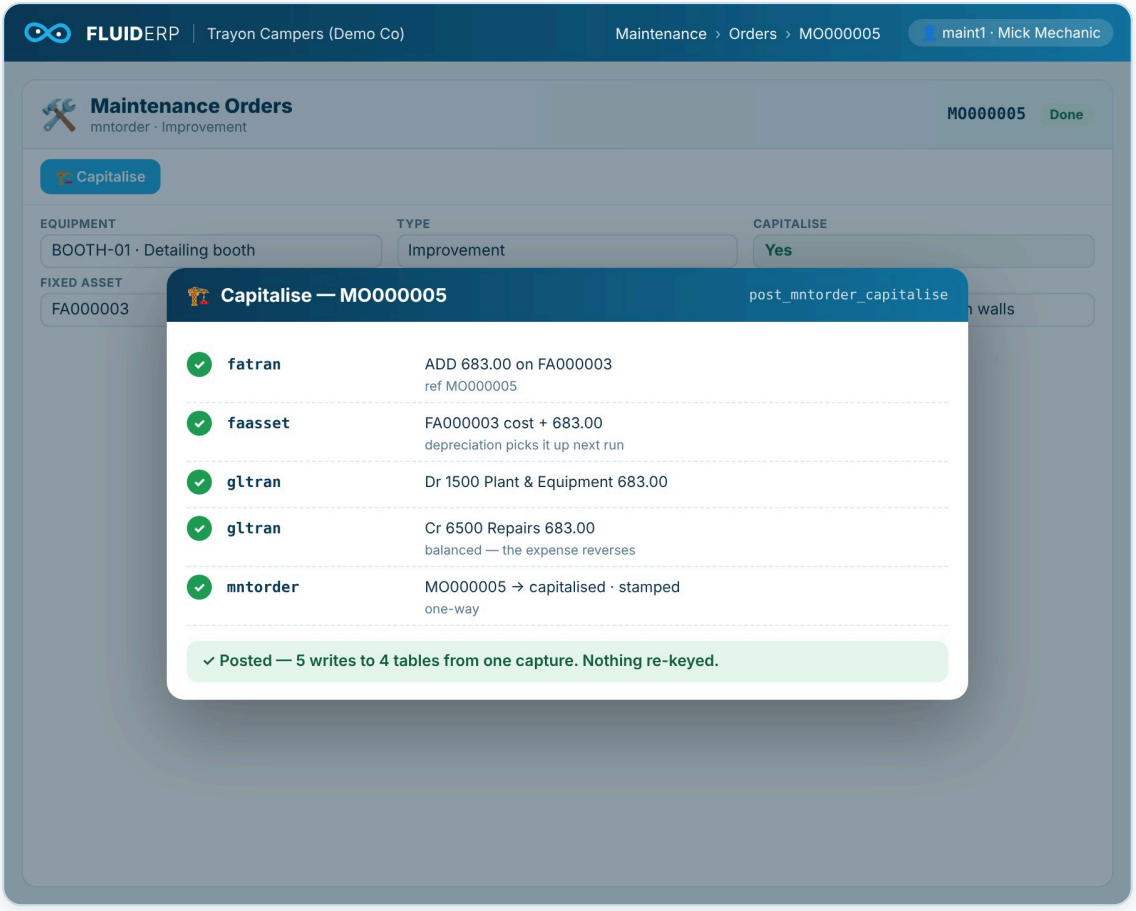
YOU SEE	WHAT IT MEANS / WHAT TO DO
Status must be Approved; approver must differ from the capturer	Every posting over purchase invoices requires a separate approver.

8 Capitalising an improvement

An order flagged as an improvement can move its cost from expense onto a fixed asset.

Capitalise an order

Who **maint1 / acct1** | Where **Maintenance › Orders › MO000005**



🏗️ Capitalise — the LED upgrade's $683 onto FA000003.

1. On a **Done** or **Closed** order with **Capitalise** set and a fixed asset chosen, press 🏗️ **Capitalise**.

What 🏗️ Capitalise writes	`post_mntorder_capitalise`
✓ **`fatran`**	ADD 683.00 on FA000003, ref MO000005
✓ **`faasset`**	cost + 683.00 — the next depreciation run picks it up
✓ **`gltran`**	Dr 1500 Plant & Equipment · Cr 6500 Repairs, 683.00
✓ **`mntorder`**	stamped capitalised (one-way)

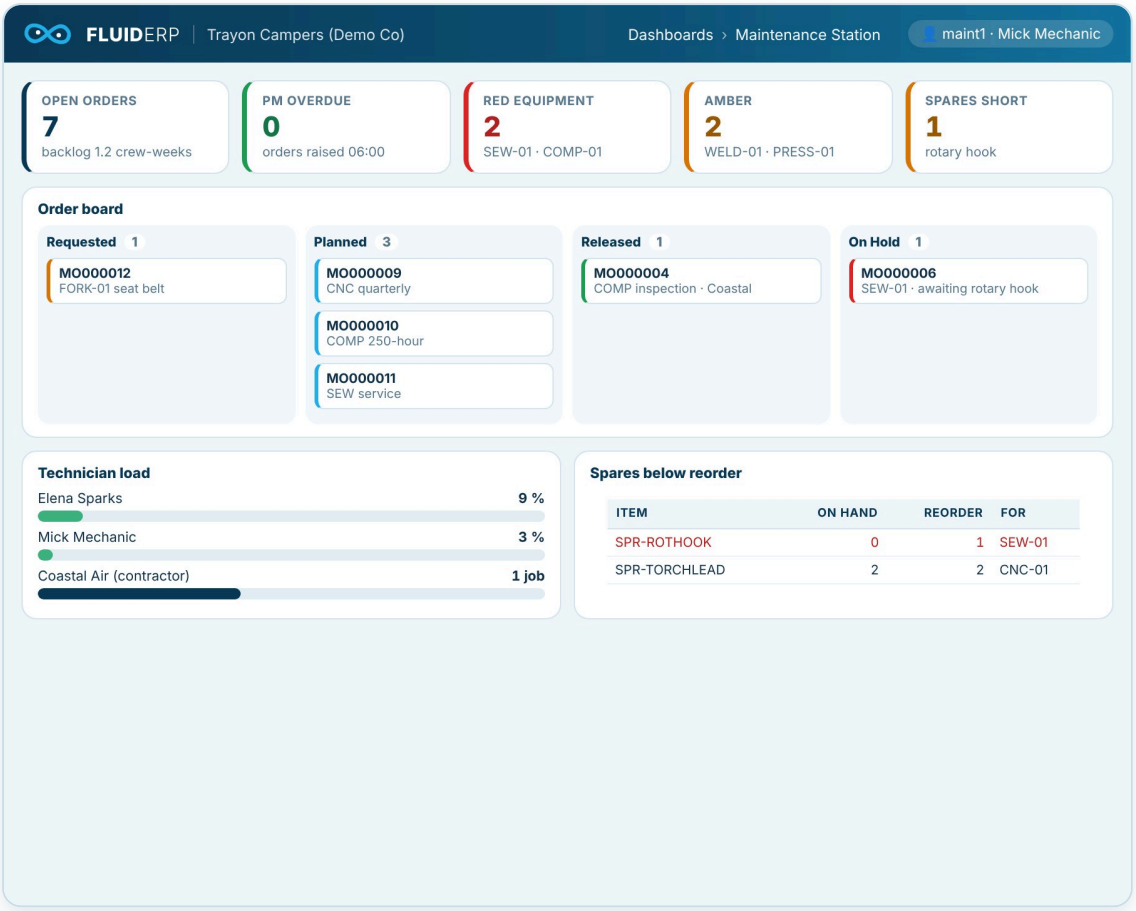
9 Critical dashboards

The station and three dashboards cover the work, reliability, the schedule and cost. Every KPI is a calculation over orders, downtime, readings and the ledger.

Maintenance Station

Where **Dashboards › Maintenance Station** Who **maint1**

The board by status, PMs due and overdue, red and amber equipment, spares short and technician load.



Maintenance Station

What each tile is and where it comes from

Open orders / backlog
Orders not closed, in crew-weeks.
`calc_mnttech_load`

PM overdue
Due plans with no open order.
`calc_mntplan_due`

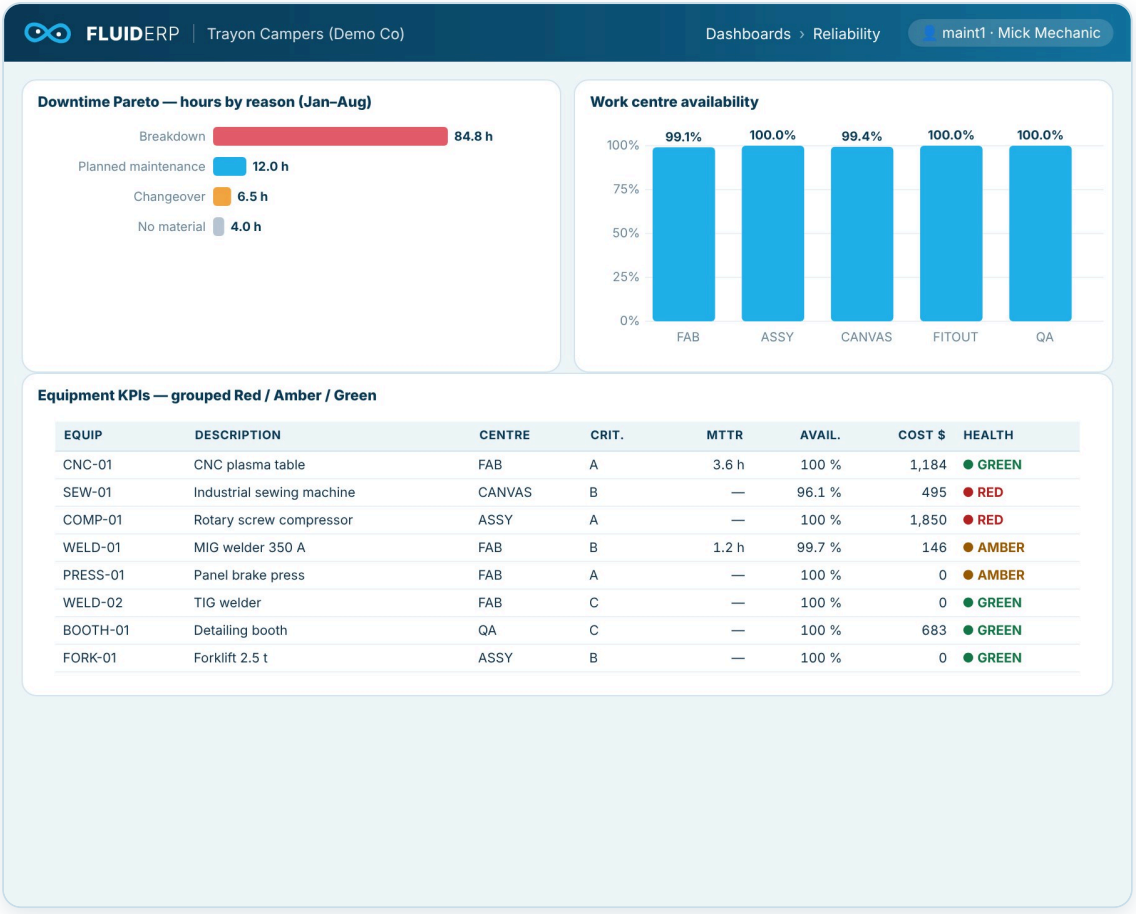
Red / Amber equipment
RAG health from open breakdowns, overdue PM, spares short.
`calc_mntequip_kpis`

Spares short
On hand below reorder for equipment spares.
`calc_mntspare_stock`

Reliability

Where **Dashboards › Reliability** Who **maint1, production**

MTBF, MTTR, availability and the downtime Pareto.



Reliability

What each tile is and where it comes from

MTBF

Operating hours ÷ breakdowns.

`planned h/week × weeks − downtime ÷ breakdowns`

MTTR

Mean repair hours of closed breakdown orders.

`CNC-01: 3.6 h`

Availability

Operating ÷ (operating + unplanned downtime).

`CANVAS centre 99.4 %`

Downtime Pareto

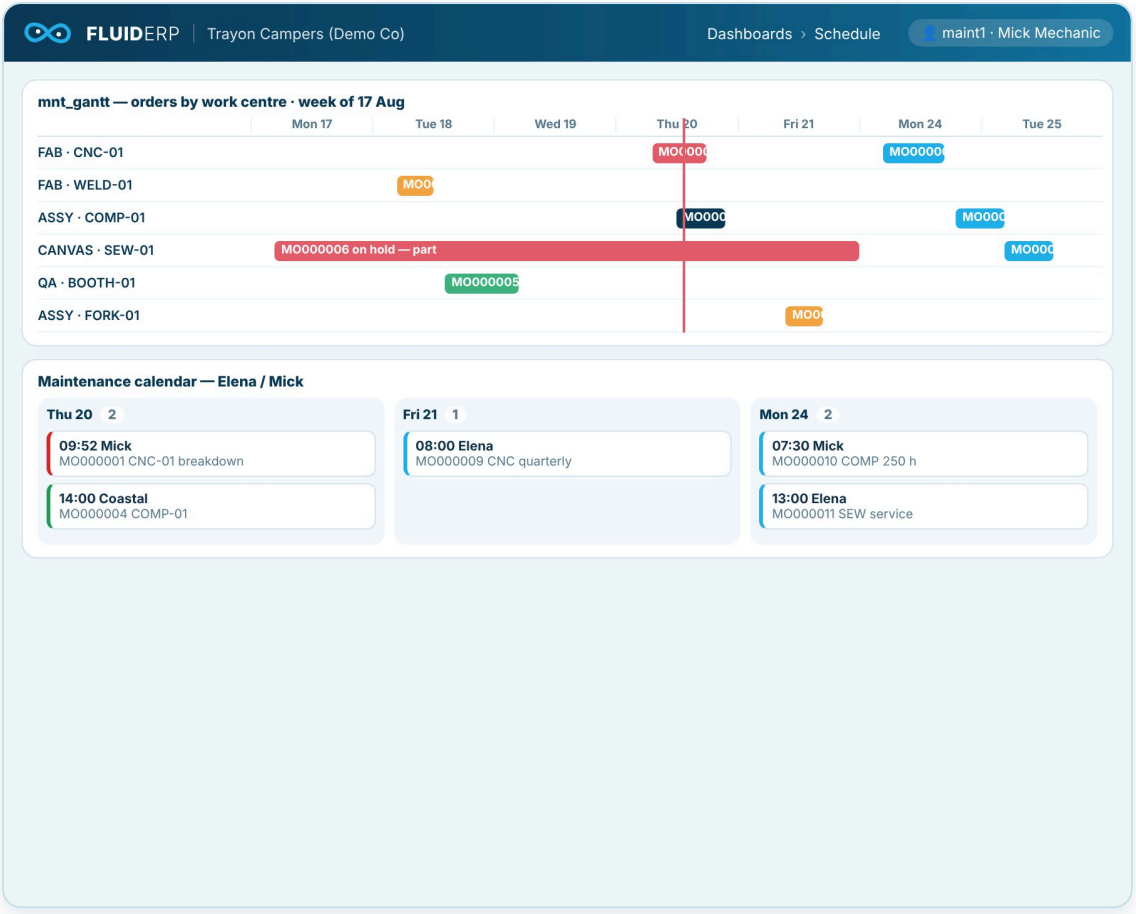
Hours by reason (Nakajima's losses).

`Σ mntdown.hours by reason`

Schedule

Where **Dashboards › Schedule** | Who **maint1, production planner**

The maintenance Gantt by work centre (breakdowns red) and the technicians' calendar.



Schedule

What each tile is and where it comes from

Gantt
Orders on each centre over the week.
`mnt_gantt`

Calendar
Orders by technician and day.
`mnt_calendar · 📅 Put on Calendar`

Maintenance cost

Where **Dashboards › Maintenance Cost** | Who **acct1, maint1**

Labour, spares and contractors by machine and month; expensed against capitalised.



Maintenance cost

What each tile is and where it comes from

Cost by equipment
Labour + spares + contractor per machine.
`calc_mntorder_costs`

Cost by kind
Labour, spares, contractor.
`Σ by kind`

Capitalised
Moved to fixed assets.
`fatran ADD from 🏗`

10 Reports

Fifteen reports ship with the module, including board, Gantt and calendar shapes.

Running and printing a report

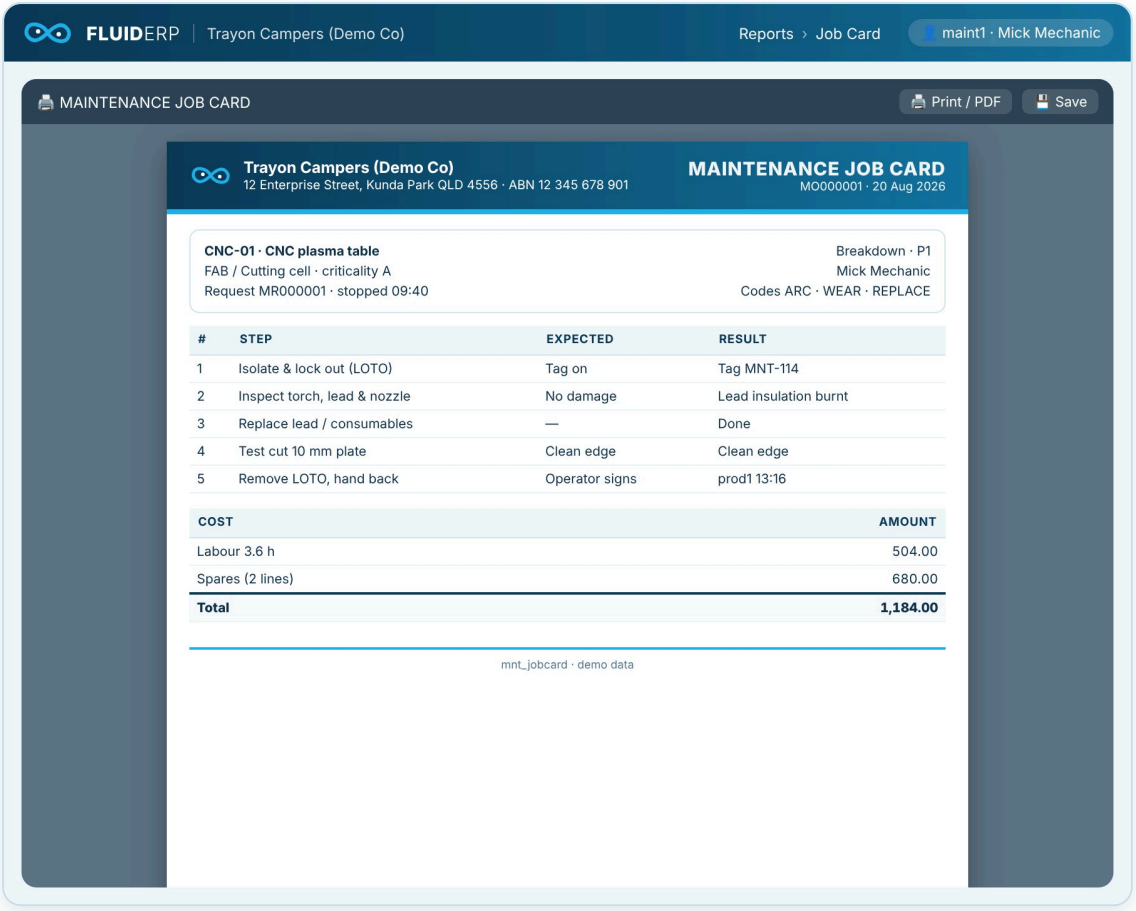
1. Open 🖨️ **Reports** on the launcher (or the report's own tile) and pick the report.
2. Fill the parameters. Required parameters are marked; dates accept `$today`-style defaults and lookups show “value — description”.
3. Press ▶ **Run**. The report draws its shape — grid with subtotals, chart, pivot, calendar, Gantt, board or tree — from the live rows.
4. Press 🖨️ **Print** to open the printed copy (letterhead, document header, footer) in the document viewer, then 🖨️ **Print / PDF** or 💾 **Save**. Reports with a file layout also offer 💾 **Export File**.

Report catalogue

REPORT	SHOWS	TYPICAL READER
Maintenance Board / Gantt / Calendar	Orders by status, centre and technician	planners
PM Due	Due and overdue plans with open orders	planners
Backlog	Open orders and crew-weeks	supervisors
Equipment KPIs	MTBF, MTTR, availability, cost, RAG	managers
Downtime Pareto / Failure Pareto	Hours by reason; orders by ISO 14224 mode	reliability
Cost by Equipment	Labour, spares, contractor	accounts
Spares Short	Below reorder	stores
Meter Readings	Readings and out-of-limit	technicians
Requests	Requests and response times	supervisors
Contractor Orders	Jobs by contractor and status	planners, accounts
Job Card	Printed order with checklist and costs	technicians
Centre Availability	Production's view of maintenance	production

Job card

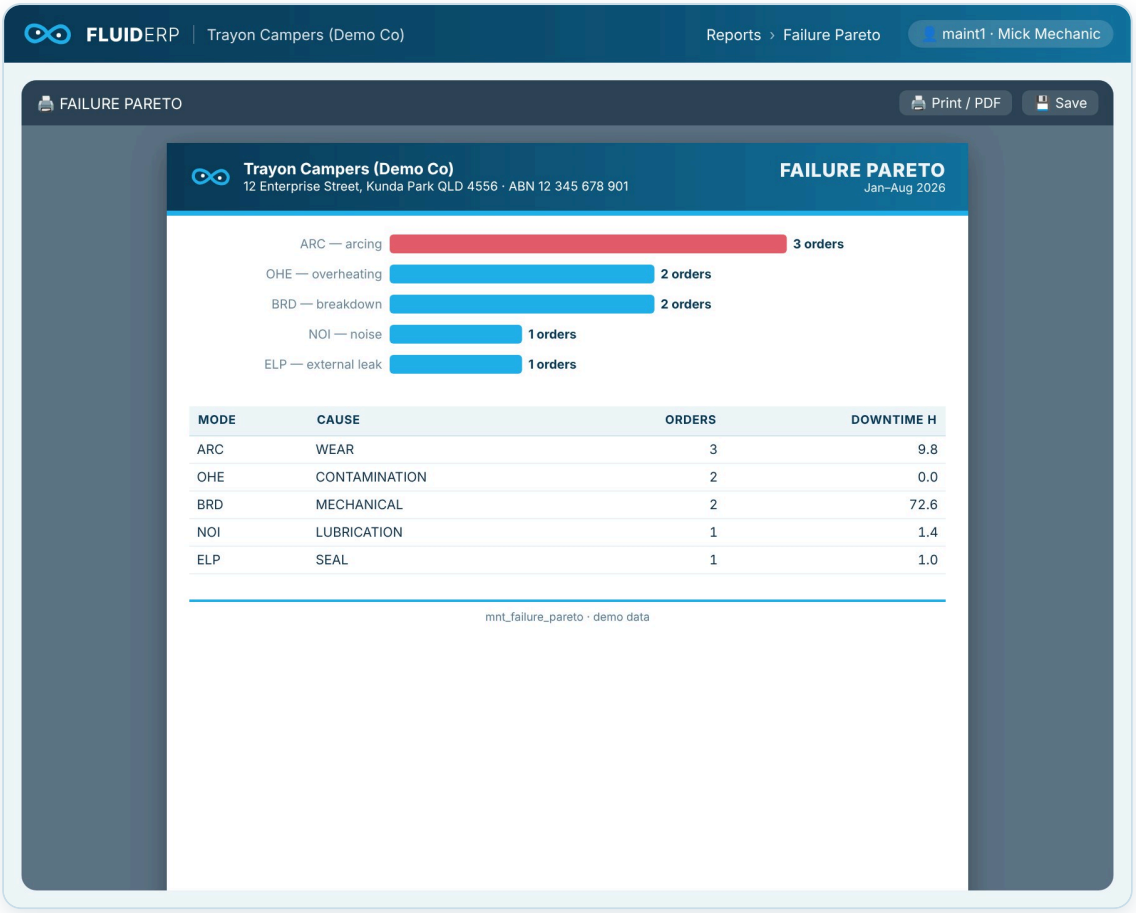
The order's document header, the checklist with results, spares and hours — the record the technician signs.



Job card

Failure Pareto

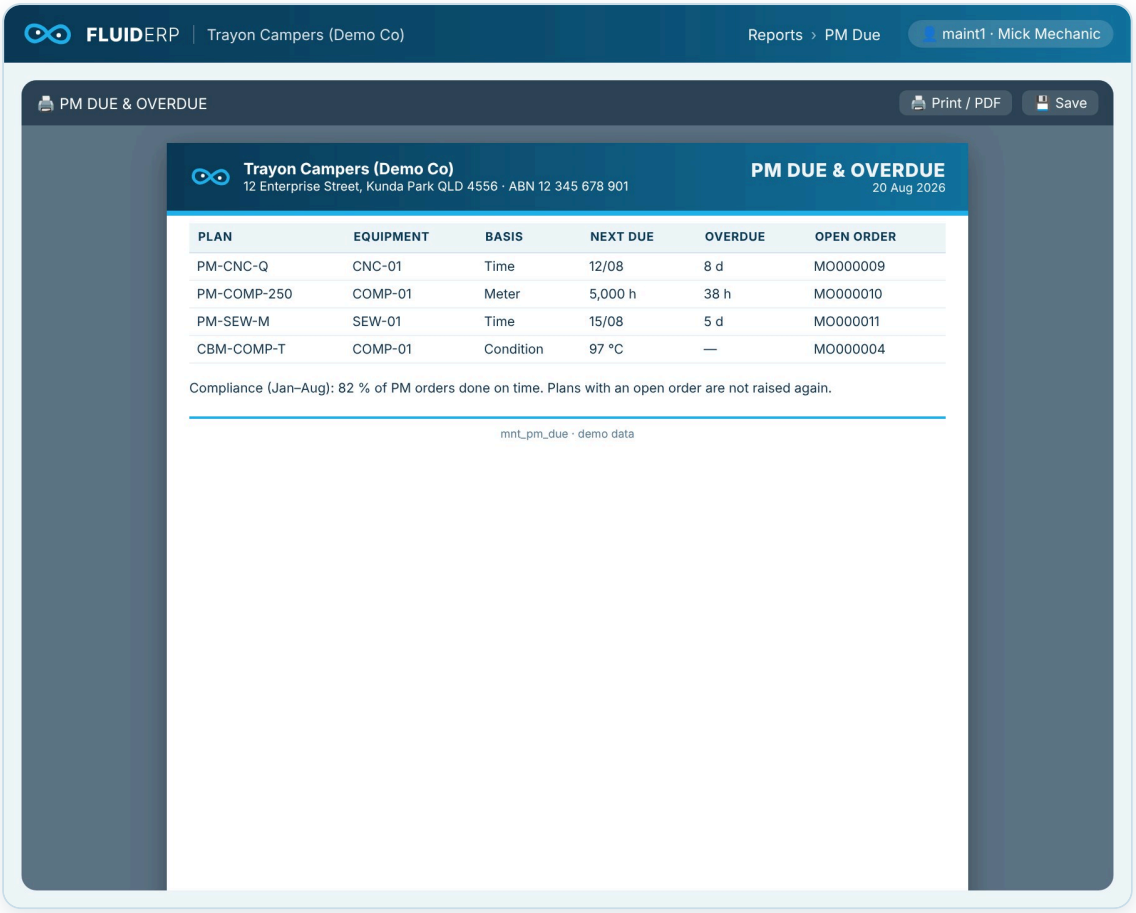
Because every order closes with codes, the failure Pareto prints itself and points at what to fix first.



Failure Pareto

PM due and overdue

The same rows as the plans screen, for the morning meeting.



PM due and overdue

Work centre availability

Each work centre gains maintenance downtime, breakdown hours, availability and cost, so production reads maintenance in its own report.



Work centre availability

11 Matching

A matching screen sets two lists side by side — for example bank statement lines and cashbook entries — and pairs the rows that belong together. Matching is a definition (`tblmatchdef`): it names the two sides, the fields to compare, the tolerances and what a confirmed match writes.

How every matching screen works

1. Open the matching screen from its 🔀 tile on the launcher, or with 🔀 **Match** on a form that takes part in it.
2. 🔄 **Auto Match** runs when the screen opens. Rows that meet the rule are paired and arrive **pre-ticked** in green.
3. Read the **summary strip**: how many pairs, how many unmatched on each side, and any outstanding amount.
4. Untick any pair you do not accept. To pair rows by hand, tick one row on each side and press 🔗 **Match Selected**. ✖ on a pair puts both rows back.
5. Press ✔ **Confirm Ticked**. Both sides are stamped (so they never match again), a session record is written, and the reconciliation report opens in the document viewer.

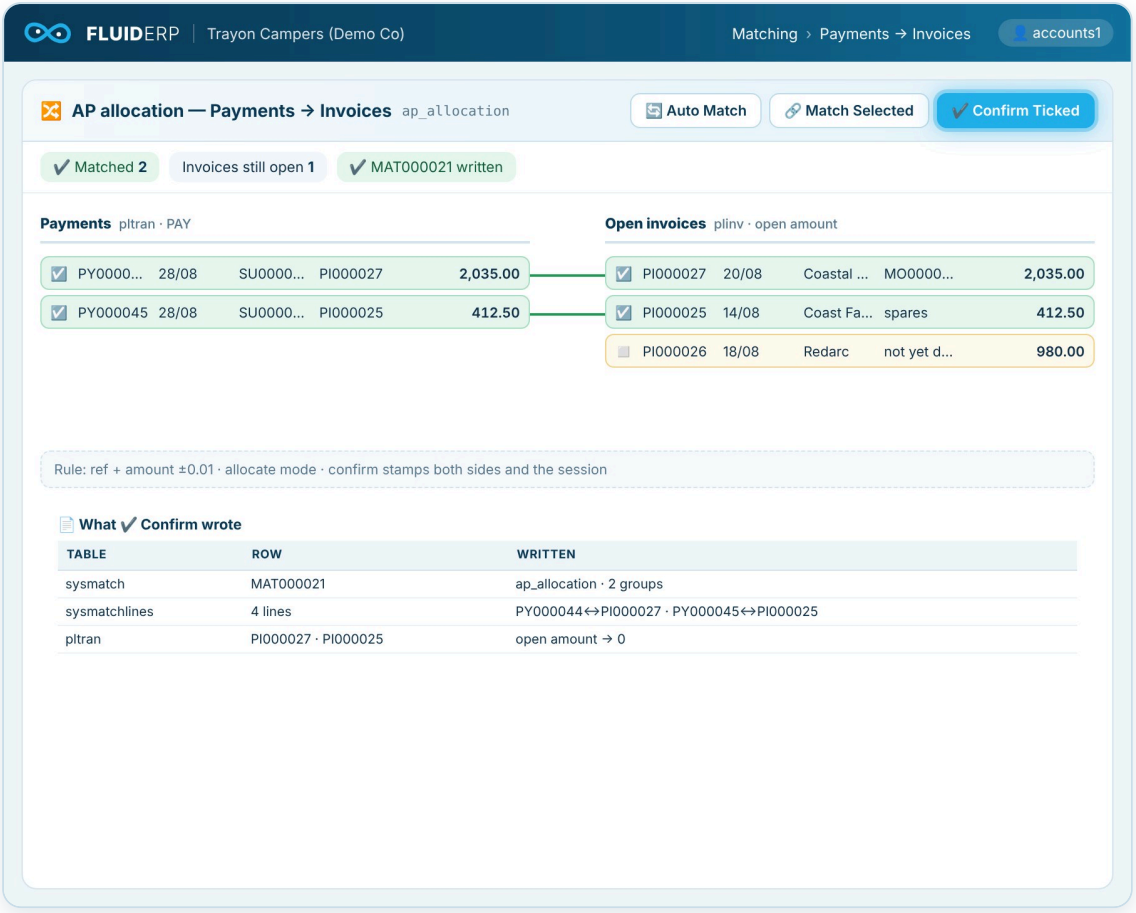
What ✔ Confirm Ticked writes	matching definition · confirm
✓ `sysmatch`	the session header — definition, who, when, number of groups (MAT number)
✓ `sysmatchlines`	one line per row in each confirmed group
✓ `each side`	the stamp field (for example matched_ind = M) and any counterpart fields the definition copies across

UNMATCHED IS INFORMATION

What is left after Auto Match is the exception list: an unexplained bank fee, a payment with no reference, a count that disagrees. Deal with those, not with the hundreds of rows that matched.

Allocate the contractor's payment

Who **acct1** | Where **Matching › AP allocation**



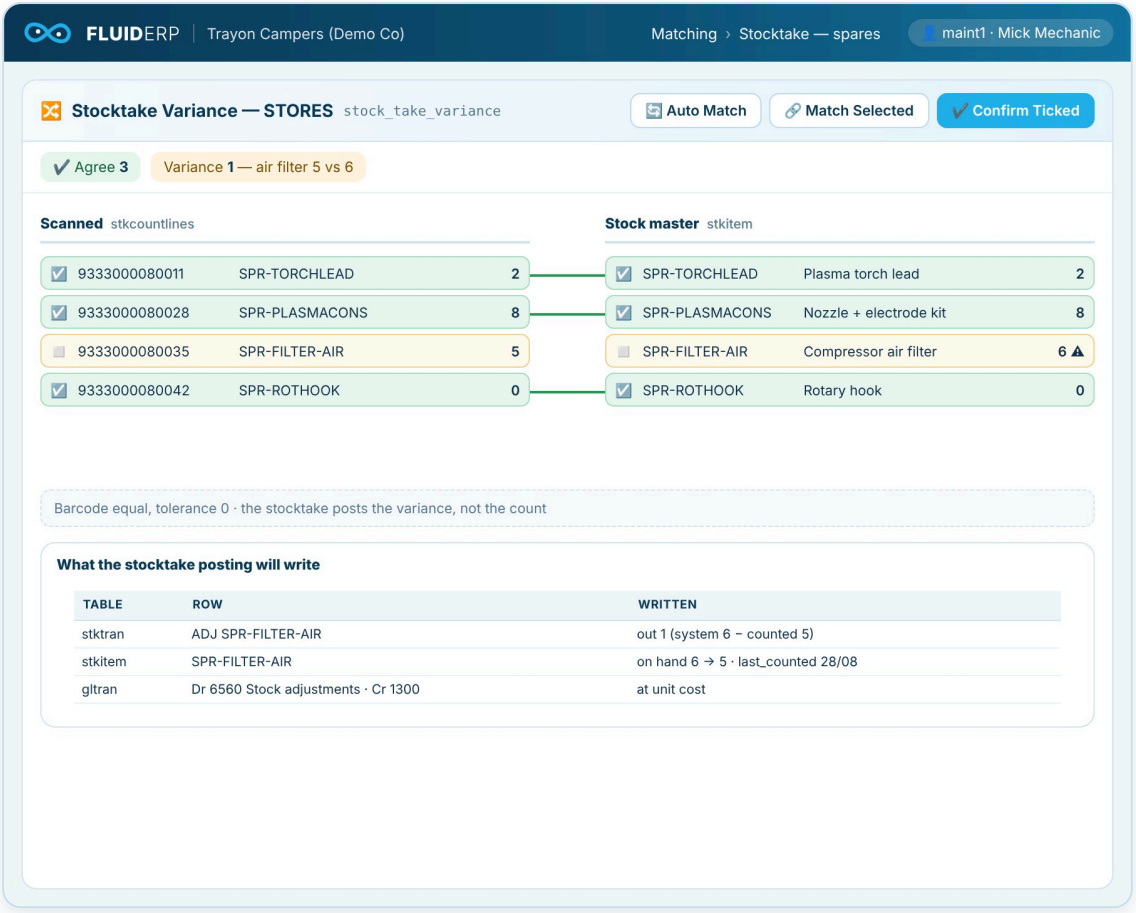
PY000044 pairs with PI000027; the Redarc invoice is not yet due and stays open.

1. Run the supplier payment run and post the payments.
2. Open **AP allocation — Payments → Invoices**; pairs arrive on reference and amount.
3. Press ✔ **Confirm Ticked**.

What ✔ Confirm Ticked writes	ap_allocation
✓ **sysmatch · sysmatchlines**	MAT000021 and its lines
✓ **pltran**	PI000027 and PI000025 open amount → 0

Count the spares cage

Who **stores** | Where **Matching › Stocktake — STORES**



Scans against the stock master: the air filter counts 5 against 6.

1. Scan the cage on a stocktake batch.
2. Open **Stocktake Variance**; review the variances.
3. Post the stocktake: it writes the adjustment (Dr stock adjustments · Cr 1300) for the difference only.

12 Messages, refusals and what to do

FLUID refuses an action rather than let a figure go wrong. Every refusal names its rule. These are the ones you will meet on this platform.

MESSAGE (IN SHORT)	WHY	WHAT TO DO
Failure codes required to close	Closing needs mode, cause, activity.	Choose them from the failure catalogue.
Order must be Done or Closed and flagged to capitalise	🏗 Capitalise rule.	Finish the order; set Capitalise and the asset.
Purchase invoice not Approved / same approver	Segregation of duties.	A second person approves.
Plan already has an open order	PMs are never raised twice.	Finish the open order first.
Spare short	Not enough on hand to issue.	Put the order On Hold — awaiting parts; buy the spare.
Already capitalised / already issued	One-way buttons.	Nothing to do.

13 Data reference and glossary

Tables you will meet

TABLE	HOLDS	WRITTEN BY
`mntclass · mntfailcode`	Equipment classes and the ISO 14224 failure catalogue	administrator
`mntloc · mntequip`	Functional locations and equipment (with derived KPIs)	planners; calculations
`mntspare`	Spares per equipment	planners
`mnttech`	Technicians and crafts	supervisors
`mntproc · mntproclines`	Procedures and steps	planners
`mntmeter · mntreading`	Meters and readings	operators; feeds; postings
`mntplan`	Preventive plans	planners; 🔁
`mntreq`	Requests	operators
`mntorder · mntorderlines · mntorderparts · mntlabour`	Orders, checklist, spares, hours	technicians; 🛠️ 📋 📦
`mntdown`	Downtime log	🛠️ 🛑
`stktran · gltran`	Stock ledger and GL	posting buttons only
`plinv · pltran`	Purchase invoices and supplier ledger	🧾; 📮
`fatran · faasset`	Fixed-asset transactions and register	🏗️

Glossary

TERM	MEANING
MTBF	Mean time between failures.
MTTR	Mean time to repair.
LOTO	Lock-out / tag-out isolation before work.
ISO 14224	The standard for failure mode, cause and activity codes.
PM	Preventive maintenance.
Condition-based plan	A plan that falls due when a reading crosses its limit.

TERM	MEANING
Capitalise	Move an improvement's cost from expense to a fixed asset.
Capture	Recording something once — by form, phone app, file or feed.
Posting button	An aqua button that writes rows to other tables from the record in front of you.
Posted · sealed	The record has been posted; the button is one-way and a second press writes nothing.
Calculated field (ƒ)	A read-only field the calculation engine keeps up to date.
Reference	The source record number carried on every row a button writes.
Matching definition	The rule for pairing rows of two tables and what confirming writes.
Session (MAT…)	The record of one confirmed matching run.
Gateway	The server that checks every read and write against the rules for your group.

A Walk it once on the demo package

The Trayon package carries the maintenance story at 20 August 2026. Sign in as maint1 and follow these steps.

The station (maint1)

1. 🧭 Maintenance Station — the board, Due & Overdue PMs (three), the Gantt, equipment KPIs, the downtime Pareto, spares short (the rotary hook), the backlog.
2. Preventive Plans → 🔁 Raise Due PM Orders → MO000009, MO000010, MO000011 appear Planned.

An order end to end (maint1)

1. MO000009 → Released → 📋 Load Checklist → 🔩 Spares: add SPR-PLASMACONS → 📦 Issue Spares → Stock Ledger MNT-ISS row; GL Dr 6500 · Cr 1300.
2. ⏱️ Hours; In Progress → Done (findings) → Closed (mode, cause).
3. MR000004 (forklift seat belt) → ⏱️ Now → 🛠️ Create Work Order → 📅 Put on Calendar.

Contractor and capitalise

1. Portal (service@coastalair.example.com): MO000004 Accepted → In Progress → Done; staff side Closed with codes → 🧾 Raise Contractor Invoice → PI to SU000012 for 1,850 + GST; accounts approves and posts.
2. MO000005 (LED strips, Done, capitalise on) → 🏗️ Capitalise → FA000003 cost + 683; GL Dr 1500 · Cr 6500.
3. ⚙️ Work Centres → CANVAS: maintenance downtime, breakdown hours, availability.

For the administrator

TASK	HOW
Generator	`python3 tools_gen_maint.py` merges the module into an adopting institution by _id.
Posting service	Runs 🔁 Raise Due PM Orders at 06:00 and the mnt_backlog_daily notification.
Capacity clusters	⛔ Block Work Centre is generated only where the capacity engine runs (Sceptre Foods).