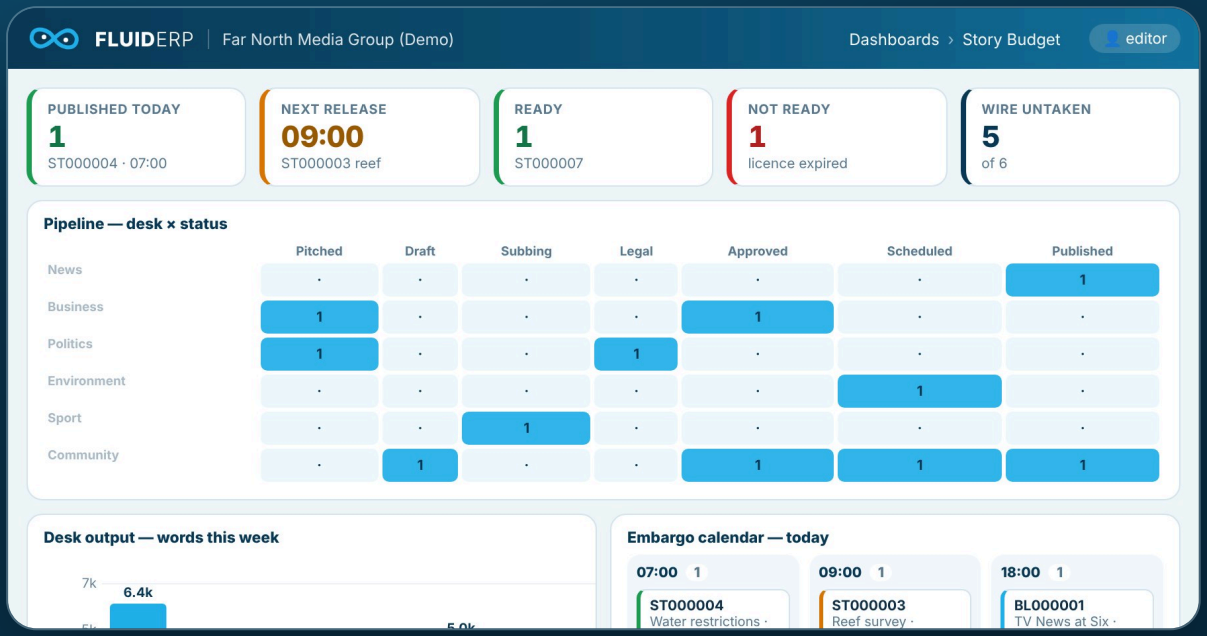
FLUID ERP · USER MANUAL

FLUID Newsroom

Wire to air.

Agency wire, the story budget, editor signatures, embargo and scheduled release, outbound feeds, corrections on sealed stories, the bulletin rundown clock, contributor fees and syndication.

NewsML-G2 · RSS · NITF wire | Signed & sealed stories | Embargo & scheduled release | Rundown clock | RSS · Atom · JSON Feed · NewsML out | Fees & syndication



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 The wire 9

Capture the wire 9

Re-send without duplicates 10

Take a wire item 11

5 Pitching and filing 12

File from the field (contributor app) 12

6 Editing, signing and rights 13

Work the budget 13

Approve with a signature 14

Fix a story that is not ready 15

7 Embargo, release, feeds and corrections 16

Schedule (embargo) a story 16

Let the release worker publish 17

Publish outward through feeds 18

Correct a published (sealed) story 19

8 The bulletin rundown 20

Time the rundown 20

Trim and re-time 21

9 Fees and syndication 22

Pay a contributor fee 22

Invoice a syndication licence 23

10 Critical dashboards 24

Story budget 24

Rights and corrections 25

Commercial 26

11 Reports 27

Running and printing a report 27

Report catalogue 27

Story budget 28

Corrections register 29

Running order 30

Syndication revenue 31

12 Matching 32

How every matching screen works 32

Wire items against stories 33

Allocate syndication receipts 34

13 Messages, refusals and what to do 35

14 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 a newsroom on FLUID Newsroom: taking wire copy, filing from the field, moving stories through the budget, signing and scheduling them, publishing to feeds and the reader app, correcting published stories, timing the evening bulletin and paying contributors and invoicing syndication. The worked examples follow Far North Media Group at 07:30 on 13 October 2026.

Who it is for

READER	READ THESE CHAPTERS
Editors	3 – 7, 10, 11
Sub-editors	3, 4, 6, 7
Reporters and contributors	3, 5 (filing from the app)
Producers	3, 8 (the rundown)
Legal	6 (Legal status, signatures)
Commercial	9 (fees and licences), 10, 11, 12

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

Stories ST000001–ST000010, the embargo at 09:00, the 07:00 release, the correction CR000001 and the bulletin timing (25:40 against a 25:00 hard out) follow the Stage 70 Far North story. Some wire headlines, fee and licence amounts and dashboard histories are illustrative and differ from the shipped package — Appendix A uses the shipped records. Screenshots come from the FLUID Newsroom demonstration film.

2 How FLUID works: capture once, post everywhere

A story is one record from pitch to archive. The wire item becomes it; the editor signs it; the release worker publishes it at its time; the feeds and the reader app carry it; the rundown times it; the fee and the licence are invoiced from it.



One wire item or story fans out to the budget, the signature, the release, the feeds, the reader app, the rundown and the ledger.

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.	📥 Take · ✍️ Approve (sign) · ⏱️ Time the Rundown · 💳 Pay Fee

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 = ST000004`). 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.

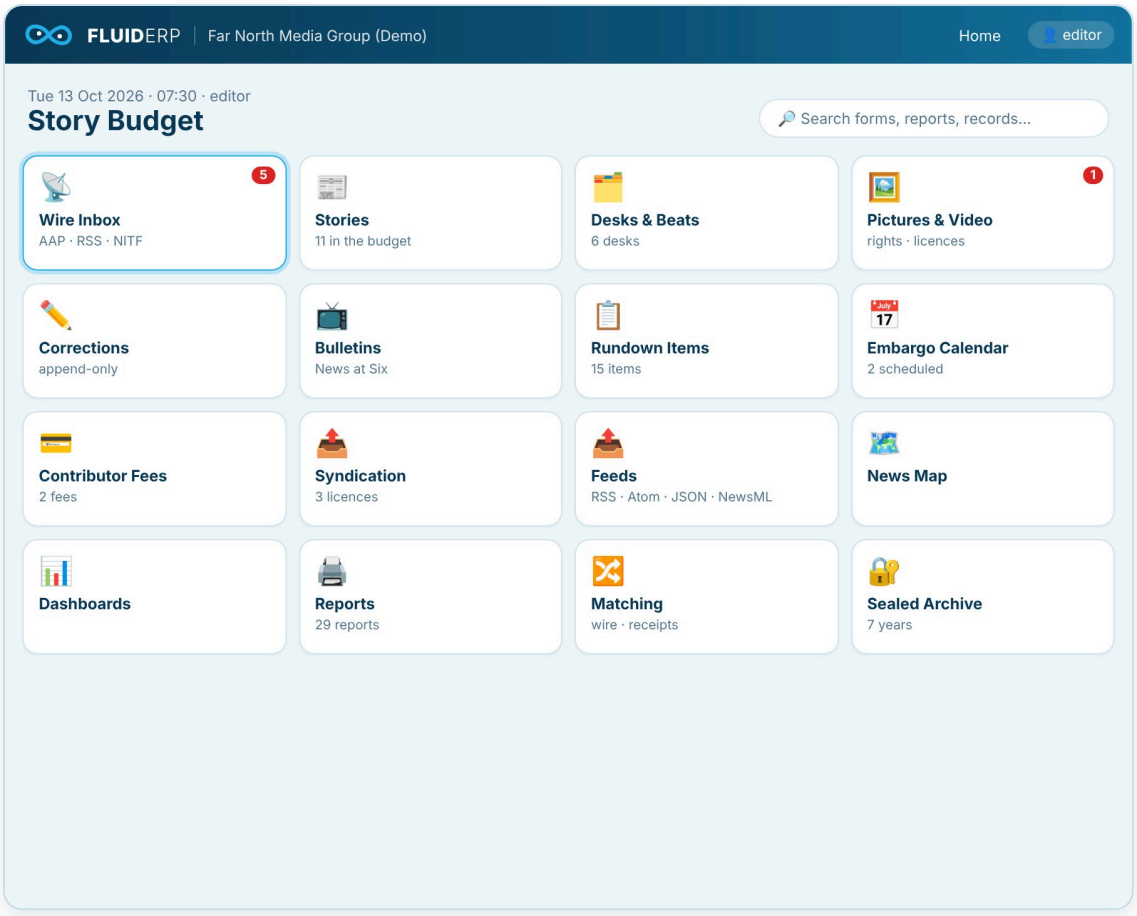
SIGN-IN	GROUP / DESK	USES FLUID TO ...
editor	Editor	Budget, signing, scheduling, corrections, fees approval
sub_editor	Sub-editor	Wire inbox, subbing, corrections
reporter	Reporter	Own stories and drafts
producer	Producer	Bulletins and rundowns
legal	Legal	Stories in Legal, risk and notes
commercial	Commercial	Contributor fees, syndication licences, invoices
kelsie@marrphoto.example.org · desk@tropicstringers.example.org	Contributors (app)	Pitch or File a Story, My Stories, My Fees, Latest News

DEMO SIGN-INS

Staff passwords are the same as the user name in the demo package; contributors use contributor123. Change every password at go-live.

The launcher

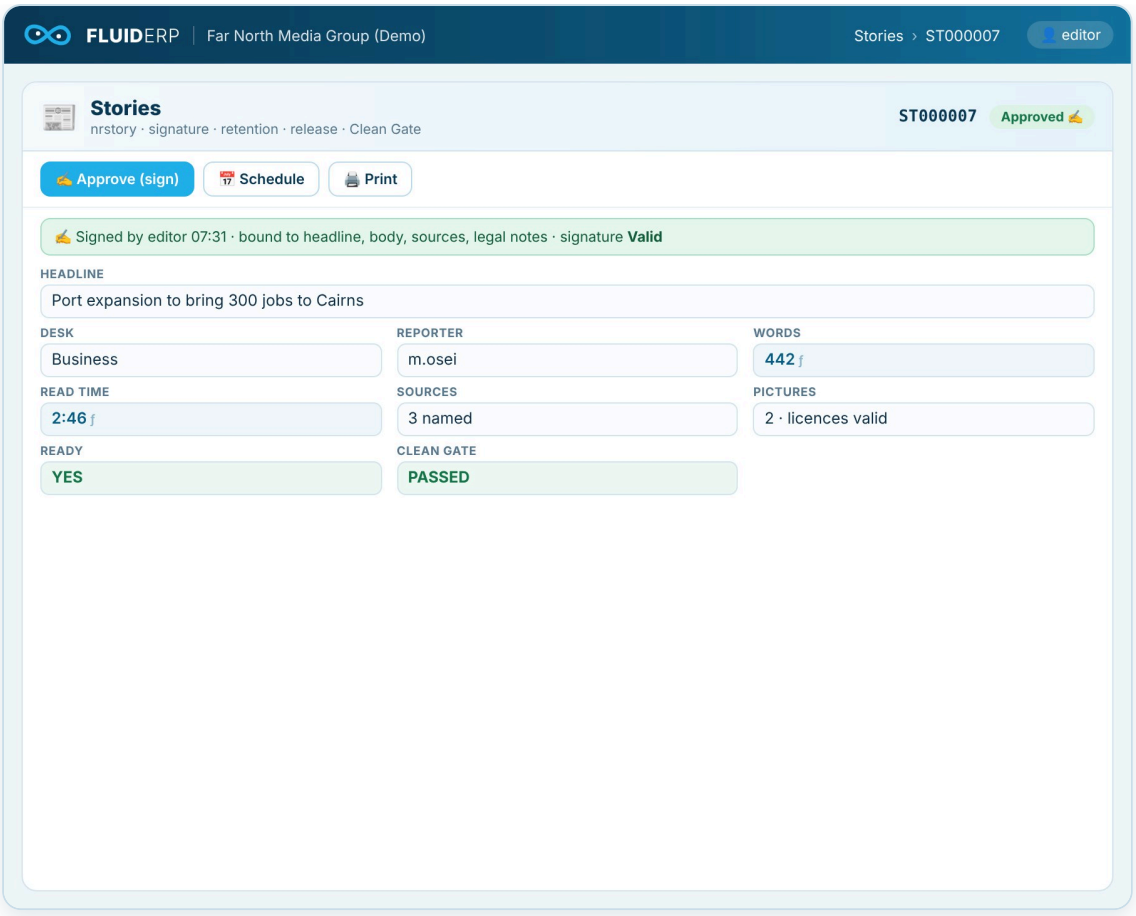
After sign-in you see the launcher: a grid of tiles for the screens, dashboards, reports and matching screens your group may open. A red badge on a tile counts records waiting for you. Use the search box to jump straight to a form, a report or a record number.



The story budget launcher: five untaken wire items and a picture with an expired licence 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 story: status is the workflow; words, read time and ready are calculated; the banner shows the editor's signature.

Everyday actions

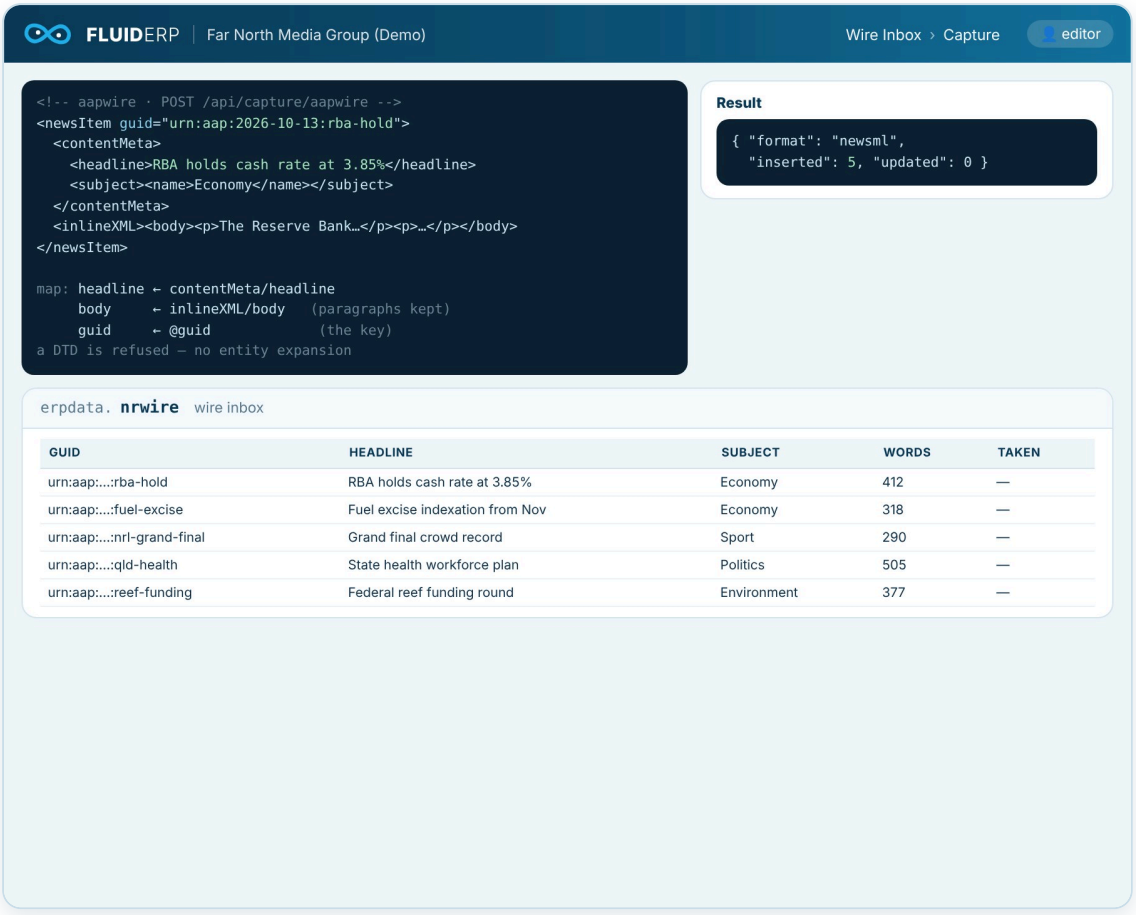
TO ...	DO THIS
Create a record	Open the tile, press ➕ **New**, fill the white fields, press 💾 **Save**. The number is issued on first save.
Find a record	Use the search box on the launcher or the filter row above any list. Record numbers, names and (where enabled) full text are searchable.
Refresh calculated 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 The wire

Agency copy is captured in the agencies' own XML. Each item is keyed by its guid, so a re-sent file never duplicates.

Capture the wire

Who **administrator / scheduled task** | Where **Wire Inbox › Capture · POST /api/capture/aapwire** | Table **nrwire**



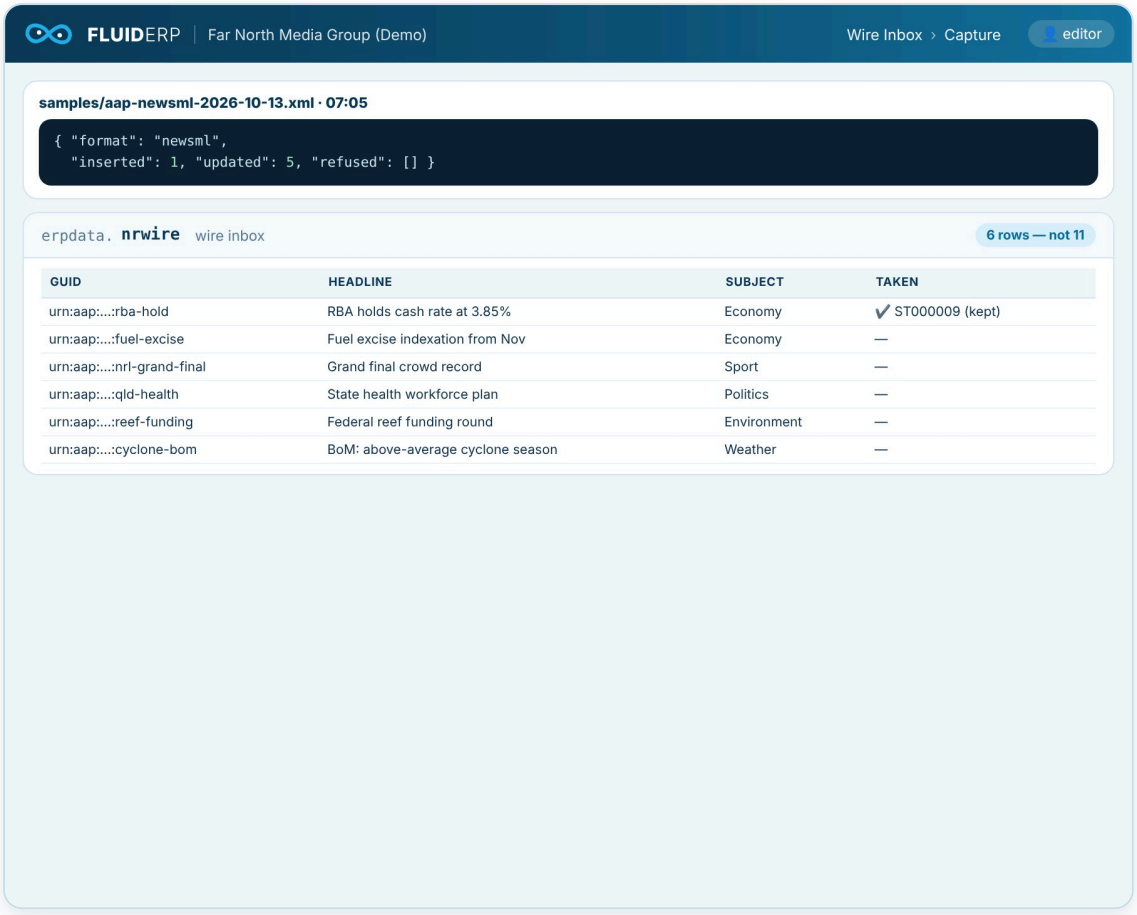
AAP NewsML-G2: headline, body paragraphs and the guid mapped; five items become five rows.

1. Send the agency file to its capture definition: `aapwire` (NewsML-G2), `rsswire` (RSS), `nitfwire` (NITF).
2. The definition names the repeating element and maps headline, body (paragraphs kept), subject and the guid.
3. A file with a DTD is refused with the reason (no entity expansion).

Re-send without duplicates

Who **administrator** Where **Wire Inbox › Capture**

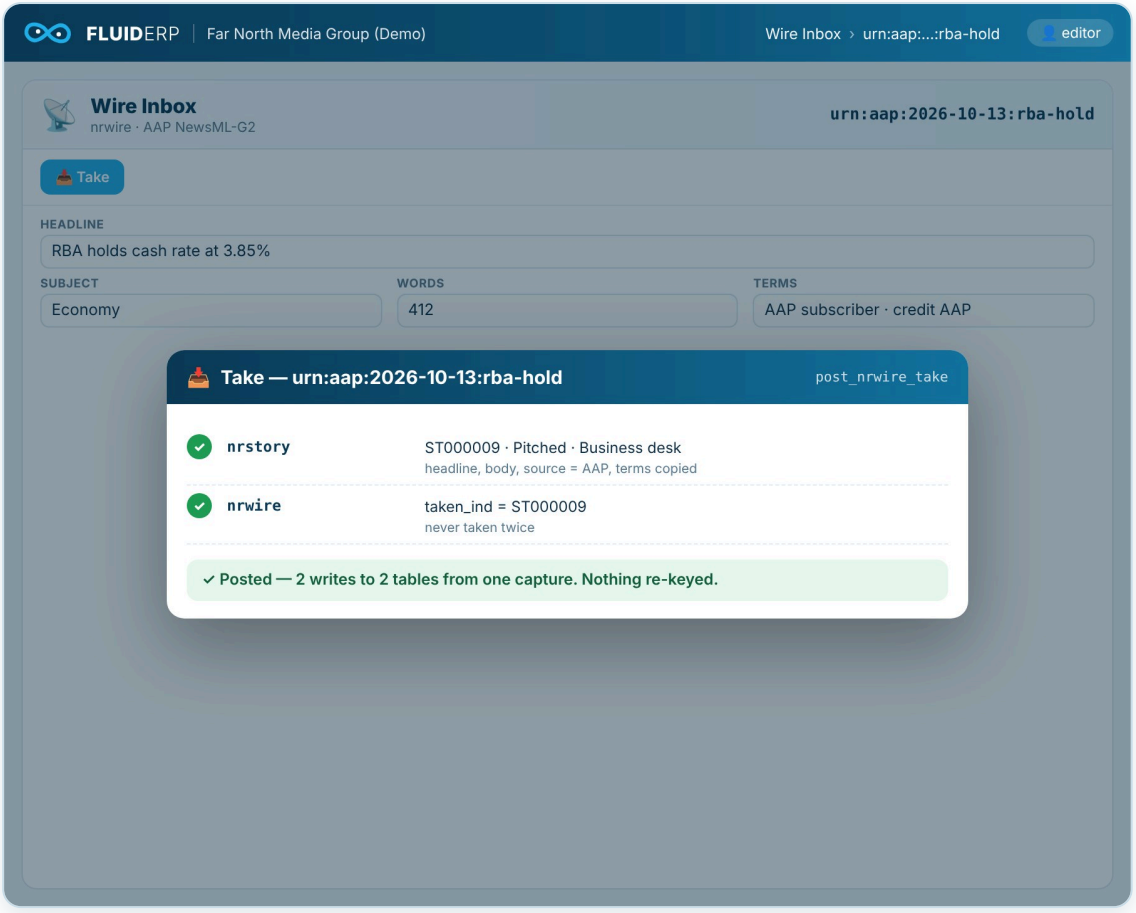
Capture upserts with $set: it updates what the file carries and keeps what it does not, such as an editor's taken mark.



The agency re-sends with a sixth item: one inserted, five updated — six rows, not eleven; the taken flag is kept.

Take a wire item

Who **editor, sub_editor** Where **Wire Inbox › item**



📥 Take — a Pitched story with the wire's headline, body, source and terms; the wire row marked taken.

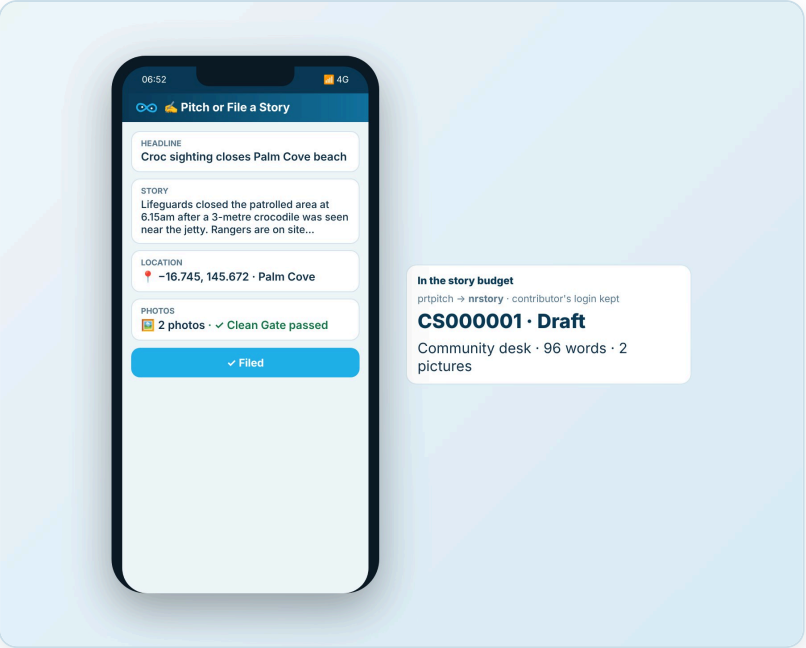
1 Open the item and press 📥 **Take**.

What 📥 Take writes	post_nrwire_take
✓ **nrstory**	a Pitched story on the right desk with headline, body, source and usage terms
✓ **nrwire**	taken_ind = the story number — an item is taken once

5 Pitching and filing

File from the field (contributor app)

Who **contributors** | Where **App › ✍️ Pitch or File a Story** | Table **prtpitch → nrstory**



A stringer files with the phone's GPS and two photos; the Clean Gate checks text and pictures.

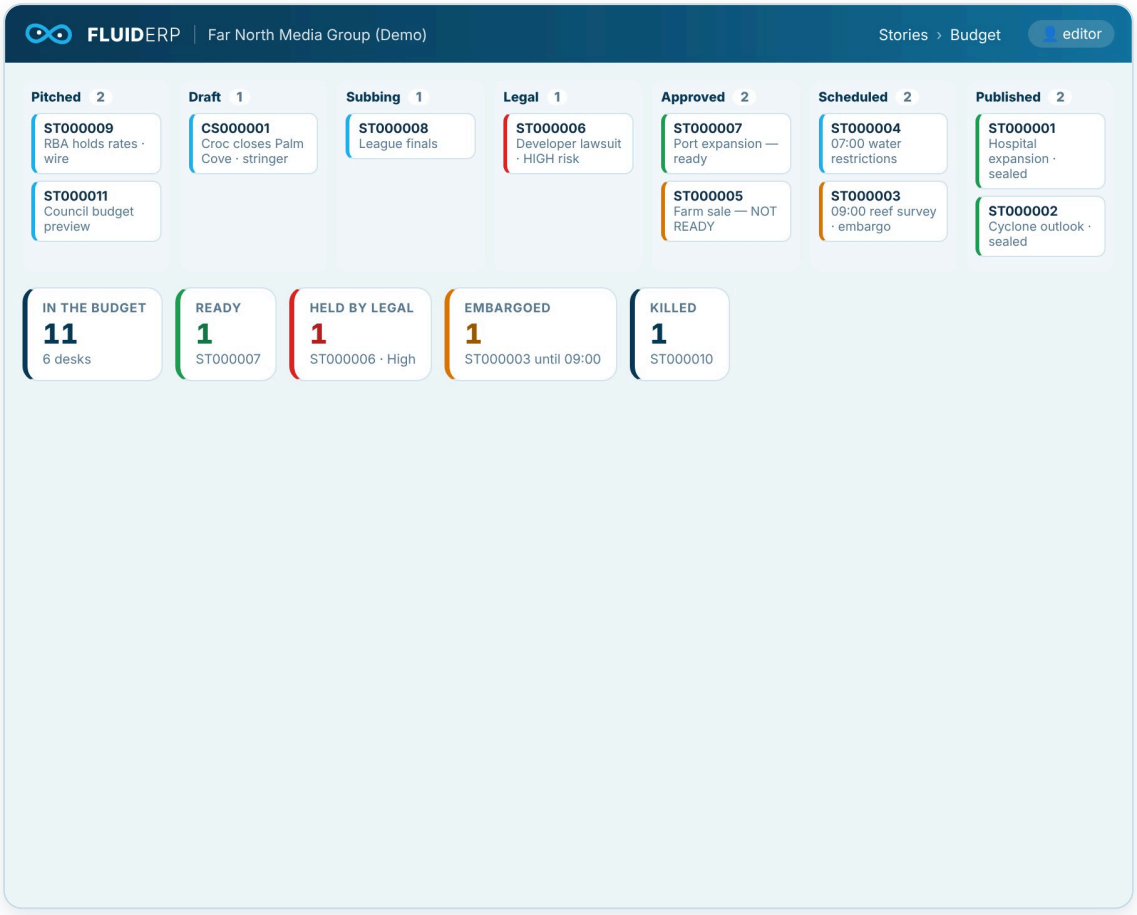
1. Open ✍️ **Pitch or File a Story**; type the headline and copy.
2. The phone adds the location; attach photos.
3. Press **File**. The Clean Gate validates it and it arrives in the budget as **Draft** (or Pitched) with your login.
4. Follow it in 📰 **My Stories** and your payments in 💳 **My Fees**.

6 Editing, signing and rights

The status is the workflow: Pitched → Assigned → Draft → Subbing → Legal → Approved → Scheduled → Published; Killed from anywhere; Withdrawn after publication.

Work the budget

Who **editor** | Where **Stories › Budget** | Table **nrstory**

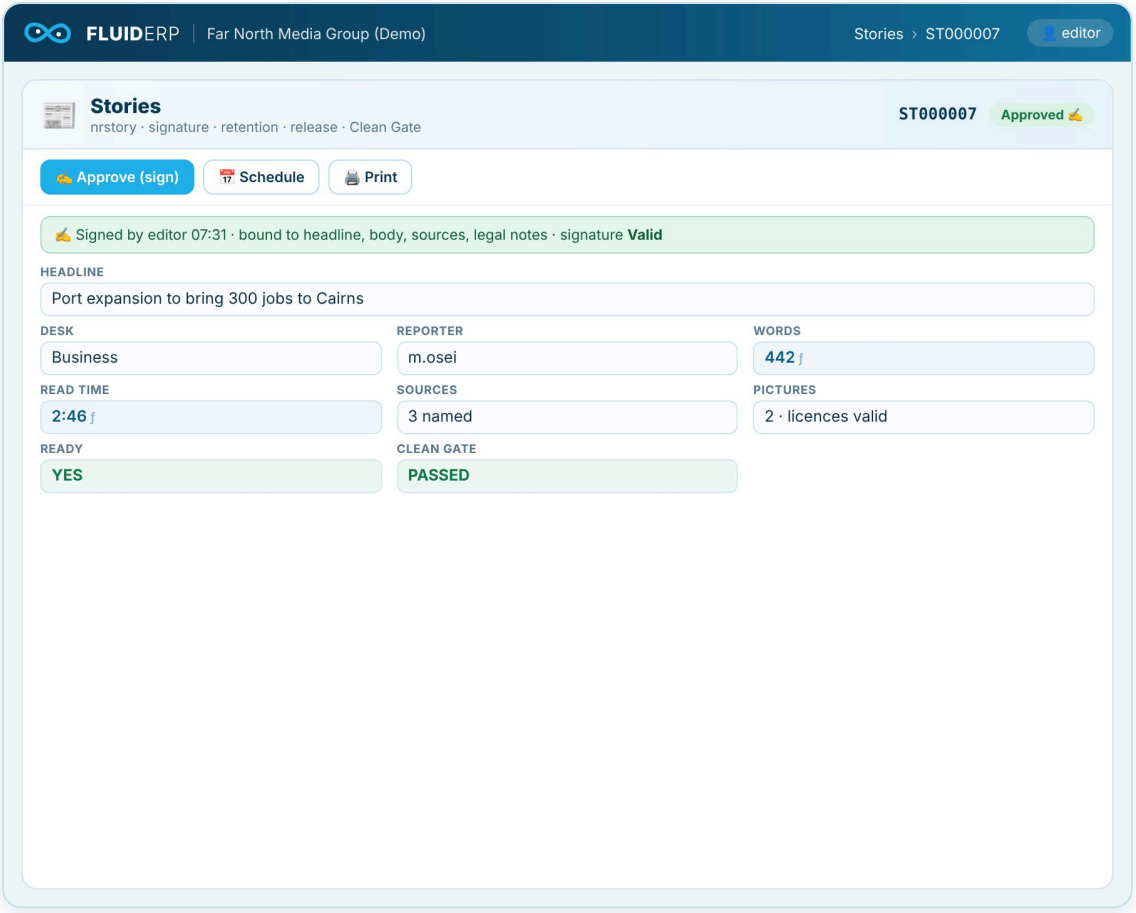


Eleven stories on one board, with ready, legal, embargoed and killed counts.

1. Assign pitched stories to a reporter and desk.
2. Move drafts to **Subbing**, then **Legal** where needed (risk High / Medium / Low with notes).
3. Words and broadcast read time (160 words a minute) are calculated as the copy changes.

Approve with a signature

Who **editor** | Where **Stories › ST000007**



Approved — the editor re-enters the password; the signature is bound to the text, sources and legal notes.

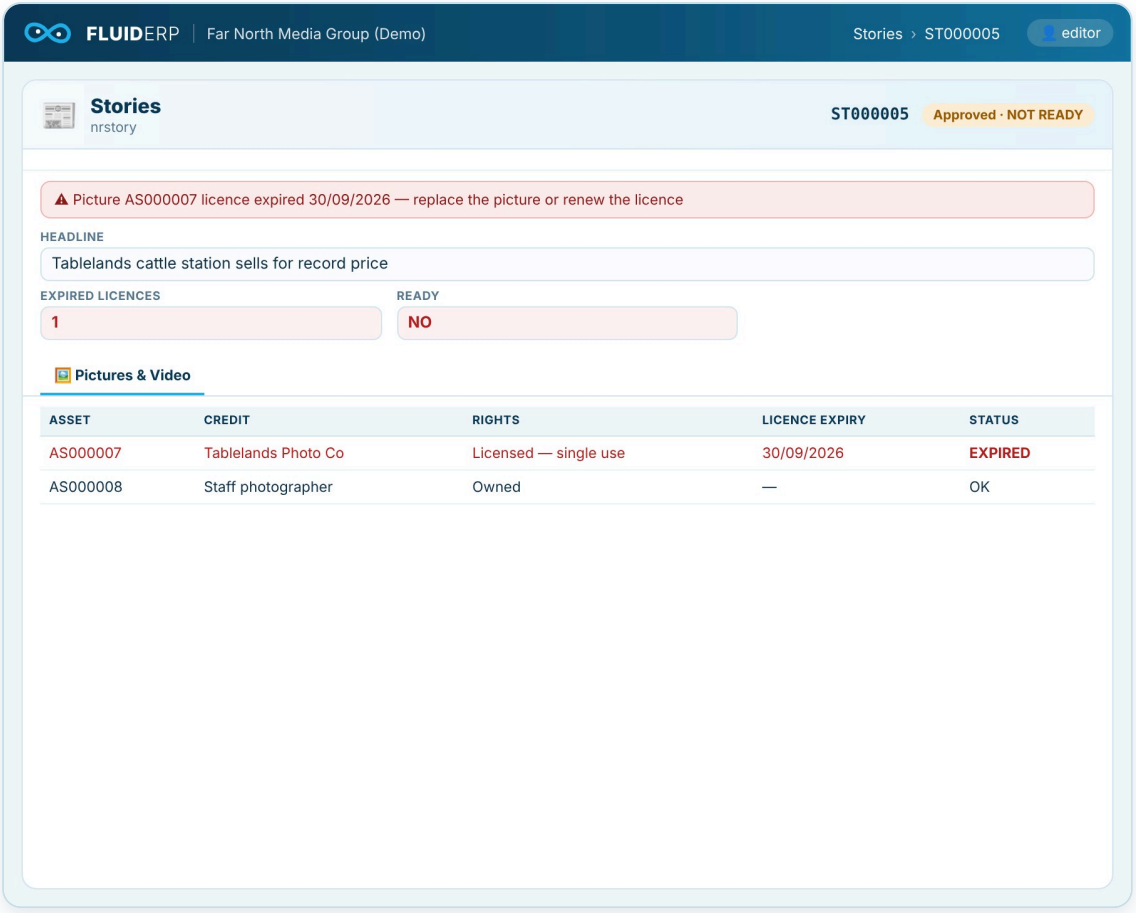
1. Press ✍ **Approve (sign)**, re-enter your password and confirm the printed meaning.
2. The signature stands through Scheduled and Published (signature covers); a later change to the body demands a reason and supersedes it.

CHECK IT

Ready = Approved, sourced and no expired picture licence.

Fix a story that is not ready

Who **editor, picture desk** Where **Stories › ST000005**



ST000005 is Approved but not ready — its picture's licence expired on 30 September.

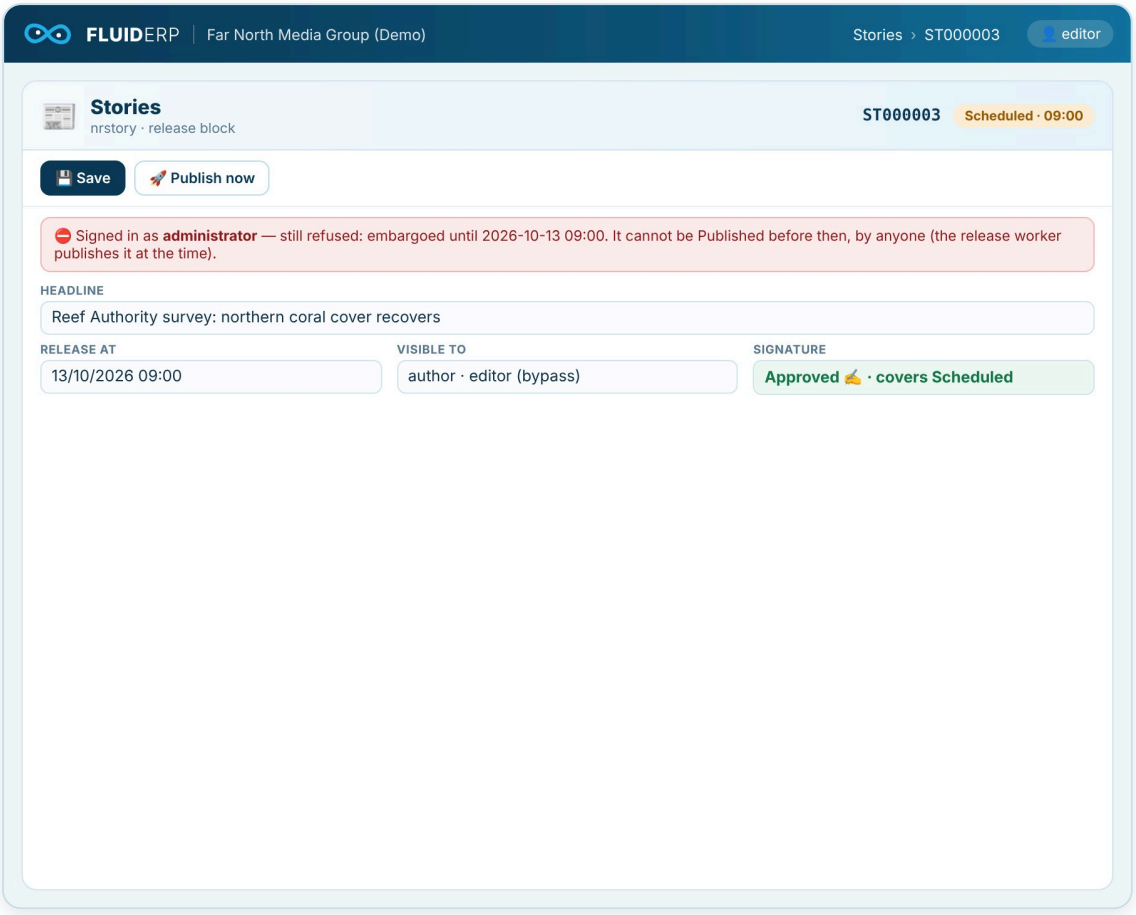
1. Replace the picture or renew its licence in 🖼️ **Pictures & Video**.
2. Ready recalculates; the story can then be scheduled.

7 Embargo, release, feeds and corrections

A scheduled story is embargoed: invisible and unpublishable until its time, for everyone. The release worker publishes it at the time.

Schedule (embargo) a story

Who **editor** | Where **Stories › ST000003**



ST000003 is embargoed until 09:00; publishing now is refused — even for the administrator.

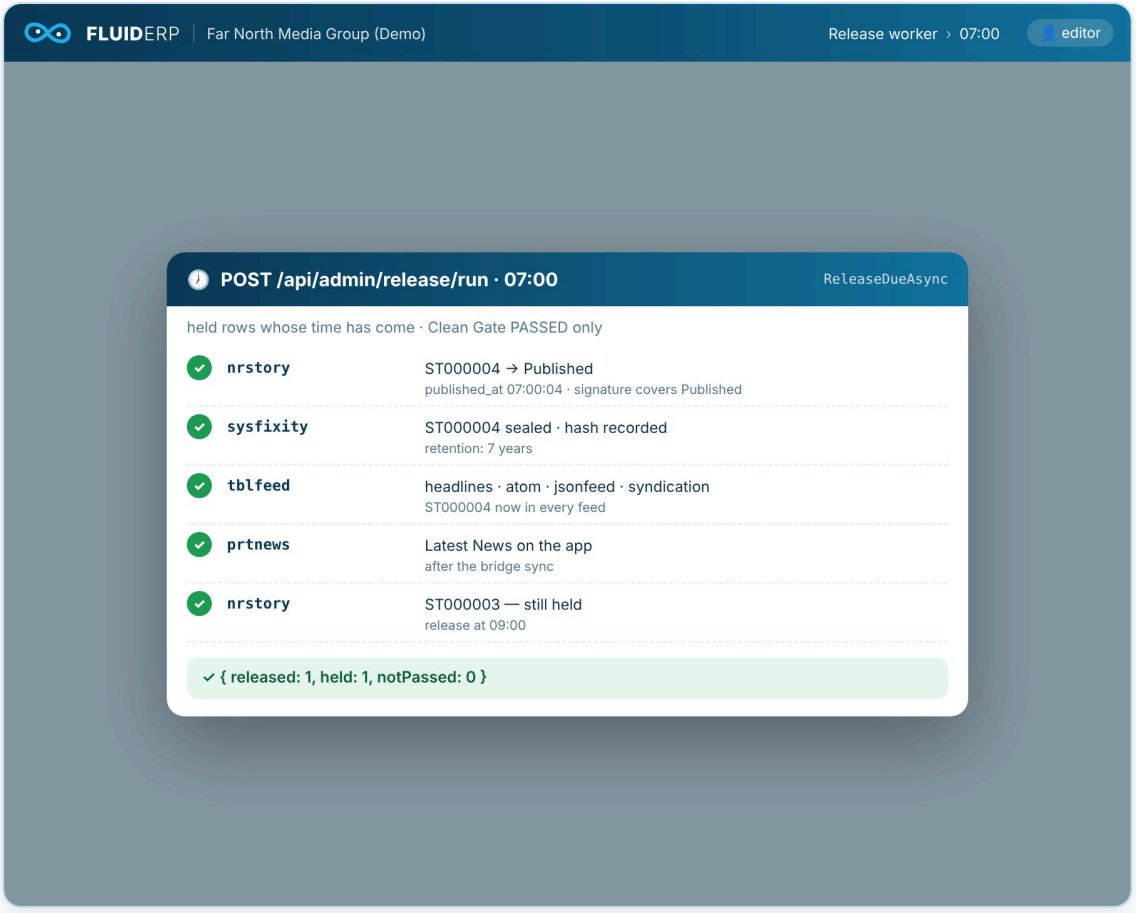
1. Set **Release at** and the status **Scheduled**. Save.
2. Until that time the story is visible only to its author and the bypass groups; it is in no feed and not in the reader app.

If the system refuses

YOU SEE	WHAT IT MEANS / WHAT TO DO
Embargoed until … — it cannot be Published before then, by anyone	Wait for the release worker, or change Release at.

Let the release worker publish

Who **scheduled task** Where **Release worker · POST /api/admin/release/run**



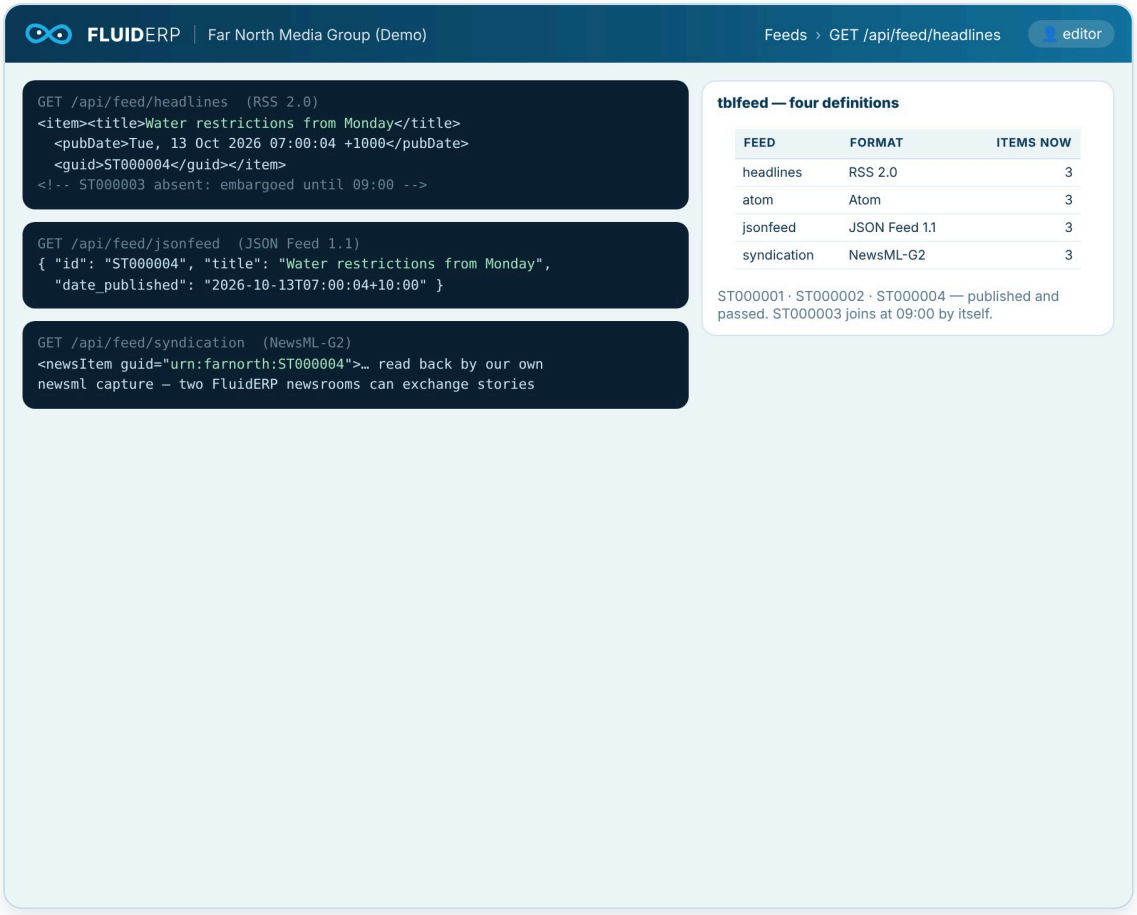
At 07:00 the worker publishes ST000004, seals it and puts it in the feeds; ST000003 stays held.

What Release worker writes	ReleaseDueAsync
✓ **nrstory**	due stories → Published, published_at stamped — only those the Clean Gate has passed
✓ **sysfixity**	the fixity row — a published story is sealed for seven years
✓ **tblfeed / prtnews**	the story appears in the feeds and, after the bridge sync, in the reader app

Publish outward through feeds

Who **administrator** | Where **Feeds · GET /api/feed/{id}** | Table **tblfeed**

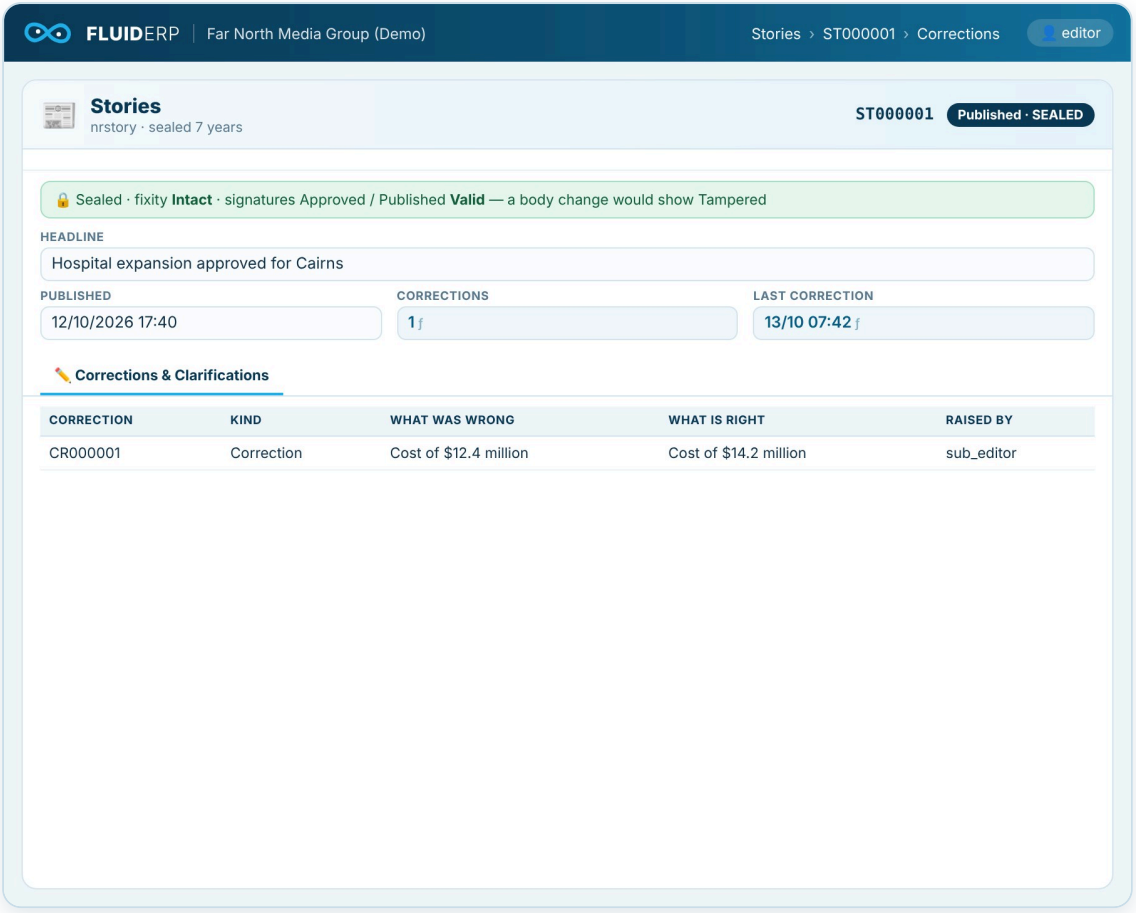
Feeds read through the caller's rights, so an embargoed or un-passed story is never in a feed. The NewsML-G2 feed is read back by the newsml capture, so two FLUID newsrooms can exchange stories.



RSS 2.0, Atom, JSON Feed 1.1 and NewsML-G2 from one definition each.

Correct a published (sealed) story

Who **sub_editor, editor** | Where **Stories › ST000001 › Corrections** | Table **nrcorrection**



CR000001 — the cost was $14.2 million, not $12.4 million; the story's fixity stays Intact.

1. Open the published story; its body cannot be edited.
2. On ✏️ **Corrections & Clarifications** add the correction: kind, what was wrong, what is right.
3. The register is append-only; the story's correction count and last correction update. These allowed fields sit outside the sealed hash.

If the system refuses

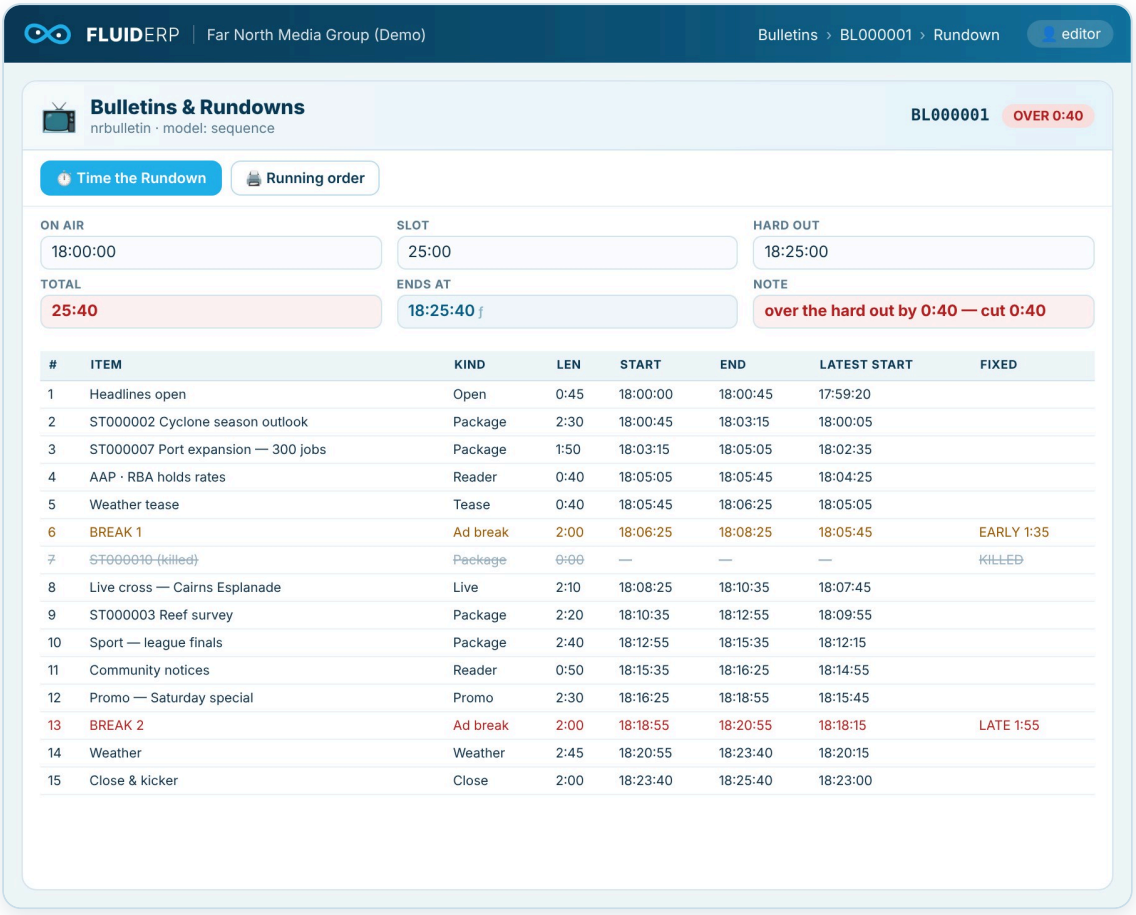
YOU SEE	WHAT IT MEANS / WHAT TO DO
Sealed — the body cannot be changed	Use a correction, clarification, update or editor's note.

8 The bulletin rundown

The rundown is an ordered list of items with lengths that must fit the slot. The sequence model times it.

Time the rundown

Who **producer** | Where **Bulletins › BL000001 › Rundown** | Table **nrbulletin · nrrunitem**



#	ITEM	KIND	LEN	START	END	LATEST START	FIXED
1	Headlines open	Open	0:45	18:00:00	18:00:45	17:59:20	
2	ST000002 Cyclone season outlook	Package	2:30	18:00:45	18:03:15	18:00:05	
3	ST000007 Port expansion — 300 jobs	Package	1:50	18:03:15	18:05:05	18:02:35	
4	AAP · RBA holds rates	Reader	0:40	18:05:05	18:05:45	18:04:25	
5	Weather tease	Tease	0:40	18:05:45	18:06:25	18:05:05	
6	BREAK 1	Ad break	2:00	18:06:25	18:08:25	18:05:45	EARLY 1:35
~~7~~	~~ST000010 (killed)~~	~~Package~~	~~0:00~~	—	—	—	~~KILLED~~
8	Live cross — Cairns Esplanade	Live	2:10	18:08:25	18:10:35	18:07:45	
9	ST000003 Reef survey	Package	2:20	18:10:35	18:12:55	18:09:55	
10	Sport — league finals	Package	2:40	18:12:55	18:15:35	18:12:15	
11	Community notices	Reader	0:50	18:15:35	18:16:25	18:14:55	
12	Promo — Saturday special	Promo	2:30	18:16:25	18:18:55	18:15:45	
13	BREAK 2	Ad break	2:00	18:18:55	18:20:55	18:18:15	LATE 1:55
14	Weather	Weather	2:45	18:20:55	18:23:40	18:20:15	
15	Close & kicker	Close	2:00	18:23:40	18:25:40	18:23:00	

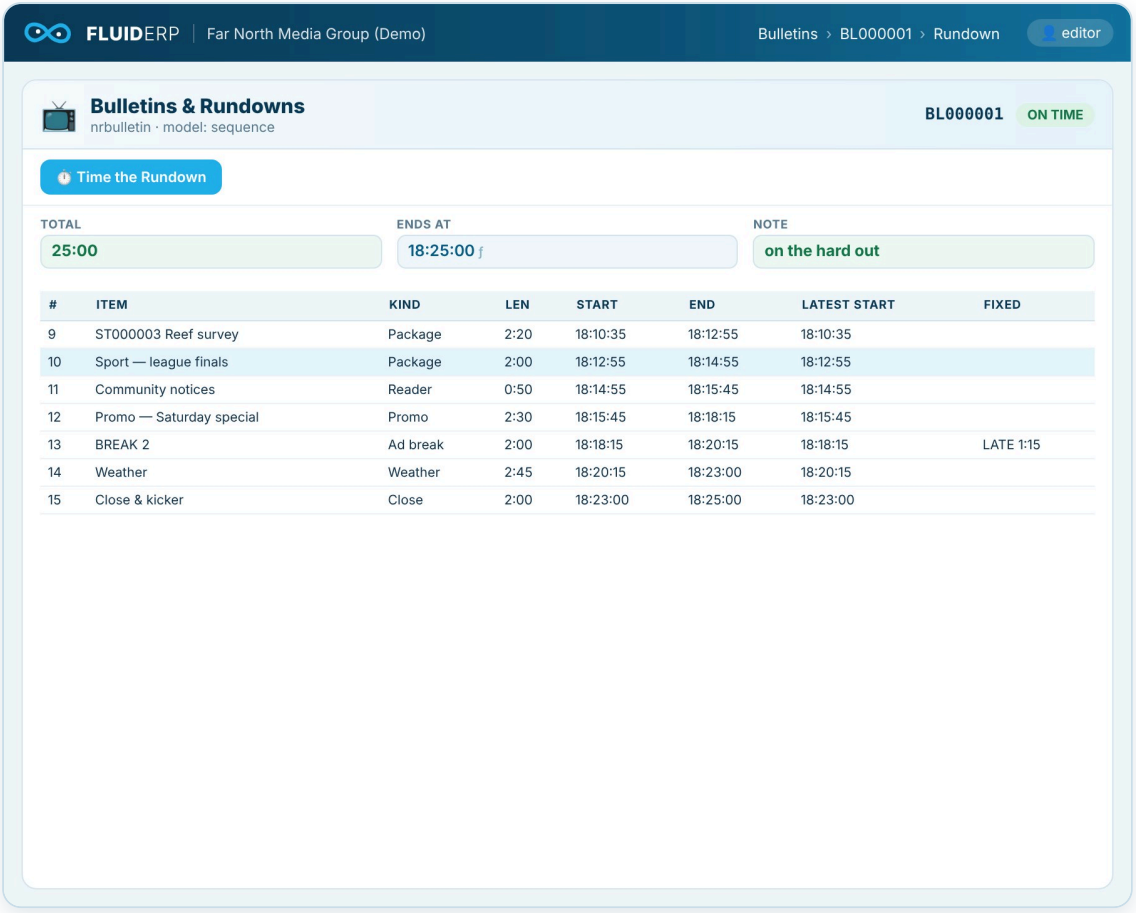
Fifteen items, two fixed-time breaks, one killed item: 25:40 against a 25:00 hard out — cut 0:40.

1. Add the items in order: story (its read time is the default length), kind, length, and a fixed time for breaks or live crosses. Mark killed items Skip.
2. Press ⏱️ **Time the Rundown**.

What ⏱️ Time the Rundown writes	model: sequence
✓ `nrrunitem`	each item's start, end, running time, latest start (back-timed to the hard out) and, for fixed items, EARLY / ON TIME / LATE
✓ `nrbulletin`	total, over / under, the clock it ends at, seconds to cut, and the producer's note

Trim and re-time

Who **producer** | Where **Bulletins › BL000001 › Rundown**



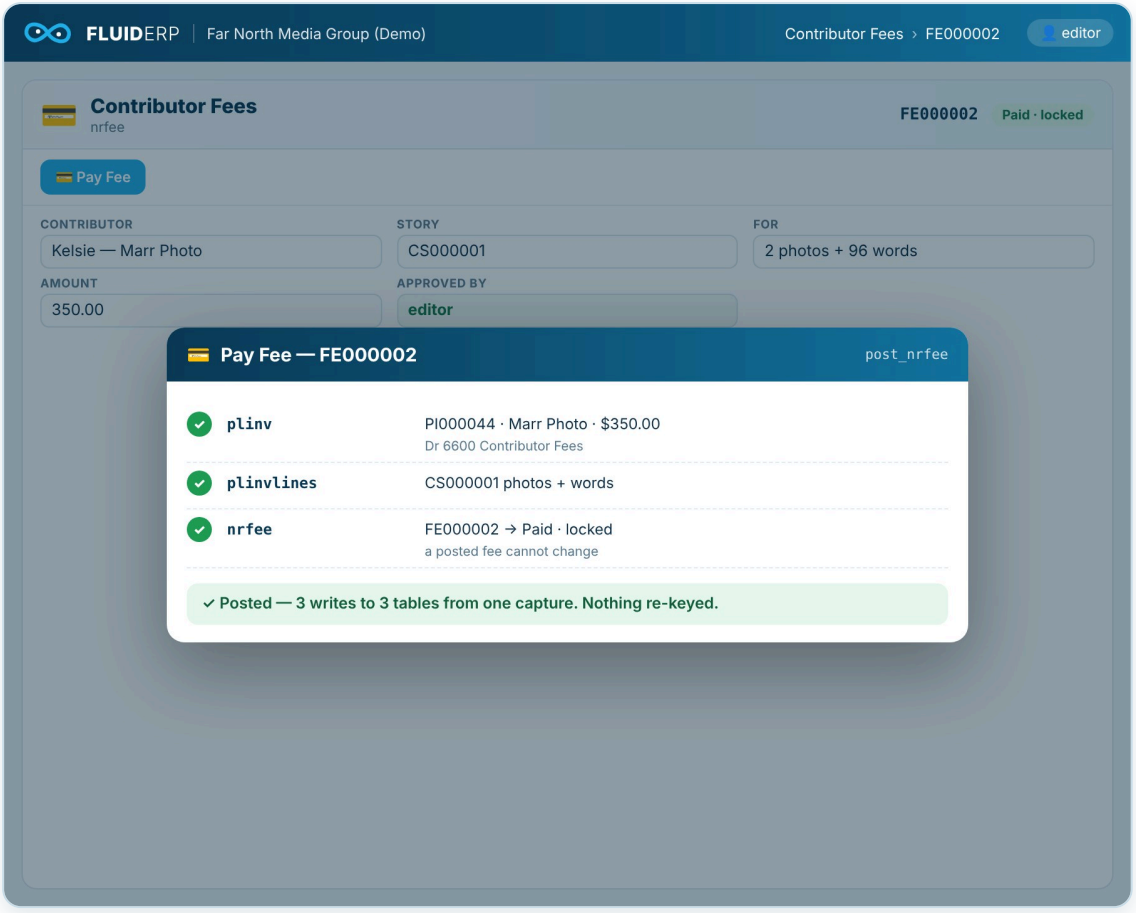
Sport trimmed to 2:00 — the bulletin ends at 18:25:00, on the hard out.

1. Change an item's length (or kill an item) and press ⏱️ **Time the Rundown** again. Every later time moves.

9 Fees and syndication

Pay a contributor fee

Who **commercial (after editor approval)** | Where **Contributor Fees** | Table **nrfee**



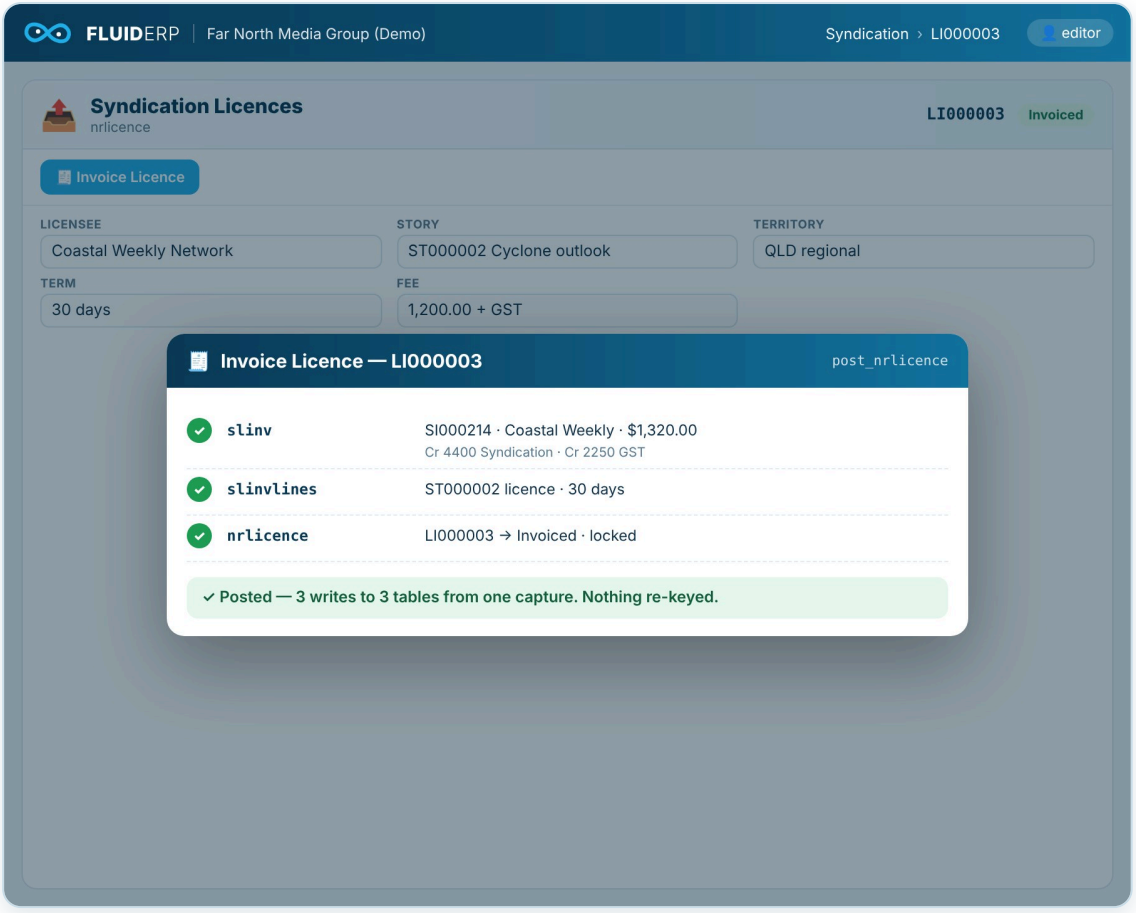
💳 Pay Fee — a purchase invoice to the contributor against 6600 Contributor Fees; the fee locks.

1. The editor approves the fee.
2. Press 💳 **Pay Fee**.

What 💳 Pay Fee writes	post_nrfee
✓ **plinv**	a purchase invoice to the contributor, Dr 6600 Contributor Fees
✓ **plinvlines**	the story and what was supplied
✓ **nrfee**	Paid · locked — a posted fee cannot change

Invoice a syndication licence

Who **commercial** | Where **Syndication** | Table **nrlicence**



🧾 Invoice Licence — a sales invoice to the licensee, Cr 4400 Syndication & Licensing Income.

1. Record the licensee, story, territory, term and fee.
2. Press 🧾 **Invoice Licence**.

What 🧾 Invoice Licence writes	post_nrlicence
✓ **slinv**	a sales invoice with GST, Cr 4400
✓ **slinvlines**	the licensed story and term
✓ **nrlicence**	Invoiced · locked

10 Critical dashboards

Three dashboards cover the day's budget, rights and corrections, and the commercial desk.

Story budget

Where **Dashboards › Story Budget** Who **editor**

Published today, next release, ready and not ready, untaken wire, the pipeline by desk × status, desk output and the embargo calendar.



Story budget

What each tile is and where it comes from

Ready / Not ready
Approved stories with and without everything they need.
`nrstory.ready`

Next release
Earliest scheduled release time.
`min nrstory.release_at`

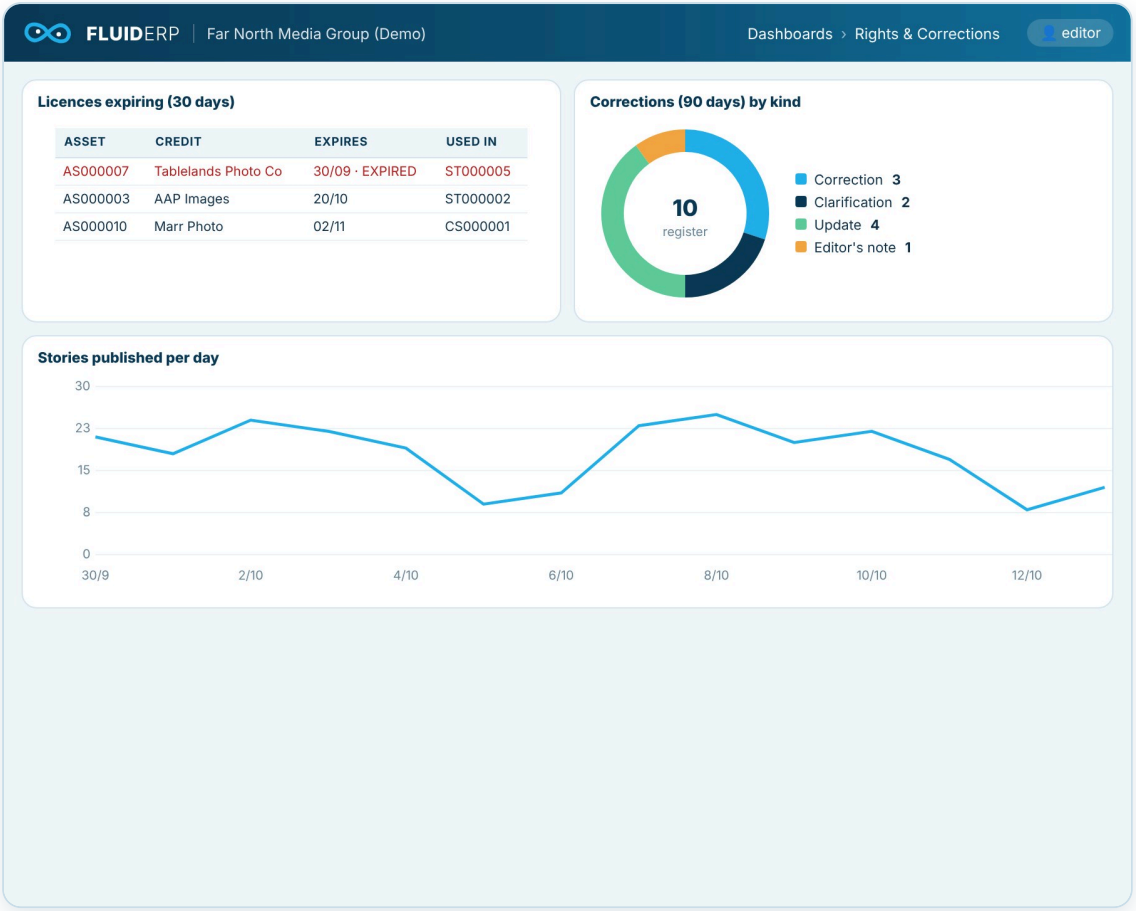
Pipeline
Stories by desk and status.
`nrstory`

Desk output
Words this week by desk.
`Σ nrstory.words`

Rights and corrections

Where **Dashboards › Rights & Corrections** Who **editor, picture desk**

Licences expiring, corrections by kind and stories published per day.



Rights and corrections

What each tile is and where it comes from

Licences expiring
Assets whose licence ends within 30 days or has ended.
`nrasset.licence_expiry`

Corrections by kind
The append-only register.
`nrcorrection.kind`

Commercial

Where **Dashboards › Commercial** | Who **commercial**

Syndication income and contributor spend from the ledger the buttons posted to.



Commercial

What each tile is and where it comes from

Syndication
Invoiced licences this month.
`Cr 4400`

Contributor fees
Paid fees this month.
`Dr 6600`

Licences by licensee
Share of income.
`nrlicence`

11 Reports

Eleven newsroom reports ship with the edition, together with the adopted service-desk reports.

Running and printing a report

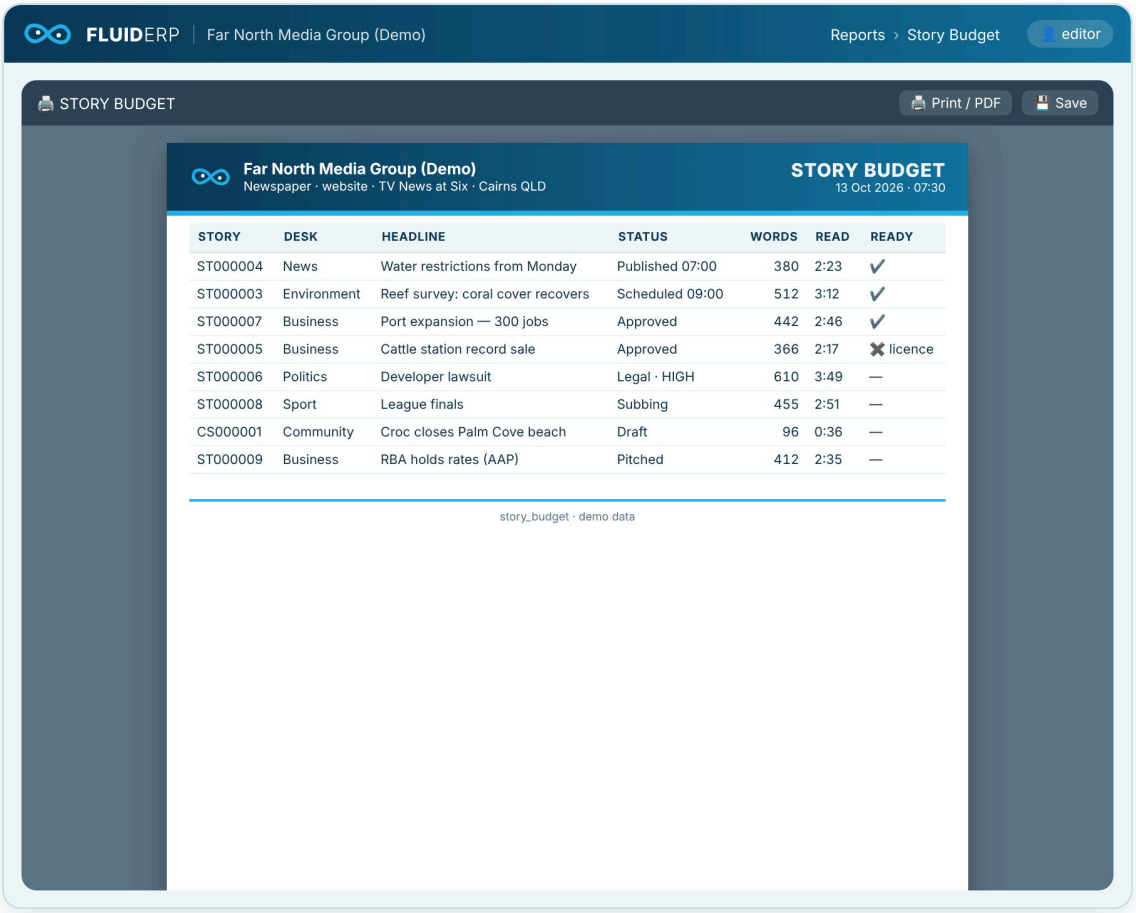
1. Open 🖨️ **Reports** on the launcher (or the report's own tile) and pick the report.
2. Fill the parameters. Required parameters are marked; dates accept `$today`-style defaults and lookups show “value — description”.
3. Press ▶ **Run**. The report draws its shape — grid with subtotals, chart, pivot, 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
Story Budget	The morning budget: desk, status, words, read time, ready	editors
Pipeline Heatmap	Desk × status	editors
News Map	Stories on the map	editors
Embargo Calendar	Scheduled releases	editors
Desk Output	Words and stories by desk	editors
Corrections Register	All corrections and clarifications	public, editors
Rights Expiring	Licences ending	picture desk
Wire Inbox	Items received and taken	sub-editors
Bulletin Rundown	The running order with times	producers, gallery
Contributor Fees	Fees by contributor	commercial
Syndication Revenue	Licences invoiced	commercial

Story budget

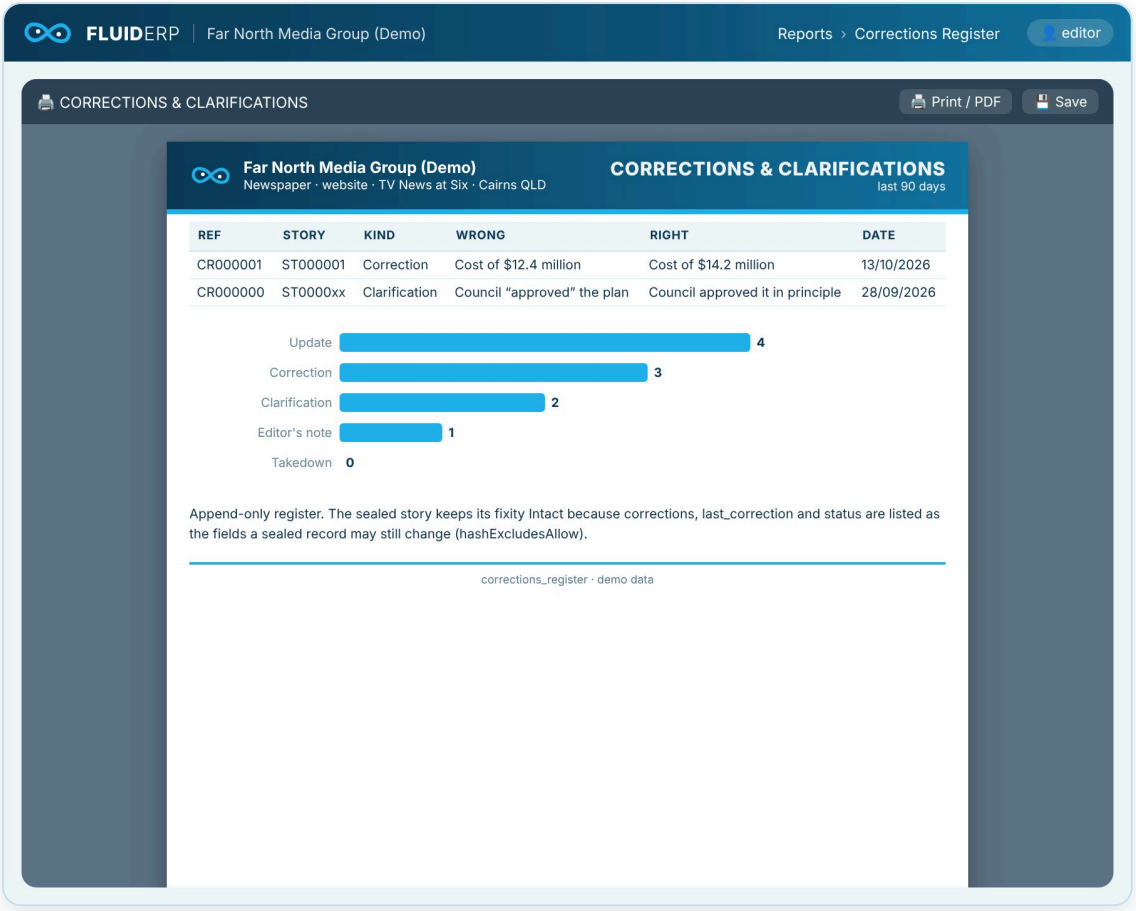
The page the news conference works from.



Story budget

Corrections register

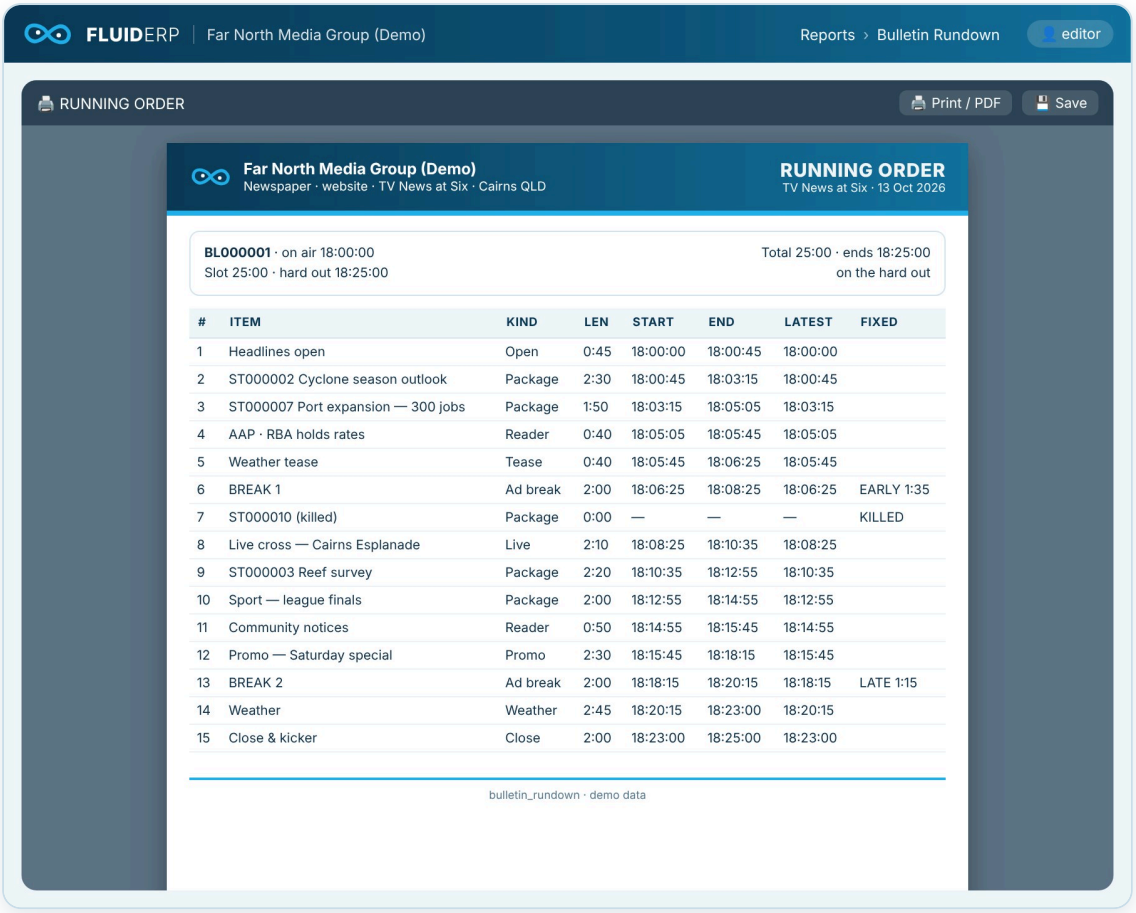
Append-only: what was wrong, what is right, who raised it and when — the newsroom's public accountability.



Corrections register

Running order

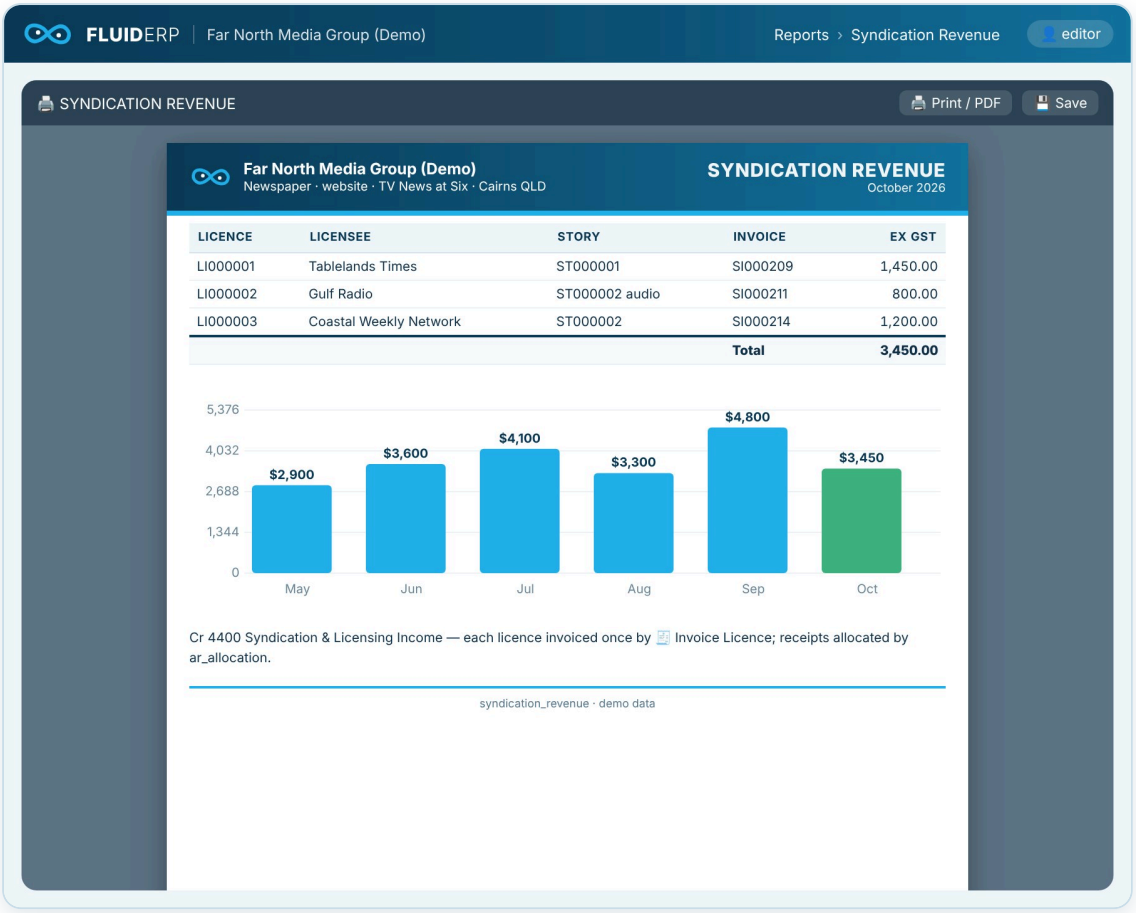
The rundown as the director and presenter see it, every time from the sequence model.



Running order

Syndication revenue

Licences invoiced by licensee, from the sales invoices the button wrote.



Syndication revenue

12 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.

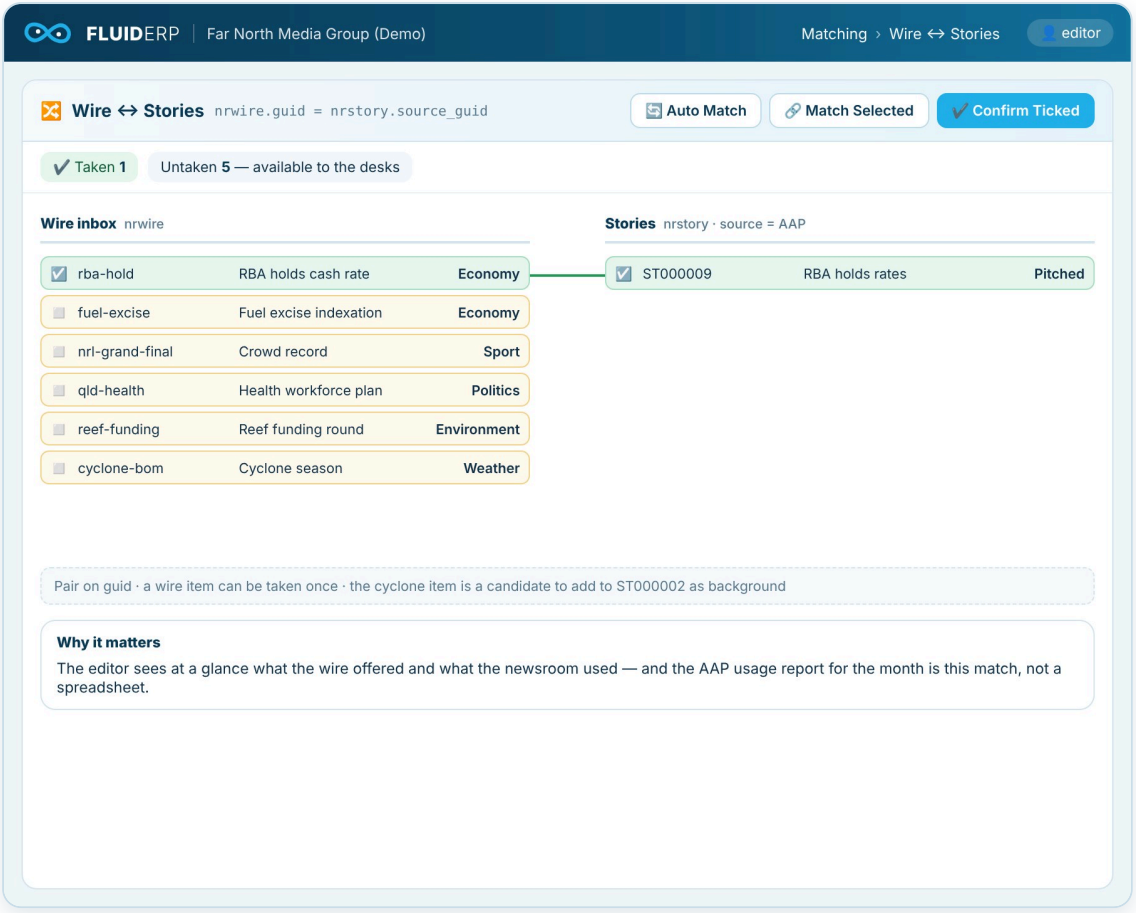
What ✔ Confirm Ticked writes	matching definition · confirm
✓ `sysmatch`	the session header — definition, who, when, number of groups (MAT number)
✓ `sysmatchlines`	one line per row in each confirmed group
✓ `each side`	the stamp field (for example matched_ind = M) and any counterpart fields the definition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

Wire items against stories

Who **editor** | Where **Matching › Wire ↔ Stories**



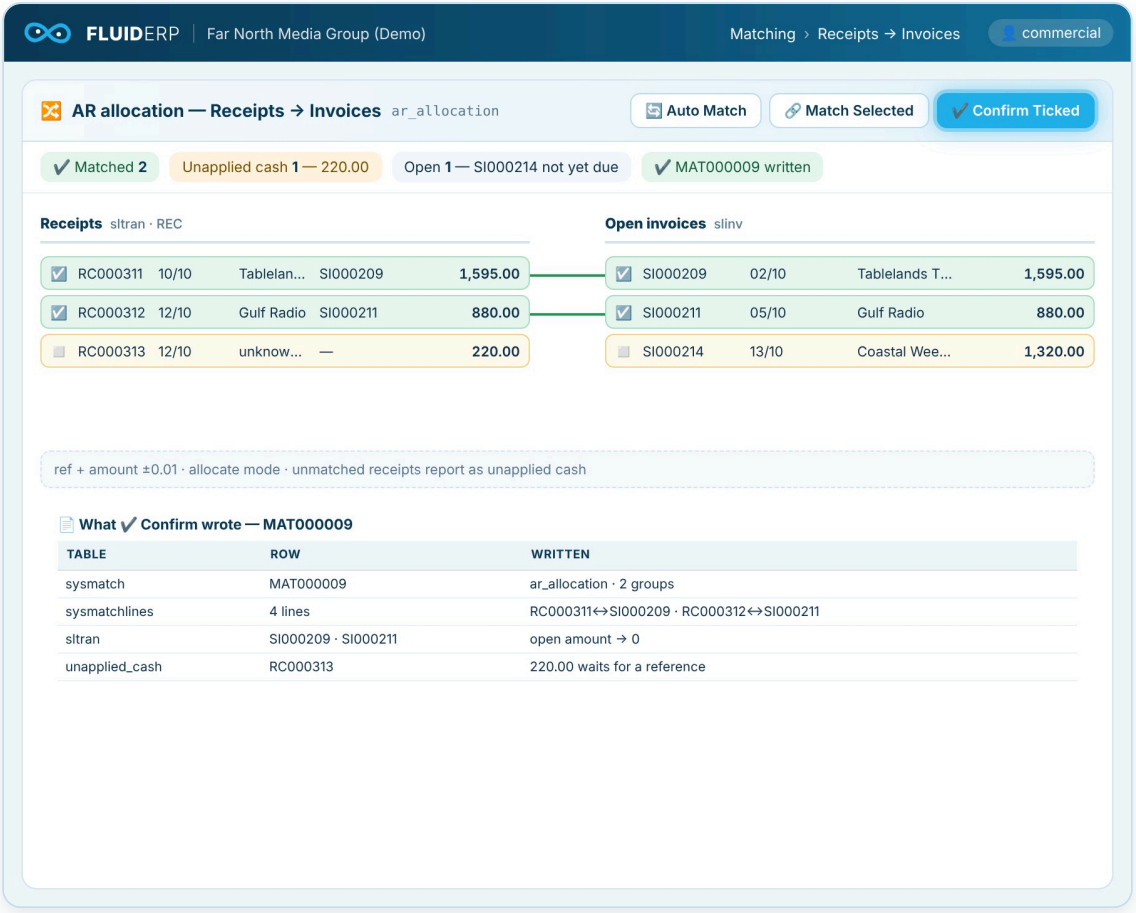
One item taken, five untaken; no story exists twice.

1. Pairs wire items and stories on the guid (written back by 📥 Take).
2. Untaken items are available to the desks; the match is the agency usage report for the month.

> **NOTE**
> This pairing is on the guid field written by Take; it is presented in the matching layout for review.

Allocate syndication receipts

Who **commercial** | Where **Matching › AR allocation**



Two receipts pair with licence invoices; one EFT with no reference waits as unapplied cash.

1. Import the bank receipts.
2. Open **AR allocation**; receipts pair with open invoices on reference and amount.
3. Press ✔ **Confirm Ticked**; unmatched receipts report as unapplied cash.

What ✔ Confirm Ticked writes	ar_allocation
✓ **sysmatch · sysmatchlines**	the session and its lines
✓ **sltran**	invoices' open amount → 0

13 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
Embargoed until … — cannot be Published before then, by anyone	The release block binds every user, the administrator included.	Let the worker publish at the time.
Clean Gate has not passed this story	Text or pictures not yet validated.	Fix what the gate reports; the worker will not publish it.
Sealed — the body cannot be changed	Published stories are sealed for seven years.	Add a correction.
A reason is required — the signature will be superseded	You changed a signed story.	Give the reason; the editor re-signs.
Not ready — expired licence	A picture's licence has ended.	Replace or renew the picture.
Wire item already taken	Take is one-way.	Open the existing story.
A DTD is refused	Capture does not expand XML entities.	Ask the agency for a file without a DTD.
Fee or licence is locked	Posted fees and licences cannot change.	Raise a credit note.

14 Data reference and glossary

Tables you will meet

TABLE	HOLDS	WRITTEN BY
nrdesk	Desks and beats	editor
nrstory	Stories — workflow, signature, release, seal	everyone in the newsroom; 📥 Take; the release worker
nrcorrection	Corrections and clarifications (append-only)	sub-editors, editors
nrwire	Wire inbox	capture definitions; 📥 Take stamps it
nrasset	Pictures, video and audio with rights	picture desk
nrbulletin · nrrunitem	Bulletins and rundown items	producers; ⏱️
nrfee	Contributor fees	commercial; 💳
nrlicence	Syndication licences	commercial; 🧾
tblfeed · tblcapture	Feed and capture definitions	administrator
prtpitch · prtnews	Contributor filings; published news in the app	the app; bridges

Glossary

TERM	MEANING
Budget	The list of stories planned and in progress.
Embargo	A release time before which nobody can see or publish the story.
NewsML-G2	The IPTC XML format agencies exchange news in.
Rundown	The ordered running order of a bulletin.
Hard out	The time the bulletin must end.
Back-timing	The latest each item may start so the show ends on the hard out.
Syndication	Licensing your stories to other publishers.
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.

TERM	MEANING
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 shipped Far North package at 07:30 on 13 October 2026. Its records differ slightly from the film's examples — use these steps on the installed system.

Embargo (editor)

1. Stories › ST000003 (Scheduled 09:00): try Status → Published — refused “embargoed until 2026-10-13 09:00 … by anyone”.
2. Sign in as commercial: ST000003 is not in the list at all. reporter (its author) sees it.
3. As administrator: POST /api/admin/release/run (dry run lists ST000004), then ?dryRun=false → Published, stamped, sealed; GET /api/records/verify/nrstory/ST000004 → Intact.

Signature, seal, correction (editor)

1. ST000006 (Legal, High) → Approved: re-enter the password; the ✍ Signatures tab shows the meaning.
2. ST000001: even the administrator cannot change its body; CR000001 is its correction; it still verifies Intact.

Wire and clock (sub_editor, producer)

1. Wire Inbox: five AAP items; capture samples/aap-newsml-2026-10-13.xml → the sixth (Federal Court / Mareeba); the taken flag on the first is kept.
2. 📥 Take on the Qantas item → a Pitched story.
3. producer: BL000001 → ⏱ Time the Rundown: 25:40 vs 25:00 — cut 0:40; break 1 1:35 early, break 2 1:55 late. Trim the seawall package to 60 and re-time → 24:10, fill 0:50.

Commercial

1. 💳 Pay Fee on FE000001 → purchase invoice $150 + GST, Dr 6600.
2. 🧾 Invoice Licence on LC000001 → sales invoice $180 + GST, Cr 4400.

For the administrator

TASK	HOW
Deploy	`./deploy-newsroom.sh "<staff URI>" "<contributor URI>"` ; re-import tblfeed and tblcapture with the package.
Release worker	Schedule POST /api/admin/release/run?dryRun=false every minute, then the bridge sync so Latest News follows.
Feeds	GET /api/feed/headlines (RSS), /atom, /jsonfeed, /syndication (NewsML-G2).