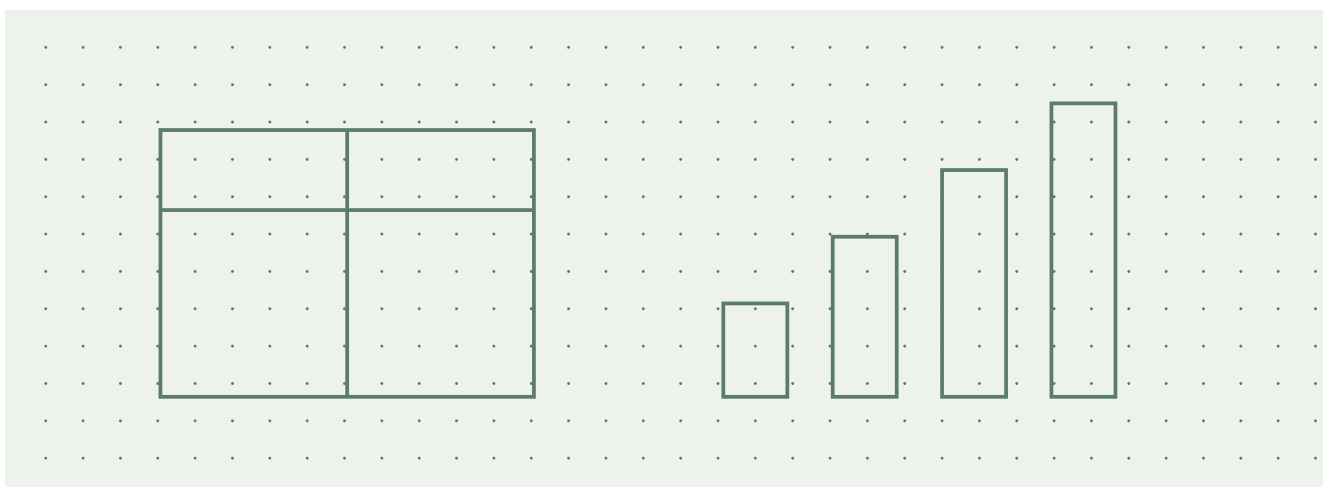


SYSTEM 03 / COMMERCE OPERATIONS

Shop Desk

Less chasing orders.
More growing the shop.



BRING

Orders and available stock

LEAVE WITH

A shop operations plan

A practical tool, a repeatable routine, and a clear handoff. Start with the fictional example, prove it against a small set of your own records, then make it part of the day.

START HERE

From the first run to a useful routine.

- 01. Try the working example on the system page. Use the example date in the runbook so you can compare the result.
- 02. Bring a small current export. Keep the column headers, confirm the mapping, and reconcile the result with its source.
- 03. Download the brief and queues. Review each proposed action. Keep the original records current as work is completed.
- 04. Install the standalone kit if you want repeatable local runs. Use the common runner for dated history and the scheduling guide for your chosen environment.

Your installation record

Client / team

Daily operator and backup

Source records and refresh time

Review cadence and timezone

Private output location

Pilot dates and agreed success criteria

THE OPERATING RUNBOOK

Outcome: a shop owner starts the day with a reviewed shipping queue, a short stock action list, and a creator outreach decision for each product.

Operating owner: commerce operator. **Backup:** owner or operations lead. **Input owners:** fulfillment lead and inventory lead. **Working cadence:** once each morning, plus a refreshed run before the day's final carrier pickup during campaigns.

The first client session

1. Open a real order in Seller Center and identify its actual dispatch deadline, timezone, fulfillment owner, and status. Agree on the distinction between a dispatch deadline and a delivery deadline.
2. Export a small, complete set of relevant orders using the account's available order-export function. Preserve IDs as text and remove unused customer information.
3. Map the file to `order_id`, `sku`, `quantity`, `status`, `ship_by`, `source`. Use the deadline shown for each order. If an export does not contain that deadline, copy it from order details or use a separately verified source; never infer it from purchase date plus a blanket rule.
4. Confirm which identifier will be shared by orders and inventory: Seller SKU or TikTok SKU ID. Choose one consistently.
5. Obtain stock and seven-day unit sales. Confirm that stock means available-to-sell **after** reservations. Consolidate warehouse rows without double-counting shared inventory. Agree on supplier lead time and buffer for each SKU.
6. Run the sample first. Then run a small live snapshot locally and compare at least five order lines and three SKU calculations with the source, or every item if the sample is smaller.
7. Assign an action owner to each urgent shipping line and each stock risk. Review one creator draft together; complete the placeholders, product facts, sample budget, recipient, and commercial terms before any message is sent.
8. Record the input timestamp and operator's approval in the operating log below. Refresh the order state before taking action in Seller Center.

See README.md (./README.md) for the complete mapping contract, formulas, supported statuses, and official policy sources. Start with seller-managed physical-product workflows the client understands; account-specific fulfillment modes and split orders may need separate operational review.

Daily operation: capture to review to act

1. Capture fresh inputs

Create a dated input folder. Save normalized UTF-8 `orders.csv` and `inventory.csv`, plus a short `snapshot.txt` containing export time, timezone, filters, source file names, and the person who prepared them. `snapshot.txt` is an operator record; the engine does not read it.

Completeness check: ensure date filters and order tabs cover the intended queue. Verify that split orders have not been omitted by an export path. A header-only file produces a warning, not evidence of an empty shop.

2. Generate the operating pack

Use the browser tool or run from the repository root:

```
node automations-library/systems/shop-desk/cli.mjs \
  --orders /path/to/client/inputs/orders.csv \
  --inventory /path/to/client/inputs/inventory.csv \
  --output /path/to/client/reports/2026-09-07 \
  --as-of 2026-09-07T09:00:00-04:00
```

Replace every path and timestamp. A repeated run replaces the four report filenames in the selected output folder. Use dated folders to retain history. Invalid input exits with a nonzero code before writing new reports; never present an older report as today's successful run. Each file is replaced atomically, but the four files are not a transactional set: if a storage failure interrupts the run, fix it and rerun before sharing the pack.

3. Review in this order

1. **Data warnings:** resolve missing inventory, held/unpaid lines, empty exports, and input freshness before concluding anything is safe.
2. **Overdue and due-now dispatch:** confirm the current order status; assign the fulfillment owner and pickup/scan follow-through. The report does not establish a policy violation or automatically cancel orders.
3. **Due-soon dispatch:** verify allocated stock and carrier capacity; track through the actual dispatch deadline.
4. **Stock risks:** verify inbound orders, supplier lead time, minimum order quantities, and planned campaigns. Convert a reviewed suggestion into a purchase request only through the client's normal approval process.
5. **Creator brief queue:** hold risky SKUs. For covered products, choose an actual creator already known to the team or found through the client's normal research process. Replace placeholders with verified facts and agree on samples, terms, disclosure, and timing before sending.

4. Close the loop

Record action owners and next check times in the client's existing task system. Re-export and rerun to confirm changes; editing the generated brief does not change the store. Archive reports under the client's agreed retention policy and remove outdated copies from shared devices.

Acceptance criteria

Go live only when the client can demonstrate all of these:

- A fresh, complete export can be prepared by both the owner and the backup operator.
- Every reviewed open line agrees with its order-specific dispatch deadline and timezone in Seller Center.
- Shipped/cancelled lines do not appear in the dispatch queue. Held/unpaid lines appear as review warnings.
- Duplicate order/SKU or inventory rows stop the run; the operator knows how to reconcile them.

- Stock is confirmed to exclude existing reservations, and the operator can reproduce the target and cover calculation for three SKUs.
- Zero sales produces unknown cover. Stock-risk products receive hold briefs, not outbound creator promises.
- A Markdown brief and both CSV queues can be downloaded or generated and opened.
- No message, inventory purchase, or fulfillment change occurs from pressing "Build my shop brief."
- A refreshed snapshot confirms at least one completed operating action; the backup operator can repeat the workflow.

During the first week, record preparation time, review time, stale-source incidents, urgent-line ownership, and false-positive stock flags. Set a client-specific time-saving target after measuring the baseline. The product does not promise revenue or compliance outcomes.

Scheduling the prepared workflow

The CLI is ready to run repeatedly, but **no schedule is installed by this package**. A schedule is only useful after someone or an authorized connector reliably supplies fresh input files.

For a local automation-capable assistant, use this setup prompt after replacing all bracketed values:

Every weekday at [time] in [timezone], review the Shop Desk input folder [absolute path]. Read snapshot.txt and confirm that both normalized CSVs belong to the intended current snapshot and contain the expected filters. If they are missing, stale, incomplete, or ambiguous, report the specific missing input and do not reuse old reports as a new result. Read automations-library/systems/shop-desk/RUNBOOK.md. If this environment can execute local Node.js and access these files, run the documented CLI with the actual snapshot timestamp and a new dated report directory. Check its exit status, read report.json, and surface overdue or due-now lines, stock risks, and data warnings with source references. Return the brief and report paths for operator review. Stay quiet when the supplied snapshot and actionable results have not changed. Do not send messages, purchase inventory, alter order status, or infer missing deadlines. If this environment cannot access local files or execute the CLI, ask for the exported report or fresh files in the task; do not claim to have run it.

This prompt can be adapted to a Claude scheduled task or another assistant with the necessary runtime and file access. A scheduled chat is not automatically a local file runner or Seller Center connector. Verify those capabilities in the actual client environment before promising unattended operation.

For an initial low-friction rollout, create a daily reminder to upload the two fresh files into the browser workbench. Upgrade to unattended input delivery only after a specific authorized adapter and its failure handling are implemented.

Failure and recovery

Symptom	Operator response
Missing column	Rename using the mapping sheet. Do not guess at a similar field.
Timestamp rejected	Copy the actual deadline and append its verified timezone offset or Z. Check daylight-saving time.
Duplicate order/SKU	Remove repeated exports or reconcile legitimate split lines against current remaining quantity, state, and earliest deadline.
Duplicate inventory SKU	Consolidate inventory across warehouses once; avoid summing the same shared stock multiple times.
Unknown status	Check its meaning in Seller Center and map it deliberately; do not label it "shipped" to bypass validation.
No inventory for an order SKU	Check the identifier mapping and add a verified inventory row; confirm that the order already has an allocation.
Reorder suggestion looks too high	Review campaign effects, the sales window, supplier lead time, inbound stock, and buffer.
Everything suddenly overdue	Check as-of, per-order timezone, source freshness, and whether shipped statuses were refreshed.
CLI failed or output incomplete	Preserve the error, correct the source or storage problem, and rerun. Do not circulate an older report as a fresh one.

Operating log template

Record for each run: Run; Snapshot time / zone; Source files and filters; Operator; Urgent lines / owner; Stock decisions / owner; Creator briefs reviewed; Next check.

Client handoff package

Deliver this runbook, README, engine/CLI, browser access or deployment instructions, blank schema headers, fictional sample files, one approved client report, the named owner and backup, and a recorded mapping walkthrough. Agree separately on where client inputs and reports live, who can access them, and how long they are retained.

Policy maintenance

Official source review: **September 7, 2026**, US market. Policy context is documented in the README with direct links to TikTok's Fulfillment Policy and order-export guidance. Recheck those sources and the client's order UI at every rollout and after a fulfillment-policy change. The software deliberately uses per-order supplied deadlines instead of storing a global TikTok dispatch SLA.

Working tool: outerscope.ai/automations-library/shop-desk

Website links refer to the intended published routes. A local preview is available at the same path on your development server; publication is a separate step.