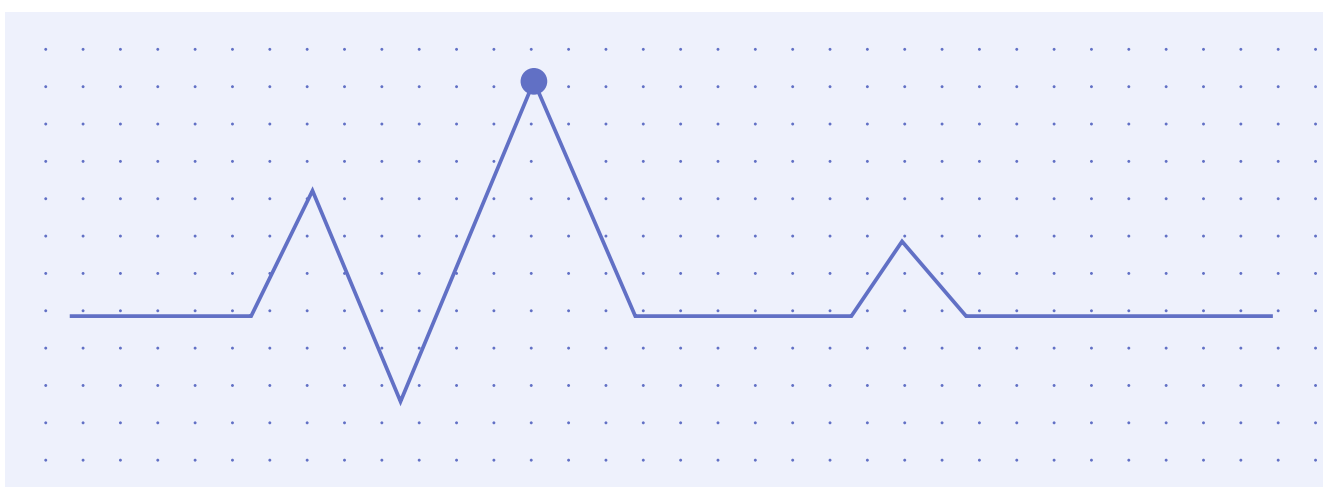


SYSTEM 01 / BUSINESS OPERATIONS

Morning Brief

Know what needs you.
Before the day does.



BRING

Your operational checks

LEAVE WITH

A prioritized action brief

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

Client outcome: one short, evidence-backed daily brief that shows which operating checks need attention and makes the owner and next step explicit.

Accountable role: operations lead. **Backup:** founder or a named delegate. **Operators:** the people who export source reports and the person who runs the brief. Assign actual names before rollout.

Deliverables: a reviewed check dictionary, a daily CSV, a Markdown brief, a CSV action queue, and an archive of dated reports. This package includes the processing tool and example data. Client source exports and delivery channels are configured during installation.

1. Choose the smallest valuable pilot

Pick 5-10 checks that somebody already reviews and can act on. Good candidates are old support tickets, leads waiting for a first response, invoices awaiting follow-up, deliveries at risk, failed jobs, and a founder's unclosed commitments.

Do not start with every available metric. A check needs a decision: "If this is outside target, who will do what?" Give every check an owner, a source report, the measurement period, units, a target, a direction, and a freshness limit. A weekly check may need a 168-hour limit; a dispatch queue may need two hours. Include these units and periods in the check name or the client's accompanying metric dictionary.

2. Map inputs once

1. Export the current source reports into a client-controlled folder. Keep only the aggregate values needed for the check.
2. Map each reading into `sample.csv`'s columns. For example, a support export's count of open tickets becomes `current`; the client's agreed ceiling becomes `target`; `direction` is `lower`.
3. Supply `updated_at` from the actual measurement/export time. Do not replace an old measurement's timestamp with the time you copied it.
4. Put a report name or file reference in `source`. No source is fetched by this tool.
5. Use raw numbers. Represent 91% as 91 in both `current` and `target`, and label the metric with (%) consistently. Do not mix percentages with fractions.
6. Leave unknown readings blank. The brief will flag them. Never replace missing data with zero.
7. Have the operations lead approve targets and P1 gap thresholds. A zero-target breach is P2 because percentage math is undefined; explicitly agree on how the team will manually escalate those checks.

Store the template, metric dictionary, and owner roster beside the CSV. Retain the original source exports for the client's agreed retention period. Avoid raw customer names, message content, and sensitive records when aggregate counts are sufficient.

3. Run a 15-minute pilot

Minutes 0-3: run the bundled example at 2026-09-07T09:00:00Z. Confirm six checks and five actions, four of them P1. Download both formats.

Minutes 3-7: replace the sample with five real aggregate checks. Use a current UTC report cutoff and review timestamps and freshness limits.

Minutes 7-11: run the brief. Compare at least one healthy check, one target breach, and any data-quality finding to the source exports. If there are no breaches, temporarily test a separate copy with a known threshold breach and restore the approved values afterward.

Minutes 11-15: read the top three findings aloud with the decision owner. Confirm the evidence, choose one next step per applicable finding, and copy those approved steps into the client's existing task system. Save the brief to a dated folder.

The pilot passes when the operator can explain every finding using the supplied evidence and the owner knows what to do next. A correct report with no exceptions also passes; never invent exceptions to make a report interesting.

4. Daily operating loop

1. **Before the agreed cutoff:** owners refresh source exports; the operator updates the CSV and records the actual measurement timestamps.
2. **At cutoff:** run the workbench or CLI with an explicit UTC --as-of. Save each run in a new dated folder.
3. **Triage:** fix missing/stale evidence first. Re-run after correction. Review target breaches against the source before choosing an action.
4. **Assign:** every approved action gets a person, a due time, and a task-system record. The exported queue intentionally leaves those workflow decisions to the operator; it does not create tasks.
5. **Distribute:** after review, paste the Markdown brief into the existing team channel or email draft. Sharing is manual; this tool sends nothing.
6. **Close the loop:** on the next run, inspect yesterday's action records. A recurring exception should have an explicit next step or an approved target change. The tool itself does not track completion across runs.

Keep the brief to what needs a decision. The complete check register remains in the downloaded report for traceability.

5. Optional scheduled run setup

Scheduling is not installed or enabled by this package. A local scheduler or a Claude environment that actually has access to the local checkout, Node, and the input folder can run the CLI. Verify those capabilities in the chosen environment first. A browser-only chat cannot be assumed to run local commands or read local files.

Use this job specification when creating the client's schedule:

Setting	Client choice
Time	Agreed daily cutoff and named timezone; record the UTC equivalent for each run.
Input	Explicit path to the refreshed UTF-8 CSV.
Command	<code>node <repo>/automations-library/systems/morning-brief/cli.mjs --input <checks.csv> --output <new-dated-folder> --as-of <ISO-UTC-cutoff></code>
Success	Exit 0 and all three reports exist. Findings are a successful result, not an execution failure.
Failure	Nonzero exit, missing input, or missing outputs: stop and notify the operator through the separately authorized channel.
Retention	Client-defined archive and deletion policy for CSVs, exports, and reports.
Permissions	Read the input; write the report folder. No permission to send messages, update source systems, or change targets.

Prompt template for a scheduled Claude task

Prepare our Morning Brief at [TIME AND TIMEZONE]. Use the local project at [REPOSITORY PATH], the operational CSV at [INPUT PATH], and a new dated output folder below [REPORTS PATH]. This task requires local file and command access; if this environment lacks either capability, explain what setup is missing.

Check that the input exists. Run the Morning Brief CLI with the current run's explicit ISO UTC cutoff and a new output folder. Do not replace the measurement timestamps, invent missing values, change targets, or overwrite past reports.

Read report.json and report.md. Prepare a short draft of the P1 findings followed by the most useful P2 findings. Keep metric names, values, owner names, source references, and uncertainty faithful to the generated report. Distinguish missing/stale evidence from a verified target breach. If the CLI fails, return its error and the smallest repair the operator can make; do not write a fake successful brief.

Save the generated artifacts and return their paths. Leave sharing and task creation for human review. Do not send email or Slack messages or change source systems. No exceptions in the supplied checks is a valid result; say that and state that systems absent from the CSV were not assessed.

Fill the placeholders and verify one manual run before saving a schedule. Test one deliberate stale-data case, one missing-file case, and one successful scheduled run. Confirm daylight-saving-time behavior in the chosen scheduler. No schedule is considered live until the operator has observed its first run.

6. Incident handling

Situation	Operator response
Invalid CSV or nonnumeric reading	Correct the named row/column using the original export. Re-run; do not coerce unknown values to zero.
Input file absent or access denied	Restore the export/access path. Do not substitute the bundled example for client data.
Stale or future-dated reading	Confirm source export time and timezone. Refresh from the source, then re-run.
Unexpected P1 target gap	Verify units, the measurement period, target direction, and raw source before escalating.
Source or owner missing	Assign the source reference or accountable person, then update the mapping.
Repeated noisy finding	Review whether the threshold reflects a real decision. Only the accountable client owner approves target changes.
Report cannot be written	Choose a writable new output folder. Existing reports are protected; use --force only when intentionally replacing that run's output.
No report at the daily cutoff	Operator runs manually, records the missed run, and repairs the separately configured scheduler.

The brief is an operating aid. An actual outage or critical business event follows the client's existing incident process; do not wait for this report to confirm it.

7. Rollout success criteria

- All pilot checks have approved definitions, units, targets, sources, and named owners.
- Every reported reading can be traced to a client export; no missing reading is silently treated as zero.
- A seeded breach, stale reading, missing value, and future timestamp are correctly surfaced before rollout.
- Five consecutive operating days produce dated reports, with missed runs recorded and repaired.
- The operations lead completes daily review within the agreed time budget; measure it during the pilot rather than promise time savings.
- Every approved P1 action has a person and due time in the client's existing task system.
- The client can run, inspect, export, and troubleshoot the package without Outerscope present.

8. Handoff and limitations

Hand over the tool folder, client CSV template, metric dictionary, approved thresholds, owner roster, observed pilot reports, and any separately configured scheduler instructions. Conduct one operator-led run during handoff.

The engine evaluates one supplied snapshot per metric. It does not connect to dashboards, authenticate, ingest exports automatically, discover anomalies, compare historical runs, forecast, identify causes, verify source truth, send messages, manage tasks, or prove business health. Changing --as-of reinterprets freshness; it does not retrieve historical data. A report with no exceptions says only that the checks supplied were fresh and within their configured targets.

For a wider rollout, add verified data extraction and delivery as explicit client-specific work, keep this engine as the inspectable decision layer, and re-run the acceptance tests whenever its rules change.

Working tool: outerscope.ai/automations-library/morning-brief

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.