PLAYBOOK · USER GUIDE

GTM AI OS

From the webinar to a running system

Install one plugin, type one line, answer what it asks. Everything you need to take the four OS plugins from zero to a scored, governed, learning loop on your own pipeline.

Version 0.2 · September 2026
Jonathan "Coach K" Kvarfordt · therevenueaireport.com

Covers: `gtm-ai-os-core` · `os-know` · `os-decide` · `os-act` · `os-learn`

Contents

1. Quick Start: eight steps, eight prompts
2. What you're building
3. Tools to have ready
4. Install the core
5. Sign the fence
6. KNOW: win/loss and the Account Context
7. DECIDE: score the list
8. ACT: draft, evaluate, approve
9. LEARN: the weekly review
10. The rhythm
11. When something goes wrong
12. Glossary and first-week checklist

> The OS asks questions on purpose. Answering them is the work. It never sends anything on its own.

1 · Quick Start

If you do nothing else, do this. In this order.

1

Add `gtm-ai-os-core.plugin` (Customize → Browse plugins → upload)

```
› Set up my workspace.
```

When it asks for folder access, select the folder in the picker and wait for the confirmation. When it asks which track, answer: **GTM AI OS**.

2

No install. Sign the fence.

Open `GTM-AI-OS/context/fence.md`, read the table, and put your revenue leader's name and date on the `signed by:` line.

3

Install `os-know.plugin`

```
› Run win loss.
```

When it asks where your closed deals live, name the tool, upload the export, or paste the list. Then, for your best open account:

```
› Build the account context for [account name].
```

Repeat for three to five accounts.

4

Install `os-decide.plugin`

```
› Score my accounts.
```

When it asks where your list lives, name it, upload it, or paste 10 to 50 accounts.

5

No install. Fill the proof register.

Open `GTM-AI-OS/context/messaging.md`. Write your thesis, tone rules, and every proof point you're allowed to cite.

6

Install `os-act.plugin`

```
› Work this account: [account name].
```

Read the passed draft. If you approve, set `draft status: approved` in the file and send it from your own tools. The OS never sends.

7

No install. Record the result.

When you get a reply or a meeting, add it to the account record's Touches and Outcome sections.

8

Install `os-learn.plugin`

```
› Weekly review.
```

Every Friday. Say yes or no to each proposal. Nothing changes without a yes. Once a month:

```
› Graph report.
```

ANY TIME YOU'RE LOST

```
› What's my status?
```

2 · What you're building

A loop that runs on one folder on your computer.

Step	What happens	Plugin
KNOW	Pull why you win and lose in the buyer's words. Build one record per account.	`os-know`
DECIDE	Score every account on FIT × MOMENT × CONFIDENCE. PAIN decides whether it's ready for outreach.	`os-decide`
ACT	Write outreach the evidence allows, have a separate judge check it, send only with your approval.	`os-act`
LEARN	Every week, review what happened and approve changes so next week is better.	`os-learn`

You do not write code. You answer questions, edit a few text files about your company, and approve what the system proposes.

> **Time to first result:** about 90 minutes to install and score one account. About a week to run the full loop once.

3 · Tools to have ready

Required

1. **A Claude account with Cowork access.** Cowork is the desktop app where the OS runs. Check claude.com for the plan that includes it. If your company provides Claude, ask your admin whether Cowork is enabled.
2. **Claude Cowork installed** on the computer you'll use. Sign in and open it once so it finishes setup. (Using ChatGPT, Codex, or Grok instead? See the table below.)
3. **The zip file** with five plugin files: core plus `os-know`, `os-decide`, `os-act`, `os-learn`.
4. **A folder location.** The OS creates `GTM-AI-OS`. Your Desktop is the default. Avoid a folder other tools constantly reindex.

Your data, in any tool

The OS never assumes which products you use. It asks "where does this live?" and takes whatever you name.

Have ready	What it needs	Minimum
Accounts and deals	18 to 24 months of closed-won and closed-lost with outcome, segment, size, notes or close reasons. CRM, spreadsheet, or export.	50 deals workable; 20 is the floor (marked Inferred)
Calls	Loss and churn transcripts or notes from any call tool	Optional, strongly recommended
Email	The mail client you use with buyers. Read only.	Optional
Enrichment or signals	Any provider, alert, or export you already pay for	Optional
Account list	The accounts you want scored, with whatever you know	10 to 50 accounts

How to add a plugin in Claude (from the zip file)

You received a zip. Inside it is a `plugins` folder with six `.plugin` files. Unzip it somewhere you'll find again.

1. Open the **Claude Desktop app** and switch to the **Cowork** tab.
2. Click **Customize** in the left sidebar (plugins, skills, and connectors live there).
3. Click **Browse plugins**. In that window, choose the option to **upload a custom plugin file**.
4. Select `gtm-ai-os-core.plugin`. Confirm. Repeat for the four OS plugins, in the order in section 1.
5. Start a new Cowork session and type **What's my status?** The router lists what it can see.

> Plugins you add this way are saved on your machine. There's no marketplace and no auto-update: when a new version ships, you'll get a new zip and repeat these steps. Nothing in your workspace folder is touched by re-installing a plugin.

IF YOUR COMPANY BLOCKS CUSTOM PLUGINS

On Enterprise plans, an admin may have disabled custom plugin uploads. Ask them to allow it, or use the "any other AI tool" path below with the SKILL.md files as instructions.

Using this with ChatGPT, Codex, Grok, or another AI tool

The plugin files are zip archives. Inside each is a `skills/` folder, and inside each skill is a `SKILL.md`: plain-text instructions plus templates. Those instructions work in any capable AI assistant. What changes is who handles the files.

Tool	How to run the OS there	What you do by hand
Claude Cowork	Install the plugins as above. Claude reads and writes the `GTM-AI-OS` folder on your machine and asks for connector access.	Approve sends and connector writes.
Claude on the web	Install the plugins the same way from Customize. Skills load; the folder does not.	Upload context files at the start of a session; download artifacts at the end.
ChatGPT	Create a Project (or a custom GPT). Paste the core `CLAUDE.md` and the SKILL.md of the step you're on into the instructions. Upload the templates as files.	Keep the `GTM-AI-OS` folder yourself. Upload `context/` files each session; save outputs back to `artifacts/`.
Codex	Point it at the `GTM-AI-OS` folder as the working directory. Put the core `CLAUDE.md` contents into its instructions file (AGENTS.md) and paste the current SKILL.md as the task.	Review every file change before accepting it.
Grok or another assistant	Same as ChatGPT: instructions from `CLAUDE.md` plus the current SKILL.md, templates uploaded as files.	Same: you are the file system.

> **What only works in Claude Cowork:** the router (`gtm-chief-of-staff`), automatic folder reads and writes, connector requests, and the fence check before any external action. In every other tool, you run one skill at a time and you are the fence. For the OS, that means: in any tool but Cowork, run `os-message-evaluator` yourself on every draft, and never send without reading `fence.md` first.

Connectors

When a skill asks where your data lives and you name a tool, it tells you whether a Cowork connector exists and how to enable it. If none exists, you paste or upload. Nothing is blocked by a missing connector.

THE OS WILL NEVER, THROUGH ANY CONNECTOR

Send an email · Book a meeting · Change a CRM stage · Change pricing · Delete history

4 · Install the core

4.1 Add the plugin

Claude Desktop → Cowork tab → Customize → Browse plugins → upload `gtm-ai-os-core.plugin`. Always first; the four OS plugins depend on it.

4.2 Start the setup

```
› Set up my workspace.
```

The `os-install` skill will:

1. Ask where `GTM-AI-OS` should live. Type `default` for your Desktop, or a path.
2. Ask you to grant Cowork access. A folder picker appears. Select the folder or its parent. **Wait for the confirmation.** Skip this and nothing can be saved.
3. Create the structure: `CLAUDE.md`, `INDEX.md`, `context/`, `artifacts/`, `signals/`, `evolve/`, `evals/`.
4. Ask which systems hold your accounts and deals, calls, and email. Name whatever you use. It writes them to `context/stack.md` and tells you which connectors to enable, if any.
5. Ask which track. Answer **GTM AI OS**.
6. Ask three questions about your company: what you sell and to whom, your role and who you report to, your systems. Plain sentences. These become `context/company.md`.

4.3 Check it worked

```
› What's my status?
```

You should see: workspace created, `company.md` and `stack.md` at draft, everything else not started, and "next: install os-know."

4.4 Open the folder once

Find `GTM-AI-OS` and open `context/company.md`. Fix anything wrong in a text editor. You'll edit these files for the life of the system.

5 · Sign the fence

The step people skip and regret.

Open `GTM-AI-OS/context/fence.md`. It's a table of what the OS may do for each bucket of accounts, in three tiers: **Auto** (it just does it), **Approve** (it asks first), **Never**. Below it is the never list.

1. Read every row. Defaults are conservative: research and draft on its own; ask before any send; never book a meeting or change a stage.
2. If you're the revenue leader, put your name and the date on `signed by:`. If you're not, send the file to the person who is.
3. Until it's signed, every skill treats the fence as a draft and will not present anything for external send.

> Do not loosen the fence in week one. The rule at the bottom says how an action earns its way from Approve to Auto. Let the data earn it.

6 · KNOW: win/loss and the Account Context

6.1 Install `os-know.plugin`

6.2 Run win/loss

```
› Run win loss.
```

1. Asks where your closed deals live. Name the system, say "upload" and drop the export in, or paste.
2. Pulls 18 to 24 months: won, lost, disqualified, churned. Reports the sample per group. **Under 20 in a group is marked Inferred.** Honest, not a failure.
3. Asks for loss and churn transcripts or notes. Give what you have.
4. Extracts reasons in the buyer's words, quoted where possible, each tagged Verified, Inferred, or Unknown.
5. Proposes buying signals and, for each, shows wins with it, losses with it, win share, base rate, sample. **Only signals that beat the base rate with a sample of 5 or more enter the library.** The rest go to "tested and dropped."
6. Writes your ICP FIT band and disqualifiers.

YOU'LL HAVE `context/icp.md`, `context/deal-history.md`, `context/signal-library.md`, `artifacts/os-icp.md`

CHECK

Open `signal-library.md`. Every row has win-vs-loss numbers and an expiry (default 45 days). No numbers, no row.

6.3 Build your first Account Context

```
› Build the account context for [account name].
```

1. Scores FIT against your ICP, 1 to 5, with reasons.
2. Gathers evidence in order: your notes, calls, and email with the account; then enrichment or web for leadership changes, funding, hiring, tool switches, public statements. Every fact gets six fields: source, source date, event date, expiry, contradicts?, real or synthetic.
3. Tags every fact. **Verified** only with a named source. **Inferred** for patterns. **Unknown** written out instead of guessed.
4. Scores PAIN, MOMENT, CONFIDENCE.
5. Writes one hypothesis: "We believe [account] will [act] because [verified pain] and [moment]."
6. Writes "Why we might be wrong" with at least two real reasons, and lists contradicting evidence above it. **If it can't write the doubt, it refuses to mark the record complete.** That's the system working.
7. Fills the persona map: workflow owner, economic buyer, technical owner.

8. Proposes any new signal for your library. You decide.

> Build three to five records before scoring. The scorer calibrates on real ones.

7 · DECIDE: score the list

7.1 Install `os-decide.plugin`

7.2 Score

```
› Score my accounts.
```

1. Asks where your account list lives. Name it, paste it, or upload it.
2. Scores FIT, MOMENT, CONFIDENCE, PAIN using `readiness.md`, the single source of truth for every threshold. Expired signals don't count. Demo data caps CONFIDENCE at 1.
3. Priority = FIT × MOMENT × CONFIDENCE (max 125). PAIN is not in the formula; it decides readiness.
4. Every account gets exactly one bucket, first match wins.
5. Shows the table with the threshold that decided each account and the message path allowed. Explains any account you expected higher.
6. Creates draft records for Action Priority accounts that lack one.
7. Offers to write bucket and priority to a CRM field you name, with approval. Never stage.

Bucket	Rule (applied top to bottom)	What's allowed
Deprioritize	FIT ≤ 2	Nothing
Action Priority	FIT ≥ 4, MOMENT ≥ 4, CONFIDENCE ≥ 3, PAIN ≥ 4	Draft; send with approval
Research Priority	FIT ≥ 4 and (CONFIDENCE ≤ 2 or PAIN ≤ 3)	Research and update the record
Monitor	Everything else	Weekly refresh; nurture on Auto

CHECK

No account in Action Priority with PAIN under 4. If there is, something's wrong.

8 · ACT: draft, evaluate, approve

8.1 Install `os-act.plugin`

8.2 Fill the proof register first

Open `context/messaging.md`. Write your thesis, tone rules and banned phrases, then the proof register: every result you may cite, with source, measurement period, scope, and external-use permission. **The drafter will not use**

a proof point that isn't in this table. Empty register means a question where the proof beat should be, which is correct.

8.3 Run the loop on one account

```
› Work this account: [account name].
```

Step	What happens
Refresh	Re-runs the account context so the record is current
Re-score	If it dropped out of Action Priority, stops and tells you why
Draft	The evidence picks the path. **Confirmed pain (PAIN 4 or 5):** states the problem in their words. **Signal only (PAIN 3):** asks whether the problem exists; never asserts. **Insufficient (PAIN ≤ 2, expired, or synthetic):** no message; a research task. Six beats, three roles, three formats.
Evaluate	A separate judge runs ten checks and writes pass or a failure code. Three rounds maximum, then it hands you the codes.
Fence	Send is Approve for Action Priority. It presents the passed draft and waits.
Log	Records the touch in the account record and the signals log. Append only.

> **Signal-only, in practice:** a new CRO and a CRM rollout support "Does your rollout include call-to-CRM handoffs?" They do not support "You're struggling with CRM hygiene."

8.4 Approve and send, yourself

Read the draft. If you approve, change `draft status` to `approved` in the file and send from your own tools. The OS may place a draft in a connected mail client. It will not press send.

8.5 Record what happens

Reply, meeting, opportunity: add a line to the record's Touches and Outcome sections. Skip this and the system can't learn.

9 · LEARN: the weekly review

9.1 Install `os-learn.plugin`

9.2 Pick a day

Friday morning. Twenty minutes. Put it on the calendar.

9.3 Run it

```
› Weekly review.
```

1. Reads the signals log since the last review and every record updated this week.
2. Links every outcome to the account, the signals cited, the persona, the path, and the draft rounds.
3. Finds patterns: repeated corrections, signals that preceded replies and signals that never did, beats that got replies, deals that contradict the ICP, fields wrong twice, steps you always skip or add.
4. Checks the graph: files nothing read this month, the same fact in two files, a skill that ran with a required file missing.
5. Writes up to seven proposals, each with the file, the exact change, the evidence, the expected effect.
6. Asks you to approve each one. **Nothing changes without a yes.** Skill-file proposals are logged for the next plugin version; the review never edits skills.

9.4 See whether it's learning

```
› Graph report.
```

Once a month. From your own log: evaluator first-round pass rate, override rate, drafts refused for weak evidence, median signal age, records with doubt filled, signals tested against losses, proposals approved per week, dead nodes. Week one against now. If the numbers aren't moving, it says so.

10 · The rhythm

When	Do	Type
Daily, 10 min	Record replies and outcomes in the account records	(edit the file)
Weekly	Score the list; run the loop on Action Priority accounts	`Score my accounts.` then `Work this account: X.`
Friday	The review; approve or reject proposals	`Weekly review.`
Monthly	Graph report; re-run win/loss on last month's closes	`Graph report.` then `Run win loss.`
Quarterly	Re-check fence thresholds; retire expired signals	(edit `fence.md`)

11 · When something goes wrong

"I need access to your GTM-AI-OS folder."
Cowork lost the folder permission. Open the picker and select the folder again.

A skill says a prerequisite is missing.
Two paths: run the earlier skill, or answer three questions for a fast-path version. Fast path is fine for a first pass; do it properly within the week.

The scorer put an account lower than expected.
It names the threshold it missed. Usually PAIN: the pain is Inferred, not Verified. Find a source or accept the bucket.

The drafter refused to write a message.
The evidence is insufficient. It gave you a research task. Do it; don't override.

The evaluator keeps returning the draft.
`proof_missing`: your register is empty for that claim. `signal_stale`: refresh the record. `path_mismatch`: someone edited the path by hand; don't.

I connected my CRM and nothing changed.
Skills ask for access when they need it. Re-run the skill and name the tool.

I want to loosen the fence.
Read "earning autonomy" at the bottom of `fence.md`. Run the graph report. If thresholds are met, the review proposes it. If not, wait.

A context file looks wrong.
Fix it in a text editor. The files are yours.

12 · Glossary

Account Context · one file per account: scores, verified pain with sources, signals, unknowns, hypothesis, doubt, persona map, touches, outcome.

Bucket · Action, Research, Monitor, Deprioritize. Exactly one per account.

Confidence tags · Verified (named source), Inferred (pattern), Unknown (we say so).

Evaluator · the separate skill that checks a draft and can block it. Never rewrites.

Expiry · 45 days for signals, 180 for stated pain, by default.

Fence · what the OS may do per bucket, and the never list. Signed by the revenue leader.

Graph · the workspace as connected files. Every run logs what it read; the review checks the links.

Path · what a message may claim: confirmed, signal-only, insufficient. Decided by PAIN.

Proof register · the only source of proof points, with source, period, scope, permission.

Readiness · PAIN ≥ 4 lead with pain; 3 ask; ≤ 2 research.

Signals log · one line per run: date, skill, files read, produced, what you changed, open question.

Track · the OS (external, this guide) or SCALE (internal AI transformation; shares the workspace).

First-week checklist

- ☐ Claude with Cowork installed and opened once
- ☐ Core installed; "Set up my workspace." completed; folder access confirmed
- ☐ `company.md` and `stack.md` reviewed in a text editor

- ☐ Fence read and signed by the revenue leader
- ☐ `os-know` installed; win/loss run; every signal has win-vs-loss numbers
- ☐ Three to five Account Context records built, each with doubt filled
- ☐ `os-decide` installed; list scored; no Action Priority account with PAIN < 4
- ☐ `messaging.md` thesis, tone, and proof register filled
- ☐ `os-act` installed; loop run on one account; draft passed; you approved and sent it yourself
- ☐ Touch and outcome recorded in the account record
- ☐ `os-learn` installed; first weekly review run; at least one proposal decided
- ☐ Friday review on the calendar