

MANAGING EXPECTATIONS / STAFF & AGENT OPERATIONS

Hermes Self-Improving Knowledge Base Manual

Exact staff and agent playbook for building an Obsidian/LLM Wiki knowledge system around Hermes Agent

Version 1.1 - June 14, 2026

Prepared for Chris Anderson, staff, and approved AI agents.

Source videos: *Build a Hermes Knowledge Base That Self-Improves* and *Every Hermes Concept explained for Normal People* by Jack Roberts.

Purpose: Build a private, source-grounded knowledge system where Hermes remembers stable personal facts, while an Obsidian-compatible LLM Wiki stores the broader operating knowledge of the business.

Public-share version. No credentials, passwords, or private operational secrets are included.

Table of Contents

1. 1. Executive Summary
2. 2. What We Are Building
3. 3. Roles and Responsibilities
4. 4. Required Tools and Skills
5. 4A. Hermes Concepts Staff Must Understand First
6. 4B. Operating Hermes Day-to-Day
7. 4C. Safety, Cost, and Access Rules from the Beginner Tutorial
8. 5. Folder Structure to Create
9. 6. Exact Initial Setup Steps
10. 7. SCHEMA.md Starter Template
11. 8. index.md and log.md Templates
12. 9. Build the Hermes Wiki Skill
13. 10. Source Intake Lanes
14. 11. Exact Ingest Procedure for Staff
15. 12. Exact Query Procedure for Staff
16. 13. Video and Social Media Ingest Procedure
17. 14. Meeting Notes and Calls
18. 15. Email and Inbox Knowledge
19. 16. Documents, PDFs, Scans, and Images
20. 17. Automation and Cron Jobs
21. 18. Quality Control and Linting
22. 19. Public PDF / Website Publishing Workflow
23. 20. Security, Privacy, and Data Rules
24. 21. Staff Training Plan
25. 22. Agent Operating Prompts
26. 23. Rollout Timeline
27. 24. Troubleshooting
28. 25. One-Page Quick Start for Staff
29. 26. Source Trail for This Manual

1. Executive Summary

This manual explains how to build a practical, staff-run, agent-assisted knowledge base around Hermes Agent. The idea is simple:

- **Hermes memory** remembers stable facts about the user, preferences, environment, and recurring corrections.
- **The LLM Wiki** remembers the wider world: meetings, research, email themes, SOPs, source notes, video transcripts, legal/policy notes, marketing ideas, CRM context, and lessons learned.
- **Skills** teach Hermes the exact procedure for reading, writing, checking, and using the wiki.
- **Cron jobs** can ingest approved sources on a schedule so the knowledge base improves without staff manually rebuilding it every week.

The goal is not to dump every file into AI. The goal is to create a disciplined source trail that staff and agents can trust. Every important claim should have a source, every source should have a place, and every reusable procedure should become a skill or wiki page.

Plain-English promise: if we build this correctly, staff can ask, "What do we know about this client/project/topic?" and Hermes can answer from organized source material instead of guessing from memory.

2. What We Are Building

The system in one sentence

A private, Obsidian-compatible markdown knowledge base that Hermes can read, update, cross-reference, lint, and use when answering business questions.

The core layers

1. **Raw source layer:** immutable copies of transcripts, articles, PDFs, emails, notes, screenshots, meeting summaries, and videos.
2. **Wiki layer:** cleaned pages about people, companies, projects, concepts, policies, SOPs, marketing ideas, disputes, and lessons.
3. **Schema layer:** rules that tell staff and agents how to name files, tag pages, cite sources, handle contradictions, and update the index.
4. **Hermes skill layer:** reusable instructions that tell Hermes exactly how to ingest, query, and maintain the wiki.
5. **Automation layer:** scheduled jobs that collect approved sources, create source notes, and ask Hermes to file them properly.

What this is not

- It is not a public website by default. It is a private operating memory.
- It is not a replacement for legal, accounting, engineering, medical, or regulatory review.
- It is not a place to store passwords, API keys, banking credentials, or confidential documents without approval and access controls.
- It is not a random dumping ground. The value comes from structure, source links, and review.

3. Roles and Responsibilities

Assign these roles before building. One person can hold more than one role, but every role must be covered.

Role	Who	Responsibility
Owner	Chris or appointed lead	Approves scope, access, publication boundaries, and automation risk.
Wiki Lead	Staff operations lead	Maintains SCHEMA.md, index.md, log.md, page quality, tags, and naming rules.
Source Collector	Staff/admin	Saves approved sources into the raw folders with source URLs, dates, and context.
Agent Operator	Hermes user	Runs ingest/query/lint prompts, verifies outputs, and fixes errors.
Review Lead	Manager or subject expert	Checks legal, technical, public-company, HR, or client-sensitive content before sharing.
Automation Lead	Technical operator	Creates cron jobs, API connections, sync jobs, backups, and health checks.
Privacy Gatekeeper	Owner/manager	Decides what may enter the wiki, what may leave it, and what must be redacted.

4. Required Tools and Skills

Core software

- **Hermes Agent:** the agent that reads, writes, schedules, searches, and reasons over the wiki.
- **Obsidian:** optional but recommended human interface for browsing the markdown vault and graph view.
- **Git:** version control for the wiki rules and non-sensitive pages.
- **Python 3:** scripts for linting, transcript cleanup, PDF generation, and batch processing.
- **yt-dlp / transcript tools:** for approved YouTube and social-video source capture.
- **OCR tools:** for scanned PDFs and image-only documents.
- **Cloud storage or backup:** Time Machine, external drive, private Git remote, or secured cloud backup.

Hermes toolsets to enable

- **file:** read/write/search/patch wiki files.
- **terminal:** run setup scripts, Git, Python, transcript tools, and verification commands.
- **web/browser:** capture public articles, source URLs, pages, and screenshots when needed.
- **skills:** load and maintain the wiki procedures.
- **memory:** save durable user/environment facts only, not giant documents.
- **session_search:** find relevant past Hermes conversations before creating duplicate notes.
- **cronjob:** schedule approved recurring ingestion and linting jobs.
- **vision:** inspect screenshots or image-only documents when OCR is insufficient.

Hermes skills to install/load

- **llm-wiki:** core wiki architecture, ingest, query, lint, schema, and cross-link rules.
- **youtube-content:** capture and transform video transcripts into source notes and wiki pages.
- **obsidian:** optional skill for reading and editing an Obsidian vault.
- **ocr-and-documents:** extract text from scans, PDFs, images, and business documents.
- **source-based-research-deliverables:** keep claims grounded and preserve uncertainty.
- **google-workspace / himalaya / teams-meeting-pipeline:** only if those connectors are approved and configured.

Model note: The video mentions using Claude/Opus as one possible interface. This manual does not require that. Use Hermes with the currently approved provider/model stack. Do not add new paid model providers or expose private data to third-party tools without owner approval.

4A. Hermes Concepts Staff Must Understand First

The second Jack Roberts source, *Every Hermes Concept explained for Normal People*, adds the beginner layer that staff need before they can safely build the knowledge-base system. The earlier manual explains the wiki. This section explains the operating brain around the wiki.

Training rule: no staff member should be asked to run ingestion, cron jobs, email access, MCP connectors, or publishing until they can explain these concepts in plain language.

#	Concept	Plain meaning for staff	Manual action
1	Agent, not chatbot	A chatbot answers. An agent can use tools, inspect files, browse, run commands, schedule tasks, and create outputs.	Give Hermes concrete outcomes and permission boundaries, not vague chatter.
2	Long-term assistant	Hermes is closer to a long-term assistant that learns how you work than a one-off coding tool.	Use Hermes for recurring business operations, not just single questions.
3	One brain, many mouths	The same Hermes can be reached through Telegram, terminal, dashboard, Discord, WhatsApp, Slack, and other gateways.	Keep instructions consistent across interfaces; do not create separate contradictory workflows.
4	Where Hermes lives	Hermes can run locally on a Mac/computer or remotely on a VPS. Local is simple and tangible; VPS is always-on but needs security.	Choose local for pilot/training; use VPS only when uptime, remote access, or automation requires it.
5	OAuth vs API keys	OAuth means sign in and approve access. API keys are secret strings that unlock services.	Prefer OAuth where possible. Never paste API keys into public chats, manuals, or wiki pages.
6	Model as the brain	Hermes is the agent framework; the selected model is the thinking engine behind it.	Pick the right model for the job: cheap for routine work, stronger for complex/legal/research planning.
7	Local-hosted models	The model itself can run on your own hardware for privacy, but power is limited by the machine.	Use local models for privacy-sensitive drafts where quality is acceptable; use approved cloud models for heavier reasoning.
8	Never start from zero	Hermes memory remembers durable user facts and past sessions; the wiki remembers source-grounded external knowledge.	Before asking staff to re-explain context, search memory/session/wiki first.
9	Character Bible / persona	System/persona instructions define tone, values, style, escalation rules, and how Hermes should behave.	Create a business persona file for staff-facing outputs: plain, careful, source-grounded, no secrets.
10	Integrations	Hermes becomes more useful when connected to approved tools: calendar, meeting notes, email, docs, drives, CRM, websites.	Connect only approved sources and document each connector in the wiki.
11	Computer access	When Hermes runs on a computer, it can read/write files, use browser tools, run shell commands, and produce PDFs.	Keep a clear working directory and backup before allowing file-changing actions.
12	MCP connectors	MCP is a standard way to give Hermes structured access to external tools.	Use MCP for repeatable connectors, but test each one with least-privilege access first.
13	Skills as muscle memory	Skills are reusable workflows Hermes can load when a task repeats.	Turn recurring staff processes into skills or SOP pages, not scattered chat history.

14	Slash-command controls	Commands such as queue, background, stop, reset, compress, and status control the session.	Train staff on these commands before allowing autonomous workflows.
15	Safety and house rules	Agents are powerful; give the least access needed and keep secrets out of memory/wiki/chat.	Use the privacy gate before connecting email, drives, websites, CRM, or payment systems.
16	North Star / goals	Instead of one question at a time, Hermes can organize work around a goal and break it into steps.	Use goal prompts for projects: launch a page, prepare a dossier, build a source library.
17	Subagents	Hermes can split work among smaller agents working in parallel.	Use subagents for research batches, audits, and parallel content review; verify their outputs.
18	Cron / heartbeat	Scheduled jobs let Hermes check or do things daily, weekly, or after a delay.	Use cron for briefings, lint checks, backups, and approved source ingestion.
19	Token and cost discipline	Every request has overhead; long contexts, wrong models, and runaway jobs waste money.	Compress/reset sessions, choose model tiers carefully, and keep cron prompts focused.
20	Operating system	Hermes is not just chat; it can become the interface across business knowledge, agents, tools, and workflows.	Design the system as an operating layer, with rules, ownership, backups, and review.
21	One brand / one operating brain	Hermes, code agents, wiki, notebooks, meetings, CRM, and websites should work under one governed system.	Centralize source trails and procedures so staff do not build isolated mini-systems.

4B. Operating Hermes Day-to-Day

This section turns the beginner tutorial into daily staff behaviour. The point is to make Hermes useful without letting it become messy, expensive, or unsafe.

Daily operator checklist

1. **Start with the outcome.** State the result wanted: report, PDF, source note, website update, CRM entry, briefing, or task list.
2. **Name the source lane.** Tell Hermes whether the material is a meeting, email, document, video, public article, CRM note, or SOP.
3. **Load the right skill.** For wiki work, use the company knowledge wiki skill. For videos, use YouTube/social-source skills. For PDFs, use the PDF manual workflow.
4. **Set boundaries.** Say whether it is internal only, public-share, legal-adjacent, financial, HR, family/private, or needs human review.
5. **Use the right interface.** Telegram is good for quick instructions and approvals. Terminal/dashboard is better for setup, debugging, file-heavy work, and long runs.
6. **Use queue/background carefully.** Queue follow-ups when Hermes is busy. Use background only for non-critical tasks that can be checked later.
7. **Stop bad runs early.** If Hermes is looping, spending too much, or doing the wrong task, use stop/reset and restate the task cleanly.
8. **Verify outputs.** For PDFs, verify text. For websites, verify HTTP and visible content. For wiki pages, verify index and log updates.

Beginner-safe command meanings

Control	Use	Staff warning
Reset / new session	Start clean when context is stale or confused.	Use before major new projects to avoid wrong assumptions.
Compress	Summarize long context so the session can continue.	Good for long builds, but verify important details survived.
Stop	Interrupt a bad or unsafe run.	Use immediately if Hermes starts changing the wrong files or exposing private data.
Queue	Line up a follow-up prompt for the next turn.	Do not queue conflicting instructions.
Background	Run a separate non-blocking task.	Do not use for sensitive writes unless the prompt is very specific.
Cron	Schedule recurring work.	Only create cron jobs with self-contained prompts and privacy limits.
Subagent/delegation	Split parallel research or review tasks.	Subagent claims must be verified before publication.

Daily staff prompt pattern

```

Use Hermes as an agent, not a chatbot.
Outcome wanted: [PDF / wiki ingest / source note / website update / CRM dossier / briefing]
Source lane: [meeting / email / document / video / article / SOP]
Privacy level: [public-share / internal / private / review-required]
Skills to use: [company-knowledge-wiki, youtube-content, pdf-generation, etc.]
Boundaries: do not store secrets; do not publish externally without approval; cite sources;
verify the result.
Report back with files changed, live links if any, and review flags.

```

4C. Safety, Cost, and Access Rules from the Beginner Tutorial

The second video is especially useful because it explains why the operating system must be governed. Hermes becomes powerful when connected to tools, but every connection creates responsibility.

Least-access rule

Rule: give Hermes the minimum access required for the current job. Do not connect email sending, payment systems, public posting, domain/DNS changes, or client folders just because the connector exists.

- **Read before write:** start with read-only access when testing a new connector.
- **One lane at a time:** connect meetings first, then documents, then approved email labels, then website publishing.
- **No secrets in chat:** API keys, passwords, tokens, seed phrases, SSH keys, and recovery codes stay in approved secret stores only.
- **Human review gates:** legal, finance, public-company, HR, medical, family/custody, and reputational claims require human review.
- **Public-share separation:** never publish from raw private wiki folders directly.

Model and cost rules

- Use cheaper/faster models for routine formatting, extraction, and drafts.
- Use stronger models for complex synthesis, legal-adjacent drafting, strategy, debugging, and high-impact public content.
- Reset or compress long sessions before asking small questions.
- Keep cron prompts narrow so they do not reload the whole world every morning.
- Review scheduled jobs monthly and remove stale ones.

Connector approval checklist

Question	Yes/No	Action
Does this connector expose private or client data?		If yes, owner/privacy approval required.
Can it write, send, delete, post, buy, or change records?		Start read-only or require explicit approval per action.
Is the connector needed for the current workflow?		If no, do not connect it.
Is there a source note or SOP for how it is used?		If no, create one before automation.
Can the output be verified?		If no, do not automate it yet.

How this strengthens the wiki build

The self-improving knowledge base should not become an uncontrolled memory dump. The beginner tutorial adds the missing operating rules: understand the agent, pick the right brain, connect only the right tools, schedule only safe recurring jobs, and keep the wiki as the source-grounded layer that staff can audit.

5. Folder Structure to Create

Use one clear root folder. Recommended default:

```
~/knowledge-wiki
```

Inside it, create this structure:

```
knowledge-wiki/  
├── SCHEMA.md  
├── index.md  
├── log.md  
├── README.md  
├── raw/  
│   ├── articles/  
│   ├── transcripts/  
│   ├── meetings/  
│   ├── emails/  
│   ├── documents/  
│   ├── videos/  
│   └── assets/  
├── entities/  
│   ├── people/  
│   ├── companies/  
│   ├── projects/  
│   └── places/  
├── concepts/  
├── operations/  
│   ├── sops/  
│   ├── checklists/  
│   └── automations/  
├── comparisons/  
├── queries/  
├── source-notes/  
├── reviews/  
└── _archive/
```

Why this matters

- **raw/** is the evidence. Do not rewrite it after capture.
- **entities/** is for who/what: people, companies, projects, assets, domains, properties.
- **concepts/** is for ideas: policy, strategy, marketing, law, AI, operations.
- **operations/** is for staff playbooks and repeatable workflows.
- **queries/** saves valuable answers that would be expensive to re-create.
- **source-notes/** explains where each source came from and how reliable it is.

6. Exact Initial Setup Steps

Step 1 - Confirm Hermes is healthy

```
hermes --version
hermes doctor
hermes status --all
hermes tools list
hermes skills list
```

If this is a messaging-platform Hermes, restart the gateway after config/tool changes:

```
hermes gateway status
hermes gateway restart
```

Step 2 - Create the wiki folder

```
mkdir -p ~/knowledge-
wiki/{raw/{articles,transcripts,meetings,emails,documents,videos,assets},entities/{people,compani
es,projects,places},concepts,operations/{sops,checklists,automations},comparisons,queries,source-
notes,reviews,_archive}
```

Step 3 - Tell Hermes where the wiki lives

Add this line to `~/ .hermes/ .env` or the active Hermes profile env file:

```
WIKI_PATH=/Users/YOUR-USER/knowledge-wiki
```

On Chris's Mac, the path pattern will normally be:

```
WIKI_PATH=/Users/macfebruary-20-2026/knowledge-wiki
```

Step 4 - Restart Hermes / gateway

```
hermes gateway restart
# or, for terminal sessions, exit and reopen hermes
```

Step 5 - Initialize Git for version history

```
cd ~/knowledge-wiki
git init
git add SCHEMA.md index.md log.md README.md operations source-notes entities concepts comparisons
```

```
queries  
git commit -m "init: create knowledge wiki structure"
```

Do not commit private raw sources unless the Git remote is private and approved. Add a `.gitignore` if the raw folders contain sensitive documents.

Step 6 - Open in Obsidian

1. Install Obsidian from [obsidian.md](#).
2. Open Obsidian.
3. Choose **Open folder as vault**.
4. Select `~/knowledge-wiki`.
5. Turn on backlinks, graph view, and daily notes if useful.
6. Set attachments to save under `raw/assets/`.

7. SCHEMA.md Starter Template

Create SCHEMA.md first. Staff and agents must obey it.

```
# Knowledge Wiki Schema

## Domain
This wiki stores approved operational knowledge, source notes, meetings, research, CRM-adjacent context, public-policy notes, marketing ideas, and staff procedures for Chris Anderson's business ecosystem.

## Hard Rules
- Never store passwords, API keys, recovery codes, bank credentials, or private login details.
- Raw sources are immutable. Do not edit raw files after capture.
- Every non-raw page must have YAML frontmatter.
- Every claim that may be disputed must cite a source file or URL.
- Every new or updated page must be added to index.md.
- Every ingest, query worth saving, lint, and archive action must be logged in log.md.
- Use [[wikilinks]] to connect pages.
- Use confidence labels: high, medium, low.
- Mark contradictions instead of hiding them.

## Frontmatter
---
title: Page Title
created: YYYY-MM-DD
updated: YYYY-MM-DD
type: entity | concept | operation | comparison | query | source-note | review
tags: [approved-tag]
sources: [raw/path/or/source-note]
confidence: high | medium | low
contested: false
---

## Approved Tags
- ai-operations
- staff-training
- marketing
- legal-adjacent
- public-company
- crm
- property
- mining
- restaurant
- website
- automation
- source-watch
- privacy
- finance
- policy
- meeting
- email
- youtube
- document
- procedure

## Page Rules
```

- Create a page when a person/company/project/concept is central to one source or appears in two or more sources.
- Add to an existing page when the topic already exists.
- Split pages over about 200 lines.
- Do not create pages for passing mentions.
- Prefer short, scannable pages with links to deeper pages.

Contradictions

When sources disagree:

1. Keep both claims.
2. Note date, source, and context.
3. Add contested: true.
4. Add a "Contradictions / open questions" section.
5. Ask the owner/review lead before public use.

8. index.md and log.md Templates

index.md

```
# Knowledge Wiki Index

> Every wiki page must be listed here with a one-line summary.
> Last updated: YYYY-MM-DD

## People

## Companies

## Projects

## Concepts

## Operations / SOPs

## Comparisons

## Saved Queries

## Source Notes
```

log.md

```
# Knowledge Wiki Log

> Append-only record of actions.
> Format: ## [YYYY-MM-DD] action | subject

## [YYYY-MM-DD] create | Wiki initialized
- Created folder structure.
- Added SCHEMA.md, index.md, log.md, README.md.
- Owner approval: pending/approved.
```

README.md

```
# Knowledge Wiki

This is a private source-grounded knowledge base for staff and Hermes agents.

Start here:
1. Read SCHEMA.md.
2. Read index.md.
3. Check recent log.md entries.
4. Save raw sources before creating analysis pages.
5. Never store secrets.
```

9. Build the Hermes Wiki Skill

Create a dedicated Hermes skill so every agent follows the same process. Skill name suggestion:

```
company-knowledge-wiki
```

Skill file content

```
---
name: company-knowledge-wiki
description: Use when reading, querying, ingesting, maintaining, linting, or publishing from the
company knowledge wiki.
tags: [wiki, knowledge-base, operations, staff]
---

# Company Knowledge Wiki

## Trigger
Use this skill when the user mentions the wiki, knowledge base, source notes, meeting notes,
staff manual, CRM context, business research, or asks "what do we know about...".

## Location
WIKI="${WIKI_PATH:-$HOME/knowledge-wiki}"

## Orientation - mandatory every session
1. Read $WIKI/SCHEMA.md.
2. Read $WIKI/index.md.
3. Read the last 30 entries of $WIKI/log.md.
4. Search existing pages for the topic before creating anything new.

## Ingest workflow
1. Capture raw source into raw/ with source URL, date, and hash.
2. Create a source note under source-notes/.
3. Identify entities, concepts, projects, and claims.
4. Update existing pages before creating new pages.
5. Add wikilinks and confidence labels.
6. Update index.md.
7. Append log.md.
8. Report every file created/updated.

## Query workflow
1. Orient first.
2. Read relevant pages.
3. Answer with source-page citations.
4. If the answer is reusable, save it under queries/ and update log.md.

## Public output rule
Before publishing or sharing externally:
- Remove private contact details unless approved.
- Remove secrets and credentials.
- Mark uncertain claims.
- Include source links or source-note references.
- Ask for human review on legal, finance, public-company, HR, health, and family/custody topics.
```

How to create it in Hermes

Ask Hermes:

```
Create a new Hermes skill called company-knowledge-wiki using the manual's skill content. Save it under ~/.hermes/skills/company-knowledge-wiki/SKILL.md. Then list the skill and confirm it loads.
```

Verification:

```
hermes skills list | grep company-knowledge-wiki
```

10. Source Intake Lanes

Every source must enter through a lane. This stops staff from mixing private notes, public articles, and unverifiable social content into one messy folder.

Lane	Examples	Raw folder	Review needed
Meetings	Granola, Teams, Google Meet, Zoom notes	raw/meetings	Yes if client/legal/HR
Email	Approved email threads or summaries	raw/emails	Yes; redact private info before public output
Documents	PDFs, scans, contracts, letters, reports	raw/documents	Yes; legal/financial sensitivity
Videos	YouTube, TikTok, Instagram, Facebook	raw/videos or raw/transcripts	Fact-check before treating claims as true
Articles	Web pages, blogs, news, policy docs	raw/articles	Source-quality review
CRM context	Referral notes, public professional bios	entities/people + source-notes	Keep private contact data internal
Operations	SOPs, checklists, staff instructions	operations/sops	Manager approval before staff rollout

Minimum source metadata

```

Source title:
Source URL or origin:
Date captured:
Captured by:
Why this matters:
Public/private/internal:
Reliability: high / medium / low
Known caveats:
Related people/companies/projects:
Recommended wiki pages to update:

```

11. Exact Ingest Procedure for Staff

1. **Decide if the source is allowed.** If it contains secrets, private family matters, bank data, medical data, HR issues, or privileged legal advice, get approval first.
2. **Save the raw source.** Use the correct raw folder. Do not edit the original after saving.
3. **Create a source note.** Put a short explanation in source-notes/.
4. **Ask Hermes to orient.** Hermes must read SCHEMA.md, index.md, and recent log.md first.
5. **Ask Hermes to ingest.** Give the source file path and what matters.
6. **Review Hermes' file list.** Confirm which pages were created or updated.
7. **Check index.md and log.md.** If Hermes missed either, send it back to fix them.
8. **Review sensitive claims.** If legal/finance/regulatory/public-company content is involved, have a human review.
9. **Commit safe changes.** Use Git for version history, excluding private raw files if necessary.

Staff prompt

```
Please ingest this source into the company knowledge wiki.
```

```
Source path:  
[insert path]
```

```
Context:  
[why this matters]
```

```
Follow company-knowledge-wiki exactly:  
- orient first by reading SCHEMA.md, index.md, and recent log.md;  
- save or verify the raw source;  
- create/update source note;  
- update existing pages before creating new pages;  
- add wikilinks and confidence labels;  
- update index.md;  
- append log.md;  
- report every file changed.
```

```
Do not publish externally. Do not store secrets.
```

12. Exact Query Procedure for Staff

Use this when staff want Hermes to answer from the wiki instead of general memory.

Query prompt

```
Use the company knowledge wiki to answer this.
```

```
Question:
```

```
[insert question]
```

```
Required steps:
```

1. Read SCHEMA.md, index.md, and recent log.md.
2. Search the wiki for relevant pages.
3. Read the relevant pages before answering.
4. Cite the wiki pages and source notes used.
5. Separate confirmed facts from assumptions.
6. If the answer would be useful later, save it under queries/ and update log.md.

Good questions

- What do we know about this person/company/project?
- What was decided in the last meeting about this?
- What public claims have we already fact-checked?
- What is our standard process for publishing a source card?
- What are the open risks before we send this letter or publish this article?

Bad questions

- "Tell me everything" - too broad.
- "Is this definitely legal?" - requires human legal review.
- "Find a password/API key" - secrets do not belong in the wiki.
- "Publish this private file" - needs owner approval and redaction.

13. Video and Social Media Ingest Procedure

The source video that triggered this manual is a good example: first capture the transcript, then turn it into an internal guide. Do not rely on the title alone.

YouTube transcript capture

```
python3 ~/.hermes/skills/media/youtube-content/scripts/fetch_transcript.py "YOUTUBE_URL" --
timestamps > raw/transcripts/source-name-transcript.json
```

Fallback using yt-dlp subtitles

```
mkdir -p /tmp/youtube_transcript
cd /tmp/youtube_transcript
yt-dlp --skip-download --write-auto-sub --write-subs --sub-lang en --sub-format vtt -o '%(id)s.%(
ext)s' 'YOUTUBE_URL'
```

Ask Hermes to process

```
Ingest this YouTube transcript into the wiki as a source.
Create:
- raw transcript file record;
- source note with URL, title, channel, date captured, limitations;
- concept page if the idea is central;
- operations/SOP page if the source contains a repeatable workflow;
- index and log updates.
Keep claims source-grounded. Do not treat a creator's opinion as proven fact.
```

For Instagram/TikTok/Facebook reels

- Download only if allowed and needed for analysis.
- Verify audio/video with ffprobe.
- Transcribe locally where possible.
- Create a contact sheet if visuals matter.
- Fact-check policy, law, finance, medicine, or public claims externally.
- Label uncertain claims clearly.

14. Meeting Notes and Calls

External steps

1. Choose the approved meeting recorder: Granola, Teams, Google Meet, Zoom, or manual notes.
2. Tell participants if recording/transcription is required by law or company policy.
3. Export the transcript or summary after the meeting.
4. Remove private side comments that do not belong in a business knowledge base.
5. Save to raw/meetings/YYYY-MM-DD-meeting-topic.md.
6. Create or update pages for decisions, tasks, people, companies, and projects.

Meeting source note template

```
---
title: Meeting - [topic]
date: YYYY-MM-DD
participants: [names or roles]
source_type: meeting
privacy: internal
confidence: medium
---

## Purpose

## Decisions

## Action items

## Open questions

## Pages to update
```

Daily automation idea

Once connectors are approved, create a daily job that checks the meeting-notes export folder and asks Hermes to ingest only new approved files.

```
Schedule: every weekday at 9:00 AM
Prompt: Check the approved meeting-notes folder for new files since the last run. For each new file, ingest it into the company knowledge wiki using company-knowledge-wiki. Do not publish externally. Report files updated.
```

15. Email and Inbox Knowledge

Email is sensitive. Do not bulk-ingest entire inboxes without approval. Start with summaries, selected threads, or labels/folders created for AI processing.

External steps

1. Create a dedicated email label/folder such as AI-Knowledge-Approved.
2. Staff move only approved threads into that folder.
3. Automation reads only that folder.
4. Summarize threads into source notes rather than copying unnecessary private details.
5. Update relevant entity/project pages.

Email source note template

```
Title: Email thread - [topic]
Date range:
Participants:
Mailbox/label:
Reason saved:
Private details removed:
Key facts:
Decisions:
Follow-up required:
Related wiki pages:
```

Safe automation prompt

```
Check only the approved email label/folder named AI-Knowledge-Approved. Summarize new threads into source notes without storing passwords, payment details, private family details, or unnecessary personal data. Update project/company pages only where the thread contains durable business facts. Append log.md.
```

16. Documents, PDFs, Scans, and Images

Exact document workflow

1. Save the original file under `raw/documents/`.
2. If it is a scan, run OCR and save extracted text beside it.
3. Create a source note with date, origin, sender, recipient, and purpose.
4. Ask Hermes to extract key facts, deadlines, obligations, names, amounts, and open questions.
5. For legal/financial/regulatory documents, mark analysis as **not legal/accounting advice** and require human review.
6. Update entity/project pages with cautious wording.

OCR command examples

```
# If text-based PDF
python3 -m pymupdf gettext input.pdf -o extracted.txt

# If using OCR/document extraction tools, save output as:
raw/documents/YYYY-MM-DD-document-name.extract.txt
```

Prompt

```
Extract the operational facts from this document for the wiki.
Separate:
- confirmed facts;
- deadlines;
- people/entities;
- obligations/requests;
- risks;
- questions for human review.
Do not provide legal advice. Update the wiki with confidence labels and source references.
```

17. Automation and Cron Jobs

Automation should be added only after the manual process works. Start small and add one lane at a time.

Recommended jobs

Job	Schedule	Purpose	Risk
Daily meeting ingest	Weekdays 9:00 AM	Process approved new meeting summaries.	Medium - privacy/consent.
Weekly wiki lint	Friday 4:00 PM	Find broken links, missing index entries, low-confidence pages.	Low.
Weekly source digest	Monday 8:00 AM	Summarize new approved sources and ask what to ingest.	Low/medium.
Monthly backup check	1st day monthly	Verify wiki backup, Git status, and archive readability.	Low.

Hermes cron command pattern

```
hermes cron create '0 9 * * 1-5'
# Then paste a self-contained prompt with source folders, wiki path, privacy rules, and delivery target.
```

Safe cron prompt template

```
You are maintaining the company knowledge wiki at $WIKI_PATH.
Task: ingest approved new meeting files only.
Rules:
- Read SCHEMA.md, index.md, and recent log.md first.
- Process only files in raw/meetings/_approved-inbox/.
- Do not process private/unapproved folders.
- Do not store secrets.
- Update source notes, relevant pages, index.md, and log.md.
- If a file contains legal, HR, medical, family/custody, or financial advice, mark it review-required and do not summarize sensitive details.
- Deliver a concise report with files changed and review flags.
```

18. Quality Control and Linting

Weekly lint checklist

- Every page listed in index.md.
- No broken `[[wikilinks]]`.
- No orphan pages with zero inbound links.
- Every page has valid frontmatter.
- Tags all appear in SCHEMA.md.
- Low-confidence pages are visible and reviewed.
- Contested pages are not silently treated as facts.
- Raw files have source notes.
- log.md is current.
- Large pages over about 200 lines are split or summarized.

Lint prompt

```
Run a health check on the company knowledge wiki.  
Check for:  
- broken wikilinks;  
- orphan pages;  
- pages missing from index.md;  
- invalid frontmatter;  
- tags not in SCHEMA.md;  
- source notes missing for raw files;  
- pages over 200 lines;  
- contested or low-confidence pages needing review.  
Report issues by severity and append the lint result to log.md.
```

Publication review checklist

- No passwords, keys, credentials, or private tokens.
- No unnecessary private addresses, phone numbers, or family details.
- Legal/finance/medical/public-company claims reviewed by a human.
- Quotes and claims trace to source notes.
- Uncertainty is labeled.
- Images and screenshots are allowed for sharing.

19. Public PDF / Website Publishing Workflow

The wiki is private. Public PDFs are outputs created from reviewed, redacted wiki material.

Steps to publish a PDF

1. Ask Hermes to draft the manual/report from approved wiki pages only.
2. Include source links and a source note section.
3. Remove private operational paths if the audience does not need them.
4. Remove private names/contact details unless approved.
5. Generate an HTML version first for review.
6. Export the HTML to PDF.
7. Extract text from the PDF to verify key sections are present.
8. Send to owner/review lead for final approval.
9. Upload to the website only after approval.

HTML to PDF command on Chris's Mac

```
CHROME='/Applications/Google Chrome.app/Contents/MacOS/Google Chrome'  
"$CHROME" --headless --disable-gpu --no-sandbox --print-to-pdf=/tmp/manual.pdf  
file:///tmp/manual.html
```

Website publishing idea

For Managing Expectations, store public downloads under:

```
assets/downloads/
```

Then link from a resource page or article after approval.

20. Security, Privacy, and Data Rules

Never store these in the wiki

- API keys, passwords, seed phrases, recovery codes, SSH passwords, or private keys.
- Banking credentials or payment card data.
- Private legal advice unless the wiki is explicitly approved for privileged material.
- Medical information unless required and access-controlled.
- Private family/custody details unless explicitly approved and access-controlled.
- Unredacted identity documents.

Access rules

- Give staff access only to the folders they need.
- Use private sync, private Git, or encrypted backup.
- Keep public website outputs separate from private wiki raw files.
- Review before connecting any third-party SaaS integration.
- Log automations so staff know what is being ingested.

Agent rules

- Hermes may summarize sensitive material internally only when approved.
- Hermes must not publish or email private material without explicit instruction.
- Hermes memory is for durable facts and preferences, not large documents.
- Wiki pages are for sourced knowledge. Skills are for procedures.

21. Staff Training Plan

Day 1 - Orientation

- Explain Hermes memory vs wiki knowledge.
- Show folder structure.
- Show Obsidian graph and backlinks.
- Review privacy rules.
- Practice saving one approved article into raw/articles.

Day 2 - Ingest practice

- Have each staff member create one source note.
- Ask Hermes to ingest one source.
- Check index.md and log.md.
- Fix one missing link or tag.

Week 1 - Pilot

- Pick one narrow domain: e.g., Managing Expectations content sources, staff SOPs, or CRM referral notes.
- Ingest 10-20 approved sources manually.
- Run lint at the end of the week.
- Review what helped and what created noise.

Weeks 2-4 - Automate cautiously

- Add one cron job for low-risk sources.
- Add meeting notes only after consent and folder rules are clear.
- Add email ingestion only from an approved label/folder.
- Hold weekly review with owner and wiki lead.

22. Agent Operating Prompts

Prompt: create the wiki

```
Create a new company knowledge wiki at ~/knowledge-wiki using the LLM Wiki pattern. Build the folder structure, SCHEMA.md, index.md, log.md, README.md, and an operations/sops folder. Use privacy-first rules: no credentials, no secrets, no private-public mixing. After creating it, verify files exist and report the structure.
```

Prompt: ingest a video

```
Use youtube-content and company-knowledge-wiki. Fetch the transcript for [URL], save it under raw/transcripts, create a source note, then turn the useful procedure into operations/sops/[slug].md. Update index.md and log.md. Include source URL and limitations.
```

Prompt: create a staff SOP

```
Create a staff SOP from these source notes. The SOP must include purpose, scope, roles, prerequisites, exact steps, verification, escalation, privacy rules, and a checklist. Save it under operations/sops, add it to index.md, and log it.
```

Prompt: answer from wiki

```
Answer using only the company knowledge wiki and clearly mark anything not found. Cite page names/source notes. If the answer requires legal/financial/regulatory judgment, state that human review is required.
```

Prompt: prepare public PDF

```
Draft a public PDF from approved wiki pages only. Remove private names, credentials, internal paths not needed by readers, and sensitive details. Include source links and caveats. Save HTML and PDF. Verify PDF text contains the table of contents and all major headings.
```

23. Rollout Timeline

Phase	Time	Outcome
Preparation	Day 0	Owner approves scope, access, and first pilot domain.
Setup	Day 1	Hermes checked, folder created, Obsidian connected, schema written.
Manual pilot	Days 2-7	10-20 sources ingested manually; staff trained on source notes.
Quality gate	End of Week 1	Lint report, duplicate cleanup, tag cleanup, privacy review.
First automation	Week 2	One low-risk cron job added.
Operations rollout	Weeks 3-4	SOPs, query prompts, publishing workflow, backups.
Scale	Month 2+	Add approved email/meeting/document lanes and department-specific pages.

24. Troubleshooting

Hermes cannot find the wiki

- Check `echo $WIKI_PATH` in the same environment where Hermes runs.
- Check `~/ .hermes/ .env` or the active profile's env file.
- Restart Hermes gateway or terminal session.

Duplicate pages appear

- Agents skipped orientation. Re-read `SCHEMA.md`, `index.md`, and `log.md` before ingesting.
- Search before creating pages.
- Merge duplicate pages and archive the weaker duplicate.

Index is stale

- Run a lint job.
- Ask Hermes to compare filesystem pages against `index.md`.
- Update index and log the repair.

Answers sound invented

- Require page/source citations.
- Ask Hermes to separate "found in wiki" from "general knowledge".
- Do not let unsourced claims harden into facts.

Staff are dumping too much

- Restrict to approved intake folders.
- Require source notes.
- Use page thresholds: central to one source or appears in two or more sources.

25. One-Page Quick Start for Staff

When you find something worth saving

1. Ask: is it approved and safe to store?
2. Save the raw source in the correct raw folder.
3. Create a source note with URL/origin, date, context, reliability, and caveats.
4. Ask Hermes to ingest it using company-knowledge-wiki.
5. Check that Hermes updated index.md and log.md.
6. Flag anything legal/finance/HR/private for human review.

Never do this

- Never save passwords or API keys.
- Never publish from the private wiki without approval.
- Never treat social media claims as proven without checking.
- Never let Hermes create unsourced public claims.

Best staff prompt

```
Please ingest this approved source into the company knowledge wiki. Orient first, create/update the source note, update relevant pages, add wikilinks, update index.md, append log.md, and report every changed file. Do not store secrets or publish externally.
```

26. Source Trail for This Manual

- **Primary video 1:** Build a Hermes Knowledge Base That Self-Improves
- **Creator/channel:** Jack Roberts
- **URL:** <https://youtu.be/D3dQqqDx2V4?si=Pin2uw73qMYV5Kvo>
- **Captured transcript 1:** research/ai-operations/hermes-self-improving-knowledge-base-video-transcript-2026-06-14.json
- **Source note 1:** research/ai-operations/hermes-self-improving-knowledge-base-source-note-2026-06-14.md
- **Primary video 2:** Every Hermes Concept explained for Normal People
- **Creator/channel:** Jack Roberts
- **URL:** <https://youtu.be/L0tQrTQBmjl?si=LbeQ5MGqtCeCnyJA>
- **Captured transcript 2:** research/ai-operations/hermes-every-concept-normal-people-transcript-2026-06-14.json
- **Source note 2:** research/ai-operations/hermes-every-concept-normal-people-source-note-2026-06-14.md
- **Related method:** Andrej Karpathy LLM Wiki / Obsidian-compatible markdown knowledge base pattern.
- **Hermes skills used for this manual:** hermes-agent, llm-wiki, youtube-content, source-based-research-deliverables, pdf-generation-macos.

Public-share caveat: This guide describes a general method. Before using it with real client, family, legal, financial, medical, HR, or public-company material, get approval and apply privacy controls.