

Hermes Agent Cheat Sheet

Hermes Agent Cheat Sheet

A practical quick-reference guide for Hermes Agent by Nous Research — the OpenClaw replacement with a closed learning loop for self-improving AI automation.

Installation

Official Bash Installer (Recommended)

```
curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh
source ~/.bashrc
hermes --version
```

Handles Python 3.11, Node.js v22, uv, ripgrep, and ffmpeg automatically. Git is the only prerequisite.

Docker

```
docker run -d \
  --name hermes \
  --restart unless-stopped \
  -v ~/.hermes:/opt/data \
  -p 8642:8642 \
  nousresearch/hermes-agent gateway run
```

Place `.env` in `~/.hermes/` with `ANTHROPIC_API_KEY` or `OPENAI_API_KEY`, plus platform tokens like `TELEGRAM_BOT_TOKEN`.

From Source

```
git clone https://github.com/NousResearch/hermes-agent.git
cd hermes-agent
python3 -m venv .venv
source .venv/bin/activate
pip install -e ".[all]"
hermes --version
```

Use `pip install -e .` for minimal install without platform connectors and memory backends.

Migration from OpenClaw

Auto-Detect and Migrate

```
# Preview then migrate (always shows preview first)
hermes claw migrate

# Preview only, no changes
hermes claw migrate --dry-run

# Full migration including API keys, skip confirmation
hermes claw migrate --preset full --migrate-secrets --yes

# Migrate without secrets
hermes claw migrate --preset user-data
```

Migration Flags

Flag	Description
<code>--dry-run</code>	Preview only — no files modified
<code>--preset {user-data,full}</code>	Migration preset (default: full). <code>user-data</code> excludes secrets
<code>--overwrite</code>	Overwrite existing files (default: skip conflicts)
<code>--migrate-secrets</code>	Include allowlisted secrets
<code>--source PATH</code>	Custom OpenClaw directory
<code>--workspace-target PATH</code>	Copy AGENTS.md to this path
<code>--skill-conflict {skip,overwrite,rename}</code>	Handle skill conflicts (default: skip)
<code>--yes</code> , <code>-y</code>	Skip confirmation prompts

What Gets Migrated

- **Persona:** `SOUL.md` → `~/.hermes/SOUL.md`
- **Memory:** `MEMORY.md` , `USER.md` , `HEARTBEAT.md` → `~/.hermes/memories/`
- **Skills:** `workspace/skills/` → `~/.hermes/skills/openclaw-imports/`
- **Command allowlist:** → `config.yaml`
- **Messaging settings:** Platform configs, allowed users → `~/.hermes/.env`
- **API keys:** Telegram, OpenRouter, OpenAI, Anthropic, ElevenLabs → `~/.hermes/.env`
- **TTS assets:** → `~/.hermes/tts/`
- **Workspace instructions:** `AGENTS.md` → `--workspace-target path`

OpenClaw → Hermes Config Mapping

OpenClaw	Hermes
<code>agents.defaults.model</code>	<code>config.yaml</code> → <code>model</code>
<code>models.providers.*</code>	<code>config.yaml</code> → <code>custom_providers</code>
<code>channels.telegram</code>	<code>config.yaml</code> → <code>platforms.telegram</code>

OpenClaw	Hermes
agents.defaults.compaction	config.yaml → compression

Setup and Verification

```
# First-time setup wizard (auto-detects OpenClaw)
hermes setup

# Check agent, auth, and platform status
hermes status

# Diagnose any config or dependency issues
hermes doctor

# Run the gateway (Telegram, Discord, etc.)
hermes gateway
```

Memory Management

```
# Rebuild memory index (run after migration)
hermes memory reindex

# List all memory entries
hermes memory list

# Search memory
hermes memory search "query"

# Delete specific memory entry
hermes memory delete <entry-id>

# Clear all memories
hermes memory clear --yes

# Export memories to JSON
hermes memory export --output ./memories.json

# Import memories from JSON
hermes memory import ./memories.json
```

Skills

```
# List all loaded skills
hermes skills list

# Enable a skill
hermes skills enable <skill-name>

# Disable a skill
hermes skills disable <skill-name>
```

```
# Install skill from URL or path
hermes skills install <source>

# Update a skill
hermes skills update <skill-name>

# Remove a skill
hermes skills remove <skill-name>

# Show skill details
hermes skills info <skill-name>
```

Model Selection

```
# Interactive model selector
hermes model

# Set model explicitly
hermes model set qwen-3.5

# Recommended models:
# - Qwen 3.5 (free on OpenRouter) – routine automation
# - Claude Opus 4.6 / GPT 5.4 – complex reasoning (thinking mode)
hermes model set claude-opus-4.6
```

Configuration

Key Config Locations

Purpose	Path
Config file	~/.hermes/config.yaml
Persona	~/.hermes/SOUL.md
Memories	~/.hermes/memories/
Skills	~/.hermes/skills/
Platform tokens	~/.hermes/.env
Logs	~/.hermes/logs/
TTS assets	~/.hermes/tts/

Environment Variables (.env)

```
# Required: at least one API key
ANTHROPIC_API_KEY=sk-...
# or
OPENAI_API_KEY=sk-...

# Optional: platform tokens
TELEGRAM_BOT_TOKEN=...
DISCORD_BOT_TOKEN=...
```

```
SLACK_BOT_TOKEN=...
ELEVENLABS_API_KEY=...
OPENROUTER_API_KEY=...
```

config.yaml Structure

```
model: qwen-3.5
custom_providers:
  openrouter:
    api_type: openrouter
    api_key: ${OPENROUTER_API_KEY}

platforms:
  telegram:
    enabled: true
    bot_token: ${TELEGRAM_BOT_TOKEN}
    allowed_users:
      - 123456789

compression:
  enabled: true
  interval: 7d

thinking:
  default: medium
  medium_threshold: 2000 tokens
```

Gateway and Runtime

```
# Start the gateway (required for platform integrations)
hermes gateway

# Start with specific platform only
hermes gateway --platform telegram

# Run gateway in background
hermes gateway &

# Stop the gateway
hermes gateway stop

# Restart gateway
hermes gateway restart

# View gateway logs
hermes logs gateway

# View real-time gateway output
hermes logs gateway --follow
```

Self-Improvement Loop (GEPA)

Hermes uses a **Goal-Evaluation-Planning-Action** reflection loop. The agent distills completed tasks into persistent memory automatically.

```
# Force a reflection cycle on recent memories
hermes reflect --depth recent

# Force reflection on specific memory entry
hermes reflect --memory <entry-id>

# View GEPA loop statistics
hermes stats reflection-loop

# Disable auto-reflection (if needed)
hermes config set reflection.auto false

# Set reflection frequency
hermes config set reflection.frequency 24h
```

Troubleshooting

```
# Full system diagnostics
hermes doctor

# Check version
hermes --version

# Update Hermes to latest
hermes update

# Rollback to previous version
hermes update --rollback

# Validate config file
hermes config validate

# Show full config (secrets redacted)
hermes config show

# Reset to defaults (keeps memories)
hermes reset --keep-memories

# Complete reset
hermes reset --full --yes
```

Docker Compose (Recommended for Self-Hosting)

```
services:
  hermes:
    image: nousresearch/hermes-agent:latest
    container_name: hermes
    restart: unless-stopped
    ports:
      - "8642:8642"
    volumes:
```

```

- ~/.hermes:/opt/data
env_file:
- ~/.hermes/.env
environment:
- ANTHROPIC_API_KEY=${ANTHROPIC_API_KEY}
command: gateway run

# Optional: nginx reverse proxy for HTTPS
nginx:
image: nginx:alpine
ports:
- "80:80"
- "443:443"
volumes:
- ./nginx.conf:/etc/nginx/nginx.conf:ro
depends_on:
- hermes

```

Quick Reference Card

Command	Purpose
hermes setup	First-time setup + migration
hermes claw migrate	Import OpenClaw config
hermes status	Check system status
hermes doctor	Diagnose issues
hermes gateway	Start platform integrations
hermes memory reindex	Rebuild memory after migration
hermes skills list	Show loaded skills
hermes model	Interactive model selector
hermes reflect	Force GEPA reflection cycle
hermes update	Update to latest version
hermes logs gateway --follow	Real-time logs
hermes config validate	Validate configuration