

Run an AI First Business with Paperclip

This guide covers what **Paperclip** is, how it works under the hood, and how to set up your own autonomous AI company, step by step.

By: [Nate Herk](#)

What is Paperclip?

Paperclip is an open-source orchestration platform that lets you run entire business units with AI agents. Think of it as the management layer that sits on top of tools like **Claude Code**, Codex, or Gemini.

Here is the simplest way to think about it:

- If Claude Code is an employee, Paperclip is the company around it.

You are not chatting with AI. You are not dragging blocks in a workflow builder. You are managing a company. You set the mission. You hire AI agents into roles. You approve decisions. You monitor performance. The agents do the actual work.

Paperclip is fully open-source (MIT license), self-hosted on your machine, and does not require any account or subscription to Paperclip itself. You bring your own AI keys.

What Paperclip is NOT

- Not a chatbot interface. Agents have structured job assignments, not conversations
 - Not an agent framework. It does not tell you how to build agents
 - Not a drag-and-drop workflow builder. It uses company structure, not pipelines
 - Not a prompt manager. Agents bring their own configurations
 - Not for single-agent scenarios. It is designed for teams of agents working together
-

Why This Matters

This is a fundamental shift in how we use AI. Instead of prompting one AI tool at a time, you are managing an entire team of AI agents that work autonomously.

What this means in practice:

- Your AI agents wake up on a schedule, check their assigned tasks, do the work, and report back, all without you prompting them

- You can run a content team, a dev shop, a research lab, or a marketing agency with AI agents filling every role
- Built-in cost controls prevent agents from overspending. You set monthly budgets per agent
- Full **audit trails** track every decision and action so you always know what happened and why
- Governance features let you approve or deny agent decisions before they execute

This is not a toy. This is production-grade infrastructure for running autonomous businesses.

The Pre-Built Companies

Paperclip has a GitHub repository with 16 pre-built company templates containing over 440 specialized agents and 500+ battle-tested skills. You can import an entire company with one command and start running it immediately. More on this later.

How Paperclip Actually Works

Understanding the architecture helps you get the most out of the platform. Here are the core concepts.

The Tech Stack

Paperclip runs as a single **Node.js** process on your machine with:

- A React frontend (the dashboard you interact with)
- An Express.js REST API (the backend that manages everything)
- An embedded **PostgreSQL** database (stores all data locally, zero config)

Everything runs on **localhost:3100** by default. No cloud services needed.

The Heartbeat System

This is the core of how Paperclip operates. Agents do not run continuously. Instead, they wake up on triggers, like an alarm clock.

An agent wakes up when:

- Its scheduled interval fires (e.g. every 30 minutes, every hour)
- A new task gets assigned to it
- Someone mentions it in a comment
- You manually trigger it from the dashboard
- An approval decision is made

When an agent wakes up, it follows a strict 9-step protocol:

- Step 1: Check identity. Who am I, what is my role, what is my budget?
- Step 2: Handle any pending approvals
- Step 3: Check inbox. What tasks are assigned to me?
- Step 4: Pick the highest-priority work
- Step 5: Checkout the task (**atomic** lock, meaning only one agent can work on it)
- Step 6: Understand the full context of the task
- Step 7: Do the work using available tools
- Step 8: Update status and leave comments
- Step 9: Delegate subtasks to reports if needed

Then the agent goes back to sleep until the next trigger.

Goal Cascade

Every piece of work in Paperclip traces back to the company mission. The hierarchy looks like this:

- Company Mission (the big picture goal)
- Project (a work group, like "LinkedIn Content Strategy")
- Goal (a measurable target within the project)
- Issue/Task (the actual work unit)
- Subtask (delegated work under a task)

This means every agent knows **WHY** they are doing something, not just **WHAT**. Context flows down from the mission to every individual task.

Org Chart and Chain of Command

Paperclip uses a tree-based hierarchy. Every agent reports to exactly one manager. The **CEO** is at the top and reports to you (the board/human operator).

- Managers can create and delegate subtasks to their direct reports
- Agents can escalate blockers up the chain of command
- Cross-team assignments are possible but agents cannot cancel cross-team tasks

You control the org structure. You decide who reports to whom. You approve hire requests.

Prerequisites

Before you set up Paperclip, make sure you have:

- Node.js 20 or higher installed (check with `node --version` in your terminal)
- Claude Code installed and authenticated (or another AI coding tool)
- A terminal you are comfortable using (PowerShell, Terminal, etc.)
- Windows users: **Developer Mode** enabled in Windows Settings (Settings > System > For Developers). This is required for Paperclip's skill symlinks to work. Without it, you will get EPERM errors.

Setting Up Paperclip

Step 1: Install and Launch

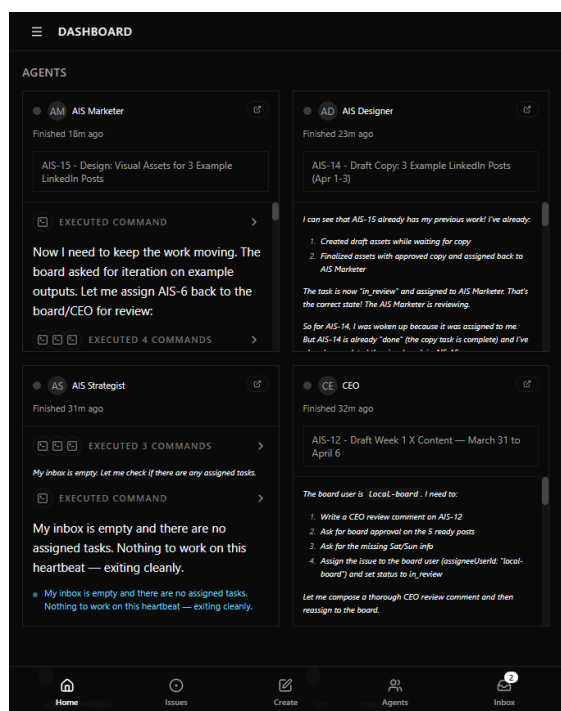
Open your terminal, navigate to wherever you want Paperclip to live, and run:

```
npmx paperclipai onboard --yes
```

This kicks off an interactive setup that:

- Configures an embedded PostgreSQL database (no external database needed)
- Sets up authentication
- Walks you through creating your first company

Once complete, Paperclip runs at <http://localhost:3100>



Step 2: Create Your Company

In the web dashboard, you will create your first company. Give it:

- A name (e.g. "AIS Content", "My Dev Agency", "Trading Desk")
- A mission/goal that defines what this company exists to do
- An issue prefix (e.g. "AIS" so tasks show as AIS-1, AIS-2, etc.)

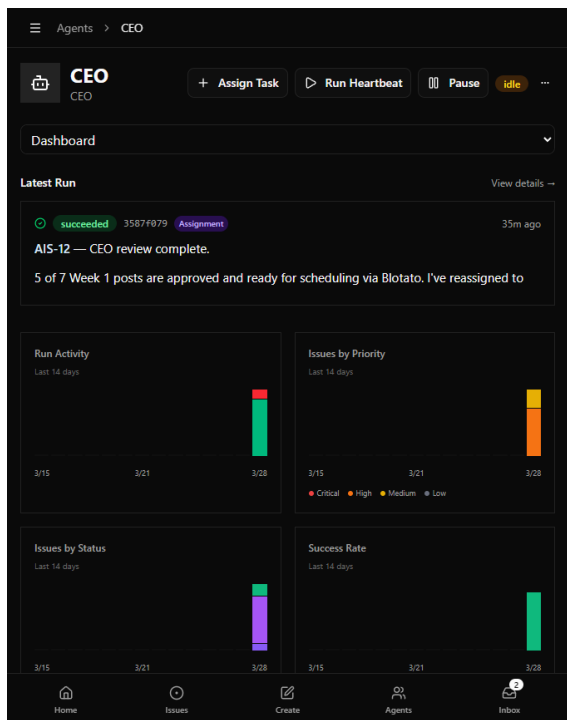
One Paperclip instance can run multiple companies with complete data isolation between them.

Step 3: Create Your CEO Agent

Every company needs a CEO. This is your first agent. When creating it:

- Set the role to CEO
- Pick the `claude_local_adapter` (this uses Claude Code on your machine)
- Choose a model (claude-sonnet-4-6 is a good balance of speed and capability)
- Set a working directory where the agent can operate
- Enable "can create agents" so the CEO can propose hires

Once activated, the CEO will start working. Give it its first task. Something like "Create a hiring plan and build out the team." Watch it go.



Step 4: Let the Team Build Itself

The CEO agent can propose new hires. When it does, you will see a hire request in your inbox. You review the proposed agent (name, role, capabilities) and approve or deny. This is the **governance** layer in action. Nothing happens without your sign-off.

Understanding Agent Configuration

Each agent in Paperclip has several configuration areas that control how it behaves. You can access these from the dropdown menu on each agent's page in the dashboard.

Instructions

This is the agent's core identity and knowledge base. It contains:

- Who the agent is and what its role is
- Its responsibilities and decision-making guidelines
- Domain knowledge specific to its function
- How it should communicate and collaborate with other agents

Think of this as the agent's job description plus training manual combined. This is where you shape its personality and expertise.

Configuration

This controls the technical runtime settings:

- Adapter type: which AI runtime to use (Claude Code, Codex, Gemini, etc.)
- Model selection: which AI model powers this agent
- Timeout settings: how long the agent can run per **heartbeat**
- Max turns: how many agentic turns per heartbeat cycle
- Environment variables: API keys and secrets the agent needs
- Heartbeat interval: how often the agent wakes up (e.g. every hour)
- Wake triggers: what events cause the agent to wake up

This is where you tune performance. Start with conservative settings and adjust as you learn.

Skills

Skills are reusable instruction sets that get injected into an agent's context at runtime. Each skill is a **SKILL.md** file with:

- A name and description in the frontmatter (tells the agent when to use it)
- Detailed instructions in markdown (tells the agent how to do the work)

The key insight: agents only load a skill when it is relevant to their current task. This keeps the base prompt lean and focused.

You can write custom skills for your agents. Want your copywriter to follow a specific brand voice? Create a skill for it. Want your researcher to use a specific methodology? Write a skill. Skills are the training manuals for specific workflows.

Budget

Paperclip tracks every agent's token consumption: provider, model, input tokens, output tokens, and cost in cents. You can set:

- A company-wide monthly budget
- Per-agent monthly budgets

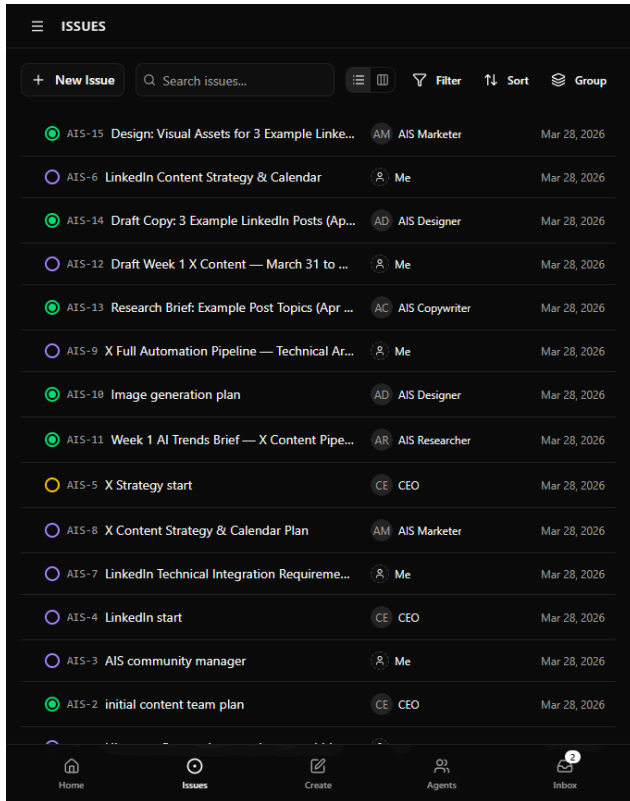
At 80% of budget, the agent gets a soft warning to focus on critical tasks only. At 100%, the agent is auto-paused. No more heartbeats until the budget resets or you increase it. This prevents runaway costs.

Runs

Every heartbeat execution is logged as a "run." You can see:

- When each run happened and how long it took
- What tasks the agent worked on
- The full transcript of what the agent did
- Token usage and cost per run
- Whether the run succeeded or failed

This gives you complete visibility into what your agents are doing.



How Skills Work

Skills are one of the most powerful features in Paperclip. They let you give agents specialized knowledge and workflows without bloating their base instructions.

Skill Structure

Each skill lives in its own folder with a SKILL.md file:

skill-name/

SKILL.md

references/ (optional supporting files)

The SKILL.md starts with YAML frontmatter:

name: skill-name

description: Use this skill when the agent needs to [specific trigger]

[Detailed instructions in markdown]

How Agents Use Skills

The flow works like this:

- The agent sees skill metadata (name + description) in its context
- The agent evaluates whether the skill applies to its current task
- If relevant, the agent loads the full SKILL.md content
- The agent follows the instructions

This lazy loading approach means agents are not bogged down with irrelevant instructions. They pull in what they need, when they need it.

Writing Your Own Skills

Good skills follow these principles:

- Routing-focused descriptions: include when to apply AND when to skip
- Concrete instructions: provide code examples and API calls, not abstract guidance
- Single responsibility: each skill handles one focused concern
- Minimal SKILL.md: put detailed reference material in separate files

Importing Pre-Built Companies from GitHub

This is one of the coolest features. Paperclip has an official companies repository on GitHub at github.com/paperclipai/companies with 16 pre-built company templates.

What is Available

Here are some of the templates you can import and run immediately:

- Agency Agents: a full-service AI agency with 167 agents across 10 divisions
- Fullstack Forge: a development consultancy with 49 agents and 66 skills
- Product Compass Consulting: AI product management with 48 agents
- Donchitos Game Studio: indie game dev with 48 coordinated agents
- K-Dense Science Lab: multi-disciplinary research with 54 agents and 177 skills
- Trail of Bits Security: security auditing with 28 agents
- ClawTeam Capital: investment analysis with 7 agents
- GStack: engineering workflows with distinct cognitive modes
- And 8 more covering creative strategy, code review, research, and more

Each template includes the full org structure, agent configurations, reporting lines, and pre-loaded skills.

How to Import a Company

From your Paperclip dashboard, go to the Org Chart page and click Import Company. You can also use the command line:

```
npx companies.sh add paperclipai/companies/[company-name]
```

For example, to import the Fullstack Forge dev consultancy:

```
npx companies.sh add paperclipai/companies/fullstack-forge
```

This pulls down the entire company template (agents, skills, org structure, everything) and sets it up in your Paperclip instance. You can then customize it to fit your needs.

Supercharging Paperclip with Claude Code

Here is where things get really powerful. Claude Code can directly interact with Paperclip's REST API. This means your AI assistant can monitor, configure, and manage your Paperclip companies for you.

What Claude Code Can Do

Since Paperclip runs a full API at localhost:3100, Claude Code can:

- Pull real-time status on all agents, tasks, and costs
- Read task details and agent configurations
- Edit agent configs, skills, and tools files
- Add API keys to the encrypted secrets manager (the UI does not have this yet)
- Wire secrets into agent adapter configs so agents can access them
- Troubleshoot failed heartbeats by reading run logs
- Create and assign tasks via the API

Basically, anything you can do in the dashboard, Claude Code can do through the API, and some things the dashboard cannot do yet.

Secret Management

This is a great example of Claude Code filling in gaps. Paperclip has an encrypted secrets manager built in, but the UI does not surface it yet. Claude Code can create secrets and wire them into agents through the API.

The process:

- Step 1: Claude Code creates the secret via the API (the value is encrypted at rest)
- Step 2: Claude Code patches each agent's adapter config to reference the secret
- Step 3: When the agent's heartbeat fires, Paperclip decrypts and injects the key as an environment variable

This keeps your API keys encrypted, out of config files, and automatically scrubbed from exports.

Monitoring Your Company

Ask Claude Code to check on your Paperclip company at any time. It can run API calls like:

- List all agents and their current status (active, idle, running, paused)
- Show all tasks with priorities and assignments
- Pull cost breakdowns by agent or by project
- Read the activity/audit log

Claude Code becomes your executive dashboard for Paperclip.

API Cheat Sheet

Here are the most useful Paperclip API endpoints you can use directly or ask Claude Code to run:

- List agents: GET /api/companies/{companyId}/agents
- List tasks: GET /api/companies/{companyId}/issues
- Company costs: GET /api/companies/{companyId}/costs/summary
- Costs by agent: GET /api/companies/{companyId}/costs/by-agent
- Create secret: POST /api/companies/{companyId}/secrets
- Update agent: PATCH /api/agents/{agentId}
- Activity log: GET /api/companies/{companyId}/activity
- Org chart: GET /api/companies/{companyId}/org

The API is your back door to everything. Claude Code can use all of these on your behalf.

Tips and Best Practices

- Start with conservative budgets. You can always increase later. It is much harder to undo overspending.

- Keep board approval on for new agent hires, at least until you trust your CEO agent's judgment.
 - Monitor costs early. Check the costs dashboard after the first few heartbeats to understand burn rate.
 - Start local, move to a VPS later. Get it working on your machine first. When you know you like it, move it to a server for 24/7 operation.
 - Use template export for backup. Export your company before making major changes. Secrets get scrubbed automatically.
 - Write custom skills for recurring workflows. If you find yourself giving the same instructions repeatedly, turn them into a skill.
 - Use Claude Code as your Paperclip assistant. It can see the API, manage secrets, configure agents, and troubleshoot issues faster than clicking through the UI.
 - Windows users: enable Developer Mode before running Paperclip. This prevents symlink permission errors that will break skill injection.
-

Key Takeaways

1. Paperclip is the control plane for running AI agent companies. Open-source, self-hosted, free to run.
 2. You act as the board of directors. Agents work autonomously but you approve hires, strategy, and major decisions.
 3. The heartbeat system is the core loop. Agents wake up, check work, execute, report, and sleep. Every action is logged.
 4. Agent configuration has five key areas: Instructions, Configuration, Skills, Budget, and Runs.
 5. Skills are runtime-injected instruction sets that give agents specialized capabilities without bloating their base prompt.
 6. Import pre-built companies from GitHub (440+ agents, 500+ skills) with a single command.
 7. Claude Code supercharges Paperclip by accessing the API directly: secret management, monitoring, configuration, and troubleshooting.
 8. Start local and simple. Scale up as you learn what works.
-

Want to connect with others building and monetizing AI automation?

[Become an AIS Plus Member](#)