SCORED.TOOLS · ROUNDUP

Best Open Source AI Desktop Agent Tools 2026

Seven open source AI desktop agents that actually run local, control your machine, and don't lock you into a vendor. Ranked by what works today.

Desktop agents that run on your machine, read your files, and execute commands are having a moment. The proprietary options (Claude Code, Cursor, Windsurf) get most of the attention, but the open source stack has quietly caught up on capability while keeping the two things closed tools can't offer: full local execution and no vendor lock-in.

This roundup covers seven open source desktop agents worth your time in 2026. Ranked by what actually works in daily use, not by GitHub stars or Twitter noise.

1. Open Interpreter — Score: 9.2/10

Open Interpreter runs an LLM in your terminal and lets it execute code locally to complete tasks. It's the most mature project in this space — the abstraction of "just let the model write and run Python/shell/AppleScript" turned out to be the right one, and everything else copied it. The OS mode gives it full desktop control (mouse, keyboard, screen), which is either exactly what you want or terrifying depending on the task.

Best for: Data wrangling, file operations, one-off automation scripts you'd rather describe than write.

Pricing: Free, open source (AGPL-3.0). Bring your own API key or point at a local model.

2. Aider — Score: 9.0/10

Aider is a terminal-based pair programmer that edits files in your git repo and commits each change. Not a full desktop agent in the OS-control sense, but it's the tool most developers actually use for agentic coding — and it's ruthlessly focused. The repo map, automatic commits, and multi-file edits work better than you'd expect for a project this lean. Works with any model that follows instructions well.

Best for: Serious codebase work where you want a real git history of every AI edit.

Pricing: Free (Apache-2.0). BYO key — pairs well with Claude Sonnet or DeepSeek for cost.

3. Continue — Score: 8.7/10

Continue is the open source IDE agent — VS Code and JetBrains extensions with chat, autocomplete, and agent modes. Not a standalone desktop agent, but it's the closest open source analog to Cursor and it's genuinely good. Full config-as-code (YAML), swappable models per role (chat/edit/autocomplete/embed), and a growing catalog of custom tools. The trade-off vs. Aider is you get IDE integration but lose Aider's git discipline.

Best for: Developers who want Cursor's UX without the subscription and without giving up model choice.

Pricing: Free (Apache-2.0). BYO key or run locally via Ollama.

4. Cline — Score: 8.5/10

Cline (formerly Claude Dev) is a VS Code agent that reads your workspace, writes code, and executes terminal commands with per-step approval. The plan/act split, checkpoint diffs, and MCP support make it the most capable in-editor agent for actual multi-file work. It's louder than Aider (asks you to approve everything) but that's the right default for an agent with shell access.

Best for: Multi-file refactors and greenfield features where you want to watch each step.

Pricing: Free (Apache-2.0). BYO Anthropic, OpenAI, or local key.

5. Goose — Score: 8.2/10

Goose is Block's open source agent — a desktop app and CLI with a clean extension model built on MCP. It ships with useful defaults (memory, computer control, developer tools) and the desktop UI is the nicest of anything in this list. Slightly less battle-tested than Open Interpreter or Aider for coding specifically, but the architecture is the most future-proof.

Best for: Users who want a polished desktop app and plan to lean on MCP extensions.

Pricing: Free (Apache-2.0). BYO key across most major providers.

6. Openhands — Score: 8.0/10

OpenHands (formerly OpenDevin) is the ambitious one — a full autonomous software engineer that runs in a sandboxed Docker environment, browses the web, edits code, and runs tests. Closer to Devin's demo than any other open source project. When it works it's impressive; when it doesn't,

debugging what the agent tried is a whole activity. Best treated as a research-grade tool you spot-check, not a daily driver.

Best for: Longer autonomous tasks where you'll review the final PR rather than each step.

Pricing: Free (MIT). BYO key — token spend can add up on long runs.

7. Autogpt — Score: 7.0/10

AutoGPT was the project that made "AI agent" a household phrase in 2023, and it's still around — now as a full platform with a block-based workflow builder. It's more "visual agent workflows" than "desktop agent" at this point, and the pivot away from the original CLI loop makes it a different tool than most people remember. Solid if you want a self-hosted no-code agent platform; overkill if you just want a coding assistant.

Best for: Teams building self-hosted agent workflows with a UI non-engineers can touch.

Pricing: Free (Polyform Shield / MIT split). BYO key. Cloud option available.

Comparison table

Tool	Score	Interface	Desktop control	License	Best for
Open Interpreter	9.2	Terminal	Full (OS mode)	AGPL-3.0	Local automation
Aider	9.0	Terminal	Repo files only	Apache-2.0	Serious code work
Continue	8.7	IDE (VS Code / JetBrains)	Editor scope	Apache-2.0	Open source Cursor
Cline	8.5	VS Code	Workspace + shell	Apache-2.0	Multi-file refactors
Goose	8.2	Desktop app + CLI	Full via extensions	Apache-2.0	Polished UX + MCP
Openhands	8.0	Web UI + sandbox	Sandboxed VM	MIT	Autonomous PRs
Autogpt	7.0	Self-hosted platform	Via blocks	Polyform / MIT	No-code workflows

Final picks

- **If you want the safest starting point:** Aider. It commits every change, so nothing is unrecoverable, and it's the tool most heavy users still reach for.
- **If you want an actual desktop agent:** Open Interpreter or Goose. Open Interpreter is more battle-tested; Goose is the nicer product.
- **If you want Cursor without the subscription:** Continue for autocomplete + chat, Cline when you want it to drive.
- **If you want to see how far autonomy goes:** Openhands. Set a budget cap and let it run.

All seven are free to run against your own API key or a local model. The real cost is inference, and that's a knob you control. Start with one, get it working on real tasks, then add the next when you feel a specific gap.

Generated 2026-07-29 · Read the live article at scored-tools.netlify.app/blog/best-open-source-ai-desktop-agent-tools-2026