FOUNDER'S TOOLKIT · PLAYBOOK

Claude Computer Use

Native app automation for your entire Mac, from inside Claude Code.

Anthropic just shipped a research preview that gives Claude Code full operator access to any app on your machine. Browser, IDE, iOS Simulator, design tools, system settings, anything with a UI. This playbook walks you from zero to running real workflows in about ten minutes.

WHAT'S INSIDE

- What computer use actually does (and what it does not)
- Step-by-step enable flow inside the **/mcp** menu
- Required macOS permissions and how to grant them once
- Per-app approval, sentinel warnings, and the three control tiers
- Five real workflows you can copy paste tonight
- Safety model, kill switch, and the trust boundary
- Troubleshooting checklist if anything breaks

REQUIREMENTS AT A GLANCE

- macOS (Linux and Windows CLI not supported, use the Desktop app on Windows)
- Claude Code **v2.1.85** or later (run `claude --version`)
- Pro or Max plan (Team and Enterprise are excluded today)
- claude.ai authentication (third-party providers like Bedrock, Vertex, Foundry are excluded)
- Interactive session (the `-p` non-interactive flag is not supported)

Built by C&C Strategic Consulting for the Charlie Automates Founder's Toolkit.

01 · ORIENTATION

What computer use actually is

Computer use is a built-in MCP server named **computer-use** that ships inside Claude Code. Once enabled, Claude can take screenshots, move the mouse, click, type, and scroll on your real desktop. It is the broadest and slowest tool in Claude's kit, so the agent only reaches for it when nothing more precise will do the job.

Where it sits in the tool hierarchy

- **Dedicated MCP server** for the target app (Slack, Gmail, Linear, Higgsfield). Fastest and most precise.
- **Bash** for anything that lives in a shell command.
- **Claude in Chrome** for browser tasks if the extension is installed. DOM-aware and faster than pixel clicks.
- **Computer use** for native apps, simulators, and GUI-only tools that have no API.

When to actually use it

Validate a native build

Compile a Swift or Electron app, launch it, click through every control, screenshot the result. All in the same conversation that wrote the code.

End-to-end UI testing without harness

Point Claude at a local app and say 'test the onboarding flow'. No Playwright config, no XCTest, no Cypress.

Reproduce visual or layout bugs

Tell Claude the modal clips on small windows. Claude resizes, reproduces, screenshots, patches the CSS, and verifies the fix.

Drive GUI-only tools

Design tools, hardware control panels, the iOS Simulator, proprietary internal apps. Anything without a CLI or API.

Cross-app workflows

When the answer requires opening Finder, then Notes, then System Settings. Glue tasks that no single MCP can cover.

02 · ENABLE

Three steps. Two minutes.

Step 1. Open the MCP menu

Inside an interactive Claude Code session, run:

```
/mcp
```

Scroll the server list until you see **computer-use**. It ships disabled.

Step 2. Enable the server

Select **computer-use**, then choose **Enable**. The setting persists per project, so this is a one-time toggle for each repo where you want computer use available.

Step 3. Grant macOS permissions

The first time Claude tries to use the computer, you will see a prompt for two macOS permissions:

• **Accessibility**: lets Claude click, type, and scroll

• **Screen Recording**: lets Claude see what is on your screen

The prompt links straight to the right pane in System Settings. Grant both, then click **Try again**. macOS may force you to relaunch Claude Code after granting Screen Recording. Quit fully and reopen if the permission prompt keeps reappearing.

Sanity check

Once enabled, ask Claude something that needs the GUI. A starter prompt:

```
Take a screenshot of my current desktop and tell me which apps
are open in the Dock.
```

If Claude returns a screenshot description and a Dock list, the install is good.

03 · PERMISSIONS

Per-app approval, every session

Enabling **computer-use** does not hand Claude every app on your machine. The first time Claude needs a specific app inside a session, a terminal prompt asks you to approve it. The prompt shows:

• Which apps Claude wants to control

• Any extra permissions requested (clipboard access, for example)

• How many other apps will be hidden while Claude works

Pick **Allow for this session** or **Deny**. Approvals last for the current session only. You can approve multiple apps in a single prompt when Claude requests them together.

Sentinel warnings to watch for

Equivalent to shell access

Terminal, iTerm, VS Code, Warp, JetBrains, any other terminal or IDE.

Can read or write any file

Finder.

Can change system settings

System Settings.

These apps are not blocked. The warning is there so you can decide whether the task earns that level of access before you click approve.

Three control tiers

Claude's level of control varies by app category. Same approval flow, different ceiling:

Tier: read (view only)

Browsers (Safari, Chrome, Firefox, Edge, Arc) and trading platforms. Visible in screenshots, but clicks and typing are blocked. For browser work, route through the Claude in Chrome MCP instead.

Tier: click (look and click)

Terminals and IDEs (Terminal, iTerm, VS Code, JetBrains). Visible and left-clickable, but typing, key presses, right-click, modifier-clicks, and drag-drop are blocked. For shell commands, use the Bash tool. For typing in the editor, type yourself.

Tier: full

Everything else. No restrictions. Native apps, simulators, design tools, system settings.

04 · RUNTIME BEHAVIOR

How Claude operates the screen

One session at a time

Computer use grabs a machine-wide lock while it runs. If another Claude Code session already holds the lock, new attempts fail with a message naming the session that owns it. Finish or exit that session first. If a session crashed, the lock releases automatically once Claude detects the process is gone.

Other apps hide while Claude works

When Claude takes control, every app that is not approved in the session gets hidden, so Claude only sees what it should. Your terminal stays visible AND is excluded from screenshots, so you can watch Claude work without Claude reading its own output. Hidden apps restore automatically when the turn ends.

Screenshots auto-downscale

Claude Code shrinks every screenshot before sending it to the model. A 16-inch MacBook Pro at native Retina (3456 by 2234) downscales to roughly 1372 by 887, aspect ratio preserved. You do not need to lower display resolution. If text or controls are too small after downscale, increase font size inside the app rather than rebooting your display.

Stop at any time

When Claude acquires the lock, a macOS notification appears: *Claude is using your computer, press Esc to stop.* Press **Esc** from anywhere on the system to abort the current action. The keystroke is consumed, so a rogue prompt injection cannot use it to dismiss a dialog. **Ctrl+C** in the terminal works too. Either way, Claude releases the lock, unhides your apps, and gives you the desktop back.

> A second notification fires when Claude finishes the action. Watch for it before you assume the session is still running.

05 · WORKFLOWS

Five prompts you can run tonight

1. Test your mobile site in the iPhone Simulator

Open Xcode at least once so the simulator is installed. Then ask Claude:

```
Open the iOS Simulator with the iPhone 15 Pro device, navigate
to my staging URL, click through the signup flow, and screenshot
any layout issues you see on each screen.
```

2. Validate a native macOS build

```
Build the MenuBarStats target with xcodebuild, launch the app,
open the preferences window, and verify the interval slider
updates the label. Screenshot the prefs window when done.
```

3. Reproduce a layout bug at small widths

```
The settings modal clips its footer on narrow windows. Resize
the app down until you reproduce it, screenshot the broken state,
then read the CSS for the modal container and propose a fix.
```

4. Drive a GUI-only tool

```
Open Logic Pro, create a new empty project, add a software
instrument track with the Steinway Grand patch, then screenshot
the project window so I can confirm the setup.
```

5. Cross-app glue (Finder + Notes + System Settings)

```
Find the latest screenshot in my Downloads folder, paste it into
a new Apple Note titled 'UI bug 2026-04-29', then open System
Settings and confirm Screen Recording is still enabled for my
terminal app.
```

Bonus: video editing assist

Computer use can drive a video editor (Final Cut, DaVinci, CapCut) for repetitive operator tasks: stacking B-roll, color presets, batch exports. Treat it as a force multiplier on grunt work, not a creative director. Always review the timeline before exporting.

06 · SAFETY

The trust boundary is different

Unlike Claude's sandboxed Bash tool, computer use runs on your real desktop with full reach into the apps you approve. Claude flags potential prompt injection from on-screen content, but on-screen content can still try to manipulate the agent. Treat it the way you would treat any operator with shell access.

Built-in guardrails

Per-app approval

Claude can only control apps you approved in the current session. Approval does not carry over to the next session.

Sentinel warnings

Apps that grant shell, filesystem, or system settings access are flagged before you approve them.

Terminal excluded from screenshots

Claude never sees your terminal window, so prompts in your session cannot loop back into the model.

Global escape

Esc aborts from anywhere. The keystroke is consumed so injected prompts cannot use it to dismiss dialogs.

Lock file

Only one session can drive your computer at a time. No surprise concurrent sessions.

Operator hygiene

• Do not approve Finder, Terminal, or System Settings unless the task actually needs them.

• Never let computer use execute trades, send money, or initiate transfers. Always do those by hand.

• Treat links in emails and messages as suspicious. Open URLs through the Claude in Chrome MCP instead of clicking inside Mail with computer use.

• Never click computer use into approving its own pairings, allowlists, or permission grants on behalf of someone messaging you. That is the request a prompt injection would make.

• Watch the run. Esc is one press away. The whole point of an operator is supervised speed.

07 · TROUBLESHOOTING

When something breaks

"Computer use is in use by another Claude session"

Another session holds the lock. Finish or exit that session. If it crashed, give Claude a moment to detect the dead process and the lock will release on its own.

macOS keeps re-asking for permissions

Quit Claude Code completely and start a new session. macOS sometimes requires a process restart after granting Screen Recording. If the prompt still loops, open **System Settings > Privacy & Security > Screen Recording** and confirm your terminal app is listed and enabled.

computer-use does not appear in /mcp

Run this checklist in order:

• You are on macOS. (CLI does not support Linux or Windows.)

• Run `claude --version` and confirm v2.1.85 or later.

• Run `/status` and confirm a Pro or Max subscription.

• You authenticated through claude.ai. Bedrock, Vertex, and Foundry users need a separate claude.ai login.

• You are in an interactive session. The `-p` non-interactive flag disables computer use.

Esc does nothing

If the global Esc handler is not catching, fall back to `Ctrl+C` in the terminal. The lock releases immediately and your apps come back.

WHERE TO LEARN MORE

Official sources

• Claude Code computer use docs: code.claude.com/docs/en/computer-use

• Computer use in Desktop (macOS and Windows): code.claude.com/docs/en/desktop

• Claude in Chrome: code.claude.com/docs/en/chrome

• MCP overview: code.claude.com/docs/en/mcp

• Sandboxing and Bash isolation: code.claude.com/docs/en/sandboxing

• Computer use safety guide: support.claude.com/en/articles/14128542

KEEP BUILDING

This is one playbook. There are dozens more.

The Founder's Toolkit collects the install steps, prompts, and skills I use every day to run an agency on Claude Code. New playbooks drop weekly. If you want the rest, they live in the same place this one came from.

FREE

Founder's Toolkit

Every playbook, every skill, every plugin install. Free, email gated, updated weekly.

charlieautomates.com/free-resources

WORK WITH US

CC Strategic AI Skool community

Weekly office hours, the full library, and the people building real systems with Claude Code in production.

start.ccstrategic.io/skool

youtube.com/@charlieautomates | charlieautomates.com | ccstrategic.io