VCI · STANDARD OPERATING PROCEDURE

How to Install Printing Press

Three steps. About five minutes total.

Why this matters

Most apps you care about don't have a clean public API. Without a CLI, your AI agent either gets stuck, burns 35× more tokens, or stops working as tasks get harder. Printing Press solves this with two tools in one:

- The Library — 70+ ready-made CLIs you plug into Claude Code (ESPN, Linear, Notion, Stripe, Shopify, Slack, Hacker News, and more).
- The Factory — a skill that turns any website or API into a brand-new CLI in about 10 minutes.

WHAT YOU'LL HAVE WHEN THIS SOP IS DONE

Claude Code will be able to talk to any of the 70+ pre-built CLIs, and you'll be able to ask Claude to build a new CLI for any tool you use, just by saying so in plain English.

What you should already have

This SOP assumes you've gone through the VCI workshop and already have:

- ✅ VS Code installed
- ✅ Claude Code installed and signed in
- ✅ Node.js installed (so npx works)

If any of those are missing, install them first using the workshop guides. Otherwise, keep going.

Step 1 Install Go

Go is the only new thing you need to install. It's the language Printing Press is built in. Free, made by Google. Install once and forget about it.

1. Open https://go.dev/dl/ in your browser.
2. Download the installer for your machine:

- Windows — the .msi file (e.g., go1.26.3.windows-amd64.msi)
- Mac (Apple Silicon — M1/M2/M3) — the .pkg file ending in darwin-arm64.pkg
- Mac (Intel) — the .pkg file ending in darwin-amd64.pkg

3. Run the installer. Click Next/Continue/Install on every screen. Defaults are fine.
4. Close any open VS Code windows so they pick up the new Go install.

To confirm Step 1 worked: open a fresh VS Code terminal (Terminal menu → New Terminal) and type:

```
go version
```

✅ **STEP 1 SUCCESS**
You see something like 'go version go1.26.3 windows/amd64' (or the Mac equivalent). Move on to Step 2.

Step 2 Tell Claude Code to install Printing Press

This is the magic step. You don't write any commands yourself. You paste a prompt into Claude Code and Claude handles the rest.

5. Open VS Code.
6. Open the Claude Code chat panel.
7. Start a fresh chat.
8. Paste this exact prompt:

COPY AND PASTE THIS PROMPT INTO CLAUDE CODE

Read these 3 links and install everything I need to use Printing Press locally on this machine:

- https://printingpress.dev/
- https://github.com/mvanhorn/cli-printing-press
- https://github.com/mvanhorn/printing-press-library

Install both the Library (the starter pack of pre-built CLIs — ESPN, Flight Goat, Movie Goat, Recipe Goat) and the Factory (the skill that builds new CLIs). Then run a quick test to prove it's working.

9. Claude reads the docs, asks permission to run install commands, and does the work. Click 'Allow' or 'Yes' each time it asks.
10. The whole step takes 2–5 minutes.

What Claude is actually running (for the curious)

You don't need to memorize these — Claude handles them. Listed only so you can recognize what's happening on your screen.

Install the starter pack of pre-built CLIs:

```
npx -y @mvanhorn/printing-press install starter-pack
```

Install the Factory skill (so Claude can build new CLIs on demand):

```
npx skills add mvanhorn/cli-printing-press/skills -g -a claude-code -y
```

If you'd rather skip the natural-language ask, run the two npx commands above directly in your VS Code terminal. Same outcome.

✅ **STEP 2 SUCCESS**
Claude says something like 'Printing Press is installed. The starter pack and factory are ready.' No red error messages.

Step 3 Test it

Last step. Ask Claude to use one of the new CLIs.

11. In Claude Code, start a fresh chat.
12. Paste any one of these:

    - "What NBA games are on tonight?" — uses ESPN
    - "Find me a flight from Lisbon to NYC in two weeks under $600." — uses Flight Goat
    - "Show me the top 5 recipes for grilled salmon." — uses Recipe Goat
    - "What's trending on Hacker News right now?" — uses Hacker News (install it first if needed)

13. Claude says something like 'I'll use pp-espn to check tonight's games' and returns the answer in 1–3 seconds.

✅ **STEP 3 SUCCESS — YOU'RE DONE**

You get a clean, readable answer back — not a giant blob of code or JSON, and no 'tool not found' errors. You now have an AI agent that can talk to 70+ apps without a public API, and a factory that can build a new CLI for any tool you use, on demand.

Bonus: build your first CLI from scratch

Once Step 3 works, the real fun starts. Pick any tool you use every day that doesn't have a clean API — your CRM, your community platform, an internal dashboard. Then say to Claude:

BUILD ME A NEW CLI

Use the Printing Press Factory to build me a CLI for [tool name]. The link is [paste the URL]. I want to be able to ask you to [list the actions — pull data, post updates, etc.] in plain English afterward.

Claude will research the tool, reverse-engineer the endpoints, generate Go code, build the CLI, and produce a Claude Code skill around it. Average time: about 10 minutes.

Troubleshooting

'go: command not found' or 'go install failed'

Go isn't on your PATH. Restart your terminal (and VS Code). On Windows, open a brand-new PowerShell. On Mac, run: source ~/.zshrc

Claude says it can't find npx

Node.js isn't installed (or isn't on your PATH). Install Node from https://nodejs.org/ (the LTS version), restart VS Code, then re-run Step 2.

Install fails with a GitHub rate-limit message (rare)

GitHub limits how many times an unauthenticated computer can pull files per hour. If you hit this, just wait an hour and re-run Step 2. If it happens repeatedly, ask Claude to 'set up a GitHub token for the Printing Press install' — it'll walk you through it on the spot.

A specific CLI won't return data

That CLI may need its own API key (Stripe, Notion, etc.). Open the CLI's README in the Library and follow its auth section. Add the env var, restart Claude Code, retry.

I'm hitting rate limits

CLIs don't bypass the underlying site's quotas. If you need higher limits, get an account on the underlying service and add your API key.

Daily use, going forward

After this SOP, you don't need to think about Printing Press most of the time. Just talk to Claude in plain English.

- To use a CLI: just ask Claude what you want. ("Pull this week's open questions from my Skool community.")
- To see what's installed: ask Claude to "list the Printing Press CLIs we have access to."
- To build a new CLI: tell Claude the site or tool. It handles the rest.
- To share a CLI with your team: ask Claude to "package this CLI into a private GitHub repo I can share."

Resources

- Printing Press homepage / catalog — https://printingpress.dev/
- The Factory (build new CLIs) — https://github.com/mvanhorn/cli-printing-press
- The Library (70+ pre-built CLIs) — https://github.com/mvanhorn/printing-press-library
- Go installer — https://go.dev/dl/
- Node.js installer (for npx) — https://nodejs.org/