

The Graphify Setup Guide

Install Graphify, map your project once, and stop Claude Code from burning usage relearning your codebase every session.

Why I built my setup around this

Every new Claude Code session starts blind. It has no memory of your project, so before it fixes anything it re-reads files and re-traces how they connect. On a \$20 plan that rediscovery is where a big chunk of your usage goes, and it is the same work over and over.

Graphify is a free tool that turns your codebase into a knowledge graph. You build the map once, Claude reads it, and it goes straight to the files that matter instead of exploring from scratch. Same plan, more real work per session.

This guide is the whole setup: the install commands, what the tool creates, the commands I actually run, and the fixes for when something goes wrong. You should be able to get it working without rewatching the video.

01

How it actually works

1. You install the Graphify package and register it with Claude Code, which adds a `/graphify` skill you can call inside any project.
2. You run `/graphify .` once. It parses your code locally with tree-sitter and builds a full map of where everything lives and how it connects.
3. When you ask Claude to make a change, it reads that map first, so it lands on the right files immediately instead of opening half the project to orient itself.

02

The one thing almost everyone gets wrong

People install it, run it once, then forget it exists and let the map go stale. A map that describes last week's code sends Claude to the wrong place.

WATCH OUT

After a batch of changes, refresh the map with `/graphify ./docs --update` so it only re-extracts what changed, or run `graphify hook install` once so the graph rebuilds automatically on every git commit. A stale graph is worse than no graph.

SOURCE CODE

The full repo, README, and examples:

<https://github.com/Graphify-Labs/graphify>

INSTALL.SH

```
# 1. Install the package (isolated environment, recommended)
uv tool install graphifyy

# Alternatives if you don't use uv:
# pipx install graphifyy
# pip install graphifyy

# 2. Register the /graphify skill with your assistant
graphify install

# 3. Build the map for the current project
# (run this inside Claude Code)
/graphify .
```

Running it once creates a `graphify-out/` folder with three files:

GRAPHIFY-OUT/

```
graphify-out/
├─ graph.html      # interactive browser visualization
├─ GRAPH_REPORT.md # highlights, connections, questions
└─ graph.json      # the full queryable graph
```

Your code is parsed locally, so only documentation and media files touch an API call. It handles 37+ languages plus docs, PDFs, and office files, so the map covers more than just source code.

Set it up in Claude Code or Codex

1. Check you have Python 3.10 or newer, and either uv (recommended) or pipx or pip. If you want to read the source first, clone the repo: `git clone https://github.com/Graphify-Labs/graphify`
2. Install the package: run `uv tool install graphify` in your terminal. If you don't have uv, use `pipx install graphify` or `pip install graphify` instead.
3. Register the skill: run `graphify install`. This wires the `/graphify` skill into Claude Code, Cursor, Codex, and Gemini CLI, so the same map works across all of them.
4. Open the project you want mapped in Claude Code (or Codex), then run `/graphify .` inside it. Give it a moment to scan.
5. Confirm it worked: you should see a new `graphify-out/` folder with `graph.html`, `GRAPH_REPORT.md`, and `graph.json`. Open `graph.html` in a browser to see the map.

04

The commands I actually run

BUILD THE MAP FOR THIS PROJECT

```
/graphify .
```

REFRESH ONLY WHAT CHANGED

```
/graphify ./[FOLDER] --update
```

ASK HOW TWO PARTS CONNECT

```
/graphify query "what connects [THING A] to [THING B]?"
```

TRACE A PATH BETWEEN TWO NODES

```
/graphify path "[START NODE]" "[END NODE]"
```

EXPLAIN ONE THING AND ITS LINKS

```
/graphify explain "[NODE NAME]"
```

When it breaks

ERROR	FIX
command not found: graphify	The tool isn't on your PATH yet. Close the terminal and open a new one. If it persists, ask Claude to add the install path to your shell profile, then reopen the terminal.
/graphify does nothing in Claude Code	You installed the package but skipped registration. Run graphify install, then restart the session so the skill loads.
uv: command not found	You don't have uv installed. Use pipx install graphify or pip install graphify instead, or install uv first.
Claude still opens the wrong files	The map is stale. Run /graphify ./docs --update, or rebuild with /graphify . after big changes.
Nothing appears in graphify-out/	You ran the command outside the project folder. cd into the project root first, then run /graphify . from there.
Python version error	Graphify needs Python 3.10 or newer. Check with python3 --version and upgrade if you're below 3.10.
Docs or PDFs skipped in the graph	Those use an API call to extract. Confirm your assistant has the required key set, then re-run the update on that folder.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>