



SETUP GUIDE · AUGUST 2026

Build your own knowledge graph with **Claude Code**

Four steps, from a policy document to a live Rainbird graph you can question.

The Rainbird skill for Claude · Community edition

The Rainbird skill teaches Claude how to turn expertise into a working Rainbird Knowledge Graph: your own policy, manual or decision process, or public regulation and research it gathers itself. Designed, authored, validated, tested and delivered into your account, with an evidence tree behind every decision.

OVERVIEW

What you are actually doing

You bring the knowledge, or just the question, and an account. Claude does the knowledge engineering: it researches and encodes the sources, designs and authors the graph, validates it against the real engine, writes evals and runs them, red-teams the result with adversarial cases, and delivers the finished project into your Rainbird account. You do not write any RBLang yourself, and you do not need to know how the engine works to start.



Fig. 1 · The whole process. Everything on the next four pages is a detail of this.

The four steps

01	Open two accounts	03
02	Install the skill	03
03	Start your project	04
04	Collect what you built	05

Open two accounts

One gives Claude access to the Rainbird platform to host the graph, the other installs Claude with the Rainbird skill. Both take a few minutes.

ACCOUNT 01

Rainbird Community, free

Sign up at app.rainbird.ai/signup. Claude uses your account to upload and live-test the graphs it builds.

Once you are in, copy your API key from Rainbird Studio, under your profile settings, and keep it handy. Claude will ask for it.

ACCOUNT 02

A Claude plan with skills

Claude Code, as the CLI, desktop or web app, or claude.ai on a plan with custom skills enabled.

We recommend the Claude app together with Claude Code, running **Opus 5** or **Fable 5** in **Ultracode mode**. Knowledge engineering rewards the strongest reasoning you have available.

Drop the skill in

You should have downloaded the skill as a zip file called Rainbird Claude Skill.zip. Install it in whichever client you use. The menu path is the same idea in both: find skills, upload the zip.

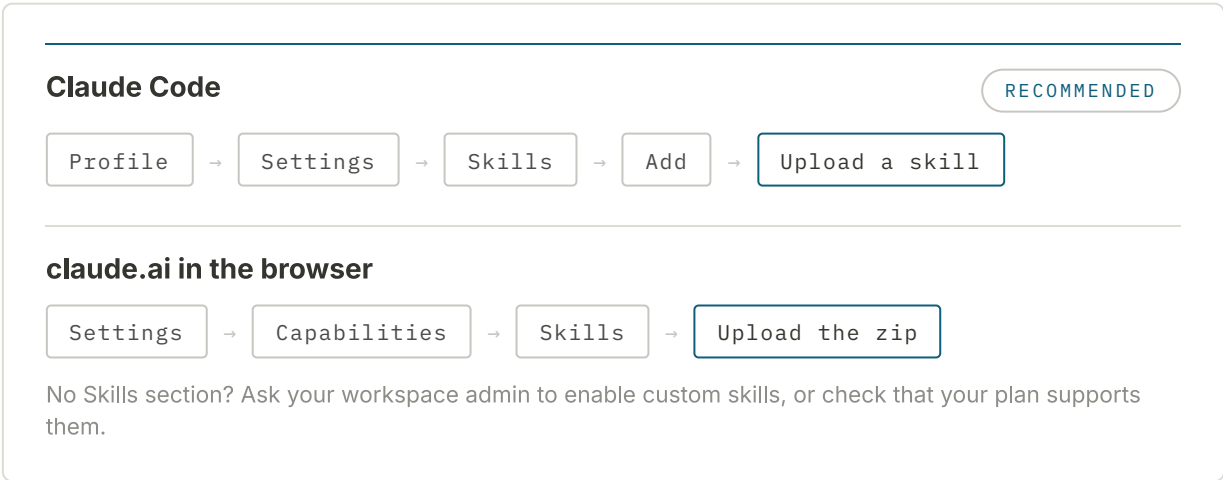


Fig. 2 · Where the upload lives in each client. Menu wording may differ slightly by version.

Start your project in plain words

There is no setup file to edit and no syntax to learn. Open a session, describe the decision you want automated, and answer the questions Claude asks you.

01 Open a fresh session

An empty folder in Claude Code, or a new chat in claude.ai.

02 Say what you want to build

For example: "**Use the Rainbird skill to** build a knowledge graph for our returns-approval policy, here is the policy document." You do not have to supply documentation. Ask instead for extensive research and the skill will encode public regulation, policy and other sources it gathers for itself, or a combination of the two.

03 Hand over your API key when asked

The key stays in your session. It is how Claude validates the graph against the real Rainbird engine.

04 Let it work

The skill makes Claude plan first, then author, then prove the graph. It uploads to your Rainbird account, runs live consultations against its own evals, red-teams the graph with adversarial cases, and hands over only once the decisions come back right.

You can watch it happen. The session reports each stage as it completes, and the test consultations run against your own account, so nothing is handed over on trust.

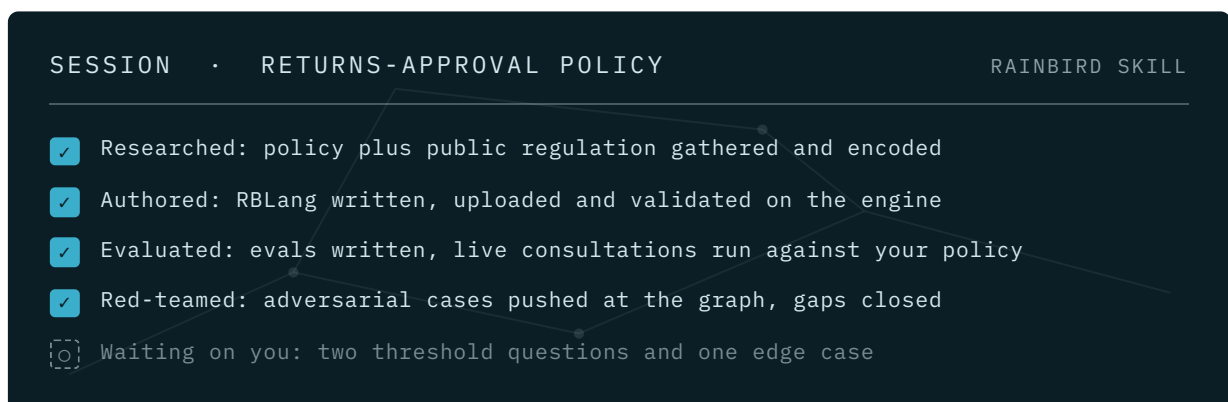


Fig. 3 · What the session reports back as it works. Illustrative.

What you get at the end

The project is delivered straight into your Rainbird account with supporting information like evals and documentation. You can iterate on your project easily by continuing to chat to Claude.

The knowledge graph

Live in your Rainbird account, with the graph ID, plus the RBLang XML source you can import into Studio.

Evals and test results

A JSON suite covering the normal cases and the adversarial ones, ready to import into Studio's automated testing screen and rerun whenever the policy changes.

Documentation

What was built, the sources it was drawn from, the decisions it makes, and example questions to try first.

How good the graph is depends mostly on what you give it and how closely you answer its questions. Three things make the difference.

TIPS

Give it a clear research direction, and/or bring real source material

Public regulation or policy, a policy PDF, a manual chapter, a spreadsheet of rules. The more complete the source, the better the graph, and Claude will ask about the gaps it finds.

Expect questions

Good knowledge engineering is a conversation, so do not be afraid to prompt Claude to come back and quiz you. It will check thresholds, edge cases, and who is allowed to decide what.

Keep your API key private

Treat it like a password. Do not put it in documents or shared files: hand it to Claude in the chat when it asks.

That is everything: community account, API key, drop the skill in, describe your project.

