

Skill to Loop Conversion

Put your working skill on a timer so a fresh batch is waiting on someone's desk before they sit down.

01 When to use this

Use this once a skill runs cleanly in a fresh chat and you want its output to exist without anyone asking for it. It is rung three of the ladder. Good for daily batches, monitoring, and scraping. Not good for anything a teammate needs to share.

02 The prompt

Convert the `<SKILL_NAME>` skill into a loop that runs on a schedule on this machine.

Schedule: `<TIME, e.g. 5:59am>` `<DAYS, e.g. every weekday>`. I want the output sitting there before `<PERSON_OR_ROLE>` starts work at `<START_TIME>`.

The important change: the skill can no longer ask me clarifying questions. It has to run unattended. So before you write it, answer these for me and I will confirm:

1. Batch size per run
2. Copy pool: does it iterate a fixed list, or generate fresh copy each run
3. Where the output goes
4. Whether old batches get cleaned up or kept
5. Daily or weekdays only

On point 2, my answer is regenerate. Do not bake a fixed list of `<COPY_TYPE>` in at build time. Each run the skill should pass through itself and write new `<COPY_TYPE>`, aware of what it already produced in previous runs, so today's batch is different from yesterday's.

Use the schedule skill and set up cron on this machine. Show me the cron entry when you are done, and tell me how to stop it.

03 What it is actually doing

A loop is just a skill with a clock attached. Same work, nobody has to ask for it.

Nick's default time is 5:59am. A creative specialist walks in at 6:00am and the batch is already there. That one minute is the whole idea: the work finishes before the human arrives.

Converting forces one real change, and it is easy to miss. A skill can stop and ask you a question. A loop cannot. Nobody is awake. So the skill needs enough autonomy to make its own choices and keep going.

The specific trap is copy. If your skill iterates a fixed list of headlines that got baked in on build day, every single run produces the same batch. Forever. You need the skill to pass through itself each run and write fresh copy, while knowing what it already made, so today does not repeat yesterday.

Claude will ask you a set of questions during the conversion: batch size, copy pool, output location, cleanup, and daily versus weekdays. Answer them in your prompt and the conversion takes one pass. It uses the built-in schedule skill and sets up cron on your machine.

Know the ceiling. A loop dies when your laptop sleeps. It breaks when something local changes, like a login expiring, a plugin updating, or an MCP server going down. And you cannot hand it to a teammate. When those limits start to hurt, you climb to a routine.

04 How it flows



05 Step by step

01

Confirm the skill is solid

Run it once more in a fresh chat before you schedule it. A broken skill on a timer just breaks quietly every morning.

02

Pick the time backwards

Start from when a human needs the output and subtract. Nick uses 5:59am for a 6:00am arrival. The output should already be waiting.

03

Pre-answer the conversion questions

Batch size, copy pool, output location, cleanup, and cadence. Put your answers in the prompt so Claude does not stall waiting on you.

04

Choose regenerate over fixed

This is the one that matters. Tell it to write new copy each run rather than iterating a list baked in at build time, or every day is a carbon copy.

05

Give it memory of past runs

The skill needs to know what it already produced. Otherwise fresh copy still drifts back to the same few ideas.

06

Let it wire cron

Claude uses the built-in schedule skill and creates the cron job on your machine. Ask to see the entry so you know what is actually running.

07

Learn how to stop it

Before you walk away, get the command that disables the loop. A loop you cannot turn off is a loop you will grow to resent.

06 Words explained

Loop	A skill that fires on a timer on your own computer, without anyone typing a command.
Cron	The scheduler built into your machine. You give it a time and a command, and it runs that command at that time.
Unattended	Runs with nobody watching. It cannot pause to ask a question, so every decision has to be settled in advance.
Copy pool	The set of headlines and calls to action a run draws from. Fixed means the same list forever. Regenerated means new writing each time.
Cadence	How often it runs. Daily, weekdays only, hourly. Pick the slowest one that still gets the job done.
MCP server	A small local service that gives Claude access to an outside tool. If it stops running, anything depending on it stops too.

07 Watch out for

- A fixed copy list baked in at build time means every run produces the exact same batch. This is the failure people notice on day two.
 - The loop dies with your laptop. Sleep, shutdown, or a closed lid and nothing runs.
-

- Local changes break it silently. An expired login, an updated plugin, or a stopped MCP server, and it fails with nobody watching.
- You cannot share a loop with your team. It lives on your machine only.
- The loop command has roughly a one minute floor for intervals, so anything faster than that is not a loop problem to solve.

08 How it connects

Prompt to Skill Conversion

A loop is only as good as the skill under it, so the skill has to pass the fresh-chat test before you put a clock on it.

Loop to Cloud Routine Conversion

Every limit of a loop, the dead laptop, the brittle local setup, the team that cannot use it, is the reason the next rung exists.

Automation Ladder – Prompt Skill Loop Routine

This is rung three, and the conversion rule it enforces is that the skill must produce something different each run.

Ad Creative Generation Pipeline

The creative pipeline is the build Nick took up this rung, firing at 5:59am so a fresh batch of ads was ready each morning.

Credit. Framework and approach by Nick Saraev, from the skill to loop rung of the automation ladder in his Zero to One course on Claude Code for marketing. Written up for the Kape vault by BrewedOps.