



Know What Worked

Tag every link so your analytics can finally tell you which post drove the traffic.

Most people post a link and never find out what it did.

The post gets some views, maybe a few clicks, then nothing. No way to tell which post sent someone to your site, and which one was just noise.

I tag every link I post. It is the only reason I can look at a week of content and say "that one drove the visit, that one drove the lead."

This is that, handed over. It tags your links the same disciplined way I tag mine, so your analytics can connect a post to a result instead of leaving you guessing.

It almost felt wrong to give away. It is part of how I prove my work. But the people who run it and realize how blind they were flying are exactly the people I want to know.

Built by Adam Sprague, Hit My Algo. @SpragueSocials.

READ THIS FIRST

This one is low-risk, and I will tell you exactly why so you are not guessing.

It only edits links you hand it. It does not touch your social accounts. It does not post anything. It does not drive your browser. It reads a link, adds tracking tags, and gives it back. That is the whole job.

Still, read this entire document before you build it, even the part written for the machine. Knowing what a tool does before you use it is the habit that keeps you safe with every tool, not just this one. Anyone who tells you to download something and run it without reading it first is the one to be careful with.

One real requirement: you need Google Analytics (GA4) on your website. The tags this adds are how GA tells you which post drove the visit. No GA4, no payoff. If you do not have it yet, that is a separate short setup, and this assumes it is already there.

PART 1: START HERE

What you're building

A tagger.

You give it a social post with a link to your own site. It hands the post back with the link tagged, so your analytics can see exactly where the visit came from.

Same post. Same link. Now it reports.

Post it wherever you want, by hand or through any scheduler. The skill just makes the link smart first.

Run more than one site? It handles all of them, and you can add a new one anytime, no rebuild.

What you need first

Two things:

1. The Claude desktop app, in Cowork mode. (Cowork is the part of the app that builds and runs skills.)
2. Google Analytics (GA4) live on your website.

That is it. No browser extension, no account access. This never leaves the chat.

The build

1. Download this file.
2. Open a new Cowork chat and attach it.
3. Type: `/skill-creator`
4. Hit enter.

(`/skill-creator` is built into Cowork. You already have it, nothing to install.)

It reads this document, asks you a couple of quick questions (your website domain, the platforms you post on), and builds your tagger.

How you'll know it worked

You paste a post with your link in it.

It hands back the same post with the link tagged, ready to copy.

It tells you the campaign name it used, and exactly where to look in GA4 to see the result.

A few days later you open that report and see which posts sent real people to your site, and which ones did not. That is the thing almost no one posting has. Now you do.

PART 2: FOR CLAUDE, BUILD INSTRUCTIONS

You are reading this because the user attached this file and ran /skill-creator. Your job is to build them a personal link-tagging skill. Follow this exactly.

Step 1: Ask the user these questions first (one short batch, then build)

Present every question that has set options as clickable multiple-choice answers, with the recommended option first and an "Other" choice so they can type their own. Only the domain needs to be typed.

1. What is your website domain? (e.g. yoursite.com) List more than one if you own several. The skill will only ever tag links on these domains, never anyone else's. They can add more sites later anytime, with no rebuild.
2. Do you have Google Analytics (GA4) on that site? (Yes / Not sure / No.) - If "Not sure": tell them how to check in one line. Open the site, view the page source, and search for "gtag" or a tag that starts with "G-". If it is there, they have GA4. - If "No": tell them plainly that this skill only pays off once GA4 is on the site, and point them to set that up first. Still build the skill so it is ready for when it is.
3. Which platforms do you post on? (X, LinkedIn, Threads, Facebook, Reddit, Instagram, Substack, YouTube, others.) This sets utm_source correctly per platform.

Use their answers to fill in the skill below. Do not invent specifics. The domain is required; if they skip it, ask once more before building.

Step 2: Build a skill with this SKILL.md (personalized from their answers)

Name it something like "link-tagger" or whatever the user prefers. First check whether a skill with that name already exists; if it does, use a distinct name (like "link-tagger-mine") or ask, so you never overwrite an existing skill.

The skill should do the following, every run, written for the user's domains and platforms.

THE JOB: Take a social post (or a bare link) plus the platform it is going on. Return the post with the owned-domain link tagged, so the user's GA4 can attribute the visit to that exact post. It tags only. It never posts.

OWNED LINKS ONLY: Tag only links whose host matches one of the user's saved domains (from the domains file below). Leave every other link untouched. Never tag a third-party or citation link. If a post has no owned link, say so and tag nothing.

DOMAINS (keep them editable, never hard-code them into the skill body): Save the user's domains to a small config file in their working folder, one per line (for example tracked-domains.txt), and read that file at the start of every run so the list is always current. Let the user change it in plain language anytime: "add yoursite2.com", "also track clientsite.com", or "stop tracking oldsite.com". Update the file and confirm the new list back to them. If the user wants to tag a link on a domain that is not in the list just once, ask whether to add it permanently or use it for this one tag only, then act accordingly. This way they never rebuild the skill to track a new site.

THE TAG (use this exact convention so the data joins): append three parameters to the owned link, preserving any existing query parameters (do not clobber them): - utm_source = the platform, mapped: X → x, Threads → threads, LinkedIn → linkedin, Reddit → reddit, Instagram → instagram, Facebook → facebook, Substack → substack, YouTube → youtube. Unknown platform → lowercase it and turn any non-alphanumeric run into a single hyphen. - utm_medium = social (always). - utm_campaign = the piece slug (see next).

THE SLUG (this is the part that makes it work): derive a stable slug for the piece. Auto-suggest it from the post's headline or main idea: lowercase, replace each run of non-alphanumeric characters with a single hyphen, trim hyphens off the ends, cap at 60 characters (empty → "untitled"). Then **SHOW** the user the suggested slug and let them rename it. Always do both: suggest, then let them override.

Whatever slug they settle on, reuse the **SAME** slug for that piece on every platform. That is what lets one piece's results add up across X, LinkedIn, and Threads, while utm_source keeps the platforms apart. Keep a short running list of slug-to-piece in their working folder so they do not accidentally reuse one or rename a piece mid-campaign.

HOW TO BUILD THE TAGGED URL (do it safely): parse the link as a URL, set utm_source / utm_medium / utm_campaign on its query string without removing existing parameters, and return the full URL. If the link has no scheme, add https:// first. Never hand-concatenate in a way that breaks an existing "?".

OUTPUT each run: - The post with the tagged link swapped in, ready to copy and paste. - The exact campaign slug used. - One line on where to read it: "In GA4: Reports → Acquisition → Traffic acquisition → set the dimension to Session campaign → find '[slug]'. That row is the sessions and key events this post drove." Give the data a day or two to populate.

PLATFORM NOTE: Links work on all of these platforms. On X, an external link in the main post can soften reach, so many people put the link in a reply instead. The tracking still works in a reply, so tag it the same way and tell the user they can move the link to a reply if reach matters to them.

KEEP IT HONEST: The skill's job ends at a correctly tagged link and a clear place to read the outcome. Never claim a result it cannot see. It tags; the user posts.

Step 3: After you build it

Tell the user, in plain language: - What the skill is called and how to use it (e.g. "paste a post with your link and tell me the platform"). - That it tags only their own domains and never posts anything. - That the same slug should ride along for the same piece across every platform, so the numbers add up. - Where to look in GA4 after a few days, and to give it a day or two to show data. - That they can add or remove a tracked site anytime by just asking, with no rebuild.

Keep it short. Then offer to tag their next post with them as a first run.

Voice note for Part 1 copy: no em dashes, no "actually", never "ChatGPT" alone, one idea per line with a blank line between where it punches, no engagement-bait, no invented numbers.