

AI Use Levels

A scheme, versioned like a licence, for saying what one piece of work owes to generative AI: a human form to type, a machine form for commit trailers and metadata, tokens that never change meaning, and four marks for the Level alone.

Revised 27 October 2026: Patches 1.1.1 to 1.1.4 changed the alt-text wording, added the length reason for withdrawing the robot-face mark, reordered the sections with Prior art last, and described the braille as spelling the level token, with the level names added in braille. No token or definition changed. Patch 1.1.5 reworded section 10 rule 5: its candidates are now for a later minor version (1.2 or after).

"I used AI for this" has stopped meaning anything. The same sentence covers a chatbot that polished one paragraph and an agent that ran all night with access to production. AI Use Levels is a tag of four words in a fixed order, Level · Hand · Door · Tool, that says in one breath what one piece of work owes to generative AI: whether the tool could change anything by itself, how much of the result is its output, whether what it did can be taken back, and what the tool was. SAE rates the car. This rates the trip. A tag describes what happened on one piece of work, never what the tool could have done.

Canonical page: <https://papers.4thcltr.com/ai-use-levels/>. Machine-readable form: spec . j son in this repository. Licence: CC BY 4.0.

1. Identity

FIELD	VALUE
Name	AI Use Levels. A tag is a <i>use tag</i> . A piece of work is "a level-two use"

FIELD	VALUE
Version	1.1. The scheme is versioned. A use tag is not dated (section 9)
Purpose	Say, in one breath, what one piece of work owes to generative AI, so two people can tell whether they are talking about the same kind of use
Principle	The same tool is L1 on Monday and L3 on Tuesday, and both tags are true
What it is not	A permission system, an audit or a liability column. It records the chance a person had to intervene, not how well they used it
The four questions	Could it change anything by itself? Did each change wait for a yes? How much of the result is its output? Could it be undone?

2. Syntax

The human form is the canonical one. Fixed order, any separator people already type:

L2 · made · undo · Claude Code

Spoken: "level two, made, undo, Claude Code". As a sentence: "Claude Code made it under supervision, all under undo."

SLOT	VALUES	REQUIRED
Level	L0 L1 L2 L3 , or the level's name	always
Hand	touched drafted made	recommended
Door	undo no undo	at L2 and L3; omitted at L0 and L1, where there is nothing to undo
Flag	unread	only when true
Tool	free text, last; optional reach in parentheses: (chat) (IDE) (app) (API) (agent)	always

The shortest honest tag is L1 · ChatGPT. The full tag is Level · Hand · Door · Flag · Tool.

The machine form goes in a commit trailer, front matter, an HTML meta or an image sidecar. Same tokens, tool in parentheses:

```
AI-Use: L2 made undo (Claude Code)
```

Pattern: AI-Use: <level> [hand] [door] [flag] (<tool>) [v<major>].

Every token identifies itself. An L and a digit is a Level, a Hand word is a Hand, a Door word is a Door, `unread` is the flag, and parentheses hold the tool. A parser reads by token, not by position. The tool runs from the first opening parenthesis to the last closing one, so a reach parenthetical nests: AI-Use: L1 touched (Copilot (IDE)). A tag with no version token is version 1. A tag under a later version carries its own after the tool: v2. The trailer sits beside git's Co-Authored-By: that trailer names the tool, and this one says what it was allowed to do.

Absence. No tag means nothing claimed. It never means L0.

3. LEVEL (the headline)

The question is whether the tool could change anything by itself, and whether each change waited for a yes. A change is an effect outside the conversation: it wrote, sent, paid, deployed, deleted, published or told someone. Reading, searching and generating text are not changes.

LEVEL	NAME	DEFINITION	TEST
L0	None	No AI touched it	None
L1	Suggested	It could change nothing by itself. A person moved every piece of its output into the work (pasted, typed, accepted, pressed send)	Did anything take effect except by a person's own action? No
L2	Supervised	It made changes itself, and each change waited for a person's yes before it took effect	Could a person have stopped every change before it counted? Yes

LEVEL	NAME	DEFINITION	TEST
L3	Unattended	It made changes itself without waiting for anyone	Did anything take effect before a person could stop it? Yes

The flag `unread` means no accountable person has read the result. It attaches to any level.

L1 · made · unread · DeepL is a translation posted blind. L3 · made · no undo ·

unread is the bot that merged to production. The flag is a state, not a level: it comes off when a person reads. A level never changes after the fact.

Rules:

One tag per piece of work. Where the steps differed, tag the highest level reached by any step that changed something. A read-only step raises nothing.

Granularity is the person's. One yes to a plan of forty changes is L2: the person could have said no before anything happened. Four hundred yeses in two hours is still L2. The tag rates the opportunity to intervene, not the diligence. A rubber stamp is still tagged L2; the scheme cannot tell diligence from its absence.

A person is someone accountable for the work. A model reviewing a model lowers nothing.

A run that changed nothing outside the conversation is L1, however long it ran. A research agent that worked for twenty minutes and produced a report is L1 · made.

The numbers do not map to SAE J3016. SAE L3 is the handover level. This L3 means only that it acted without waiting.

4. HAND (its hand in the result)

Hand says how much of the final is the machine's output. It rates production, not ideas: a transcript, a translation and a summary are all `made`, and the reader knows from the genre whose ideas they were. Hand is independent of Level. A person can press every button and still publish something the machine made whole. That is L1 · made, the case most often left untagged.

TOKEN	DEFINITION	TEST
<code>touched</code>	Corrections and local changes to a person's material: polish, autocomplete, spot fixes, formatting	Would a reader recognise anything of its output in the final? Barely
<code>drafted</code>	It produced a first version a person substantially reworked	Its shape survives; most of the words or code are a person's
<code>made</code>	The final is substantially its output; a person chose, read, maybe trimmed	Its words, code or pixels survive

Rules:

Judge the final, not the interactions. Two hundred tab-accepts that add up to most of the file are `made`.

Choosing is not reworking. One picked from forty generations is `made`. Inpainting a hand is still `made`.

In doubt between two rungs, pick the larger.

Hand has three rungs and not one label because one word cannot carry this. Meta's single "Made with AI" label landed on retouched photos and was renamed within months.

5. DOOR (what it did that cannot be taken back)

Door is tagged only at `L2` and `L3`, because at `L1` the machine changed nothing. It rates reversibility, with reach folded in.

TOKEN	DEFINITION	TEST
<code>undo</code>	Everything it changed can be reversed by the tagger alone, asking nobody: a branch, a versioned document, a trash can, a staging system nobody else saw	Could you put it all back in minutes without telling anyone? Yes
<code>no undo</code>	At least one change cannot be taken back: sent, paid, deployed, deleted without backup, published, told someone. Reaching a person or system you do not control always counts, even if a rollback exists, because they saw it	Did anything leave your control? Yes

Rules:

- Tag the worst door opened, not the typical one. One sent email makes the whole piece `no undo`.
- Door rates the machine's changes, not the person's. If the AI drafted an email and a person reworked it and pressed send, the send is the person's act: `L1 · drafted, no door`. The irreversibility belongs to the person, and the tag says so by leaving the slot empty.

6. TOOL (free text, last)

- Name the tool as specifically as you can: Claude Opus 5.5, Claude Code, Copilot (IDE), GPT-5 (API). A product name is acceptable. A model version is better, because it fixes the date. How the tool was reached goes here as an optional parenthetical: (chat), (IDE), (app), (API), (agent). Reach is a property of the tool, and its risk is already carried by Level and Door, so it earns no slot of its own.
- Several tools: name the one with the biggest hand first, then the others after a plus: `L2 · made · undo · Claude Code + Copilot`.

7. Worked examples

Each tag follows from the tests in sections 3 to 5. (tool) stands for whichever tool it was.

USE	TAG
A chatbot polished one paragraph	<code>L1 · touched · ChatGPT</code>
Autocomplete suggested a few lines	<code>L1 · touched · Copilot (IDE)</code>
I asked for the whole post and posted it as-is	<code>L1 · made · ChatGPT</code>
Grammarly fixed typos as I typed	<code>L1 · touched · Grammarly</code>
One of forty generated images, unedited	<code>L1 · made · Midjourney</code>
A research agent ran twenty minutes, changed nothing, I used the report	<code>L1 · made · Deep Research</code>

USE	TAG
Machine translation posted in a language I cannot read	L1 · made · unread · DeepL
Coding agent, permission prompt on every edit, I read each diff	L2 · made · undo · Claude Code
Agent proposed an infrastructure plan, I approved once, apply made forty changes	L2 · made · no undo · (tool)
Agent ran overnight on a branch and opened a PR I then read	L3 · made · undo · Claude Code
Lint-fixer ran unattended on a branch; I read, then merged	L3 · touched · undo · (tool)
Agent with production and email access ran unattended; I read the logs after	L3 · made · no undo · (tool)
Bot auto-merged on green CI; nobody has read the diff	L3 · made · no undo · unread · (tool)
Meeting bot emailed its summary to every attendee before the host saw it	L3 · made · no undo · unread · (tool)
AI drafted the email, I reworked it and pressed send	L1 · drafted · Claude

The same examples are in `examples/examples.json` with their machine form, and a test holds the two lists equal.

8. Marks

A mark is a symbol for the Level alone, for places with no room for the full tag: a social bio, an image overlay, a chat display name, a slide footer. The circle is the work and the dot is the tool. The four marks differ only by where the dot is, so they read without colour, without letters and in any language.

LEVEL	MARK	WHERE THE DOT IS	NAME	SPOKEN
L0 None	○	no dot	AI USE LEVEL ZERO SYMBOL	"AI use level zero, None"

WHERE	MARK	FULL TAG BEHIND IT
A blog post byline, linked	L1 mark	L1 · drafted · Claude Opus 5.5
A slide footer	L2 mark	L2 · made · undo · Claude Code in the speaker notes and the deck's sources slide
An overlay on a generated image	L1 mark	L1 · made · Midjourney in the image's alt text and sidecar
A support bot's display name	L3 mark	the dated standing tag on the help page: L3 · made · no undo, since June 2026

The robot-face mark proposed for 1.1 (🤖 2) is withdrawn. It is no shorter than L2, an emoji depends on the platform's drawing, and a screen reader reads it as "robot face".

9. Dated or versioned?

Version the scheme. Do not date a use tag. Date a standing tag.

1. The scheme is versioned: 1.0, 1.1 and so on, like CC 4.0. A tag with no version token is version 1.
2. A use tag on one piece of work is not dated. The artifact carries the date: the post, the commit, the document, the release. Tokens never change meaning (section 10), so an old tag reads correctly forever. A hand-typed tag must stay short.
3. The tool name is versioned where it can be. Claude Opus 5.5, not Claude. The model name is the date.
4. A standing tag on a system or a policy is dated. "Support bot runs at L3 · made · no undo, 2% read, since June 2026." "Production changes are L2 or lower (policy of 2026-10)." A stance gets revised, and a reader needs to know which one was in force. The standing tag is also the fleet form: ten thousand chats get one dated standing tag, not ten thousand use tags.
5. An artifact with no date of its own carries one beside the tag: a bio, an evergreen page, a screenshot that will be reposted.

The objection that the same agent suggests today and executes unattended tomorrow is answered by the principle, not by a date. The tag describes one piece of work.

10. Extension rules

1. A published token never changes meaning. A bad token is fixed by adding a new token and deprecating the old one. Nothing is redefined and nothing is renumbered.
2. Levels grow upward only. The first candidate for L4 is *it chose the task*. Every current level assumes a person set the goal. A proactive agent that decides what to do next is a new rung, not a flag.
3. A new slot carries its own closed vocabulary, and no word is shared between slots, so every token still identifies itself and order stays a reading convention. New slots go before the tool.
4. A minor version adds tokens, flags or slots and stays backward compatible. A major version is a last resort. Deprecate, never remove.
5. Candidates for a later minor version (1.2 or after), left out to keep the tag at four words: checked (tests, a fact-check or a second reader ran; the answer to whether the person could even read it), data (the material left your machine, or went to training), whose (my things, our things, other people's things), and duration.
6. Governance. The specification is public and forkable, with a changelog. Proposals are issues. A token is accepted when three real uses cannot be tagged honestly without it.
7. A mark (section 8) is not a token, so the three-uses test in rule 6 does not apply to it. The mark set is fixed for every 1.x version. A new mark is added only with a new level, and a published mark never changes its level.

11. Known limits

It rates opportunity, not diligence. A rubber stamp at L2 tags the same as a real review.

Hand is self-reported and will be shaded downward. CRediT had the same problem and spread only when publishers required it.

It says nothing about what the tool was *permitted* to do. That is a governance question (the Cloud Security Alliance framework, Zheng et al.) and belongs in a dated standing tag, not a use tag.

It does not establish liability. It does show where liability needs an owner: every L3 · no undo.

12. Route to a standard

The aim is for the four marks to be encoded in Unicode, as the Creative Commons symbols were: ☒ and ☑ entered Unicode 13.0 in 2020, years after the licences were in wide use. The Unicode Technical Committee (UTC) encodes a symbol when there is evidence that people other than its authors use it in running text. The order of work:

1. Publish the marks: the SVGs in this repository under CC0, then a small open font that maps them to Private Use Area code points until real ones exist.
2. Use them on the 4th Culture sites and in the README of each gorfednet repository whose commits carry the AI-Use trailer.
3. Keep a public log of third-party use with links and dates, opened as issues in this repository.
4. When the log shows use by unrelated publishers, submit a UTC proposal for four characters with the names in section 8, the spoken forms, the reference glyphs and the log as evidence.

Alongside that:

Machine form. Register AI-Use as an HTML meta name on the WHATWG MetaExtensions list, and propose the tag as a value set to IPTC beside its digital source type, which C2PA manifests already reference.

Commit trailer. AI-Use follows git's trailer syntax and needs no registration. Tools that read Co-Authored-By are asked to read it too.

Nothing has been submitted to any body yet. Each submission is listed here with its date when it is made.

13. Prior art

SCHEME	WHAT IT RATES	RELATION
Sheridan & Verplank 1978; Parasuraman, Sheridan & Wickens 2000 (ten levels of automation)	Who decides and who is told	The lineage. L1 to L3 compress their "executes with veto / executes and informs / does not inform"
SAE J3016 L0 to L5	The vehicle, fixed at manufacture	The analogy. This scheme rates the trip and does not share its numbers
Human in / on / out of the loop (DoD 3000.09, autonomous-weapons policy)	Oversight position	The same idea as L1 to L3. The words were dropped because "on the loop" is jargon and sounds like "in the loop" aloud
AI Assessment Scale, Perkins / Furze / Roe / MacVaugh (five levels)	What a student is permitted to use	Permission, not disclosure; no agents acting on the world
Feng, McDonald & Zhang 2025, operator to observer	The user's role in an agent system	Rates the configuration; the closest in spirit
Mitchell et al. 2025 (Hugging Face), autonomy levels	Agent systems	Argues against the top rung; rates the system
Morris et al. 2023, Levels of AGI (DeepMind)	System capability and autonomy	Rates the system
Cloud Security Alliance, Agentic AI Autonomy Levels v2 (2026)	Enterprise deployments, five dimensions, prescribed controls	Governance. The standing tag of section 9 is its small cousin
Zheng et al. 2026, capability vs permission	Separates could from may	Adopted: a use tag is about did
CRedit	Human authorship roles	Hand is a three-rung CRedit for machines
IPTC digital source type, C2PA	Machine-readable provenance of media	The machine-readable Hand for images. A use tag can sit beside it

SCHEME	WHAT IT RATES	RELATION
Creative Commons	Licence terms	The syntax model: few self-identifying tokens, versioned, a human layer and a machine layer. CC Signals (2025) is about training and is not this
Bezos's one-way and two-way doors	Decision reversibility	Door is that idiom applied to the machine's changes
"Loop engineering", "four levels of agentic loops"	Agent architecture	A name collision only, and the reason the working title "Loop Levels" was dropped

Sources read:

<https://arxiv.org/abs/2403.14692v2>

<https://leonfurze.com/2023/12/18/the-ai-assessment-scale-version-1/>

<https://labs.cloudsecurityalliance.org/research/agentic-ai-autonomy-levels-control-framework-v2-csa-styled/>

<https://www.alphaxiv.org/abs/2607.23438>

<https://medium.com/@tahirbalarabe2/what-is-loop-engineering-four-levels-of-agentic-loops-ccc48b850002>

This document's AI use: L2 · made · undo · Claude Code [\(AI Use Levels 1.1\)](#)

Cite as

Michael Duncan McArthur. AI Use Levels 1.1. 4th Culture, Toronto, 2026. <https://papers.4thcltr.com/ai-use-levels/1.1/>

Downloads and discussion

Licence: [CC BY 4.0](#). Copy, share or adapt it, with a link to this page as attribution.

Published by 4th Culture, Toronto.

papers.4thcltr.com

© 2026 MICHAEL DUNCAN MCARTHUR · 4TH CULTURE