



Terminology Governance Pack

What this is. The vocabulary layer of your knowledge system, made governable: a ~40-entry inventory of imprecise and problematic terms with working replacements, a fillable terminology standard, a gate addendum that makes language a checked property instead of a style opinion, a 30-day pilot blueprint, and the champion kit (deck, brief, and emails) for making the case inside your organization.

When you need it. The first time you notice the same concept living under three names in your knowledge base — or the first time a phrase in a client-facing asset makes you wince. Both are the same defect: ungoverned vocabulary. One costs you retrieval; the other costs you trust.

Which maturity level it serves. Any level — terminology drift starts at Level 0 and compounds quietly at every level after. The pack's enforcement point assumes a working curation gate (Module 4); without one, start with the standard and the audit, and let them motivate the gate.

The argument in one paragraph

Terms are retrieval keys. When your content says *tribal knowledge* in one article, *undocumented process* in another, and *institutional knowledge* in a third, humans get inconsistent search results and machines get three weak matches instead of one strong one — you pay for that as duplicate articles, missed answers, and a higher blended cost per answer. Idiom fails harder: a non-native speaker gets nothing from *open the kimono*, and an AI assembling an answer from your corpus can do no better. Some of these terms carry a second cost — *tribal knowledge* trivializes Indigenous identity, and the knowledge-management discipline itself has already moved off it; a *Mexican standoff* in a client-facing post attaches an ethnicity to dysfunction. You do not need two arguments for two problems. Precision fixes both: governed vocabulary is clearer for people, stronger for retrieval, safer for the brand, and cheaper to answer with.

What's in the pack

File	What it is
<code>term-inventory.md</code>	~40 terms in two tiers — replace-now and judgment-call — each with the precision case, the replacement, and where it hides
<code>terminology-standard-template.md</code>	The fillable standard: canonical terms, deprecated terms, synonym mapping, naming rules, enforcement, change process
<code>examples/meridian-terminology-standard.md</code>	The worked example — Meridian Systems' standard, filled, ratified, and in force
<code>gate-addendum.md</code>	Language line-items for the three curation gates, plus the automation note (a grep list is a linter)

File	What it is
pilot-blueprint.md	The 30-day terminology audit: 20 assets, 2 reviewers, 30 days — selection rubric, scoring sheet, outputs
champion-deck.pptx / .pdf	A 10-slide internal case deck, openly framed, ready to present
champion-brief.md	The two-page written version of the case, for the leader who reads instead of watches
champion-emails.md	Two send-ready notes: the peer-to-leader version and the leadership-forum version
references.md	Ten verified sources — standards bodies, style guides, and the business case

How it maps onto the Loop

Terminology governance is not a new loop — it is a property enforced by the machinery you already run:

- **The standard is content policy.** Your terminology standard is a sibling of the content-standards policy (Module 5): ratified the same way, amended through the same decision log, reviewed on the same cadence.
- **The gate is the enforcement point.** The addendum adds language checks to the existing checklists (Module 4) — a candidate that fails the term scan gets edited at the gate, not debated in a meeting.
- **The templates carry the vocabulary.** Canonical terms live in your metadata and section language (Module 2); the standard names which synonyms map to which canonical term, which is exactly what retrieval — human or machine — needs.
- **The workbook prices the defect.** Duplicate articles under divergent names and known issues that leak back into the assisted queue are findability failures (Module 6); the audit gives you a before/after story the workbook can carry.

Running the program — four phases

1. **Audit.** Run the pilot blueprint: twenty client-facing assets, two reviewers, thirty days. Output: a findings report with severity, not a shame list — the audit measures the corpus, not the authors.
2. **Standard.** Fill the terminology standard from what the audit found. Ratify it like any policy: named owners, a decision noted for every contested term, quarterly review.
3. **Gate.** Install the addendum line-items. From this point, language is checked where content already gets checked — no new meetings, no language police.
4. **Enablement.** The champion kit, the glossary entry in onboarding, and the linter note. The goal is that the standard becomes how people write, not a rule they remember under audit.

What this is not

- **Not a euphemism generator.** Every replacement in the inventory is *more* precise than the term it replaces, or it doesn't ship. If a proposed swap loses meaning, the standard says keep looking.
- **Not a compliance regime.** Tier 2 of the inventory is labeled *judgment calls* because they are. The pack gives you the precision case for each; the decision is yours, made once, logged, and defensible.
- **Not a rewrite of your archive.** The gate governs new writes; the audit prioritizes the twenty assets that face clients today. History gets fixed opportunistically, at update time, like any other standards drift.

Assumptions

Articles typed per Module 2 — any typed template works. A running curation gate (Module 4) for enforcement; a ratification pattern (Module 5) for the standard. None are hard dependencies: the inventory, audit, and champion kit stand alone.

Contextkeeping is a KCS-aligned operating model. KCS® is a service mark of the Consortium for Service Innovation; this material is not affiliated with or endorsed by the Consortium.