# AGENTS.md READ-ONLY Audit Protocol

# AGENTS.md READ-ONLY Audit Protocol (Public, Generalized Version)

**Positioning:** Reference material for the article *Should You Trim AGENTS.md When It Gets Long?*
**Purpose:** Present the audit method that was actually used in a form that third parties can understand, adapt to their own environment, and reuse.

## About This Public Version

This document is a public, generalized version based on the READ-ONLY audit protocol that was actually used. Only information that could identify the local environment has been generalized. The audit purpose, measurement items, no-write boundaries, decision criteria, and stop conditions are preserved.

Replacement examples:

- Local absolute path -> `<local absolute path to projects>`
- Internal project name -> `<internal project name>`
- Internal audit ID -> `<internal audit ID>`
- Artifact-specific filename -> `<artifact-specific filename>`

This public version prioritizes **human readability: a person should be able to understand what will be inspected, what is prohibited, and what outcomes are considered valid**, rather than minimizing the number of tokens sent to an AI. For that reason, it is published in Markdown rather than JSON.

## Assumed Local Environment

As an example, assume that multiple Git repositories exist under a single parent directory. Not every repository is required to contain an `AGENTS.md`.

```
<local absolute path to projects>/
├─ project-a/
│  ├─ .git/
│  ├─ AGENTS.md
│  └─ ...
├─ project-b/
│  ├─ .git/
│  └─ ...                  # Example with no AGENTS.md
└─ project-c/
```

```
    ├─ .git/
    ├─ AGENTS.md
    ├─ docs/
    │  └─ AGENTS.md         # Example with multiple AGENTS.md files in one repository
    └─ ...
```

## Prerequisites

- The specified parent directory can be searched READ-ONLY.
- For Git-managed repositories, local Git history and current state can be read.
- Write permission to the repositories is not required for the audit.
- Remote connectivity is not required. Information that cannot be obtained is reported as `UNKNOWN`.
- A repository without `AGENTS.md` is not treated as an error.
- Zero discovered `AGENTS.md` files is also a valid outcome and may end normally with `agents_files_found: 0`.
- If one repository contains multiple `AGENTS.md` files, each file is treated as an independent audit target, and repository count and file count are reported separately.
- Python or an equivalent processing environment is useful for aggregation, but the implementation method is not fixed as long as the audit conditions can be reproduced.
- Public output must not include private repository names, absolute paths, internal hostnames, credentials, or similar sensitive environment details.

---

```
# PROJECT

<internal project name>

# AUDIT

<internal audit ID>

# 1. Purpose

Recursively search under:

<local absolute path to projects>

and observe the AGENTS.md files that actually exist, READ-ONLY, across repositories.

The purpose is to record characteristics such as AGENTS.md length, structure,
history, shared text, and change-control-related wording, and to determine
whether there is clear evidence that maintenance should be considered.

The purpose of the audit is not to produce changes.
If no problem is confirmed, making no change is treated as a normal outcome.

# 2. Search Scope

Search root:
<local absolute path to projects>

Search recursively under this path.

- Not every repository is expected to contain AGENTS.md.
```

```
- Only repositories in which AGENTS.md is discovered are subject to content audit.
- Repositories without AGENTS.md may be counted as discovered repositories, but are not included in the
content audit.
- Zero AGENTS.md files is not treated as an error.
- If multiple AGENTS.md files exist in one repository, treat each as an independent file.
- Do not force the result to match a previously expected count. Report the repository count and file
count actually discovered.

# 3. Execution Mode

READ_ONLY
DRY_RUN

Do not modify repository content.

# 4. Prohibitions

Operations that change repository or Git state are prohibited.

Examples:
- editing, creating, deleting, renaming, moving, or auto-formatting files
- git add / commit / push / reset / restore / checkout / switch / stash
- branch creation / merge / rebase / tag operations
- dependency changes
- runtime / deployment / server operations

Preserve any pre-existing dirty worktree exactly as it is.

# 5. Audit Principles

## 5.1 No Forced Changes

If no problem is confirmed, return:

NO_CHANGE_RECOMMENDED

as a normal result.

Do not use deleted line count, compression ratio, finding count,
number of change candidates, or number of changed files as success metrics.

## 5.2 Length Is Not Quality

Do not treat long as equivalent to bad, or short as equivalent to good.
Do not recommend changes based on line count alone.

## 5.3 Content / Behavior Boundary

The correctness of individual rules and the operations an Agent actually performed
are outside the scope of this audit.

Examples of out-of-scope questions:
- whether prohibition / approval / security / runtime / tool rules themselves are appropriate
- causal relationships between rules and incidents
- AI or Agent performance evaluation

AGENT EXECUTION BEHAVIOR:
OUT OF SCOPE

# 6. Discovery Measurements

For each target AGENTS.md, record the following where obtainable:

- repository identifier
- relative path / full path
- Git tracked / untracked
- branch / HEAD / origin-main, if obtainable
- worktree state
- exact line count
- byte size / character count
```

```
- non-empty line count / blank line count

# 7. Structural Measurements

For each file, mechanically count the following where possible:

- Markdown heading count / heading level counts
- bullet / numbered list count
- fenced code block count
- Markdown table count
- blockquote / checklist / horizontal rule count

Do not use these as a quality score.

# 8. Aggregate Statistics

Across all target files, calculate at least the following.

Line count:
- minimum / maximum
- mean / median
- p25 / p75
- standard deviation, if possible

Character count:
- minimum / maximum
- mean / median

Also create a line-count distribution and explicitly state the buckets used.

# 9. Representative Size Cases

Identify:

- shortest file
- longest file
- file nearest the median

Do not evaluate shortest as best or longest as worst.

# 10. Exact Duplication

Mechanically identify multi-line blocks that are exact string matches.

Targets:
- exact matching blocks within the same file
- exact matching blocks across repositories

Fix the minimum block length in advance.
Example: 3 consecutive non-empty lines

Do not inflate duplicate counts using generic headings or Markdown symbols alone.

# 11. Textual Similarity

If feasible, detect near-duplicate candidates using methods such as
whitespace normalization, punctuation normalization, and string similarity.

Label them:

TEXTUAL_SIMILARITY_CANDIDATE

Do not assert that they are semantically equivalent.

# 12. Shared Blocks Across Repositories

For shared blocks, record the following where obtainable:

- block length
- occurrence count
```

```
- repository count

Record provenance only when it can be directly verified from Git history
or an explicit reference. Otherwise report UNKNOWN.

# 13. Git History Measurements

For each AGENTS.md, inspect local Git history where available.

Record:
- first known commit / first known change date
- most recent change date
- number of commits touching the file
- initial observed line count / current line count
- net line change
- largest single-commit line increase / decrease

Do not infer missing history.

# 14. Growth Pattern Classification

Using only numeric history, classify each file according to criteria fixed in advance:

- ONE_TIME_LARGE_IMPORT
- GRADUAL_INCREMENTAL_GROWTH
- MIXED
- STABLE
- INSUFFICIENT_HISTORY

Example:
ONE_TIME_LARGE_IMPORT = 50% or more of total growth occurred in a single commit

Record the thresholds actually used in the report.

# 15. Explicit Control Marker Check

Mechanically search for predefined terms that may indicate change-control-related wording.

Examples:
approval / human approval / do not modify / do not edit /
read-only / template / generated / source of truth / policy

Limit results to categories such as:

- EXPLICIT_CONTROL_MARKER_PRESENT
- EXPLICIT_CONTROL_MARKER_NOT_FOUND
- UNKNOWN

The presence of a marker does not prove that the control is effective in practice.

# 16. Referenced Document Structure

Mechanically extract references from each AGENTS.md to local documents,
and record unique referenced files and total reference count.

Do not judge whether the document placement is good or bad.

# 17. Limited Maintenance Signal Check

Check only the following as potential clear evidence that maintenance should be considered:

- Exact Duplication
- Clear Contradiction
- Definite Stale Reference
- Broken Reference Structure

Markers such as temporary / migration / date / specific work-unit names may be recorded,
but their presence alone does not make them deletion candidates.

# 18. Evidence Classes
```

```
OBSERVED:
Facts obtained directly from files, Git, or the filesystem.

DERIVED:
Results calculated arithmetically from OBSERVED values.

UNKNOWN:
Items that cannot be directly determined.

Do not turn matters that require semantic interpretation into facts by inference.

# 19. Maintenance Decision

For each file, return one of:

- NO_CHANGE_RECOMMENDED
- REVIEW_RECOMMENDED

Limit REVIEW_RECOMMENDED to cases with directly observed evidence such as
clear contradiction, definite stale reference, significant exact duplication,
or broken reference structure.

Do not recommend changes merely because a file is long, contains similar blocks,
or has been edited many times.

If there are no findings, FINDINGS: NONE is valid.

# 20. Publication-Safe Output

For public summaries, remove or anonymize private repository names,
local absolute paths, internal hostnames, credentials,
private implementation details, and similar information.

Repository identifiers may be replaced with labels such as Repository A / Repository B.

# 21. Required Questions

Using observed values only, answer at least the following:

1. What are the minimum, maximum, and median AGENTS.md line counts?
2. How much variation exists in file size?
3. Was the largest file recommended for change?
4. Is there direct Evidence that the smallest file is higher quality? If not, answer UNKNOWN.
5. Across how many repositories were shared exact blocks observed?
6. Which growth-pattern classification was most common?
7. In how many files were explicit control markers observed?
8. In how many files were references to external templates / policies observed?
9. Can this sample alone be generalized to AGENTS.md usage as a whole?

Do not predefine the expected answers.

# 22. Limitations

The public summary must state at least:

- One operational environment only.
- Not a statistical sample of all AGENTS.md usage.
- File length is not treated as a quality score.
- Rule content is not evaluated in depth.
- Actual agent behavior is not evaluated.
- Control markers do not prove effective control.
- Git history may be insufficient or incomplete.

# 23. Outputs

If audit artifacts are generated outside the repositories, record the actual save path
and filenames.

Examples:
```

```
- <artifact-specific filename>
- <artifact-specific filename>
- <artifact-specific filename>

Keep artifact generation separate from repository modification.

# 24. Reproducibility Record

Record the commands / scripts / thresholds / calculation methods actually used.

At minimum:
- search method
- line / character counting method
- duplication rule
- similarity rule, if used
- percentile calculation method
- growth classification thresholds
- actual sample count

# 25. Final Report

PROJECT:
<internal project name>

AUDIT:
<internal audit ID>

RESULT:
PASS / FINDINGS / PARTIAL / BLOCKED

EXECUTION_MODE:
READ_ONLY / DRY_RUN

DISCOVERY:
repositories_found:
agents_files_found:
repositories_without_agents:

QUANTITATIVE_SUMMARY:
...

GROWTH_PATTERNS:
...

EXACT_DUPLICATION:
...

TEXTUAL_SIMILARITY_CANDIDATES:
...

EXPLICIT_CONTROL_MARKERS:
...

MAINTENANCE_RESULTS:
NO_CHANGE_RECOMMENDED:
REVIEW_RECOMMENDED:

FINDINGS:
...

UNVERIFIED:
...

LIMITATIONS:
...

REPOSITORY_CHANGES:
NONE

GIT_CHANGES:
```

```
NONE

# 26. Stop Conditions

Do not fill gaps by inference. Return PARTIAL or BLOCKED when appropriate, including:

- insufficient Git history
- encoding or related issues prevent accurate counting
- Git command failure
- classification cannot be reproduced
- public-safe anonymization cannot be performed safely

Zero AGENTS.md files is not itself a BLOCKER.
In that case, end normally with agents_files_found: 0.

# 27. Final Principle

Success in this audit is not defined by finding problems or making AGENTS.md shorter.

- Report the numbers as observed even when they do not match an expected trend.
- Do not use length as a quality metric.
- Do not fill UNKNOWN by inference.
- If there is no problem, NO_CHANGE is a normal result.
- Do not manufacture results for the sake of producing results.
```

## How to Read This Document

This public version is **not** an example optimized to run an AI with the fewest possible tokens. It prioritizes making the audit intent, prohibited boundaries, measurement conditions, and result criteria understandable and traceable to third parties.

In operational use, headings and enumerations can be compressed as long as the same meaning is preserved. However, if shortening makes it ambiguous what may or may not be done, or where observation ends and judgment begins, brevity has no value by itself.