OpsPilot

Document Reader — User Manual

Read Any Technical Document, Six Structured Outputs · AI Engineering Co-Pilot

What this guide covers — what the Document Reader does, the outputs it produces, what to have ready, and how it protects you from misreading a document.

1. What is the Document Reader?

The Document Reader ingests any technical document — a standard, specification, procedure, manual, contract, tender, audit report, incident report, permit, vendor proposal or set of minutes — and produces structured outputs from it: a summary, a requirements extraction, a gap analysis, a change register, and more. Its discipline is what makes it trustworthy: it quotes verbatim with clause and page references, it separates what the document SAYS from what it merely IMPLIES (inferences are flagged, never smuggled in), and it flags every ambiguity rather than silently resolving it.

Crucially, it preserves modal verbs exactly — *SHALL is a requirement, SHOULD a recommendation, MAY a permission* (per ISO/IEC Directives Part 2 / RFC 2119). Paraphrasing “shall” as “should” changes the legal meaning, so it never does.

2. What the OpsPilot module does

Role	Responsibility
🎯 **AI Document Reader & Knowledge Coach (OpsPilot)**	Reads the document and produces the structured outputs you need — quoting verbatim with references, distinguishing says from implies, preserving modal verbs, and flagging every ambiguity rather than filling it.
👷 **Engineer / Operations Lead (you)**	Provide the document and tell OpsPilot which outputs you need — and you make the judgement calls the flagged ambiguities surface.

3. The outputs it can produce

- Summary — a faithful condensation with references.
- Requirements extraction — every SHALL/SHOULD/MAY pulled out, classified, with clause references.
- Gap analysis — the document against a standard or another document (per ISO 19011).
- Change register — what changed between two versions (per ISO 10007).

- Plus question-answering against the document, and ambiguity/risk flagging.

4. What you will be asked — have this ready

- The document (attach it).
- Which output(s) you need — summary, requirements, gap analysis, change register, Q&A.
- For a gap analysis, the standard or comparison document to check against.

5. What you receive — the output

One or more structured outputs (Word): each quoting verbatim with clause/page references, classifying requirements by modal verb, separating statements from inferences, and listing every flagged ambiguity for you to resolve.

6. Worked example (illustrative)

You feed in a 60-page vendor tender response and ask for a requirements extraction and a gap analysis against your specification. The Document Reader pulls out every requirement the vendor commits to, preserving the modal verbs — and that's where it earns its keep: it flags that the vendor wrote “the unit should meet the rated duty” where your spec says “shall.” That one-word difference is a commitment gap that a casual read would miss and that matters enormously in a dispute. The gap analysis lists exactly where the response falls short of the spec, with clause references on both sides, and every genuine ambiguity (“material to be confirmed”) is flagged for you rather than assumed away. You get a precise map of what the document actually commits to — not a comfortable paraphrase.

7. Getting the best result

- **Ask for the specific output.** Summary, requirements, gap analysis or change register — tell it which.
- **Trust the modal verbs.** SHALL vs SHOULD is a legal difference — the tool preserves it; so should you.
- **Read the flagged ambiguities.** They're the spots where you must decide — that's the point of flagging.
- **Use it for gap analysis.** Comparing a response or revision against a standard is where it saves the most time.