CS-DOC-0005 · COMPLIANCESUITE DOCUMENTATION

Composing with the AI Composer.

FIT-only redaction. Effective 2026-04-28.

DOCUMENT ID	VERSION	EFFECTIVE	OWNER
CS-DOC-0005	v1.0	2026-04-28	Customer Success

Public — Documentation · Review cycle: On change

01 — DOCUMENT CONTROL

Control block and metadata anchor.

The control block identifies the document, its current revision, the regulated process it supports, and the people accountable for its lifecycle. Every value below is the source of truth for any downstream record, audit trail entry, or signature block.

DOCUMENT ID	CS-DOC-0005
TITLE	Composing with the AI Composer
VERSION	v1.0
STATUS	FIT-CLEAN
EFFECTIVE DATE	2026-04-28
REVIEW CYCLE	On change
DOCUMENT OWNER	Customer Success
CLASSIFICATION	Public — Documentation
RELATED RECORDS	/output/CS-DOC-0005_AI_Composer.pdf
SUPERSEDES	— (initial release)

02 — APPROVALS

Sign-off table, ready for ink or e-signature.

The signatures below confirm review and authorisation of this document. Approvals must be recorded in chronological order. If the document is signed electronically, the e-signature record on the ComplianceSuite platform supersedes any handwritten entry on this page and carries the same legal weight under 21 CFR Part 11 and EU GMP Annex 11.

Role	Name	Function	Date	Signature
Author		Validation Lead		
Reviewer		Quality Assurance		
Reviewer		Process / System Owner		
Approver		Head of Quality		
Approver		Regulatory Affairs		

03 — CONTENTS

What's in this document.

01 — WHAT THIS EDITION COVERS

What this edition covers.

This documentation covers the AI Composer — the built-in authoring assistance system:

- Multi-provider support (OpenAI 6.10.0, Anthropic 0.71.1, Google Generative AI 0.24.1)
- Per-account and per-tenant API key configuration
- AI temperature and AI max tokens configuration
- Section-based authoring in DocumentSection
- Streaming AI responses in chat interface
- AI-expand for change descriptions
- Generic compliance knowledge context (static)

02 — WHAT THIS EDITION DOES NOT COVER (ROADMAP TOPICS)

What this edition does NOT cover (Roadmap topics).

The following concepts from the original spec are not implemented:

- **Citation generation via RAG** — no citation model in Prisma. Composer does not generate citations with anchors to SOPs, regulations, vendor docs, etc.
- **Citation types (SOP/regulation/vendor/engineering/test/external)** — not modeled.
- **Citation hash anchoring + refresh workflow** — not implemented.
- **AI-drafted markings with aiDraftStatus enum** — DocumentSection field `aiDraft` exists, but no enum for PENDING/ACCEPTED/EDITED/REJECTED.
- **Submission block for undecided AI lines** — no validator preventing submission of deliverables with undecided AI lines.
- **Dedicated AI-interaction audit log** — AiAuditLog exists, but at DocumentSection level, not as separate model with full prompt+model+response+decision tracking.

03 — WHAT THE AI COMPOSER DOES

What the AI Composer does.

The AI Composer is a built-in authoring assistance system that drafts content directly within platform deliverables — URS, risk assessment, validation plan, IQ/OQ/PQ, validation report. It is not a chat window glued to the side of the app. It runs within the regulated record, takes its inputs from system and change context, and writes every prompt, response, accept/reject decision, and human edit into the change's audit trail.

Design Principles

Principle	Meaning in practice
The human is the author	Every AI-drafted line carries an explicit accept, edit, or reject decision before becoming part of deliverable. There is no silent acceptance
Every action is auditable	Prompt, model ID, model version, temperature, top-P, response, and human decision written to audit trail in same transaction as deliverable edit
No PII or customer data leaves the boundary	AI inference runs within customer's data-residency region. Customer data not used to train shared models. Inference logs deleted after audit trail entry
The model is named	Account administrators see exact `model:version` in composer footer at all times; audit trail captures same value with every event
The model is replaceable	Account compliance lead can pin platform to specific model version, validate it as Cat 5 subsystem, and reject upgrades without change against platform itself

04 — WHERE THE COMPOSER HELPS

Where the composer helps.

The composer is not a content factory. It speeds drafting where regulated work is repetitive, well-structured, and benefits from a starting point the author then refines.

Deliverable	Composer's contribution	Author's contribution
URS	Drafts requirements from system scope, tagged by class (functional, regulatory, performance, interface) and risk relevance	Removes off-scope; adds site-specific and integration-specific requirements
Risk Assessment	Imports URS-derived risks; proposes severity/likelihood/detectability scoring with rationale; proposes mitigations linked to test cases	Confirms or revises scoring; adds risks URS does not address (data integrity, security, business continuity)
Validation Plan	Pre-populated scope, deliverable matrix, acceptance criteria, role assignments, test strategy from URS + risk	Writes schedule, deviation handling, out-of-scope statements, communication plan
IQ/OQ/PQ	Generates one test per requirement above medium risk; proposes pass criteria, evidence requirements, execution steps	Confirms each test against site reality; tunes pass criteria; adds tests model cannot derive
Validation Report	Drafts executive summary, deliverable status, test summary, deviation summary, traceability summary from change record	Writes acceptance recommendation, residual risk statement, operational handover notes — parts requiring professional judgment

Where the composer is NOT allowed

The composer is intentionally disabled on these blocks: **acceptance recommendations, deviation conclusions, residual risk statements, signature meanings**. Each is a professional judgment that must come from a named human; AI assistance there would invert the regulated obligation.

05 — HOW TO PROMPT WELL

How to prompt well.

Composer output quality is bounded by prompt quality. Two paragraphs is the sweet spot — enough specificity to bound the draft, but not so much the author does the work twice.

A good URS prompt

```
Document management system for controlled SOPs and validation packages.
Used by Quality and Validation teams across the Boston site, ~150 active
users, GAMP 5 Cat 4. Vendor: OpenText Documentum 23.4. Critical integrations:
Active Directory (SAML 2.0), the e-signature service, the training-record
system. Major use cases: SOP authoring and approval, validation package
assembly, periodic review reminders. Out of scope: batch records, raw lab
data, change control records (those live in MES and ERP respectively).
```

Note what this prompt contains: **vendor and version** (lets model use vendor-specific language), **scale** (drives performance requirements), **integrations** (drives interface requirements), **scope boundaries** (prevents model radiating into adjacent systems).

Anti-patterns

- **One-sentence prompts** — *"Generate a URS for the EDMS."* Model will draft generic EDMS URS, you'll spend more time editing than writing from scratch.
- **Pasting vendor brochure** — output reads like sales sheet, not regulated requirement set.
- **Prompting for entire deliverable** — prompt section by section. Composer can draft sections in isolation; one shot at whole document is weaker pattern.
- **Including PII or other-tenant data** — platform's prompt scrubber will reject the call.

06 — REVIEWING AI-DRAFTED CONTENT

Reviewing AI-drafted content.

Every line drafted by the AI Composer is presented to the author with three options: **accept**, **edit**, or **reject**. Audit trail captures every decision. Deliverable contains only accepted and edited lines. Platform refuses to submit deliverable to review while AI-drafted lines remain undecided.

What to look for

Problem	Example
Confabulated specifics	A line naming a vendor module that does not exist, a regulatory clause that does not say what claimed, or a pre-existing SOP you never wrote. Model is eloquent — eloquence is not accuracy
Drift from system scope	Requirements for modules you did not buy, integrations you do not have, regulators that do not apply
Generic platitudes	Lines that could appear in every URS for every system
Risk-class mismatches	A risk-relevant requirement tagged low risk, or a low-stakes convenience requirement tagged high

Reject is not an error:

An author accepting every composer suggestion does not get good service from tool. Expected accept rate on well-prompted URS in range where rejection feels frequent — closer to senior engineer reviewing junior's draft than stenographer accepting transcription. Reject often; audit trail of rejections is part credibility evidence.

Audit trail and inspector readout.

When an inspector asks *"how did you write this?"*, the answer in a composer-assisted change is same as hand-authored change: **show the audit trail**.

Audit trail content per AI-assisted line

Field	Example Value
Event type	`composer.draft`, `composer.accept`, `composer.edit`, `composer.reject`
Actor	`jane.doe@acme.com` (author deciding); `composer:<model■id>` (model drafting)
Model identifier	Stable string identifying platform-pinned model and version
Prompt hash	SHA-256 of prompt; prompt itself stored in per-change prompt log
Response hash	SHA-256 of model response
Decision	accept \
Justification	Optional free text author can attach to any decision (recommended for unusual rejects)
Timestamp	Server-side UTC, NTP-synchronized

08 — CONFIGURATION AT ACCOUNT LEVEL

Configuration at account level.

The composer is a Cat 5 subsystem of the platform. It is validated by ComplianceSuite under our internal QMS and re-validated on every model upgrade. Customers with stricter policies can pin the model and reject upgrades.

Account-level composer policy

- **Enable / disable** the composer per tenant
- **Pin the model version** — account compliance lead can pin specific model version. New model versions then require explicit tenant-level approval change before use
- **Restrict by deliverable** — disable composer on specific deliverables (e.g., validation reports) while leaving it enabled for URS drafting
- **Watermark exports** — AI-assisted records carry inspection-ready watermark in PDF exports, with AI-assisted line ratio printed in footer

When the composer is disabled

Disable the composer on a tenant or deliverable if:

- Regulator or quality agreement specifies *"human-only authorship"*
- Validated system handles regulator-confidential clinical data
- Customer's own QMS does not yet include AI-assistance procedures

Code Reference.

Prisma Models

- `AccountSettings` — `aiProvider`, `aiModel`, `openaiApiKey`, `anthropicApiKey`, `geminiApiKey`, `aiTemperature`, `aiMaxTokens`
- `TenantSettings` — same AI fields (nullable, falls back to AccountSettings)
- `DocumentSection` — `aiDraft` (string), `content` (string, final approved), `isLocked` (boolean), `chatHistory` (JSON)
- `AiAuditLog` — audit trail for AI interactions (partial, planned for extended feature)

Server Actions

- `app/actions/change/ai-expand.ts` — AI-expand for change descriptions
- `components/composer/ChatInterface.tsx` — streaming AI responses in chat UI
- `lib/ai/compliance-knowledge.ts` — static compliance knowledge context (generic, non-tenantizable)

AI Provider Integration

- **OpenAI:** via `@anthropic-sdk/sdk` (6.10.0) — `gpt-4o`, `gpt-4 turbo`
- **Anthropic:** via `@anthropic-sdk/sdk` (0.71.1) — `claude 3` family
- **Google Generative AI:** via `@google/generative-ai` (0.24.1) — `gemini` family

End of documentation

REVISION HISTORY

Every change, tracked and signed.

Add one row for every controlled revision. Minor changes (typos, formatting) increment the patch version; substantive edits trigger a fresh review cycle and a new approver round.

Version	Date	Author	Summary of Change	Approver
1.0	2026-04-28	Documentation Team	FIT-only redaction limited to codebase-verified functionality.	Head of Documentation
—	—	—	Reserved for next revision. Do not delete this row.	—

GLOSSARY

Shared language, no ambiguity.

Definitions used throughout this document. Where a term has a specific meaning inside ComplianceSuite, the platform-specific definition takes precedence over the generic regulatory term.

CSV	Computerized Systems Validation
GAMP 5	Good Automated Manufacturing Practice, Edition 5 (2nd edition, 2022)
GxP	Good 'x' Practice — covers GMP, GLP, GCP, GDP, GVP
IQ / OQ / PQ	Installation / Operational / Performance Qualification
Part 11	21 CFR Part 11 — US FDA rule on electronic records and electronic signatures
Annex 11	EU GMP Annex 11 — EU rule on computerised systems
URS	User Requirements Specification
FRS	Functional Requirements Specification
RTM	Requirements Traceability Matrix
SOP	Standard Operating Procedure
ALCOA+	Attributable, Legible, Contemporaneous, Original, Accurate (+ Complete, Consistent, Enduring, Available)
ICH Q9	International Council for Harmonisation Quality Risk Management guideline

— End of document —