CS-DOC-0005 · GXP-DESK DOCUMENTATION

Composing with the AI Composer.

FIT-only redaction. Effective 2026-06-04.

DOCUMENT ID	VERSION	EFFECTIVE	OWNER
CS-DOC-0005	v1.0	2026-06-04	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	2026-06-04
REVIEW CYCLE	On change
DOCUMENT OWNER	Customer Success
CLASSIFICATION	Public — Documentation
RELATED RECORDS	—
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 GxP-Desk 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 VERSION COVERS

What this version covers.

This documentation describes 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 ChatInterface
- AI-expand for change descriptions
- Generic compliance knowledge context (static)

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

What this version does NOT cover (roadmap topics).

The following concepts from the original specification are not implemented:

- **Citation generation via RAG** — There is no citation model in Prisma. The Composer does not generate citations with anchors to SOPs, regulations, vendor docs, etc.
- **Citation types (SOP/Regulation/Vendor/Engineering/Test/External)** — Not modelled.
- **Citation hash anchoring + refresh workflow** — Not implemented.
- **AI-drafted markings with an aiDraftStatus enum** — The DocumentSection field `aiDraft` exists, but there is no enum for PENDING/ACCEPTED/EDITED/REJECTED.
- **Submission block on undecided AI lines** — There is no validator that prevents the submission of deliverables with unresolved AI lines.
- **Dedicated AI interaction audit log** — AiAuditLog exists, but at the DocumentSection level, not as a 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 application. It runs inside the regulated record, takes its inputs from the system and change context, and writes every prompt, every response, every accept/reject decision and every 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 it becomes part of the deliverable. There is no silent acceptance
Every action is auditable	Prompt, model ID, model version, temperature, top-p, response and human decision are written into the audit trail in the same transaction as the deliverable edit
No PII or customer data leaves the boundary	AI inference runs within the customer's data-residency region. Customer data is not used to train shared models. Inference logs are deleted after the audit-trail entry
The model is named	Account administrators can see the exact `model:version` in the Composer footer at all times; the audit trail captures the same value with every event
The model is replaceable	The Account Compliance Lead can pin the platform to a specific model version, validate it as a Cat 5 subsystem, and decline upgrades without raising a change against the platform itself

04 — WHERE THE COMPOSER CAN HELP

Where the Composer can help.

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

Deliverable	Composer's contribution	Author's contribution
URS	Drafts requirements from the system scope, tagged by class (Functional, Regulatory, Performance, Interface) and risk relevance	Removes off-scope items; 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 the scoring; adds risks the URS does not address (data integrity, security, business continuity)
Validation Plan	Pre-populates scope, deliverable matrix, acceptance criteria, role assignments, and test strategy from the URS + risk	Writes the schedule, deviation handling, out-of-scope statements, and communication plan
IQ/OQ/PQ	Generates one test per requirement from Medium Risk upward; proposes pass criteria, evidence requirements, and execution steps	Confirms each test against site reality; tunes pass criteria; adds tests the model cannot derive
Validation Report	Drafts the Executive Summary, Deliverable Status, Test Summary, Deviation Summary, and Traceability Summary from the change record	Writes the acceptance recommendation, residual-risk statement and operational handover notes — the parts that require professional judgement

Where the Composer is NOT allowed

The Composer is deliberately disabled on these blocks: **Acceptance Recommendations, Deviation Conclusions, Residual Risk Statements, Signature Meanings**. Each is a professional judgement that must come from a named human; AI assistance there would invert the regulated obligation.

05 — HOW TO PROMPT WELL

How to prompt well.

The Composer's output quality is bounded by the quality of the prompt. Two paragraphs is the sweet spot — enough specificity to bound the draft, but not so much that 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 the model use vendor-specific language), **scale** (drives performance requirements), **integrations** (drive interface requirements), **scope boundaries** (prevent the model from bleeding into adjacent systems).

Anti-patterns

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

Reviewing AI-drafted content.

Every line drafted by the AI Composer is presented to the author with three options: **accept**, **edit** or **reject**. The audit trail captures every decision. The deliverable contains only the accepted and edited lines. The platform refuses to submit a deliverable for review while AI-drafted lines remain unresolved.

What to look for

Problem	Example
Confabulated specifics	A line that names a vendor module that does not exist, a regulatory clause that does not say what is claimed, or a pre-existing SOP you never wrote. The 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 are not applicable
Generic platitudes	Lines that could appear in any URS for any system
Risk-class mismatches	A risk-relevant requirement tagged as Low Risk, or a low-stakes convenience requirement tagged as High

Reject is not a failure:

An author who accepts every Composer suggestion is not getting good service out of the tool. The expected accept rate on a well-prompted URS sits in the range where rejection feels frequent — closer to a senior engineer reviewing a junior's draft than to a stenographer accepting transcription. Reject frequently; the audit trail of rejections is part of the credibility evidence.

Audit trail and inspector readout.

When an inspector asks *"how did you write this?"*, the answer in a Composer-assisted change is the same as in a 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` (the author who decides); `composer:<model■id>` (the model that drafted)
Model Identifier	Stable string identifying the platform-pinned model and its version
Prompt Hash	SHA-256 of the prompt; the prompt itself is stored in the per-change prompt log
Response Hash	SHA-256 of the model response
Decision	accept \
Justification	Optional free text the author can attach to each decision (recommended for unusual rejects)
Timestamp	Server-side UTC, NTP-synchronised

08 — ACCOUNT-LEVEL CONFIGURATION

Account-level configuration.

The Composer is a Cat 5 subsystem of the platform. It is validated by GxP-Desk under our internal QMS and re-validated for every model upgrade. Customers with stricter policies can pin the model and decline upgrades.

Account-level Composer policy

- **Enable / Disable** the Composer per tenant
- **Pin the model version** — The Account Compliance Lead can pin a specific model version. New model versions then require an explicit tenant-level approval change before they can be used
- **Restrict by deliverable** — Disable the Composer on specific deliverables (e.g. Validation Reports) while keeping it enabled for URS drafting
- **Watermark exports** — AI-assisted records carry an inspection-ready watermark in PDF exports, with the AI-assisted-line ratio printed in the footer

When to disable the Composer

Disable the Composer on a tenant or deliverable when:

- A regulator or a quality agreement specifies *"human-only authorship"*
- The validated system handles regulator-confidential clinical data
- The customer's own QMS does not yet incorporate AI assistance procedures

09 — CODE REFERENCES

Code references.

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 an expanded feature)

Server actions

- `app/actions/change/ai-expand.ts` — AI-expand for change descriptions
- `components/composer/ChatInterface.tsx` — streaming AI responses in the 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	2026-06-04	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 GxP-Desk, 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 —