




INDITION AI CORE

Administrator Guide

How to configure providers, credentials, models, request types, logging, and settings that power every AI feature on the platform.

AI Core Module · Version 1.0 · June 2026

 © 2026 Indition Corp. All rights reserved. Indition™ and the Indition logo are trademarks of Indition Corp. All other product names, logos, and brands are the property of their respective owners.

Indition AI Core — Administrator Guide

A task-oriented guide for the administrators who **set up and operate** AI Core — the layer that powers every AI feature on the Indition platform. It walks through enabling AI, connecting a provider, registering models, defining request types, and managing logs, cost, and settings.

AI Core has no end-user screens. Everything here lives under the platform's owner/administration menus. If a product module (such as AI Assistant) "can't do AI", the cause is almost always something in this guide.

Where everything lives

Menu	Use it to...
AI Settings	Turn AI on/off and reach the other AI configuration areas.
Manage AI Providers	Register vendors and their API base URLs; set the default; manage credentials.
Manage AI Models	Register models with pricing and capability flags.
Manage AI Request Types	Define structured, reusable request definitions (schema + system prompt).
Manage AI Request Logs	Inspect every AI call: status, tokens, latency, cost, attribution.
Miscellaneous AI Settings	Tune logging, retries, timeouts, file/vision limits, and email generation.

First-time setup in five steps

1. **Enable AI Core** (the master switch).
2. **Add a provider** and a **credential** (API key).
3. **Add at least one model** with its pricing and capabilities.
4. **Confirm request types** exist for the features you'll use (e.g. email generation).
5. **Run a test** from a consuming feature and check the request log.

1. Enable AI & connect a provider

Turn AI on

AI Core ships with a master switch (`AI_CORE_ENABLED`). Until it is on, no module can generate anything. Enable it in AI Settings / Miscellaneous AI Settings.

Add a provider

In **Manage AI Providers**, create a provider with:

- **Code** — a stable identifier (e.g. `openai`, `gemini`).
- **Name** — a display label.
- **Base URL** — the provider's API endpoint.
- **Status** — active/inactive, and whether it's the **default** provider.
- **Default share** — whether all customers can use it without explicit sharing.

Add a credential

- Add one or more API keys for the provider. Keys are **encrypted at rest** — you won't see the raw key again after saving.
- Give each a clear **label** (e.g. "Production key") and mark one as the **default**.
- Rotate by adding a new credential, making it default, then deactivating the old one.

Security: credentials are stored as ciphertext, and sensitive fields are redacted from request/response payloads in the logs. Treat provider keys like any other production secret.

2. Register models

In **Manage AI Models**, add the models you want available under each provider:

- **Provider** and **code** (e.g. the vendor's model id) plus a display name.
- **Context window** — maximum tokens the model accepts.
- **Pricing** — input and output cost rates (per million tokens). These drive all cost reporting, so keep them accurate.
- **Capabilities** — completion support, vision (image input), image generation.
- **Status** and **default share**.

Pricing matters: cost figures everywhere on the platform (including AI Assistant's Cost & Usage report and quotas) are computed from these per-token rates. If costs look wrong, check the model rates first.

Choosing which models to enable

- Enable a capable general model for text/answers, and (if needed) a vision-capable model and an embedding model.
- Only enable image generation models if a feature needs them.
- Deactivate models you don't want consumers selecting rather than deleting them, to preserve historical logs.

3. Request types

Request types are reusable definitions that standardize how a particular kind of AI call is made and validated. In **Manage AI Request Types** each has:

- **Code** and **name** (e.g. an email-generation type).
- **Handler class** and a **default model**.
- **Response format** — JSON schema or JSON object — with a stored schema and example.
- **System prompt** — the instruction sent with every call of this type.
- **On-invalid-JSON** policy and retry behaviour.

Most features ship with the request types they need. You generally adjust the system prompt or default model rather than creating new types from scratch.

AI email generation

AI Core can generate structured, multi-block marketing/transactional emails from a validated email schema. Related settings let you allow or block image blocks, cap the number of images, hide model selection in the email UI, and retry when the model returns invalid JSON.

4. Logging, cost & settings

Request logs

Manage AI Request Logs records every AI call. For each you can see provider/model/credential, endpoint, status (success / error / timeout / invalid JSON), HTTP status, latency, token usage, and computed cost — plus attribution back to the product, feature, operation, actor, and business object that triggered it.

- **Enable/disable logging** with `AI_LOG_ENABLED`.
- **Retention** — old logs are pruned after `AI_LOG_RETENTION_DAYS` (default 365).
- **Redaction** — fields in `AI_LOG_REDACTION_FIELDS` (`api_key`, `authorization`, `key`, `secret` by default) are masked before storage.

Reliability settings

- **Timeout** — `AI_REQUEST_TIMEOUT_SECONDS` for provider calls.
- **Retries** — `AI_REQUEST_MAX_RETRIES` on transient failures.

Documents & vision settings

- `AI_MAX_FILE_SIZE_MB`, `AI_ALLOWED_INFO_DOC_TYPES`, `AI_ALLOWED_IMAGE_TYPES`.
- `AI_MAX_INFO_DOCS`, `AI_MAX_IMAGE_DOCS`, `AI_MAX_EXTRACTED_TEXT_CHARS`.
- `AI_VISION_FORCE_BASE64` — send images inline as base64 vs. by URL.

Sharing to customers

Use the sharing grid to make specific providers, models, and request types available to particular customers, or set **default share** to expose them to everyone. This is how multi-tenant access to AI capabilities is controlled.

Operational checklist

1. AI Core enabled, with an active provider, valid credential, and active model.
2. Model pricing accurate; request types present for your features.
3. Logging on with a sensible retention window and redaction list.
4. Spot-check the request log after enabling a new feature.