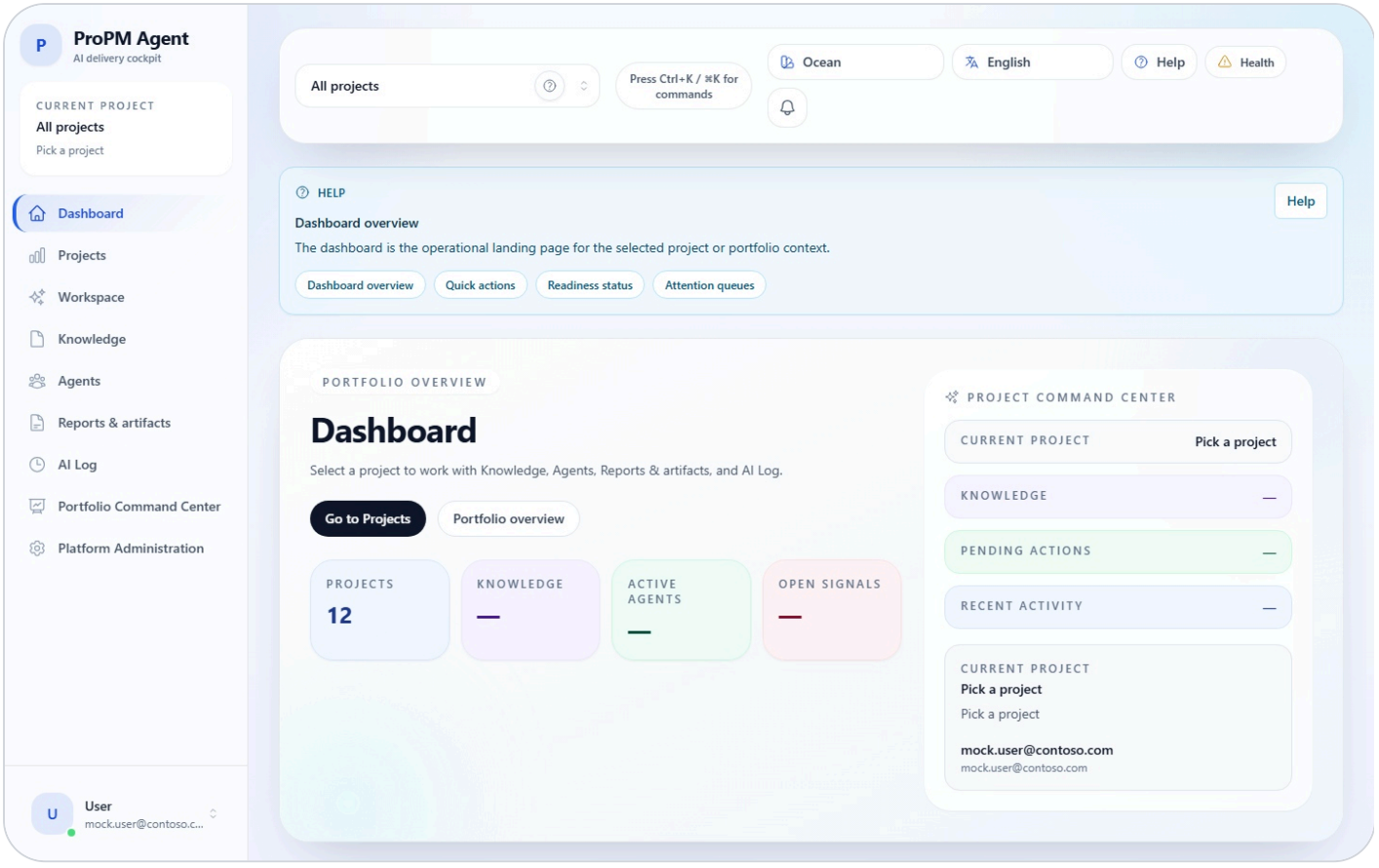
ProPM Agent User Guide

This documentation helps Azure customers use ProPM Agent without mixing daily usage, project administration, platform administration and support diagnostics.



Export PDF

Download the English PDF

Start with the right path

SITUATION	OPEN FIRST	GOAL
I am discovering ProPM Agent	Audiences, roles and scopes	Choose the right reading path
I need to sign in	Getting started	Access the app and understand empty states
I want project status	Dashboard	Read metrics, shortcuts and recent activity
I need to select or create a project	Active project	Set the working context

SITUATION	OPEN FIRST	GOAL
I need documents or evidence	Knowledge	Upload, import and search evidence
I want to use AI	Agents	Choose an agent, run it and read results
I need to review a deliverable	PM Documents	Review, approve, download, publish or add to knowledge
I need to audit a run	AI Log	Find the run and effective AI provider
I need to handle an alert	Signals and notifications	Read, snooze, dismiss or draft from a signal
I need to approve an action	Actions and approvals	Understand approved vs executed

Paths by profile

PROFILE	MAIN PAGES
Business user	Getting started → Dashboard → Knowledge → Agents → PM Documents
Project Manager	Active project → Signals → Actions and approvals → AI Log
Project Owner	Access control → Project workspace → Governance → Connectors
PMO or leadership	Portfolio → PM Documents → AI Log
Azure or tenant administrator	Azure and Entra administration → AI provider → Platform integrations → Licenses, plans and updates
Support or audit	Support, audit and diagnostics → AI Log → Maintenance, support and FAQ

Workflow to remember

The common flow is **active project → knowledge → agent → structured output → artifact → PM Document → Download / Publish / Add to knowledge → AI Log**. An agent conversation does not automatically create a final shared document. A PM Document must be reviewed, governed and published or added to knowledge according to project rights and policies.

Essential concepts

TERM	SHORT DEFINITION
Active project	Current context for Knowledge, Agents, PM Documents, Signals and AI Log
All projects	Scope limited to projects already accessible by the same account
Project binding	Link that makes a platform integration usable in a project
Structured output	Agent run result, distinct from a final document
PM Document	Reviewed, versioned and governed project deliverable
Effective AI provider	Provider actually used by a run, visible in AI Log
Entitlement	Operational blocker related to license, role, policy, integration or health

Azure and platform administration

Azure, Entra, AI provider, connector, license and update details are separated from the end-user guide:

TOPIC	PAGE
Deploy and finalize access	Azure and Entra administration
Configure Save / Validate / Test / Activate	AI provider
Define, validate and bind integrations	Platform integrations
Manage seats, Marketplace plan and updates	Licenses, plans and updates

Support

For an incident, provide page URL, active project, time, attempted action, visible message and visible IDs such as Run ID or Trace ID. Never share secrets, keys, tokens or sensitive payloads.

- Support, audit and diagnostics
- Maintenance, support and FAQ
- Glossary

Audiences, roles and scopes

Goal

Use this page to choose the right documentation path. It separates daily product usage, project administration, Azure/platform administration and support/audit diagnostics.

Who this page is for

PROFILE	USE THIS PAGE TO
Business user	Find task-oriented pages without Azure internals
Project owner	Find project roles, governance, knowledge and integration pages
Azure or tenant administrator	Find deployment, AI provider, license, integration and update pages
Support or auditor	Find diagnostic, AI Log and traceability pages

Documentation levels

LEVEL	CONTAINS	DOES NOT CONTAIN
End user	Steps, roles, expected results, common blockers	Azure variables, secrets, raw payloads or infrastructure details
Project owner	Members, roles, policies, categories and project integrations	Connector secrets or low-level auth details
Azure/platform admin	Marketplace, Entra, AI provider, licenses, integrations and updates	Daily business workflows
Support/audit	Run ID, Trace ID, effective provider, activity and symptoms	Secrets, keys or internal endpoints

Recommended paths

ROLE	START WITH	THEN
New user	Getting started	Interface, Dashboard, Active project
Contributor	Knowledge	Agents, Structured outputs, PM Documents
Project owner	Access control	Project workspace, Governance
Tenant admin	Azure and Entra administration	AI provider, Licenses and updates
Support	Support, audit and diagnostics	AI Log, FAQ

Label convention

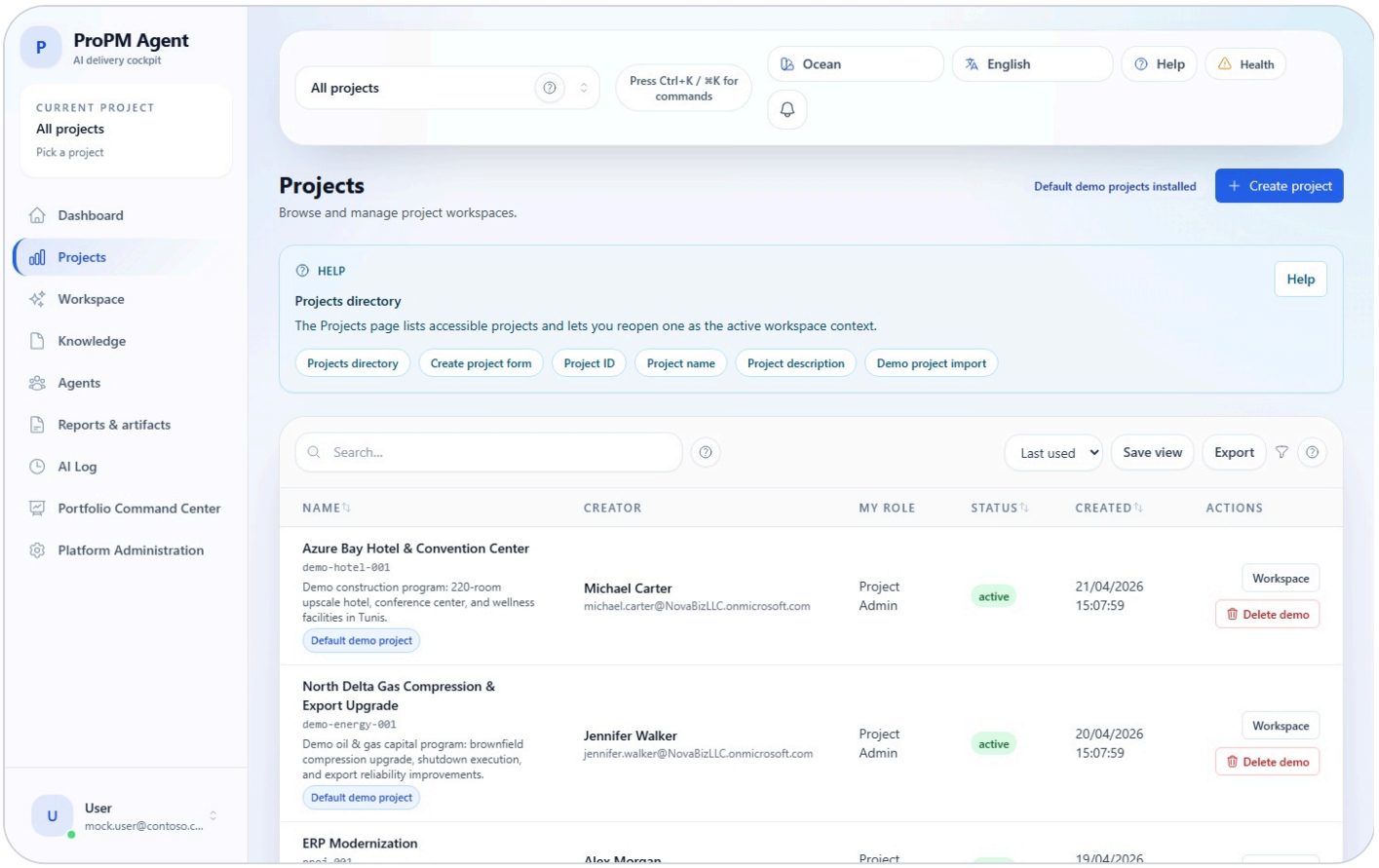
The documentation uses the business term first and keeps the exact UI label when useful, for example **Add to knowledge**, **All projects** or **Effective AI Provider**.

Security rule

Never include secrets, passwords, tokens, full sensitive payloads or unmasked customer data in support requests.

Getting Started

Home · Getting Started



Goal

This page helps a new user open ProPM Agent, sign in with Microsoft Entra ID, select the active project and recognize common empty states.

Before you start

- You have received the ProPM Agent URL from your organization.
- You have an authorized Microsoft Entra ID account.
- You are a member of at least one project or can create one.
- You use a modern browser.

If the app is not deployed or Entra access is not finalized, an administrator must first follow Azure and Entra administration.

First sign-in

1. Open the ProPM Agent URL.

2. Select **Sign in**.
3. Choose the expected work account.
4. Accept consent if your organization allows it and the screen asks for it.
5. Wait until ProPM Agent opens again.
6. Confirm that your profile appears in the top bar.
7. Select an active project if requested.

Select or create the first project

1. Open the project switcher or **Projects**.
2. Select the project provided by your team.
3. If you can create a project, use **Create project**.
4. Verify the active project in the top bar.
5. Open Dashboard to confirm project data loads.

Common empty states

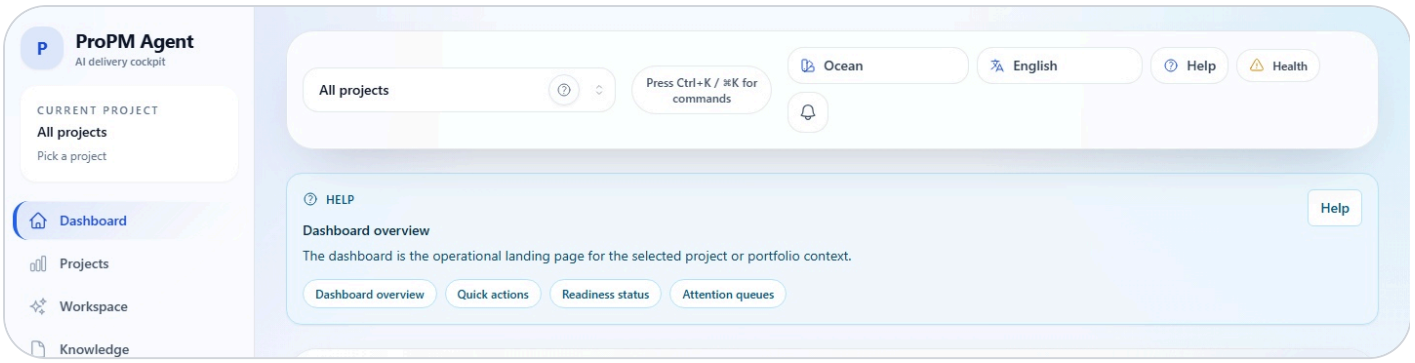
STATE	MEANING	RECOMMENDED ACTION
No project available	Your account is not yet a project member	Ask the Project Owner to add you
No active project	You are signed in but no context is selected	Select a project
Role message	Your role cannot perform the action	Ask for the right role
Sign-in loop	Tenant, consent or Entra setup issue	Contact the tenant administrator
Disabled buttons	Role, policy or integration is insufficient	Open Access control

Related pages

- Interface and navigation
- Active project and project creation
- Support, audit and diagnostics

Interface and navigation

Home · Interface and navigation



Goal

This page explains how to navigate ProPM Agent: main navigation, active project, notification center, health indicator, command palette, language, theme and help.

Who this page is for

PROFILE	USE THIS PAGE TO
Business user	Find the right page and understand empty states
Project Manager	Switch project before acting and follow notifications
Project Owner	Quickly diagnose role, project, health or integration issues
Support	Collect first-level clues without technical administration details

Main surfaces

NEED	OPEN
Read project status	Dashboard
Select or create a project	Active project
Add or search documents	Knowledge
Run AI	Agents

NEED	OPEN
Review deliverables	PM Documents
Audit a run	AI Log
Handle a signal	Signals and notifications
Approve an action	Actions and approvals

Switch active project

1. Open the project switcher in the top bar.
2. Search for the project if the list is long.
3. Select the project.
4. Confirm that its name appears in the top bar.
5. Open **Dashboard** to verify the context.
6. Continue in **Knowledge**, **Agents**, **PM Documents**, **Signals** or **AI Log**.

Notification center

1. Open the bell icon.
2. Read unread notifications.
3. Select a notification to open the linked page when available.
4. Use **Mark all read** to clear the counter.
5. Use **Clear** only when the local list is no longer needed.
6. If the linked page is empty, check active project and permissions.

Health indicator

STATE	MEANING	ACTION
API healthy	Services respond	Continue normally
API degraded	Some pages may load slowly or stay empty	Wait, note the time and retry
Realtime degraded	Notifications may lag	Refresh if needed

STATE	MEANING	ACTION
Auth issue	Session or tenant needs attention	Return to Getting started
High latency	Actions may be slow	Avoid duplicate clicks
Blocking message	Live API may be unavailable	Open Support diagnostics

Language, theme and help

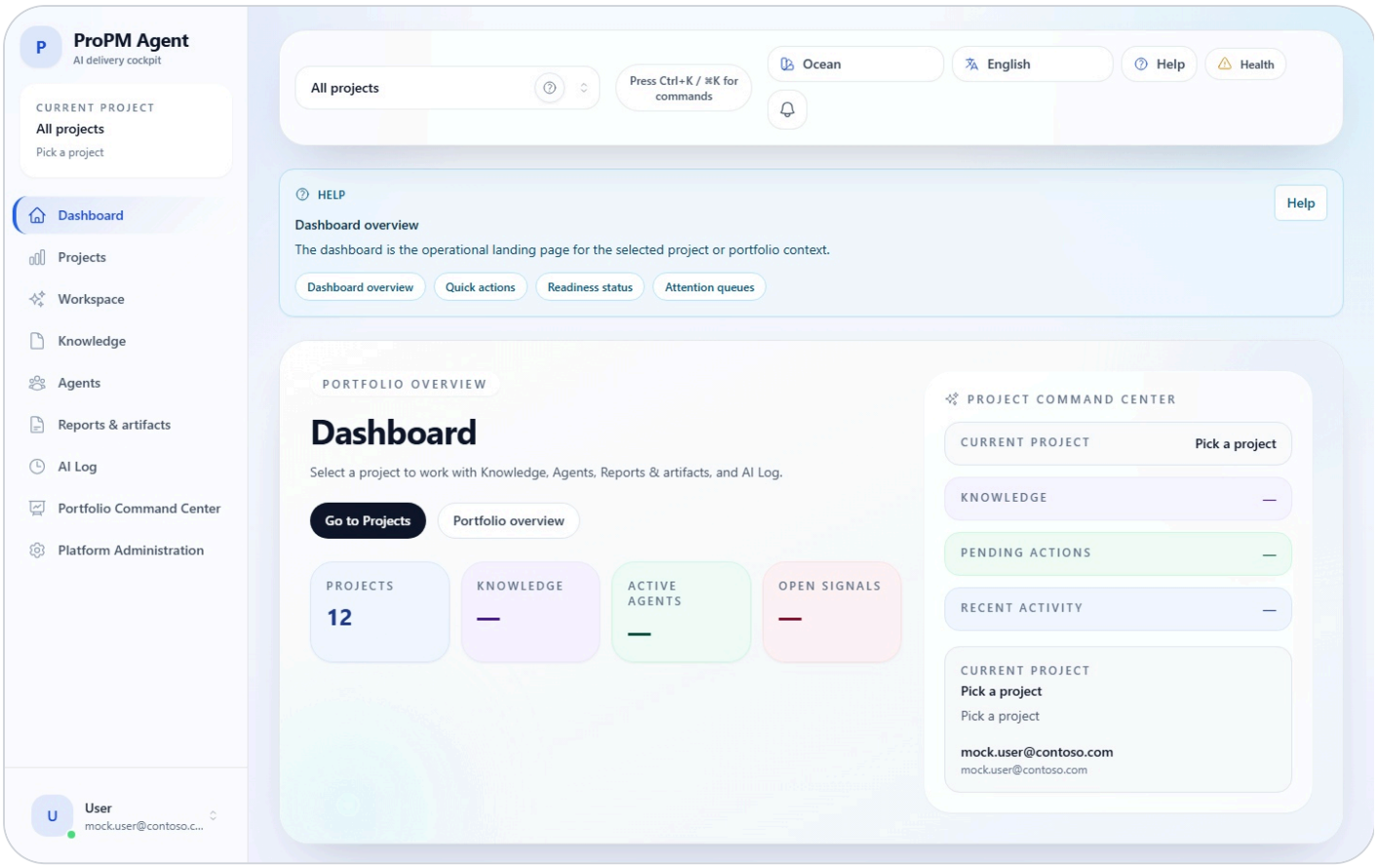
Language changes labels, not project data. Theme is local comfort. Some preferences such as recent project, language, theme and notification read state can stay local to the browser.

Related pages

- Dashboard
- Active project and project creation
- Support, audit and diagnostics

Dashboard

Home · Dashboard



Dashboard reading order

Start by confirming the project context, then read status, activity and available shortcuts.

1 Active project
Every metric, agent run and document depends on the selected project.

2 Project metrics
Check progress, blockers, risk indicators and missing evidence signals.

3 Health and alerts
Confirm platform health, notifications and warnings before troubleshooting.

4 Recent activity
Review recent runs, imports, documents and actions to understand what changed.

5 Action shortcuts
Open knowledge, agents, PM documents or actions from the current context.

If the page looks empty, first verify the active project and your role

Goal

The dashboard summarizes the active project: recent activity, knowledge, agents, PM Documents, signals, actions and shortcuts.

Before you start

- You are signed in to ProPM Agent.
- An active project is selected, unless you are intentionally reading the empty state.
- Your role gives access to the project.

Read the dashboard

1. Open **Dashboard**.
2. Verify the active project in the top bar.
3. Read the summary cards for agents, documents, runs, PM Documents, signals and actions.
4. Review recent activity.
5. Use shortcuts to **Workspace**, **Knowledge**, **Agents**, **PM Docs** or **AI Log**.

Empty states

STATE	MEANING	RECOMMENDED ACTION
No active project	Project pages have no context	Select a project or open Active project
No project available	Your account has no project access	Ask a Project Owner to add you
Access denied	You are signed in but not authorized	Ask for the right role

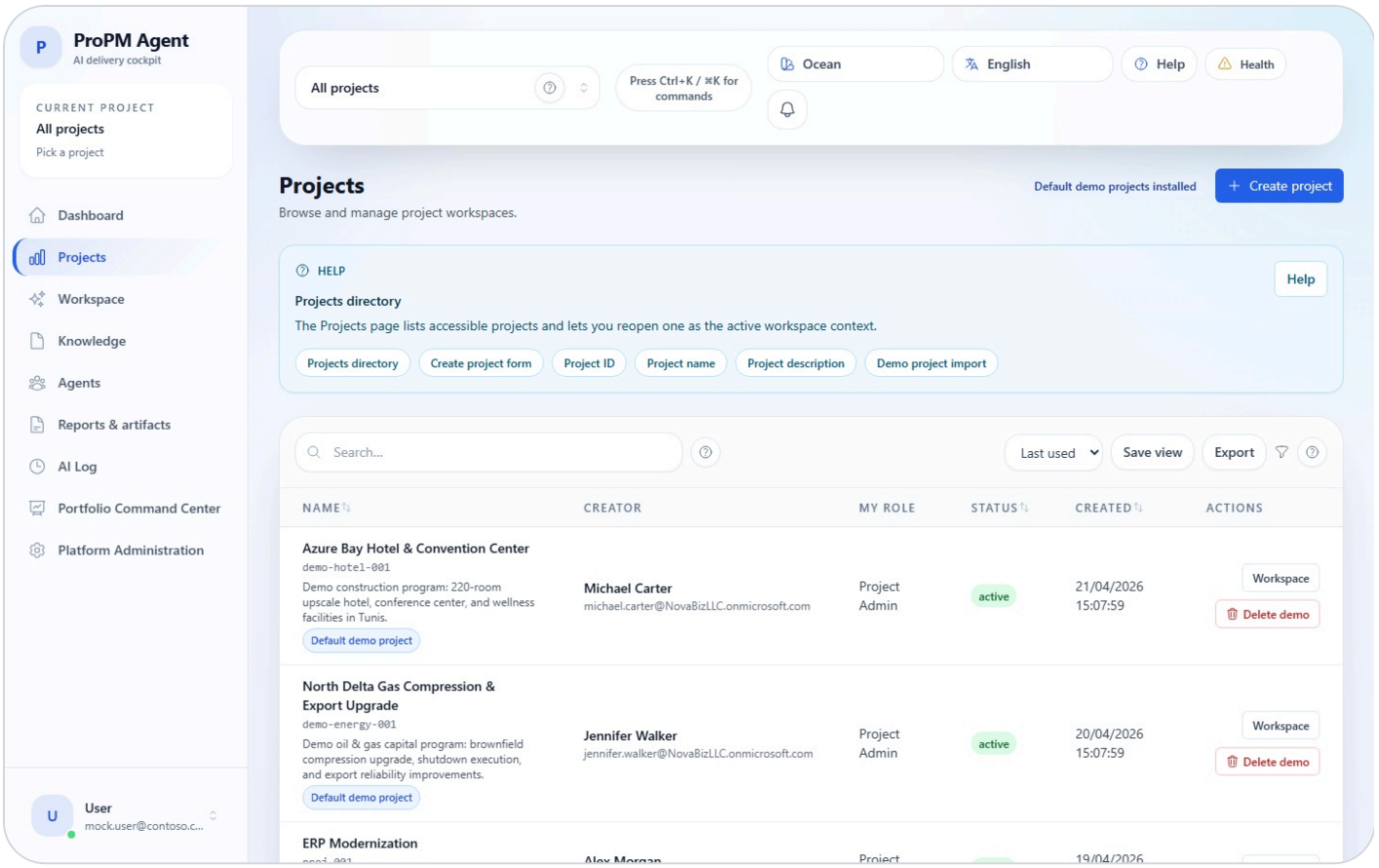
Common issues

SYMPTOM	LIKELY CAUSE	RECOMMENDED ACTION
Dashboard is empty	No active project or insufficient role	Select a project and verify your role
Numbers look stale	Data is still processing	Open the related page and refresh later
Card is visible but disabled	Missing permission	Ask the Project Owner

SYMPTOM	LIKELY CAUSE	RECOMMENDED ACTION
Global error	API health or provider issue	Open Support, audit and diagnostics

Active project and project creation

Home · Active project and project creation



Goal

This page explains how to select the active project, create a project and verify that Knowledge, Agents, PM Documents and AI Log use the right context.

Select an active project

1. Open the project switcher in the top bar.
2. Search for the project if the list is long.
3. Select the project.
4. Confirm that the project name appears in the top bar.
5. Open **Dashboard** or **Workspace** to confirm the data matches.

Create a project

1. Open **Projects**.
2. Select **Create project** if the button is available.

3. Enter name, description and required settings.
4. Choose the default data language if shown.
5. Add additional languages only when the project sources need them.
6. Save the project.
7. Confirm that it can be selected as the active project.

Key concepts

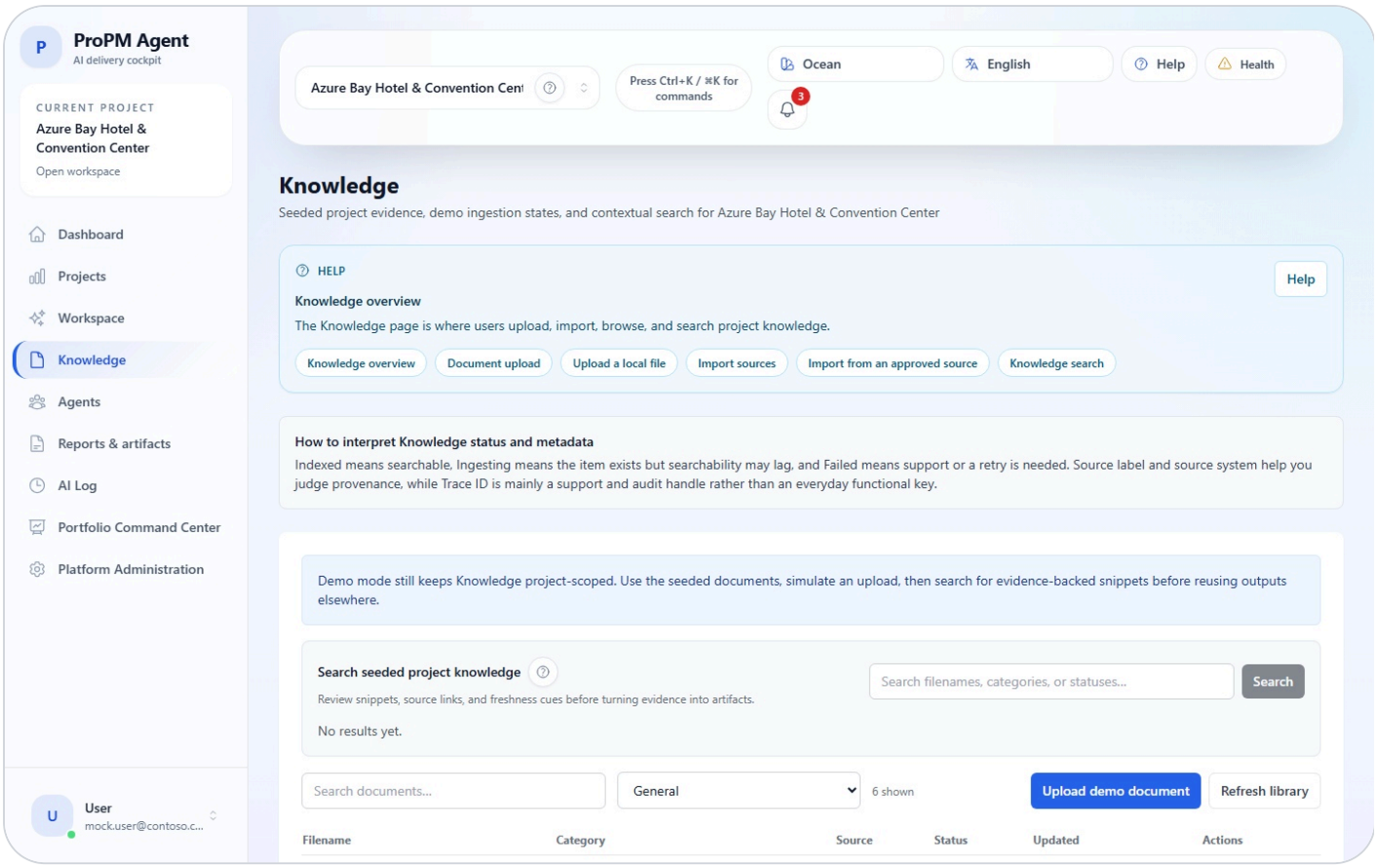
CONCEPT	MEANING
Active project	Current operational context for project pages
All projects	Scope limited to projects the same account can already access
Locally remembered project	Browser-local preference that may differ across devices

Common issues

SYMPTOM	LIKELY CAUSE	RECOMMENDED ACTION
Create button missing	Insufficient role	Ask for the required role
Project not visible	You are not a project member	Ask the Project Owner
Pages stay empty	No active project	Select the project in the top bar
Data does not match	Wrong active project	Switch project and reload

Knowledge, documents and imports

Home · Knowledge, documents and imports



Goal

This page explains how to add project knowledge, verify indexing, import from an approved source and read search results with citations.

Upload a document

1. Open **Knowledge**.
2. Select **Upload**.
3. Choose the file.
4. Enter the category or metadata requested by the UI.
5. Start the upload.
6. Verify that the document appears in the list.
7. Wait for **Indexed** before using it as evidence.

Import from an approved source

1. Open **Knowledge**.
2. Select the import action if available.
3. Choose the approved source.
4. Select the scope proposed by the UI.
5. Start the import.
6. Open import history to review imported, skipped and failed items.

Statuses

STATUS	MEANING	ACTION
Indexed	Ready for search and agents	Use the document
Ingesting	Processing is still running	Wait and refresh
Failed	Processing failed	Check format/source and contact support if needed
Not visible	Wrong project, filter or permission	Check active project and filters

Search evidence

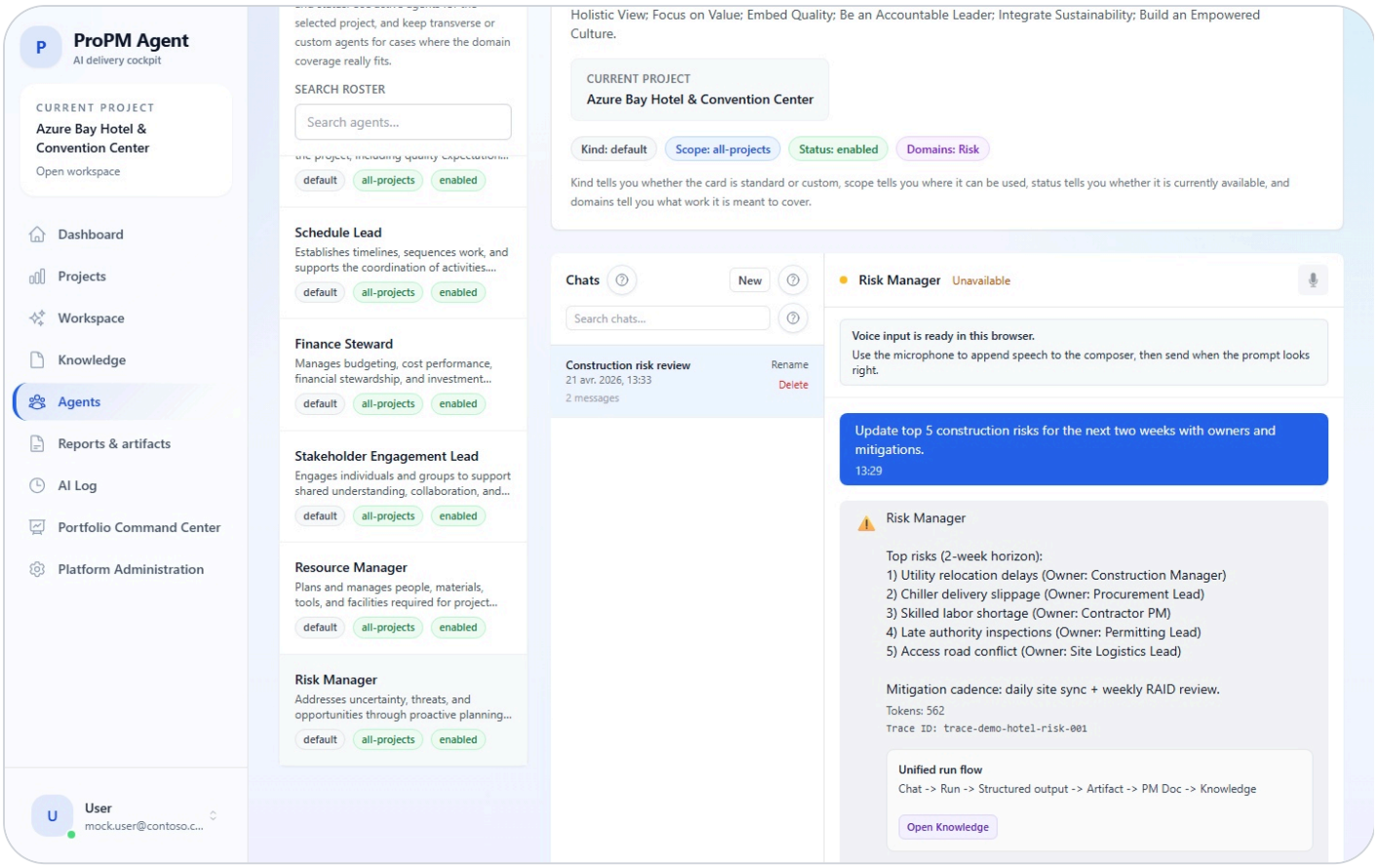
1. Open **Knowledge**.
2. Enter a business term, question or document identifier.
3. Filter by category, source or freshness if available.
4. Open a relevant result.
5. Read snippet, citation, score, source label and source system when shown.

Related pages

- Agents
- Structured outputs, evidence and freshness
- Support, audit and diagnostics

Agents

Home · Agents



Goal

This page explains how to choose an agent, run it, understand local sessions and create a custom agent when your role allows it.

Standard agents

NEED	AGENT
Cross-functional analysis	Orchestrator
Decisions, policies or approvals	Governance
Scope, deliverables and requirements	Scope
Schedule, milestones and dependencies	Schedule
Budget, cost and variance	Finance

NEED	AGENT
Communication and stakeholders	Stakeholders
Capacity and resources	Resources
Risks, issues and responses	Risk

Run an agent

1. Open **Agents**.
2. Verify the active project.
3. Select the agent.
4. Write a precise prompt with objective, context and expected format.
5. Attach or reference documents if the UI supports it.
6. Start the run.
7. Read the answer, citations, freshness and confidence.
8. Open **AI Log** if you need to audit the execution.

Custom agents

1. Open **Agents**.
2. Select the custom-agent creation action if available.
3. Name the agent and describe its business purpose.
4. Choose project scope or **All projects** when allowed.
5. Save and run a simple test.
6. Delete or disable the agent when it should no longer be used.

Important notes

- An agent response is not automatically a final PM Document.
- Browser-local chat sessions are not a central shared archive.
- Voice input is a convenience feature and depends on the browser.

Structured outputs, evidence and freshness

Home · Structured outputs, evidence and freshness

Make procurement and schedule owners explicit before the first controls review.

Owner: Procurement Lead Level: program Due: 2026-01-13

Assumptions

- Authority inspection slots remain available through the planned enabling-works window.

Missing information

- Confirm final sponsor attendance and decision authority for the kickoff SteerCo.

Watchpoints

- Long-lead equipment exposure should remain visible until procurement awards are contractually locked.

Evidence

Project Charter

`blob://demo/demo-hotel-001/01-project-charter.md`

Fresh Authority 5

Authorize and govern the delivery of a 220-room upscale urban hotel, conference center, and wellness facilities.

System: Knowledge

Master Schedule

`blob://demo/demo-hotel-001/04-master-schedule.md`

Aging Authority 4

Commissioning complete remains targeted for 2028-03-15 with a recovery trigger when SPI < 0.95.

System: Schedule

Review a structured output before reuse

Follow the steps in order: every AI answer remains a draft until human review is complete.

1. **Summary and findings**
   Read the main message, then separate verifiable facts from model interpretation.
2. **Decisions to take**
   Identify what requires approval, arbitration, project owner review or stakeholder sign-off.
3. **Evidence and citations**
   Open cited sources and verify that they really support the conclusion.
4. **Freshness**
   Classify information before reuse: Fresh | Aging | Stale | Conflict
5. **Confidence**
   Use confidence as a prioritization signal, never as automatic approval.
6. **Missing information**
   Complete knowledge, request a source, or keep the answer exploratory.
7. **Next actions**
   Choose only after review: PM Document, AI Log audit, approval action or project follow-up.

Human review is required before publication

No AI output should become a final deliverable without evidence, context and decision review.

Goal

This page explains how to review an agent structured output before using it in a decision, PM Document, publication or governed action.

Read a structured output

1. Read the **Summary**.
2. Read **Findings** and **Decisions**.
3. Open **Evidence** and citations.
4. Check **Source freshness**.
5. Read **Confidence** and **Missing information**.

6. Decide whether the output stays exploratory, becomes a PM Document or needs escalation.

Decision table

SITUATION	RISK	RECOMMENDED DECISION
No evidence	Result cannot be justified	Do not publish; add context or source
`stale` source	Source is too old	Refresh, reimport or confirm manually
`conflicting` source	Sources disagree	Human arbitration required
`unavailable` source	Evidence could not be retrieved	Treat as an alert, not a proof
Low confidence	Result is uncertain	Keep exploratory or request review
External action proposed	Impact outside ProPM Agent	Use Actions and approvals
Shareable deliverable	Versioning and governance needed	Open PM Documents and artifacts

Human review

AI outputs require human review before publication, sponsor decision, external communication, customer notification, ticket creation or governed action. Confidence is not approval.

Downstream flow

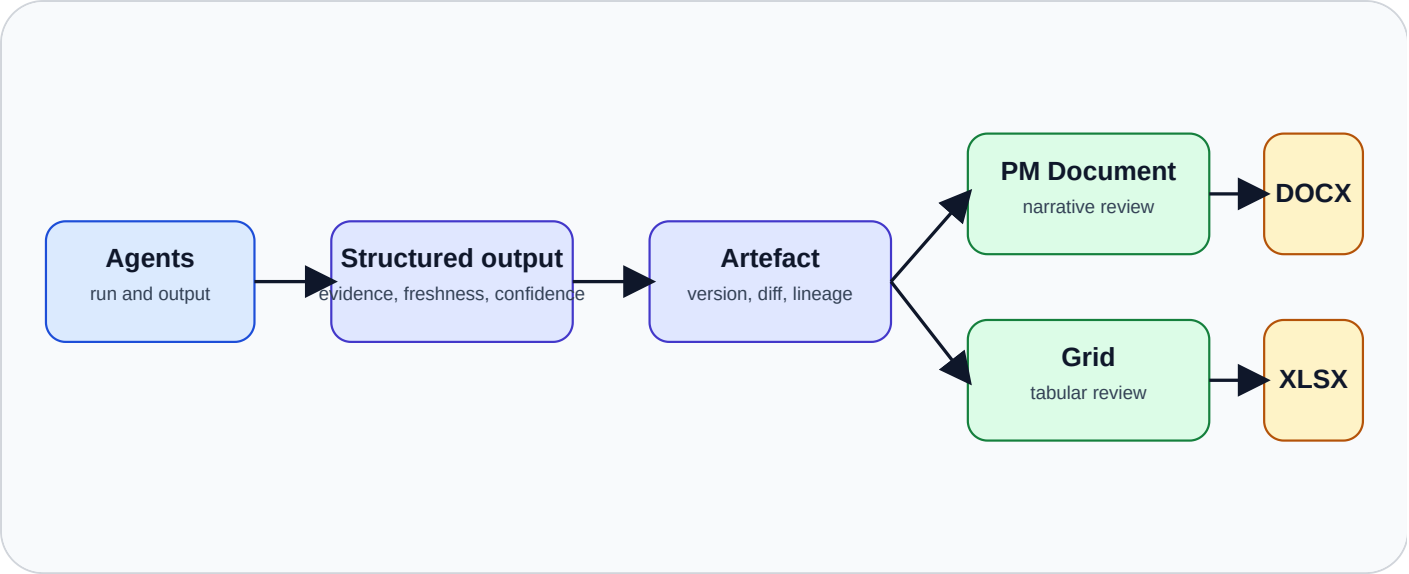
The recommended flow is **run → structured output → artifact → version → PM Document → Download / Publish / Add to knowledge**.

Support and audit IDs

Fields such as `Trace ID`, `Structured output ID` and `Context snapshot ID` are useful for support and audit. They do not replace business evidence review. Use AI Log and Support diagnostics for investigations.

PM Documents and artifacts

Home · PM Documents and artifacts



Goal

This page explains the review flow for artifacts and PM Documents: find a deliverable, review content, compare versions, download, publish or add it to knowledge.

Lifecycle

The common flow is **run → structured output → artifact → version → PM Document → Download / Publish / Add to knowledge**.

TERM	MEANING
Structured output	Agent result that still needs human review
Artifact	Governed object from a run or review
Version	Specific artifact state with history and diff
PM Document	Reviewed project document ready for downstream action

Review a document

1. Open **PM Docs**, **Reports** or **Reports & artifacts**.
2. Verify the active project.

3. Filter by status, type, date or author if available.
4. Open the document.
5. Review content, metadata, lineage and available actions.
6. Save edits if your role allows it.
7. Check the diff before approval or publication.

Actions

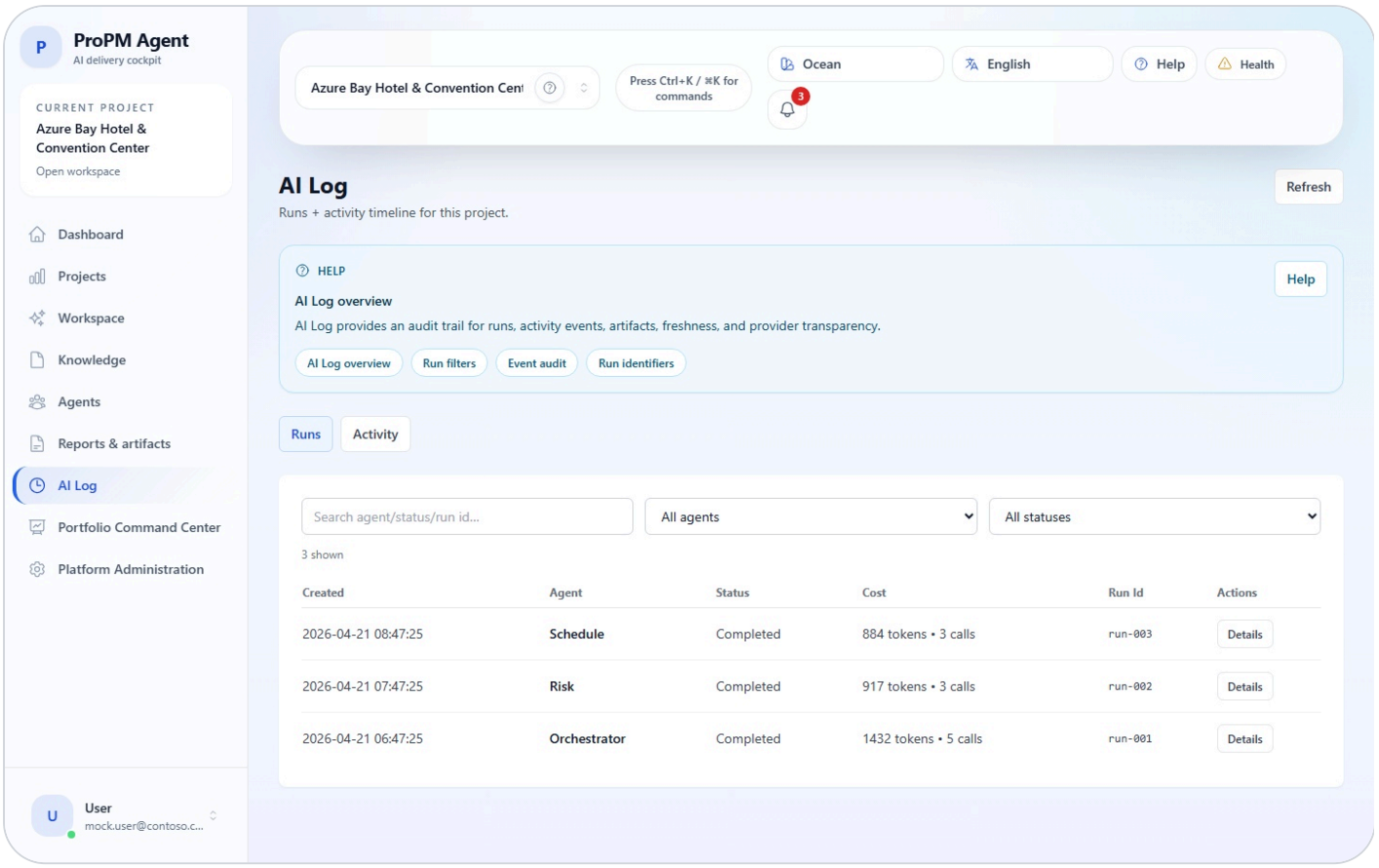
ACTION	EFFECT
Download	Downloads a local copy
Publish	Publishes to a governed destination
Add to knowledge	Adds the reviewed document back to project knowledge

Common issues

SYMPTOM	LIKELY CAUSE	RECOMMENDED ACTION
Document not found	Wrong project or filters	Check active project and reset filters
Publish unavailable	Missing destination, policy or role	Check project integrations and policy
Diff unexpected	Wrong reference version	Open lineage and compare versions

AI Log

Home · AI Log



Goal

AI Log shows what happened during an execution: run, activity, effective provider, timestamp, user, project, status and support identifiers.

Find a run

1. Open **AI Log**.
2. Select **Runs**.
3. Filter by project, status, agent, user or period when available.
4. Open the run.
5. Check status, time, effective provider and links to artifacts or PM Documents.

Verify the effective provider

1. Open the run details.
2. Find **Effective AI Provider** or its equivalent.

3. Compare it with the expected provider.
4. Note any discrepancy.
5. If it repeats, ask an administrator to check AI provider.

Selected vs effective provider

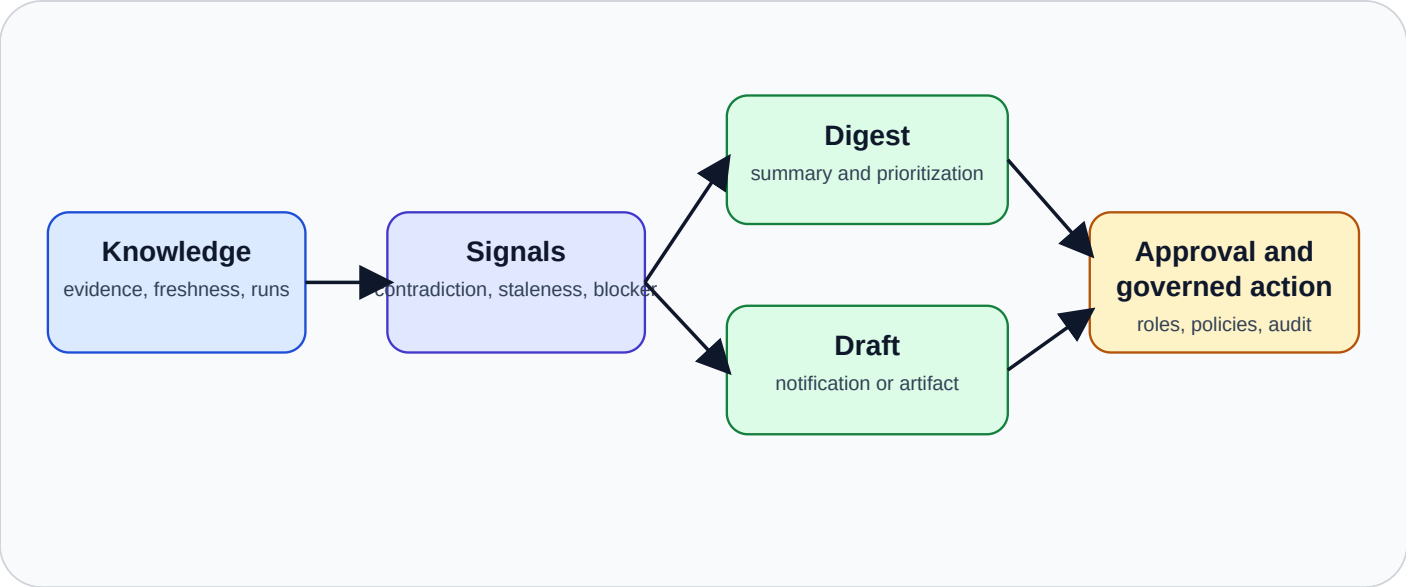
TERM	MEANING
Deployment-selected provider	Initial Marketplace choice
Configured provider	Current platform administration setting
Effective AI provider	Provider actually used by a given run

Support guidance

Collect Run ID, Trace ID, Artifact ID or Context snapshot ID when visible. Do not copy full sensitive payloads unless support asks for a targeted extract.

Signals and notifications

Home · Signals and notifications



Goal

This page explains how to handle operational signals and notifications without confusing alerts, drafts, digests and governed actions.

Read a signal

1. Open **Signals**, **Inbox** or the signal area in **Workspace**.
2. Check severity, status, mode and date.
3. Open the signal details.
4. Read evidence, sources and recommendations.
5. Decide whether to handle, snooze, dismiss, draft or escalate.

Common actions

ACTION	EFFECT
Refresh	Reloads or recalculates visible signals
Generate digest draft	Prepares a summary digest
Create draft	Creates a usable draft

ACTION	EFFECT
Snooze 24h	Temporarily delays the signal
Dismiss	Closes the signal

Notification center

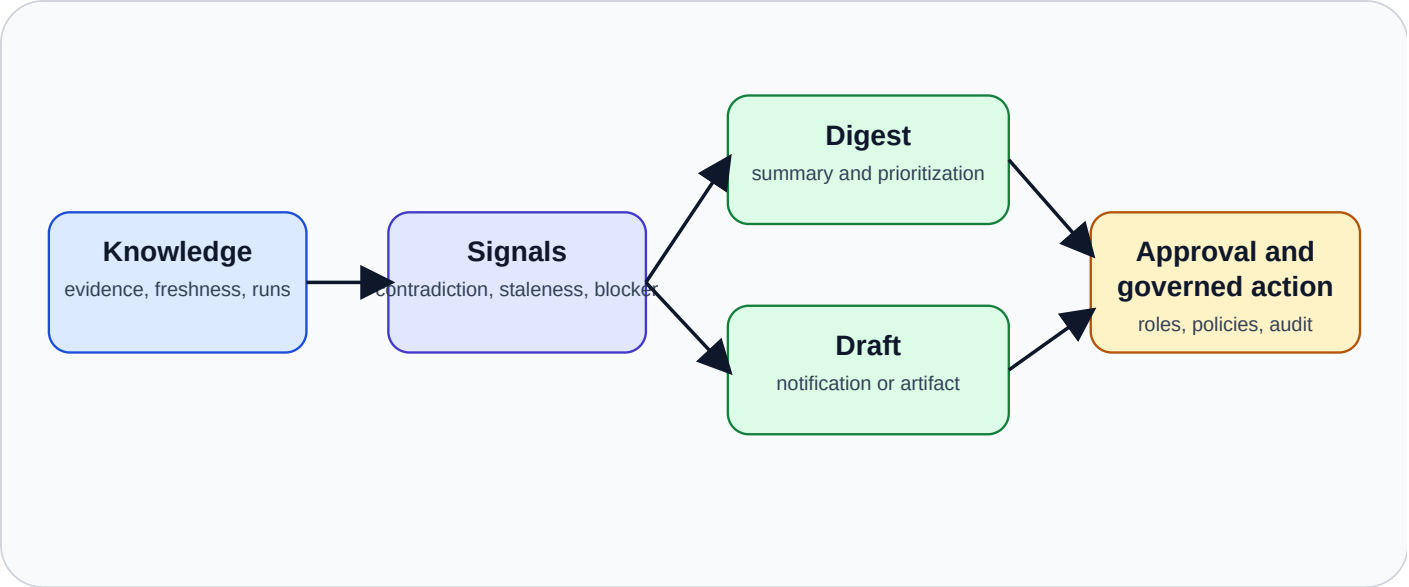
1. Open the notification icon in the top bar.
2. Select a notification to open the linked page.
3. Use **Mark all read** if available.
4. Use **Clear** only when the local list is no longer needed.

Related pages

- Actions and approvals
- Governance, decisions and actions
- Support, audit and diagnostics

Governance, decisions and actions

Home · Governance, decisions and actions



Objective

This page explains, in a simple way, how ProPM Agent moves from:

1. a **signal** detected;
2. to a **decision**;
3. then to a **governed action**;
4. possibly subject to **approval**;
5. before it is **executed** and **traced**.

This page helps you clearly understand:

- what a **governance policy** is;
- what `allow`, `require_approval` and `deny` mean;
- what `observe`, `draft`, `propose` and `execute` mean;
- how to use Actions & approvals step-by-step;
- why an action can be visible but blocked.

Very simple view of the flow

In ProPM Agent, the normal path is:

1. a **signal** grabs attention;
2. the team reads it;

3. **governance** decides what each role is allowed to do;
4. if an external output is required, an **action** is created;
5. if the project requires it, the action goes into **approval**;
6. the action is then **executed** or **rejected**;
7. the trace remains visible in **activity** and the **AI Journal**.

Part 1 — Understanding signals

A **signal** is a structured alert that says: **“this topic deserves attention.”**

Signal examples

A signal can surface:

- a **freshness** of sources that is insufficient;
- a **contradiction** between multiple pieces of evidence;
- a **project blockage**;
- a follow-up that deserves a **notification**, a **decision** or an **external action**.

What the user usually sees on a signal card

A signal card can display:

- a **title**;
- a **summary**;
- an **explanation**;
- a **severity**;
- a **status**;
- a **mode**;
- a number of **evidence** or retriggers;
- actions such as **Create draft**, **Snooze 24h** or **Dismiss** depending on the role.

Recommended step-by-step to handle a signal

When you open a signal, keep this order:

1. read the **summary**;
2. then reread the **explanation**;
3. check the **evidence** and **freshness**;
4. decide if the topic only needs monitoring, a draft or a real action;

5. if an external output becomes necessary, move to **Actions & approvals**.

Useful signal statuses

STATUS	WHAT IT MEANS
`open`	the topic remains active and still needs attention
`snoozed`	the topic is temporarily paused
`dismissed`	the topic is removed from the active queue
`resolved`	the topic is considered handled

Useful signal modes

MODE	SIMPLE READING
`inform`	the signal informs, without requiring immediate action
`suggest`	the signal suggests a next step
`draft`	the signal is already oriented toward a draft or preparation
`request_approval`	the signal calls for a governed review or approval

Part 2 — Governance policies

What is a governance policy?

A **governance policy** is a rule that answers the question:

“Who is allowed to do what, on which connector, and with what level of control?”

In other words, governance prevents an external action from leaving without a clear framework.

What a policy decides

A policy generally answers four questions:

1. **Who?** — which role is concerned;
2. **On what?** — which connector, action type or destination is concerned;

3. **How far?** — simple observation, draft, proposal or execution;
4. **With what effect?** — allowed, allowed with approval, or denied.

Very simple example

A policy might say:

- the **Contributor** can prepare a Teams draft;
- the **Project Manager** can propose a SharePoint publication;
- the **Project Owner** must approve before execution;
- no one else can directly execute that publication.

Understanding levels: `observe`, `draft`, `propose`, `execute`

The **level** describes how far a role can go in the flow.

LEVEL	WHAT THE USER CAN DO	WHAT THEY CANNOT YET DO	SIMPLE EXAMPLE
`observe`	see information, follow the topic, consult the queue	create an action or draft	a reader follows signals without preparing an output
`draft`	prepare a draft, a text, an intention to act	officially submit the action to the queue	a contributor prepares a Teams message but does not propose it
`propose`	submit a real action request in the governed queue	execute the action directly	a project manager proposes a Jira ticket
`execute`	launch the real execution if other conditions are met	bypass the policy or imposed approvals	a project owner publishes an artifact to SharePoint

Very simple reading

- `observe` = I look;
- `draft` = I prepare;
- `propose` = I officially request;
- `execute` = I actually launch.

Understanding effects: `allow`, `require_approval`, `deny`

The **effect** describes what the platform does when a role reaches this level.

EFFECT	WHAT IT MEANS	PRACTICAL CONSEQUENCE
`allow`	the action is permitted at this level	the flow can advance without an extra approval step if everything else is ready
`require_approval`	the action is possible but must be approved	the approval queue becomes mandatory before execution
`deny`	the action is forbidden for this role or scope	the user cannot go further on this flow

Very simple reading

- **`allow`** = yes;
- **`require_approval`** = yes, but after human validation;
- **`deny`** = no.

How to read a policy line

Take this reading:

- **Role**: Project Manager
- **Connector**: SharePoint publish
- **Level**: `execute`
- **Effect**: `require_approval`

This means:

- the Project Manager can go as far as an execution request;
- but the publication does not leave immediately;
- approval is required before the real execution.

Concrete policy examples

BUSINESS CASE	ROLE	RECOMMENDED LEVEL	RECOMMENDED EFFECT	WHY
Publishing a report to SharePoint	Project Manager	`execute`	`require_approval`	the output is external and must be reviewed
Creating a Jira ticket from a blockage	Project Manager	`propose`	`allow` or `require_approval`	the project can request a ticket without necessarily opening it automatically

BUSINESS CASE	ROLE	RECOMMENDED LEVEL	RECOMMENDED EFFECT	WHY
Low-risk internal Teams message	Contributor	`execute` or `propose`	`allow`	quick communication with low impact
Outlook email to sponsors	Contributor	`propose`	`require_approval`	more sensitive and formal communication
Webhook to a third-party tool	Project Owner	`execute`	`require_approval`	technical output that must remain tightly controlled
Connector not ready or not authorized	all except admin	`observe` or no usage	`deny`	we avoid any accidental departure

Step-by-step to configure a governance policy

Follow this simple order.

Step 1 — Open the right surface

From the project **Workspace**, open **Governance policies**.

Step 2 — Choose the flow to control

First ask yourself:

- is it a **publication**;
- a **ticket**;
- a **message**;
- a **webhook**;
- or another external action?

Step 3 — Choose the concerned role

Then define which role can act:

- **Contributor**;
- **Project Manager**;
- **Project Owner**;
- or another role present in your configuration.

Step 4 — Choose the action level

Decide if this role should only:

- observe;
- prepare a draft;
- propose;
- or execute.

Step 5 — Choose the effect

Decide if this level should be:

- allowed directly (`allow`);
- allowed with approval (`require_approval`);
- or denied (`deny`).

Step 6 — Verify the connector or destination concerned

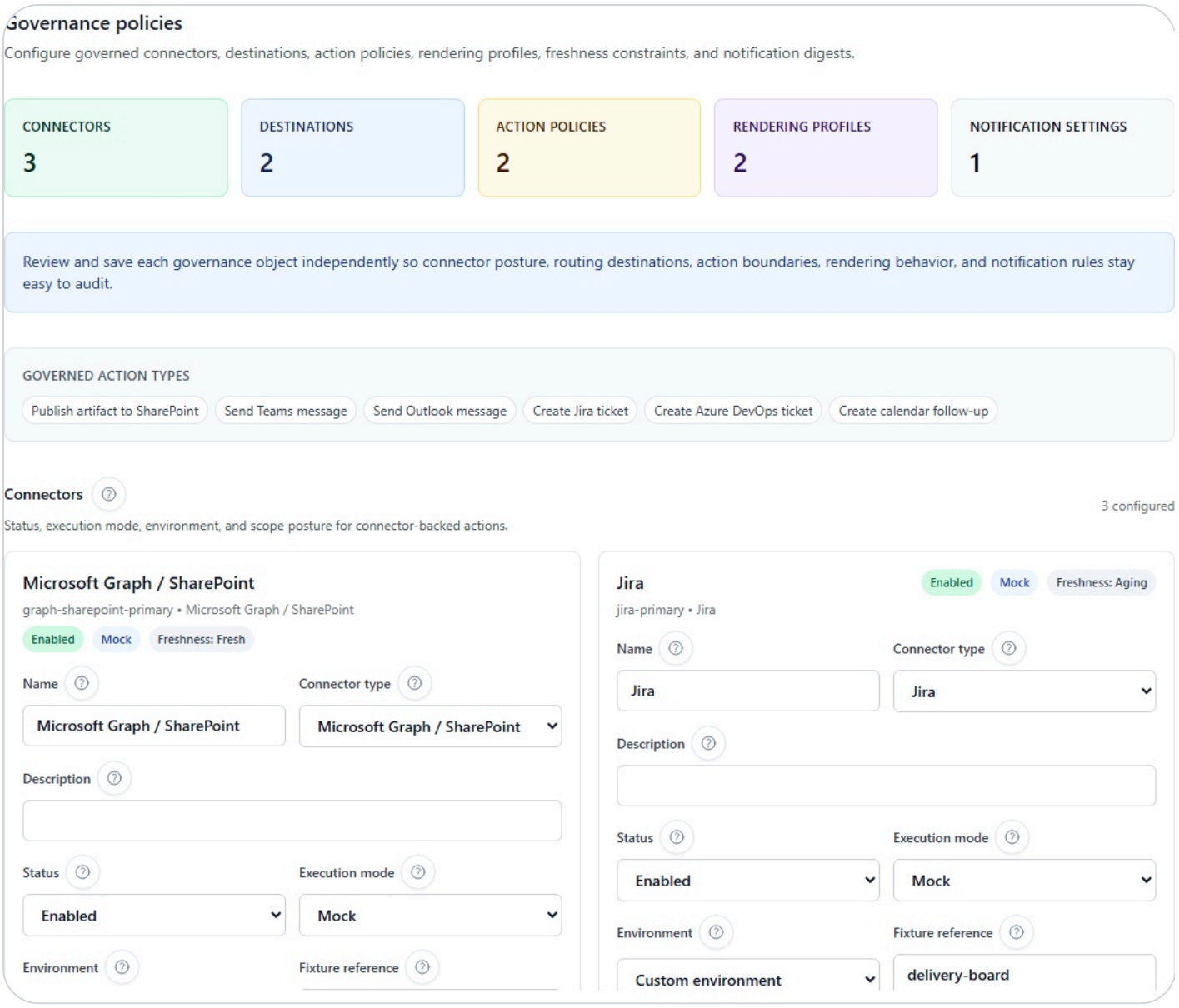
A good policy is not enough if the connector:

- is not technically ready;
- is not open to the project;
- or does not have the correct artifact destination.

Step 7 — Test with a non-admin role

The best control is practical:

1. log in with a realistic business role;
2. open **Actions & approvals**;
3. check what is visible, allowed or blocked;
4. adjust the policy if the behavior is not as expected.



Simple rules for good governance configuration

- use `allow` on `execute` only for low-risk flows;
- use `require_approval` whenever content leaves the project or modifies an external system;
- use `deny` when the connector is not ready, not authorized or too sensitive;
- keep rules consistent with the roles actually assigned;
- always test a real case before considering the policy ready.

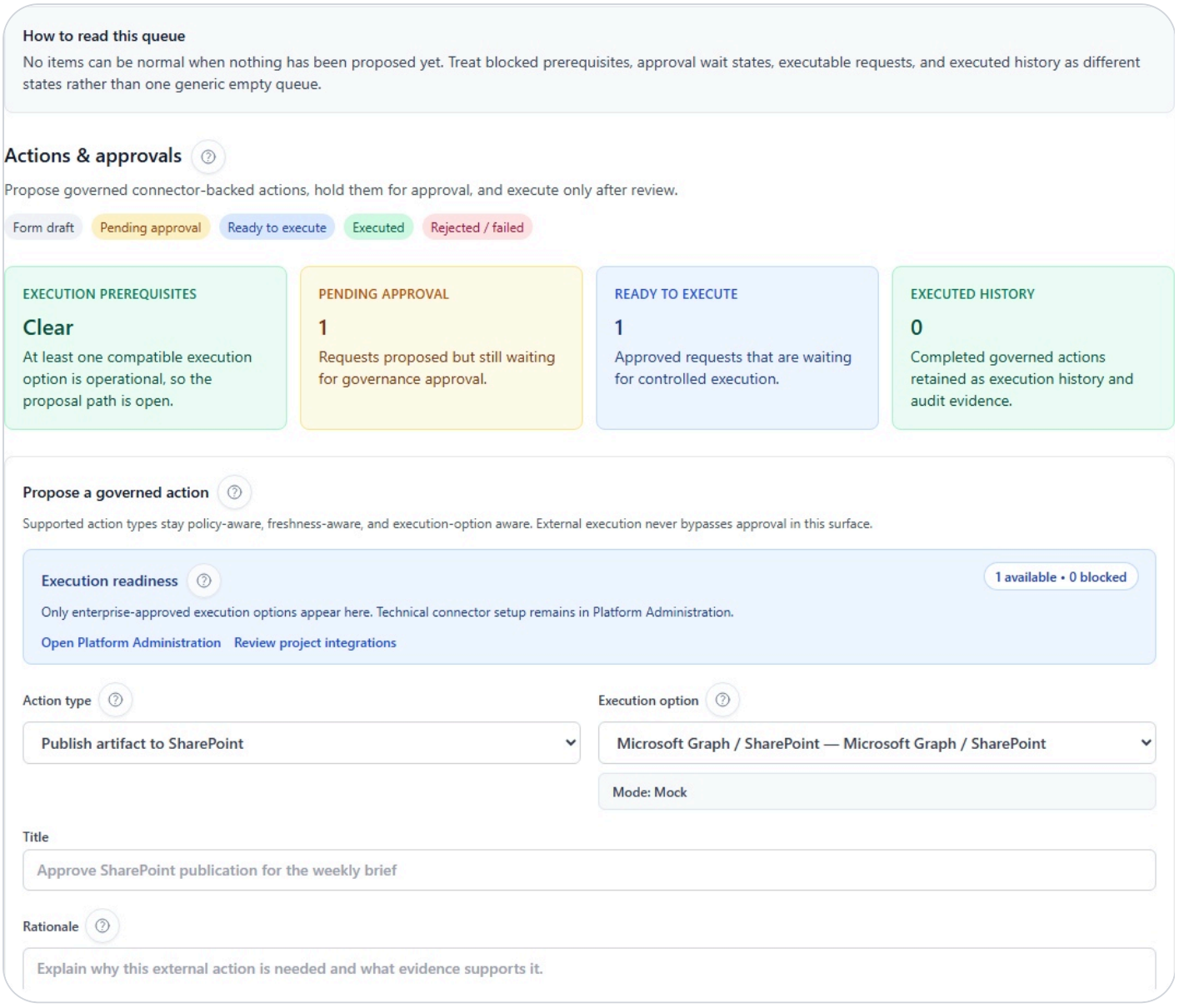
Common mistakes to avoid

MISTAKE	CORRECT READING
“I see the connector, so I can use it”	wrong: visibility does not guarantee authorization or technical health

MISTAKE	CORRECT READING
“`propose` means the action leaves”	wrong: `propose` means the request enters the governed queue
“`execute` means no control”	wrong: `execute` can still be combined with `require_approval`
“`deny` means a failure”	wrong: `deny` is often a normal governance decision

Part 3 — Actions & approvals

The **Actions & approvals** screen is used to turn an intention into a real **governed action**.



Four canonical readings to keep in mind

In the interface, this surface is easiest to read through four distinct states:

CANONICAL READING	WHAT IT MEANS
Execution prerequisites	compatible execution options may exist, but operational use is blocked or unavailable
Pending approval	a governed request was proposed and is still waiting for approval
Ready to execute	a request is already **approved**, but controlled execution is still a separate step
Executed history	the action was actually executed and remains visible as audit history

An **approved** request is therefore not yet an **executed** request.

When everything is correctly ready, what should I see?

In a nominal case, the authorized user usually finds:

- at least one relevant **action type**;
- a **compatible healthy execution option**;
- a truly active **project binding**;
- a **policy** that either allows proposal or routes to approval;
- a queue where requests then move through **pending approval**, **approved**, **executed** or **rejected** depending on the case.

If one of these links is missing, the screen can look empty, incomplete or purely informational without this necessarily being a failure.

What the user sees in this screen

A user typically finds:

- a **Propose a governed action** form;
- the choice of **action type**;
- the choice of **execution connector** or **execution option**;
- a **readiness summary** indicating what is available or blocked;
- fields such as **title**, **rationale**, **destination**, **message**, **ticket description**;
- an **approval and execution queue** with already submitted requests.

Step-by-step — create a governed action

Step 1 — Open the screen

In the **Workspace**, open **Actions & approvals**.

Step 2 — Choose the action type
First select the business intention. Visible action types include:

- **Publish artifact to SharePoint**;
- **Send Teams message**;
- **Send Outlook message**;
- **Create Jira ticket**;
- **Create Azure DevOps ticket**;
- **Webhook** depending on tenant configuration.

Step 3 — Verify the compatible execution option
The interface then searches for **compatible execution options**.
Choose an option:

- **healthy**;
- **authorized**;
- actually open to your project.

If no healthy option appears, the check usually concerns:

- the connector itself;
- its health status;
- the project binding;
- the policy;
- or your permission.

Step 4 — Read the readiness
The **Execution readiness** area is to verify you are not just preparing a theoretical action.
In practice:

- **available / healthy** = option usable;
- **blocked by health** = connector to check on the platform side;
- **blocked by entitlement** = legacy wording for a non-plan block; check configuration, binding, policy, permission, approval or health;
- **blocked by policy** = restrictive project governance;
- **blocked by permission** = your role is insufficient;
- no visible option = compatible connector missing, missing project binding or option not opened to your role.

Pedagogical reading of a screen that seems empty

When `Actions & approvals` shows nothing executable, distinguish three cases before concluding that the feature is broken:

1. execution-option availability could not be loaded;
2. compatible options exist but remain blocked or unhealthy;
3. no compatible option is configured yet for the current governed action type.

In all three cases, this is different from a queue that is simply empty because nobody has proposed an action yet.

WHAT YOU SEE	LIKELY DOMINANT REASON	USEFUL REFLEX
no selectable connector	no compatible healthy connector is ready	check **Project integrations** then **Platform Administration**
visible action but blocked button	permission, policy, or approval required	check the role, then governance
visible queue but nothing goes out	the request is still waiting for approval or downstream execution	reread the actual queue status
connector present but unusable	insufficient binding, health, configuration, policy or permission	verify the full platform → project → policy chain

Remember as well that only enterprise-approved execution options appear here. Technical connector preparation stays in **Platform Administration**, while project exposure is read through **Project integrations**.

Step 5 — Fill in the title and justification

Then complete:

- a **clear title**;
- a short but useful **justification**;
- fields specific to the chosen action type.

The justification should answer two questions:

1. **Why is this action necessary?**
2. **On what evidence or decisions does it rely?**

Step 6 — Complete the business fields

Fields change according to the action type.

ACTION TYPE	FREQUENTLY EXPECTED FIELDS
SharePoint publication	title, justification, `artifact ID`, destination, rendering profile, format
Teams message	title, justification, message body
Outlook message	title, justification, recipients, subject, message body

ACTION TYPE	FREQUENTLY EXPECTED FIELDS
Jira ticket	title, justification, ticket description, optionally project / board key
Azure DevOps ticket	title, justification, description, ticket type per connector
Webhook	title, justification and data useful to the target system

Step 7 — Propose the action

Once fields are filled, submit the request.

At this stage, the action may not yet have left. It can first enter the **approval queue**.

Step 8 — Review by the approver

If the policy requires `require_approval`, an approver must review:

- the title;
- the justification;
- the connector used;
- the payload or business detail;
- any related artifact or draft.

Step 9 — Approval, rejection or execution

Depending on the policy and the approver's role, the request can be:

- **approved**;
- **rejected**;
- then **executed** if everything is ready.

Step 10 — Verify the final trace

After execution, check:

- the action queue;
- project activity;
- the **Trace ID** if displayed;
- the **AI Journal** if the flow is reflected there;
- the presence of the artifact, ticket or message in the target tool.

How to read action statuses

STATUS	WHAT IT MEANS
`draft`	the request is still preparatory
`pending approval`	approval is awaited before real follow-up

STATUS	WHAT IT MEANS
`approved`	the request has been accepted
`executed`	the action has actually been launched
`rejected`	the request was denied
`failed`	the action was launched but did not complete correctly
`cancelled`	the request was cancelled

Step-by-step example — publishing an artifact to SharePoint

Situation

The team has reviewed a **weekly brief** and wants to publish it in SharePoint.

Path

1. open **Actions & approvals**;
2. choose **Publish artifact to SharePoint**;
3. select a healthy **SharePoint publish** option;
4. fill in the **title** of the action;
5. add a **justification**, e.g. “reviewed and approved version for weekly distribution”;
6. enter the **artifact ID**;
7. choose the **SharePoint destination**;
8. choose the **rendering profile** or **format** if requested;
9. propose the action;
0. if the policy requires it, wait for approval;
1. execute;
2. verify that the artifact is indeed published in SharePoint and traced in ProPM Agent.

Step-by-step example — create a Jira ticket

Situation

A signal reports a recurring blockage with planning impact.

Path

1. open **Actions & approvals**;
2. choose **Create Jira ticket**;
3. select an available Jira connector;

4. enter a clear title, e.g. “Supplier blockage on critical batch”;
5. complete the **ticket description**;
6. add the **justification** and useful evidence;
7. propose the request;
8. let the approver review if the policy requires `require_approval`;
9. execute;
0. then verify the external reference or created ticket.

Step-by-step example — send a Teams or Outlook message

Situation

The project must inform an internal group or a sponsor that a review is finished.

Teams path

1. choose **Send Teams message**;
2. select the authorized Teams connector;
3. draft a short, understandable message;
4. add the justification if the flow is governed;
5. propose, get approved if needed, then execute.

Outlook path

1. choose **Send Outlook message**;
2. select the Outlook connector;
3. fill in the **recipients**;
4. complete the **subject** and **message body**;
5. propose, get approved if needed, then execute.

Practical difference

- **Teams** is well suited for collaborative internal communication;
- **Outlook** is better for more formal and targeted communication.

Step-by-step example — webhook to a third-party tool

Situation

The company wants to trigger an internal flow to a custom tool.

Path

1. choose the action type or flow related to **Webhook**;
2. select a compatible webhook execution option;

3. fill in the title and justification;
4. complete the data useful to the target system;
5. propose the request;
6. get approval if the policy requires it;
7. execute;
8. check the result on the target system and in ProPM Agent audit.

Why an action may be visible but not executable

An action can be visible in the interface but remain blocked if:

- the compatible connector is not healthy;
- the project does not have the correct binding;
- the project policy forbids this level of action;
- your role does not allow proposing or executing;
- an approval is still pending;
- the connector, binding, policy, permission, approval or health state blocks the flow.

What to do if no execution option appears

Follow this order:

1. first check **Governance policies**;
2. then check **Project integrations**;
3. open **Platform administration**;
4. finally check **configuration**, **binding**, **permission** and connector health.

Full scenario — from signal to executed action

Simple case

1. an `open` signal reports a blockage;
2. the team reads the summary, explanation and evidence;
3. they decide a Jira ticket is needed;
4. the policy allows the **Project Manager** to `propose` but requires `require_approval`;
5. the Project Manager creates the request in **Actions & approvals**;
6. the **Project Owner** approves;
7. the action moves to **executed**;
8. the external ticket and internal trace remain aligned.

This scenario summarizes the product logic: **see, decide, control, execute, trace**.

Best practices

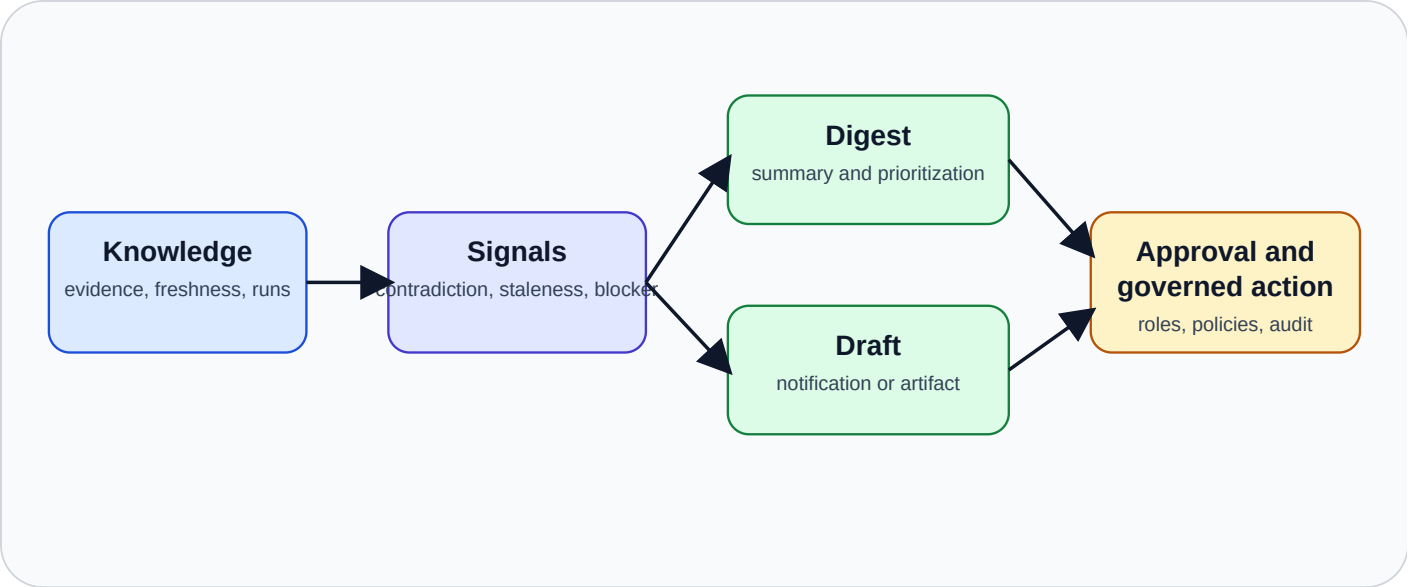
- never confuse **draft** and **real execution**;
- use policies to limit sensitive outputs;
- request approval whenever content leaves the project or changes an external system;
- always verify evidence before proposing an action;
- use the **AI Journal** and the action queue as a common audit reference.

Next

- Connectors and integrations
- AI Log
- Actions and approvals
- Maintenance, support and FAQ

Actions and approvals

Home · Actions and approvals



Goal

This page explains the governed action workflow: propose an action, get it approved, reject it when needed and understand when it is actually executed.

Propose an action

1. Open the signal, PM Document or page that proposes the action.
2. Select the available action, such as publishing to SharePoint or creating a Jira ticket.
3. Review content, destination and summary.
4. Add a justification if requested.
5. Submit the action.
6. Check its status.

Approve or reject

1. Open **Actions & approvals**.
2. Filter pending actions.
3. Open the action details.
4. Review evidence, destination, impact and requester.
5. Select **Approve** or **Reject**.

6. Add a clear reason when requested.

Approved is not executed

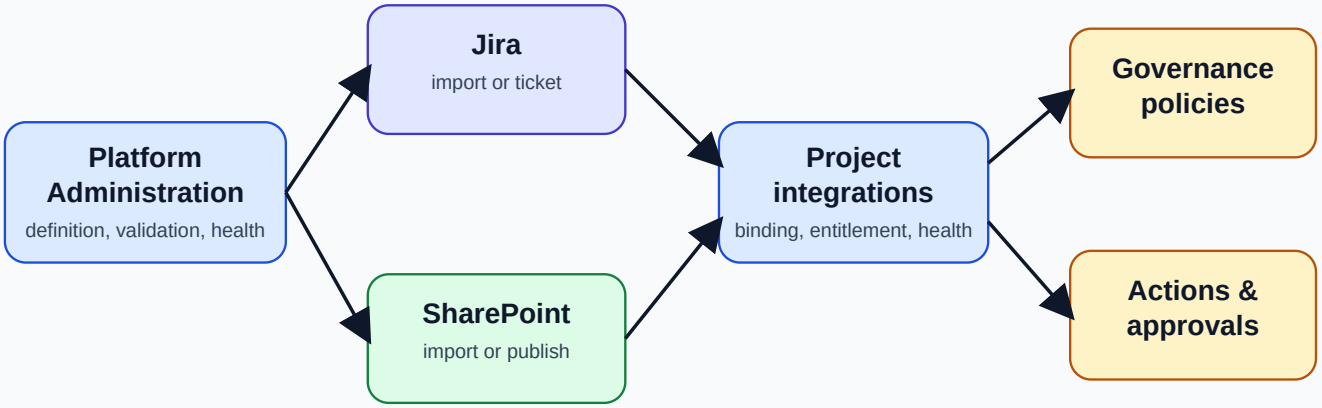
An **approved** action is authorized. **Executed** means the target action actually happened. External actions require a validated connector, permitted policy and live mode when applicable.

Common blockers

BLOCKER	MEANING
Insufficient role	Your role cannot propose or approve
require approval	Approval is mandatory
deny	The policy forbids the action
Not bound	The project is not connected to the integration
Not healthy	Connector or provider is not operational
Live mode off	No real external object will be created

Connectors and Integrations

Home · Connectors and Integrations



Goal

This page explains connectors for users and Project Owners. Technical fields, secrets and detailed authentication belong in secured administration, not in the end-user guide.

Three usages

USAGE	EXAMPLE	RELATED PAGE
Import into knowledge	Import SharePoint documents, Jira issues or Confluence pages	Knowledge
Publish a deliverable	Publish a PM Document to SharePoint	PM Documents
Create an external action	Create a Jira ticket or send a notification	Actions and approvals

Checklist before usage

CONDITION	CHECK
Platform ready	Integration is defined and validated by an administrator

CONDITION	CHECK
Project bound	Active project uses the integration
Compatible policy	Governance allows the action or requires approval
Sufficient role	Your role can import, propose, approve or publish
Healthy	Connector is ready and not blocked
Live mode	Real external actions are enabled when needed

Import from an approved source

1. Open Knowledge.
2. Choose the import action.
3. Select an available source.
4. Start import.
5. Review import history and statuses.
6. Verify that documents become **Indexed**.

Publish to SharePoint

1. Open PM Documents and artifacts.
2. Select the reviewed PM Document.
3. Check the proposed destination.
4. Submit publication if your role allows it.
5. Wait for approval if policy requires it.
6. Verify final status and trace.

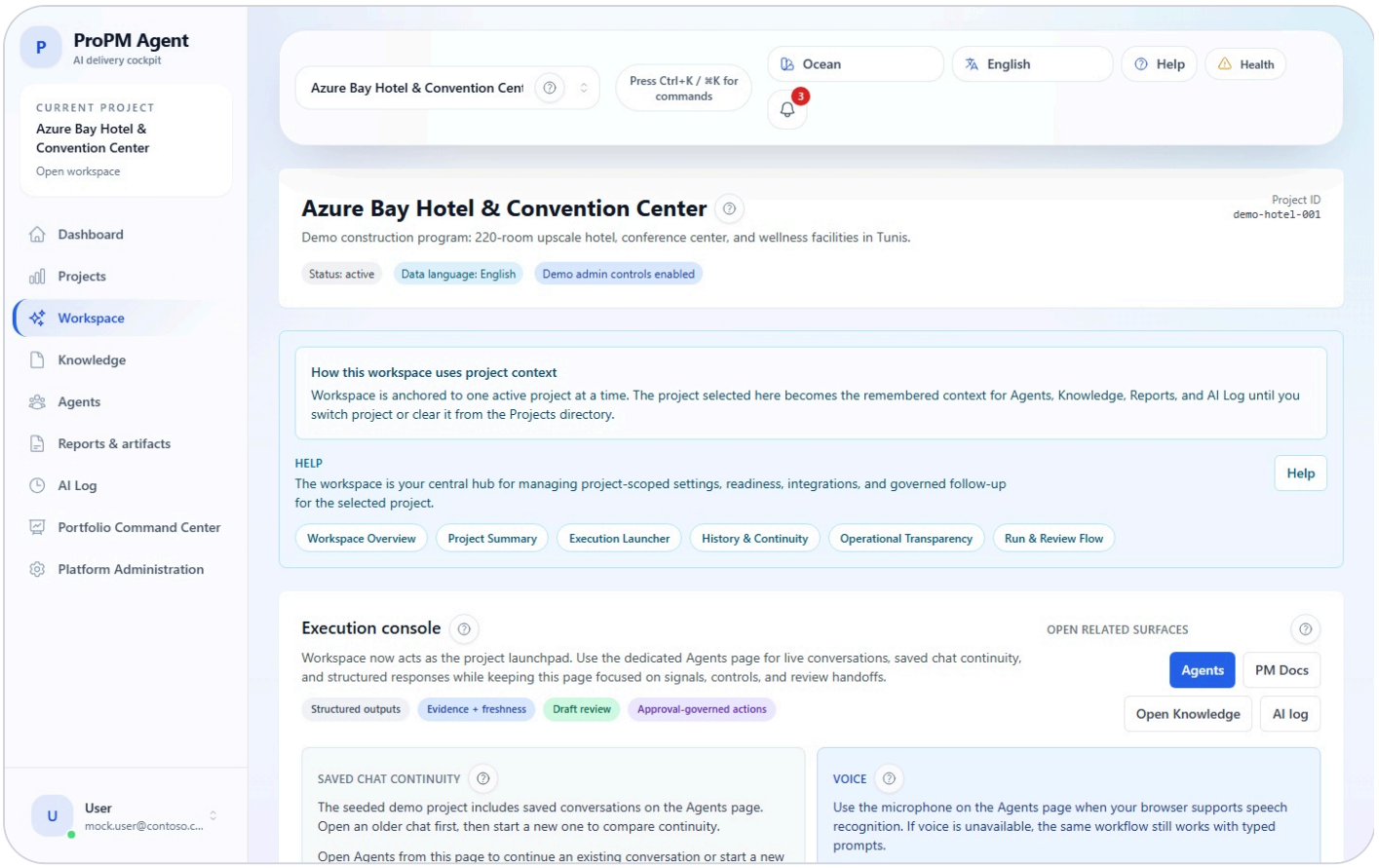
Create a Jira ticket from a signal

1. Open Signals and notifications.
2. Select the relevant signal.
3. Choose the ticket creation action if available.
4. Review title, description, priority and evidence.
5. Submit the action.

6. Track approval and execution in Actions and approvals.

Projects and workspace

Home · Projects and workspace



Goal

This page explains the difference between **Projects**, **Workspace** and **Agents**, then details the project settings actually visible in the application.

Three surfaces to distinguish

SURFACE	WHEN TO USE
Projects	Create a project, open an existing one, change context
Workspace	Read the project summary, operational transparency, signals and project-level settings
Agents	Start a live exchange with an agent and read the run’s structured output

In practice, **Projects** is used to enter the right context, **Workspace** to configure it, and **Agents** to exploit it.

The exact role of the active project

The **active project** is the context currently applied to project-scoped work pages. Concretely, it determines:

- the documents visible in **Knowledge**;
- the runs launched in **Agents**;
- the PM Docs, artifacts, and diffs visible in **Reports & artifacts**;
- the runs and events displayed in **AI Log**;
- the signals, integrations, and policies displayed in **Workspace**.

It should therefore not be confused with:

- the **active project**: current operational context;
- **Portfolio**: multi-project comparison surface;
- **All projects**: optional custom-agent scope visible across projects for the same account.

Create a project

The form contains the following fields:

- **Project ID**;
- **Name**;
- **Description**;
- **Default data language**;
- **Additional data languages**.

Input recommendations:

- choose a readable and durable ID;
- do not confuse **project data language** with **interface language**;
- correctly define the scope before opening knowledge or agents.

Project creator: initial rights and delegation

At creation, the **project creator** starts with the role **Project Owner** and all available project permissions. In practice, this is the person who can open the project, verify the initial configuration and then **delegate roles** to the rest of the team.

Recommended delegation immediately after creation

1. open **Access control**;
2. add at least one other **Project Owner** or a trusted **Project Manager**;
3. create, if needed, targeted custom roles rather than multiplying owners;
4. then assign roles to contributors, readers and auditors;
5. finally review the **Governance Policies** and **Project Integrations** to align rights, connectors and validations.

What the platform still protects

- the creator’s entry remains protected;
- the creator’s role remains fixed in the interface;
- delegation is done by assigning additional roles, not by removing creator protection;
- for detailed RBAC, see Access control and project roles.

Open and change project

A project can be opened from:

- the **Projects** page;
- the top-bar **project selector**;
- the recently remembered context in the browser.

When you change project, the following surfaces align: **Knowledge**, **Agents**, **PM Documents / Reports & artifacts**, **AI Journal**, signals and project settings.
Changing project therefore really changes the **active context** used by document search, agent conversations, reports and related traces.
The last retained project may be remembered locally by the browser to ease resumption, but this local memory is not a platform-wide shared setting.

The workspace: project command center

The **Workspace** brings together in a single surface:

- the **project summary**;
- shortcuts to **Agents**, **PM Documents** and **AI Journal**;
- a view of **operational transparency**;
- the project’s **signals**;

- project-level setting tabs.

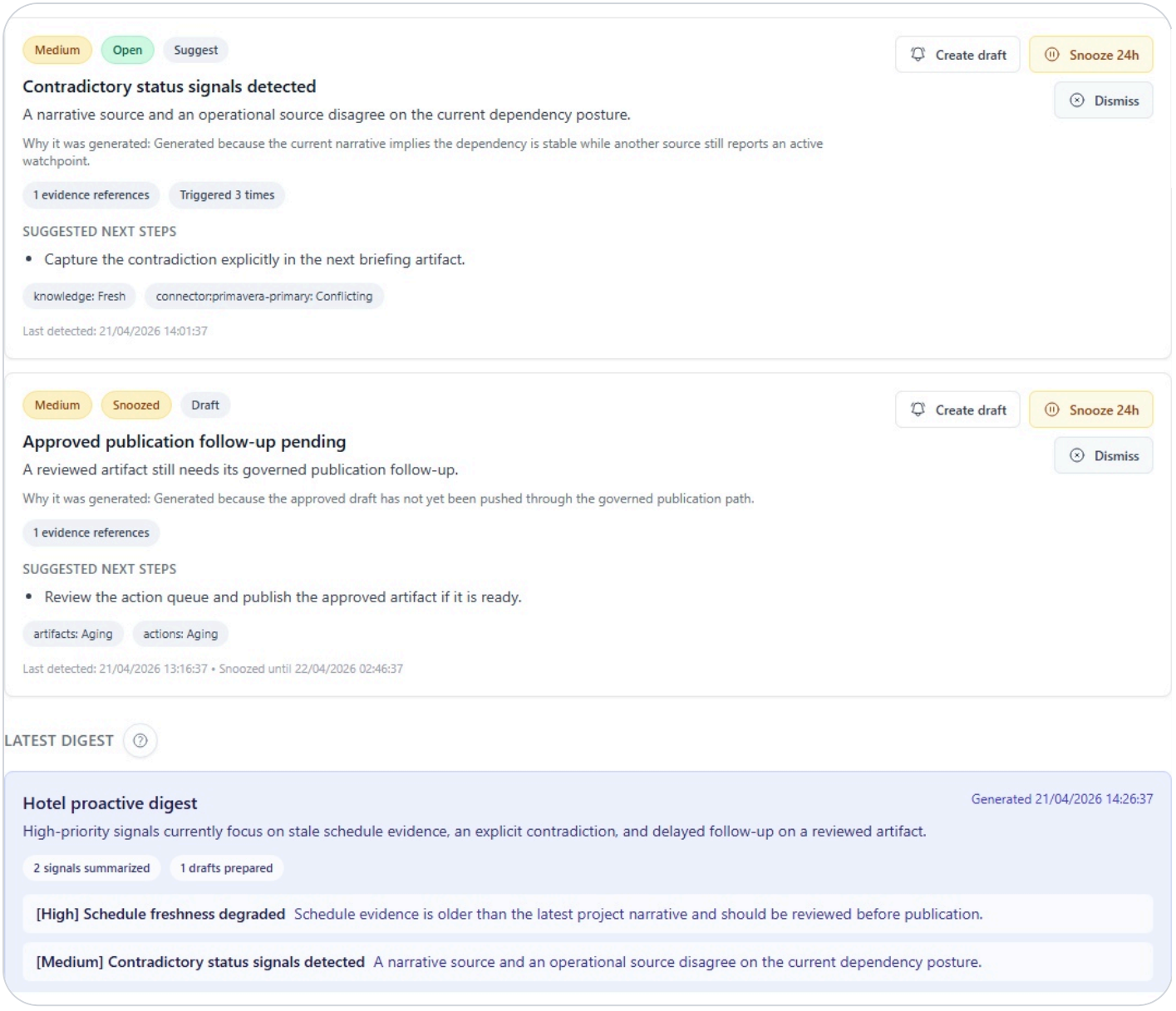
The **Workspace** no longer presents a dedicated **voice** entry point. When voice input still exists in some environments, it is available in **Agents**, not as a separate workspace entry point.

Operational transparency and preparation

The workspace is not only for summarizing the project. It also lets you see if the project is ready to act:

- presence or absence of signals;
- recent activity;
- shortcuts to related drafts or deliverables;
- preparation of project integrations when they exist;
- exposure of the actual AI provider without opening tenant configuration.

Use this area to understand **why** an action or import may be available, confirmed or blocked.



How signals, digests, and drafts arrive

In the interface, the project signals panel rereads three shared platform flows for the **active project**:

- current **signals**;
- recent **digests**;
- **notification drafts** linked to those signals.

Useful reading:

- opening the workspace loads the shared state already known for that project;
- **Refresh** explicitly asks the system to re-evaluate the project and pull the latest proactive signals;
- **Generate digest draft** builds a fresh grouped summary and can prepare `in_app` notification drafts;

- these items are therefore not just browser notes or local chat leftovers.

Project-level tabs

TAB	PURPOSE
Agent configuration	Set agents for this project
Access control	Manage members, roles and project-level permissions
Document categories	Adapt the project’s document taxonomy and propagate it to the project’s document surfaces
Governance policies	Define connectors, destinations, action policies, rendering profiles and project-level notification preferences
Project integrations	Link ready and authorized integrations to the project
Actions & approvals	Manage action requests, validations and governed execution

Agent configuration

Project-level parameters confirmed are:

- `status`;
- `temperature`;
- `max tokens`.

Visible constraints

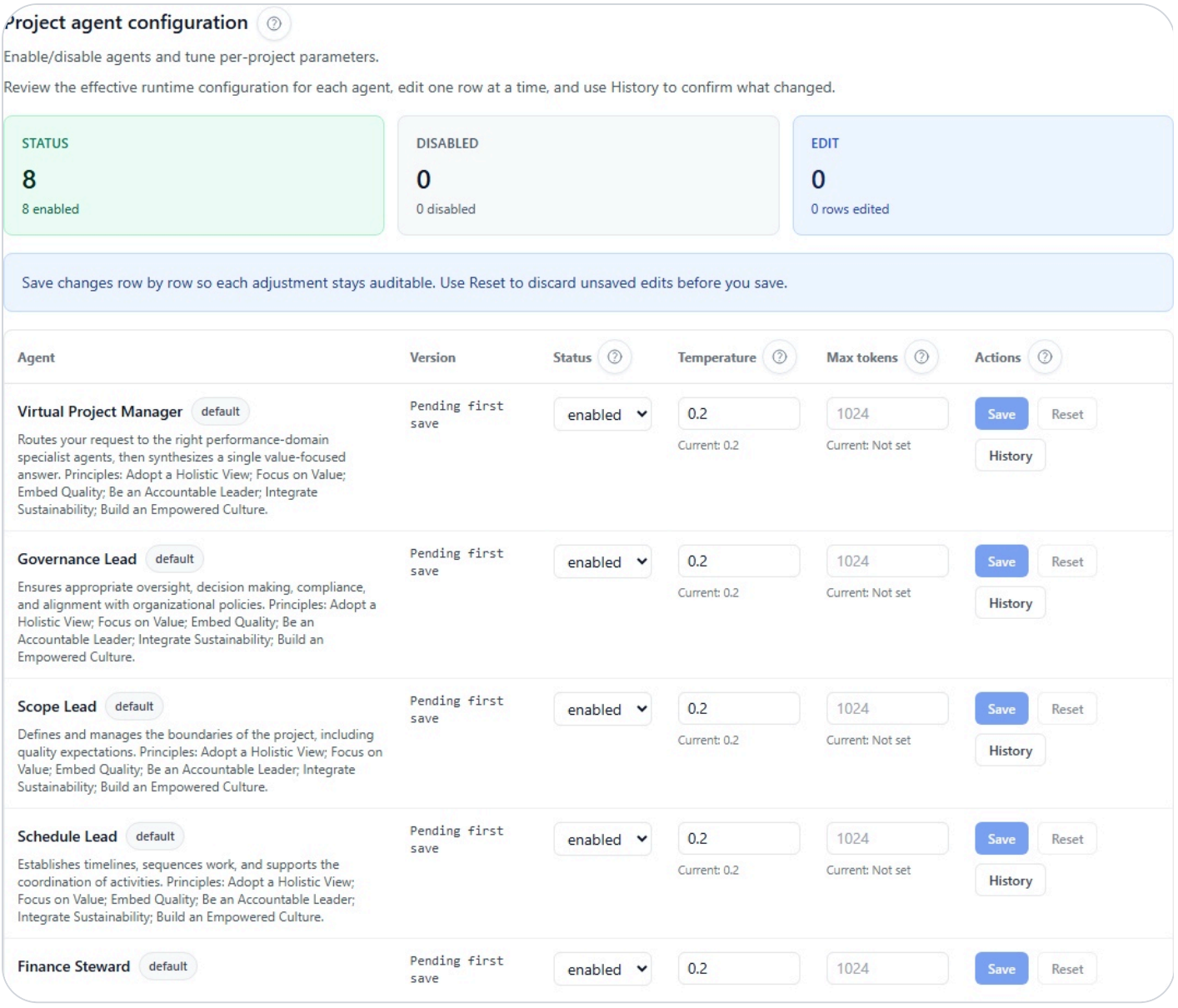
- `temperature` is expected between **0** and **2**;
- `max tokens` must be an integer ≥ **1**.

Configuration history

The interface also exposes a history by version with at least:

- version number;
- status;
- temperature;
- max tokens;
- creation date;

- author;
- associated `Trace ID`.



Access control

The **Access control** tab administers project members and roles. It supports:

- standard roles;
- custom roles;
- RBAC safeguards;
- read-only for profiles not authorized to modify.

See the dedicated page: Access control and project roles.

Document categories

This tab aligns the document classification with the project. In practice, the project taxonomy influences the categories offered during uploads and some document selectors used later in project surfaces.

Concrete effect of an update

When the category list is successfully updated:

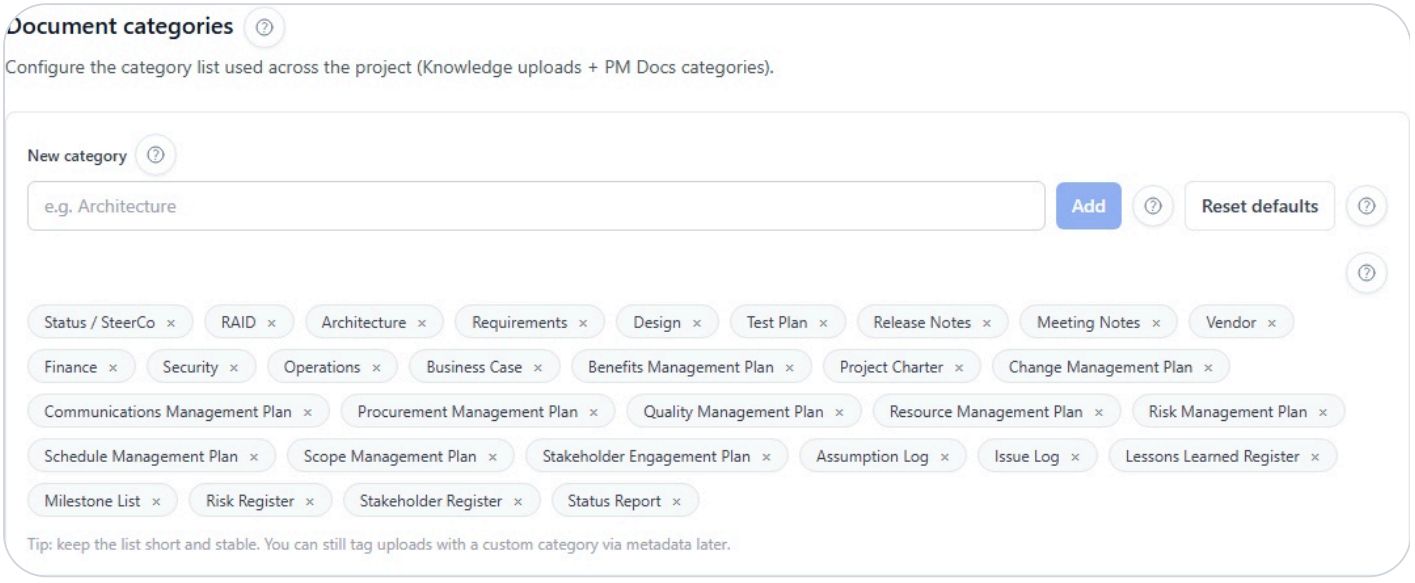
- the upload category selector in **Knowledge** is updated;
- category selectors and filters in **PM Documents** align when they use this shared taxonomy;
- the change remains limited to the **current project**.

Practical examples

Keep a short and stable taxonomy. For example, instead of multiplying close variants, prefer a few coherent categories such as:

- project charter;
- risk register;
- status report;
- procurement plan;
- communication plan.

The goal is not to encode the document version in the category, but to keep a reusable classification between **Knowledge** and **PM Documents**.



Governance policies

This tab sets the rules that frame decisions, validations and governance behavior of the project. Use it before publishing a deliverable or authorizing an external governed action.

Visible sub-surfaces in Governance policies

SUB-SURFACE	WHAT IT REGULATES
Execution connectors	Connector type, status, execution mode, environment, scopes and context parameters
Artifact destinations	Target destination of an artifact, associated connector, active or default status
Action policies	Concerned role, targeted connector, action level (`observe`, `draft`, `propose`, `execute`), effect (`allow`, `require_approval`, `deny`) and authorized scopes
Rendering profiles	Rendering profiles and output format used during governed publications
Notification preferences	Channel, notification type, digest mode, severity threshold and preference activation

Useful settings examples

- require explicit approval before publishing to **SharePoint**;
- allow creation of **Jira** tickets only at `propose` level for certain roles;
- prepare `signal_digest` preferences in `in_app` for internal tracking;
- leave external notifications `email`, `teams` or `webhook` in an approved path only when the connector is healthy;
- choose separate rendering profiles for **DOCX** and **XLSX** publications.

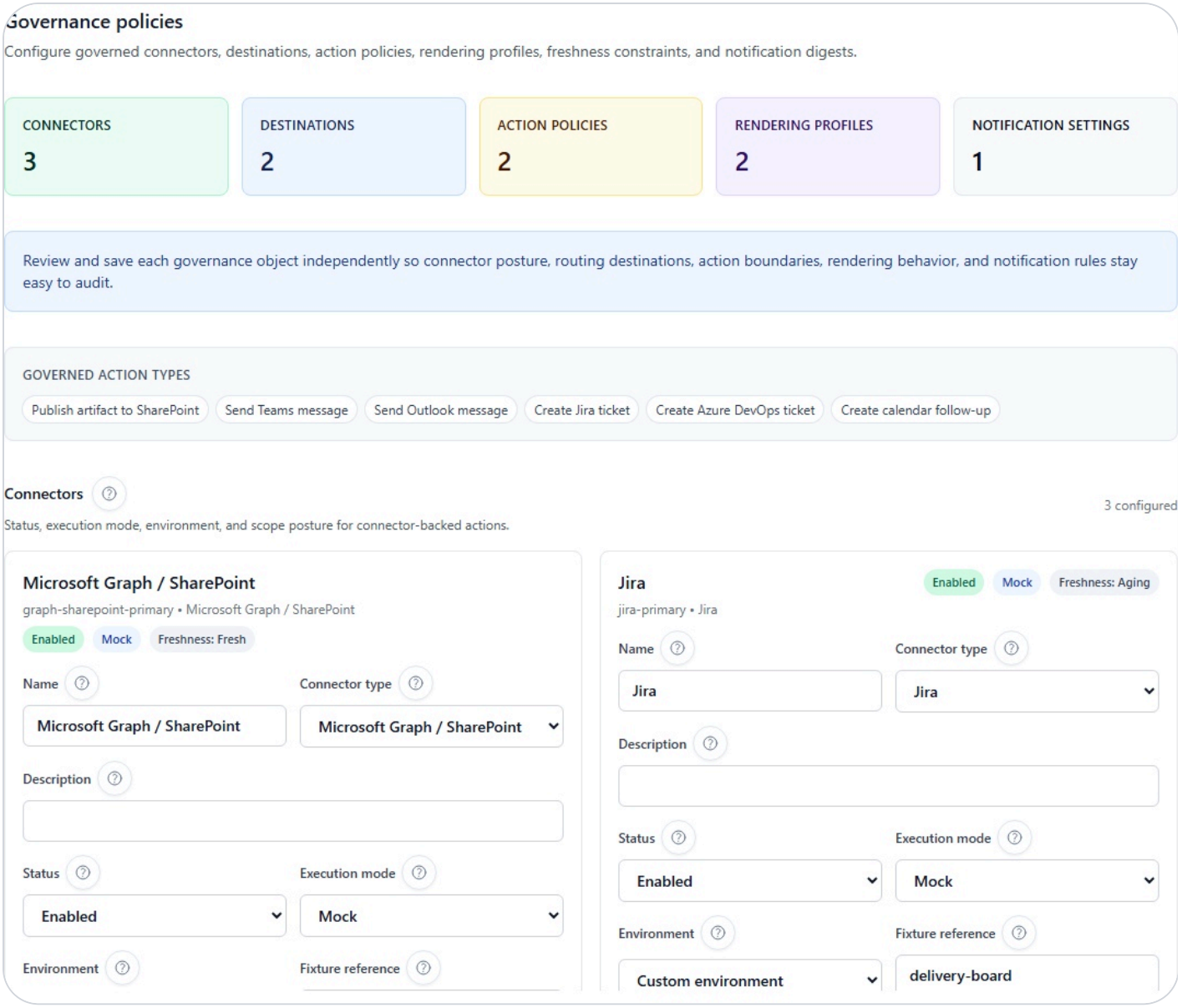
Credible scenario — sensitive project / governed distribution

For a project where all external distribution must be controlled, a coherent setting often looks like this:

1. **Artifact destinations**: active SharePoint destination with known rendering profile;
2. **Action policies**: `allow` for `observe` and `draft`, but `require_approval` for `execute` on publications and external notifications;
3. **Execution connectors**: external connectors visible only to truly authorized roles;

4. **Notification preferences**: `signal_digest` in `daily` for the team, `signal_alert` only for the most sensitive cases;
5. **Project integrations**: bindings enabled only for connectors already validated at platform level.

This combination prevents a draft, digest or action from appearing directly distributable while the project still awaits human approval.



Project integrations

This tab separates **technically defined** platform integrations from those that are **actually usable** by the project.

How to read this tab

Project integrations is not the place where you configure the full tenant-wide technical stack. It is mainly a project readiness surface: what this project can see, what is ready, and what remains blocked with an explicit reason.

You can typically read several information families there:

- **Execution connectors**: governed outbound options toward external systems;
- **Ingestion providers**: import sources later consumed by **Knowledge**;
- **AI runtime transparency**: effective AI provider and deployment-selected provider;
- **License and seat posture**: active plan, purchased licenses, used licenses and available seats visible to the project.

What the Project integrations screen shows

The screen separates technical readiness from project availability:

- platform setup, project binding, policy, permission, health and license availability for app access are **separate causes**. A connector can stay visible in read-only mode so the project team understands why it is blocked instead of assuming it is missing;
- technical setup remains in **Platform Administration**. Project settings managers can bind integrations that are enabled and ready, while sensitive information stays centrally controlled in secured administration.

AREA	WHAT YOU MAY SEE	HOW TO ACT
Execution Connectors	a current list that may be empty, plus an **Available to bind** catalog with Asta Powerproject schedule sync, Azure DevOps delivery project, Jira delivery workspace, Microsoft Project schedule sync, Microsoft Teams collaboration, Outlook executive notifications, SharePoint publication library, Smartsheet portfolio workspace, Webhook event delivery and Wrike delivery workspace	use **Bind to project** only for connectors already enabled and ready at platform level
Ingestion Providers	visible providers such as **Smartsheet sheet import** and **Azure Data Factory evidence pipeline** marked **Healthy**, plus available providers such as SharePoint knowledge import, Azure Blob document ingest, Confluence knowledge import, Jira issue import, SFTP document intake, Microsoft Project schedule import, Wrike task import and Asta Powerproject schedule import	use **Validate binding** to re-check a bound provider, **Disable** to close it for the project, or **Bind to project** for an approved provider

Provider cards can show **Ready**, **Healthy** or **Not configured**. **Not configured** means the provider exists in the platform catalog but still needs source, credential or readiness setup before the project can use it.

For required fields, probes, action mapping and live execution limits, use Connectors and integrations. **Ready** or **Healthy** means preparation is coherent, but does not by itself prove that a full external ticket, message or import has already been executed.

Blocking causes shown by the product

A project integration or import option may be blocked due to:

- policy;
- permission;
- health status to check;
- missing or disabled platform definition;
- disabled or unconfigured project binding;
- incomplete provider-specific configuration or validation.

Interpreting a binding block

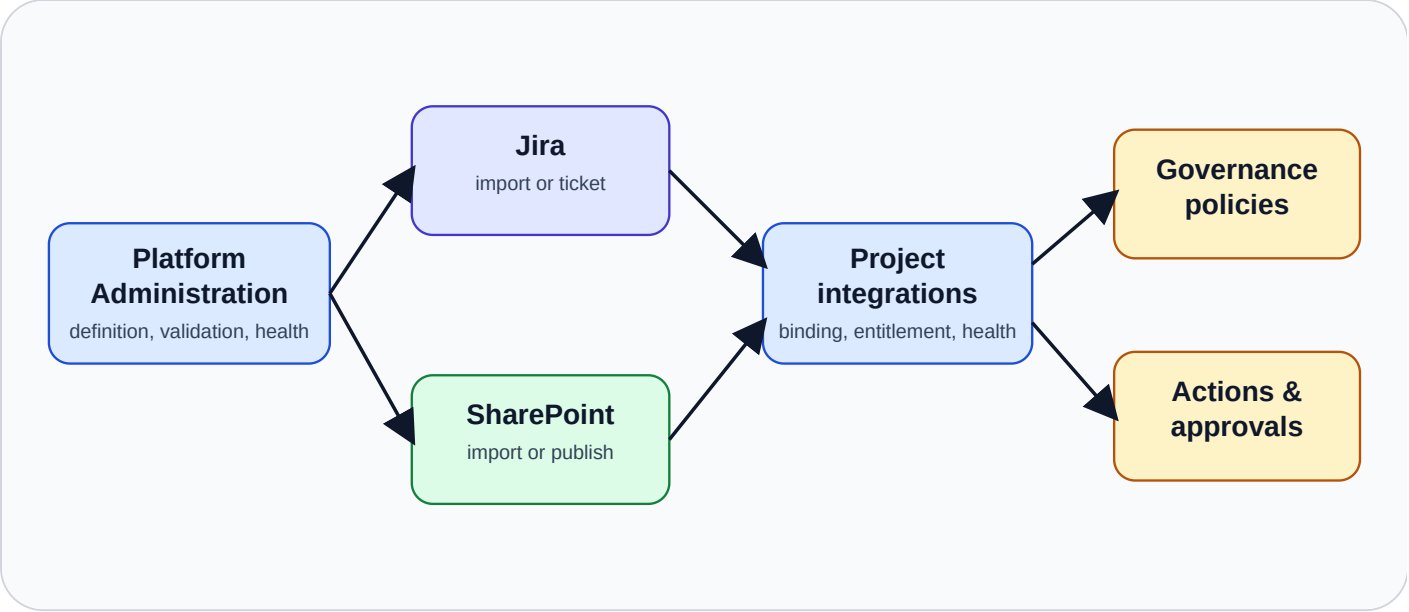
VISIBLE CAUSE	PRACTICAL READING	RECOMMENDED REFLEX
`entitlement`	legacy wording for a blocked integration, not a Marketplace feature-tier difference	check configuration, validation, health, binding, policy, role and license availability for access issues
`policy`	project governance forbids or limits this flow	reread **Governance Policies** before changing the binding
`permission`	the connector exists but your role does not allow activation or use	check the project role in Access control and project roles
`health`	the platform definition exists but its preparation or availability requires verification	open **Platform Administration** to confirm the technical definition
missing or disabled definition	nothing is actually ready at tenant level	first request setup or re-activation from the platform
missing project binding	the platform is ready but the project has not yet consumed the integration	explicitly enable the binding on the project side

Practical reading of `binding` and licensing

- **binding** means the connector or provider exists at platform level, but must still be attached and opened to the project before the project can consume it;
- **licensing** controls whether a user has an available seat to access the app. Marketplace plans do not unlock or block different connector families;
- a visible but blocked connector therefore does not automatically mean it is broken: the interface can keep it visible precisely to explain the blocking reason.

If a block persists, open **Platform Administration** to verify the technical definition, then return to the project to confirm binding and readiness.

Jira, SharePoint and connector chain



Keep this logic simple:

1. **Platform integrations** define the connector or ingestion provider;
2. **Project integrations** expose only the approved and ready binding;
3. **Governance policies** decide what each role can observe, prepare, propose or execute;
4. **Actions & approvals** then apply these rules during the real request;
5. **PM Documents** and **AI Journal** keep the flow trace.

See the dedicated page: Connectors and integrations.

Actions & approvals

This tab turns a recommendation into a controlled operation.

The real states to keep in mind

In the interface, the queue and synthesis cards mainly distinguish four states:

VISIBLE STATE	PRACTICAL READING
Execution prerequisites	compatible connectors may exist, but execution is still blocked by health, configuration, binding, permission, policy, approval or unavailable readiness

VISIBLE STATE	PRACTICAL READING
Pending approval	the request has been proposed and is still awaiting a governance decision
Ready to execute	the request is already **approved**, but execution remains a distinct step
Executed history	the action has actually been executed and remains visible as audit history

A request can therefore be **approved** without yet being **executed**.

How to read a tab that looks empty or incomplete

The tab being visible does not mean an action is already executable. When nothing concrete seems available, the most useful reading is often:

1. no **compatible healthy execution connector** is ready for that action type;
2. the **project binding** does not yet expose the option to the project;
3. a **policy** allows review but not proposal or execution;
4. your **permission** lets you see the queue, but not act on it;
5. approval is required and no decision has yet been taken.

When everything is correctly ready, you should expect at least:

- a compatible action type;
- at least one healthy execution option;
- a valid project binding;
- a coherent policy;
- a user authorized to propose, approve or execute depending on the case.

Read-only or access denied

- **read-only**: the tab remains visible but saving is blocked;
- **access denied**: the route or action is not available for your account.

This difference is especially important for **Access control**, **Project integrations** and governance settings.

Recommended path after project creation

1. open the **Workspace**;
2. first verify the **creator**, **members** and **roles** if the project is collaborative;

3. then adjust **document categories**;
4. review **Governance Policies** before any external distribution;
5. link only the truly ready **Project integrations**;
6. then load **Knowledge**;
7. finally move to **Agents**, **PM Documents** and **Actions & approvals**.

Two useful configuration scenarios

Scenario 1 — minimal new project

For a project that starts, keep a simple order:

1. add essential members and verify their roles;
2. create a short document taxonomy in **Document categories**;
3. enable only already validated and truly necessary integrations;
4. prepare minimal governance, e.g., an internal digest and a default artifact destination;
5. load **Knowledge** before opening agents.

This scenario avoids opening connectors or rules too early that will not be used immediately.

Scenario 2 — sensitive project / governed distribution

For a project exposed to external notifications or formal document publication:

1. limit roles that can access external connectors;
2. prepare a SharePoint or equivalent destination in **Artifact destinations**;
3. apply `require_approval` on action levels that can produce external distribution;
4. prioritize `signal_digest` for current tracking and reserve instant alerts for critical cases;
5. only make visible in **Project integrations** the bindings whose preparation and policy are already compliant.

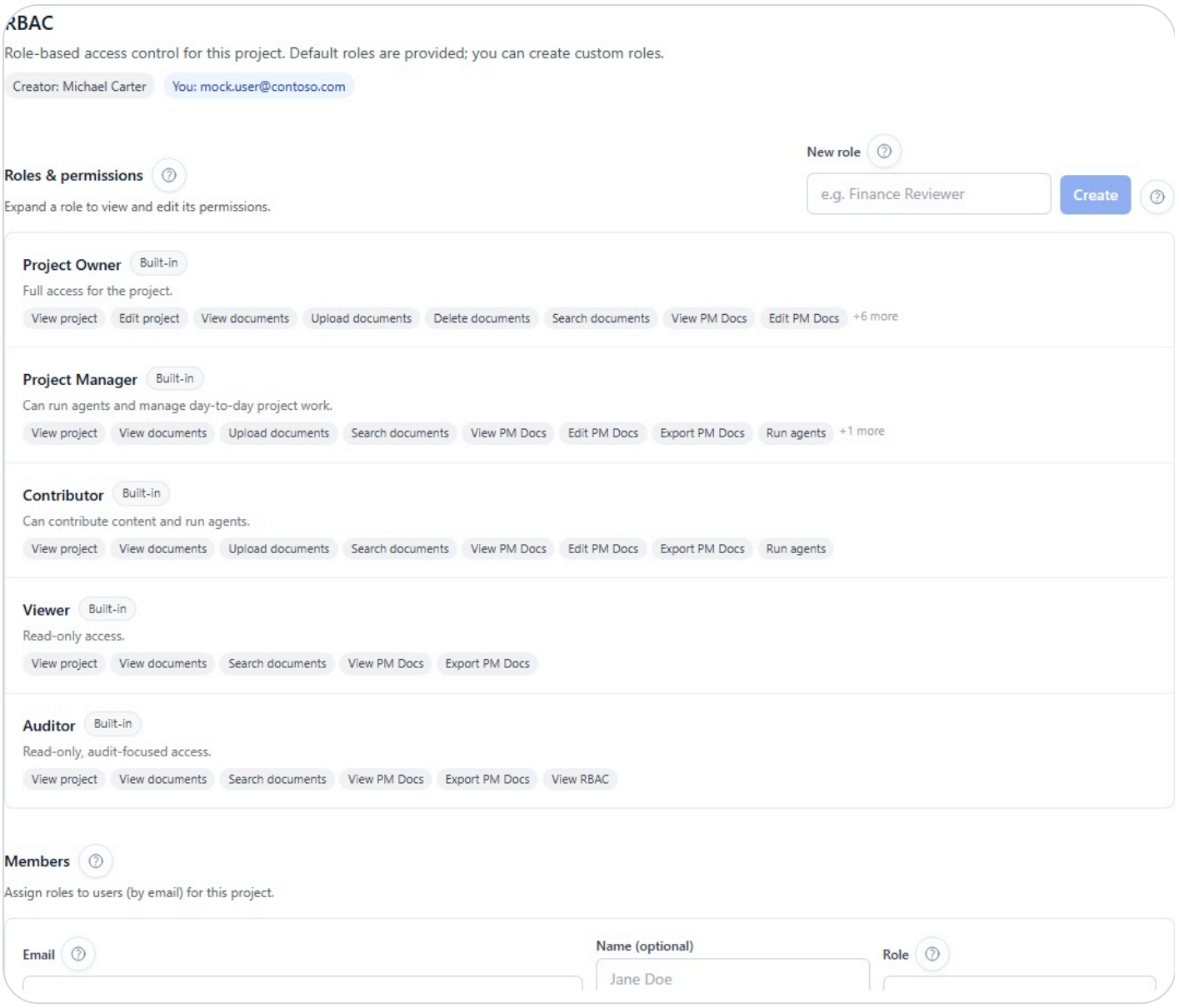
This second scenario aligns signal reading, distribution, approval and real execution instead of letting the team treat each screen as an independent surface.

Next

- Access control and project roles
- Knowledge, documents and imports
- Governance, decisions and actions
- Connectors and integrations
- Azure and platform administration

Access control and project roles

Home · Access control and project roles



Project access control

Assign business roles first, then narrow the scope only when the project context requires it.

1. **Members**
   Add users or groups to the active project before assigning permissions.
2. **Standard roles**
   Start with Owner, Manager, Contributor, Reader or Auditor before creating exceptions.
3. **Custom roles**
   Use them only when standard roles are too broad, too narrow or non-compliant.
4. **RBAC guardrails**
   Prevent self-lockout, unexpected elevation, policy bypass and excessive access.
5. **Audit trace**
   Record who changed access, when it changed and why the scope changed.

If a button is disabled, check role, policy and integration

A blocked action usually comes from the project role, RBAC policy or unauthorized integration binding.

Goal

This page explains how to administer project members and roles in ProPM Agent using business roles first, not raw permissions.

Standard roles

ROLE	TYPICAL USE
Project Owner	Administers project, members, roles and sensitive settings
Project Manager	Drives daily work, deliverables, signals and actions according to policy
Contributor	Adds knowledge, runs agents and contributes to authorized deliverables
Reader	Reads information without modifying the project
Auditor	Reviews evidence, runs, lineage and traces according to audit rights

Add a member

1. Open **Workspace**.
2. Open **Access control**.
3. Select **Members**.
4. Enter the member email address.
5. Choose a standard or custom role.
6. Save.
7. Verify that the member appears with the expected role.

Custom roles

1. Open **Roles and permissions**.
2. Create a role.
3. Use a clear business name.
4. Add a description.
5. Enable only the capabilities required.
6. Save and test with a non-critical user.

Modify or delete a custom role

To modify a custom role, open it, adjust capabilities, save and verify the impact on assigned members. To delete it, first make sure no member still uses it. System roles cannot be deleted.

Guardrails

- You cannot remove your own access from this screen.
- You cannot downgrade your own role from this screen.
- The project creator entry remains protected.
- A system role cannot be deleted.
- A custom role still assigned to members is blocked from deletion.

Technical permissions

Labels such as `project:read`, `agent:run`, `members:manage`, `roles:manage` or `settings:manage` may appear in advanced RBAC diagnostics. For daily use, start with

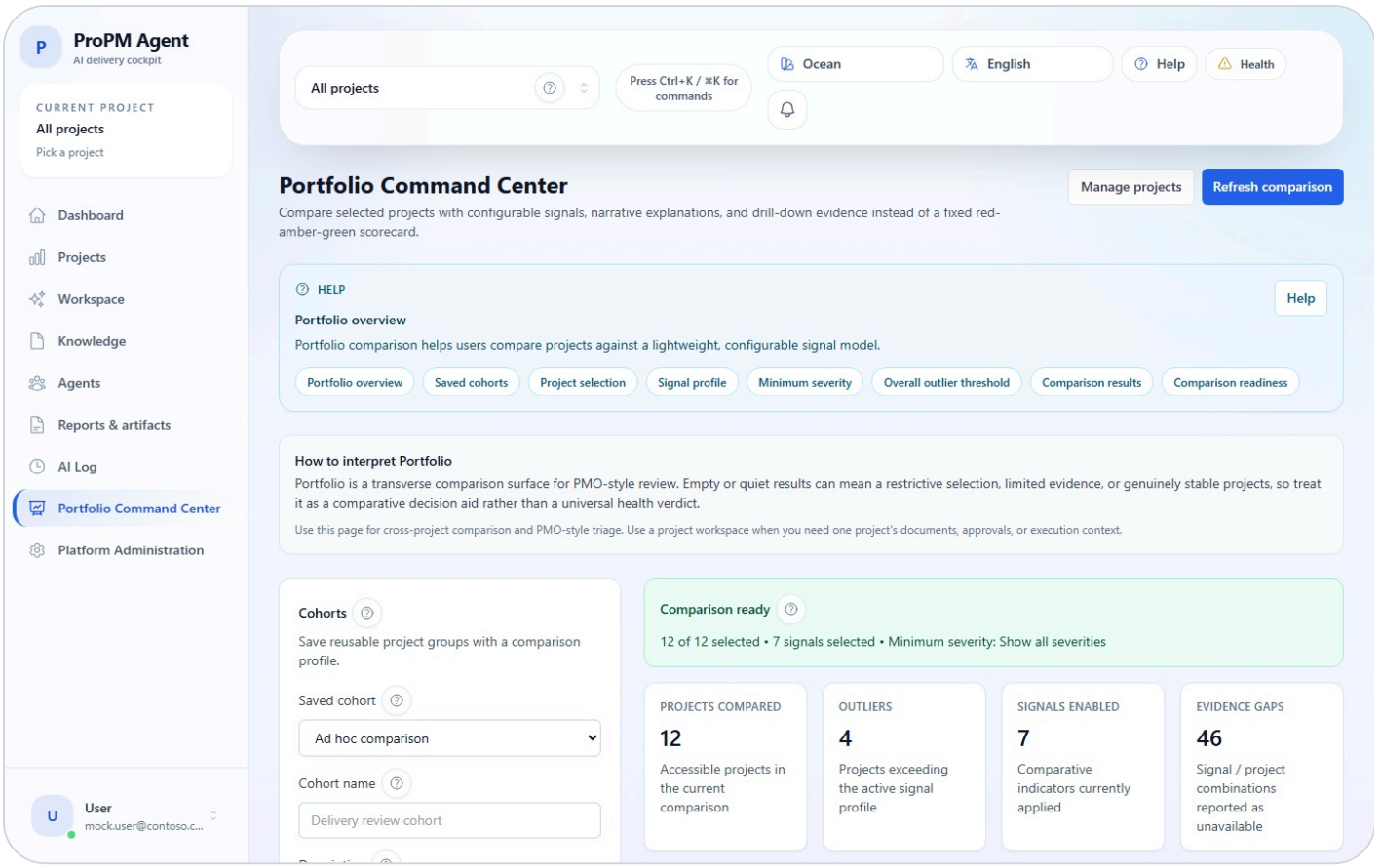
business roles and expected actions.

Related pages

- Active project
- Governance
- Actions and approvals
- Support diagnostics

Portfolio

Home · Portfolio



Portfolio review workflow

Compare projects only after confirming scope, signal profile, threshold and evidence quality.

1. **Project selection**
   Choose portfolio, scope and projects to compare before reading indicators.
2. **Signal profile**
   Select the signals that matter: risk, schedule, evidence, blockers and project health.
3. **Cohorts**
   Group comparable projects to avoid mixing non-equivalent contexts.
4. **Outlier threshold**
   Set ranking sensitivity before interpreting deviations as priorities.
5. **Outliers and evidence**
   Inspect the strongest deviations with their evidence before drawing conclusions.
6. **Follow-up decision**
   Open project details, create a review action or prepare a PM Document based on evidence.

An outlier is a review priority, not an automatic conclusion

The final decision depends on evidence, project context and human validation.

Goal

Portfolio helps PMOs compare several projects with the same signals and identify which projects deserve priority analysis.

Keywords

TERM	MEANING
`Projects`	Projects selected for comparison
`Signal profile`	Signals, weights, thresholds and minimum severity
`Overall outlier threshold`	Global threshold for a project to stand out
`Refresh comparison`	Recalculates the view with current settings

TERM	MEANING
`Reset defaults`	Restores default profile values
Cohort	Saved comparison configuration
Snapshot	Timestamped comparison result

Select projects

1. Open **Portfolio**.
2. Select projects in **Projects**.
3. Use **Select all** if all accessible projects are relevant.
4. Use **Clear all** to start over.
5. Verify the selected projects share the same review objective.

Tune signal profile

1. Open **Signal profile**.
2. Enable only useful signals.
3. Set weights and thresholds.
4. Choose minimum severity.
5. Set **Overall outlier threshold**.
6. Run **Refresh comparison**.

Manage cohorts

ACTION	STEPS
Create	Select projects, tune signals, name the cohort and save
Modify	Load, adjust projects or signals, save again
Delete	Open the cohort and use delete if available
Reload	Select the saved cohort and run **Refresh comparison**

Read an outlier

An outlier is a project that stands out under the active weights, thresholds and signals. It is not an automatic verdict. Open the project detail, read the contributing signals, then continue to Dashboard, Signals, Knowledge or AI Log.

Evidence gaps

Unavailable signals do not mean no risk. They indicate missing or insufficient evidence and should trigger verification.

Azure Marketplace Deployment

Goal

This page details how to deploy **ProPM Agent** from **Azure Marketplace**. It explains the two main Azure form screens, the purpose of every visible field, and the checks to complete before selecting **Review + create**.

The screenshots below show the **Create ProPM-50** form. The displayed name can change depending on the selected Marketplace plan, such as ProPM-50, ProPM-100, or another available plan.

Audience

PROFILE	USE THIS PAGE TO
Azure administrator	Create the Managed Application from Azure Marketplace
Tenant administrator	Prepare Entra groups and first administrator access
Platform administrator	Understand which settings must be completed after installation
Support	Verify the workflow without asking for secrets or internal variables
Business user	No, use Getting started instead

Before You Start

Prepare this information before opening the Marketplace form.

INFORMATION	WHY IT IS REQUIRED
Target Azure subscription	Carries the Marketplace transaction and deployed resources
Target resource group	Organizes the customer-side Managed Application object
Target Azure region	Defines the primary deployment location
Application name	Identifies the ProPM Agent instance in Azure

INFORMATION	WHY IT IS REQUIRED
Managed resource group	Receives the internal resources managed by the application
Entra Group Object IDs	Grant platform administration access
Optional bootstrap users	Help with first access or controlled recovery
Initial AI provider	Indicates the AI provider family to prepare for this instance
Optional CORS origins	Allow additional web domains when needed
VNet CIDR plan	Avoids conflicts with the organization's network
Initial password	Must be generated and stored as a secret

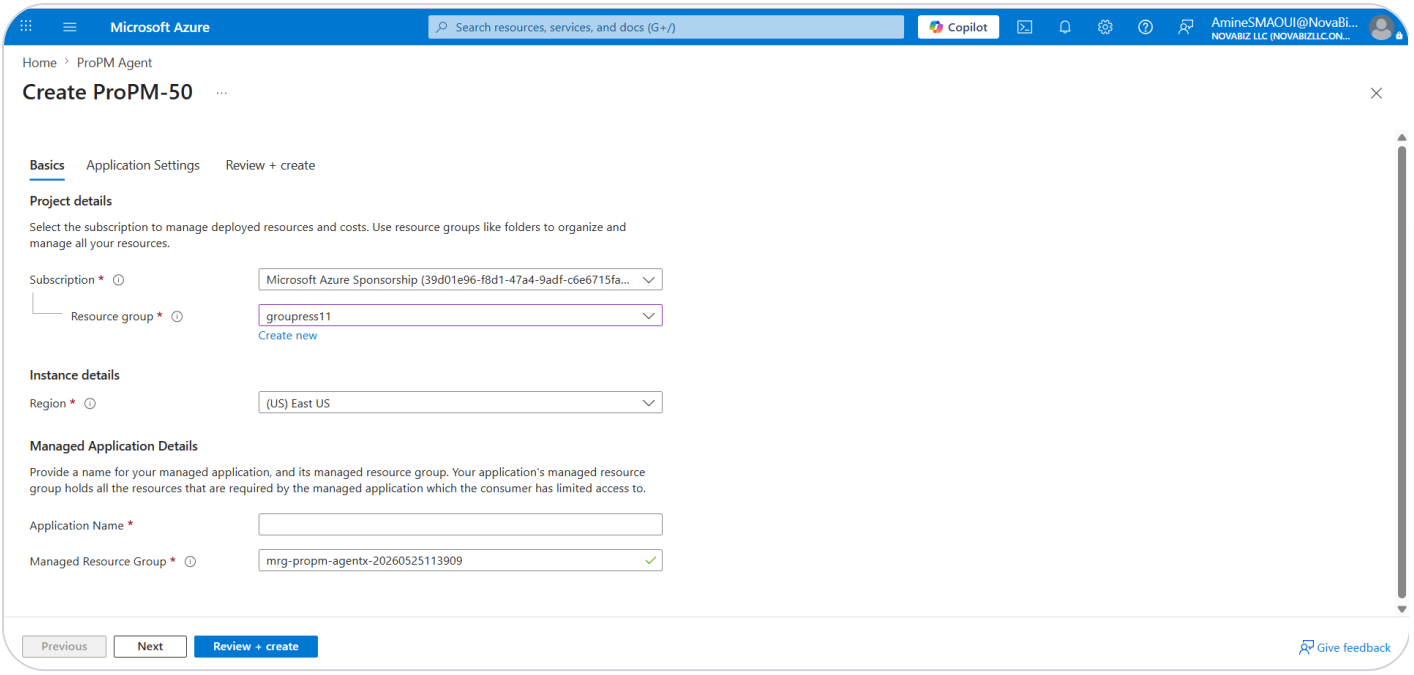
Never copy a password, secret, API key, token, or confidential value into documentation or an unsecured support request.

Deployment Flow

1. Open the ProPM Agent offer in Azure Marketplace.
2. Select the required Marketplace plan.
3. Fill in the **Basics** tab.
4. Select **Next**.
5. Fill in the **Application Settings** tab.
6. Select **Review + create**.
7. Fix any validation errors.
8. Start creation.
9. Wait for Azure provisioning to finish.
0. Open the published URL and complete administration in ProPM Agent.

Screen 1 - Basics

The **Basics** tab defines the Azure scope of the deployment: subscription, resource group, region, and Azure names for the Managed Application.



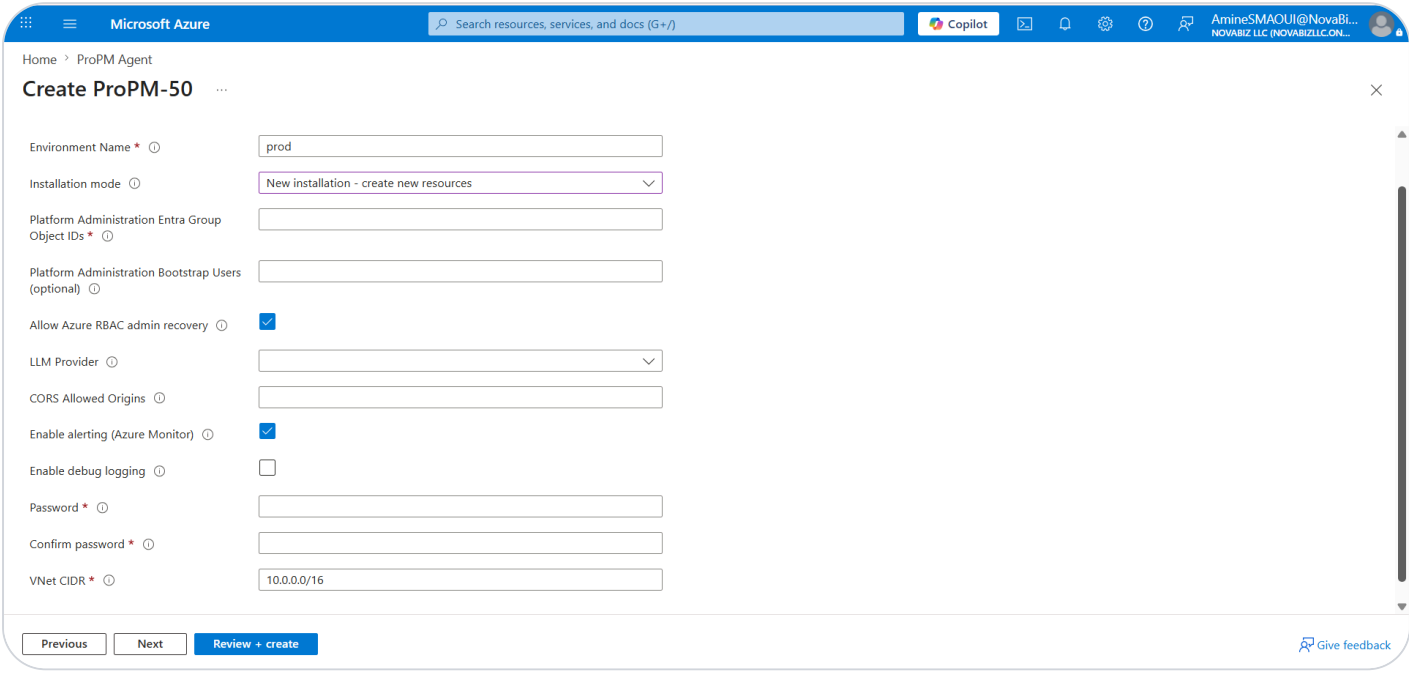
Basics Fields

FIELD	REQUIRED	WHAT TO ENTER	IMPACT AND RECOMMENDATIONS
Subscription	Yes	The Azure subscription that receives the managed application and Marketplace billing.	Use a subscription where you have permission to deploy a Managed Application and create or select resource groups. Also verify internal cost and governance rules.
Resource group	Yes	The customer-side resource group that contains the Managed Application object.	Select an existing group or use **Create new**. This is not the internal managed resource group that contains the technical ProPM Agent resources.
Create new	No	Azure action to create a new resource group when no suitable group exists.	Use a clear name related to the environment and region. Avoid temporary names for production instances.
Region	Yes	The primary Azure deployment region.	Choose a region approved by your organization, close to users, and compatible with data residency requirements. Keep it consistent with planned network and AI resources.
Application Name	Yes	The visible name of the ProPM Agent Managed Application instance.	Use a stable, readable name such as `propm-prod-eus`. This name helps identify the instance in Azure. Do not include secrets or sensitive customer data.

FIELD	REQUIRED	WHAT TO ENTER	IMPACT AND RECOMMENDATIONS
Managed Resource Group	Yes	The managed resource group that receives internal resources deployed by the application.	Azure often proposes a generated name. Keep it unique and recognizable. This group is managed by the application; direct access can be limited by the Managed Application model.
Previous	No	Returns to the previous form step.	This button does not start a deployment.
Next	No	Moves to the next tab when required fields are filled well enough.	Use it to continue to **Application Settings**. If Azure blocks progress, review required fields marked with `*`.
Review + create	No	Runs final Azure validation before creation.	Use it only after completing application settings. Azure then shows errors or warnings that must be fixed before real creation starts.

Screen 2 - Application Settings

The **Application Settings** tab configures the ProPM Agent environment itself: administration identity, installation mode, initial AI provider, CORS, monitoring, logging, initial password, and network.



Application Settings Fields

FIELD	REQUIRED	WHAT TO ENTER	IMPACT AND RECOMMENDATIONS
Environment Name	Yes	A short environment name, such as `dev`, `test`, `uat`, or `prod`.	This helps identify the environment in resources, configuration, and support exchanges. Use a short, stable, non-secret value that follows your internal convention.
Installation mode	Depends on scenario	The installation mode. For a first installation, keep **New installation - create new resources**.	This creates a new instance with new resources. Use an attach-existing-resources mode only for a plan change, major update, or recovery scenario that has been explicitly prepared.
Platform Administration Entra Group Object IDs	Yes	Object IDs of the Entra groups authorized to administer the platform.	Enter the group **Object IDs**, not only display names. These groups determine who can access **Platform Administration** after deployment. Prefer dedicated groups over broad groups.
Platform Administration Bootstrap Users (optional)	No	Bootstrap or recovery users, if your procedure requires them.	Use this to secure first access when Entra groups are not fully operational yet. Keep the list minimal and controlled.
Allow Azure RBAC admin recovery	No	Checkbox allowing administrator recovery through Azure RBAC.	Keep enabled if your operating model allows an authorized Azure administrator to recover platform access. Disable only if your governance requires strict, documented separation.
LLM Provider	Recommended	The initial AI provider family: Azure OpenAI, OpenAI, OpenRouter, or OpenAI-compatible depending on available choices.	This is an initial deployment intent. It does not prove the provider is operational yet. After installation, complete the provider in **Platform Administration > AI Provider Settings** with **Save**, **Validate**, **Test**, then **Activate**.
CORS Allowed Origins	Depends on scenario	Additional allowed web origins, for example `https://portal.contoso.com`.	Leave empty if no additional origin is required. Avoid broad wildcards. Values should be complete origins with `https://`, domain, and port when needed.

FIELD	REQUIRED	WHAT TO ENTER	IMPACT AND RECOMMENDATIONS
Enable alerting (Azure Monitor)	No	Enables or disables Azure Monitor alerts created for the environment.	Recommended for production so incidents can be detected. Afterwards, verify recipients, action rules, and cost rules in Azure Monitor.
Enable debug logging	No	Enables more detailed logs.	Keep disabled in production unless a support diagnosis requires it. Debug logging can increase log volume and expose more technical detail to authorized administrators.
Password	Yes	The initial password requested by the deployment form.	Generate a strong password and store it in an approved vault or secret manager. Do not send it by email, chat, or unsecured ticket.
Confirm password	Yes	The same value as **Password**.	Azure validates that both fields match. If validation fails, re-enter both values from the secure source.
VNet CIDR	Yes	The private network range reserved for the deployment, for example `10.0.0.0/16`.	Choose a range that does not overlap with existing networks, peerings, VPNs, or planned future ranges. Have the network team validate this field before creation because it is difficult to change after deployment.
Previous	No	Returns to the **Basics** tab.	Use it to fix subscription, group, region, or names before final validation.
Next	No	Moves to the next form step.	Use it if you want to follow the tab order before validation.
Review + create	No	Runs final validation across all parameters.	Azure does not create resources until you confirm creation after validation. Fix every error before starting.

Choosing LLM Provider

The **LLM Provider** field selects the AI family the environment should use first. It is not the end of AI configuration.

CHOICE	CHOOSE IT WHEN	STILL REQUIRED AFTER INSTALLATION
Azure OpenAI	The organization wants to stay in the Azure, Entra, and Microsoft governance ecosystem.	Confirm endpoint, region or available model, Azure OpenAI settings, tests, and activation.
OpenAI	The organization directly uses OpenAI APIs.	Enter URL, model, key or secret reference, then run **Save**, **Validate**, **Test**, **Activate**.
OpenRouter	The organization wants access to multiple models through one entry point.	Enter Base URL, key or secret reference, default model, then validate and activate.
OpenAI-compatible	The organization uses an OpenAI-compatible gateway or endpoint.	Enter exact endpoint, authentication mode, expected model or deployment, then test actual compatibility.

Simple rule: deployment **selects** the AI provider; ProPM Agent administration makes it **operational**.

Validation Before Review + Create

Check these points before starting creation.

CHECK	EXPECTED RESULT
Subscription and resource group	They match the target environment and governance rules
Region	It respects data residency, availability, and network strategy
Application Name	The name is clear, stable, and not confidential
Managed Resource Group	The name is unique and recognizable
Entra Group Object IDs	IDs are for the groups expected to administer the platform
Bootstrap Users	The list is empty or strictly limited to expected accounts
RBAC recovery	The choice aligns with the administrator recovery procedure
LLM Provider	The initial provider matches the AI strategy
CORS	Only required origins are allowed
Alerting	Enabled for environments that require monitoring
Debug logging	Disabled except during controlled diagnosis

CHECK	EXPECTED RESULT
Password	Stored in a vault and never shared in clear text
VNet CIDR	Validated by the network team and without known overlap

After Deployment

1. Wait for Azure provisioning to finish.
2. Open the published ProPM Agent URL.
3. Sign in with an account authorized by the Entra groups or bootstrap procedure.
4. Verify access to **Platform Administration**.
5. Open AI provider settings.
6. Run **Save**, **Validate**, **Test**, then **Activate** for the selected provider.
7. Verify licenses and the Marketplace plan.
8. Run a first functional test with a standard user.
9. Check AI Log after a run to confirm the provider actually used.

Common Blockers

SYMPTOM	LIKELY CAUSE	RECOMMENDED ACTION
Azure refuses to move to the next step	Required field is empty or invalid	Review every field marked * and the messages shown under fields
Administrator cannot see Platform Administration	Wrong Entra Object ID or group membership not propagated	Verify the group Object ID and the account membership
Network conflict during validation	VNet CIDR already used or overlapping	Choose another range with the network team
AI provider is not usable after creation	Deployment only selected the AI provider family	Complete the provider in administration with **Save**, **Validate**, **Test**, **Activate**
Too many logs or unexpected costs	Debug logging or alerting enabled without governance	Adjust Azure Monitor settings and disable debug outside diagnosis

Next

- To finalize Entra access, open Azure and Entra administration.

- To configure the AI provider, open AI provider.
- To verify licenses, plans, and updates, open Licenses, plans and updates.
- To invite users, send Getting started.

Azure and Entra administration

Home · Azure and Entra administration

Goal

This page is for Azure and tenant administrators. It covers Marketplace deployment, Entra access finalization and functional checks before inviting users.

Deploy from Azure Marketplace

1. Open the ProPM Agent offer in Azure Marketplace.
2. Select the plan and subscription.
3. Enter resource group, region and requested settings.
4. Declare the Entra groups or administrators.
5. Choose the initial AI provider family.
6. Start deployment.
7. Wait for provisioning to finish.
8. Retrieve the published application URL.

Finalize Entra access

1. Open the ProPM Agent URL with an authorized administrator account.
2. Complete Entra consent if required by the organization.
3. Verify that administrators can open **Platform Administration**.
4. Invite the first Project Owners.
5. Ask end users to follow Getting started.

Post-deployment checks

CHECK	EXPECTED RESULT
App URL	Sign-in and home page open
Entra	Authorized account signs in without a loop
Administration	Platform Administration is visible to admins

CHECK	EXPECTED RESULT
AI provider	Configuration still needs validation and test
Licenses	Plan and license information are visible

Related pages

- AI provider
- Platform integrations
- Licenses, plans and updates

AI provider

Home · AI provider

AI Provider Settings

Follow the administration sequence: an AI configuration is reliable only after validation, testing and controlled activation.

1. **Provider type**
   Select Azure OpenAI, OpenAI-compatible, OpenRouter or another allowed provider.
2. **Configuration**
   Enter endpoint, model or deployment, API version and requested authentication mode.
3. **Save**
   Store the settings before running any technical checks.
4. **Validate**
   Check required fields, formats, provider consistency and prerequisites.
5. **Test**
   Run a functional check before activation to confirm that the provider responds.
6. **Activate**
   Activate only the configuration that has been saved, validated and tested.
7. **Verify in AI Log**
   Confirm the provider actually used by runs and keep the audit trail.

Do not copy secrets, API keys or tokens into tickets

Use secure references and contact support only with non-sensitive information.

Goal

This page explains how to administer the AI provider and distinguish the deployment-selected provider from the effective provider observed on a run.

Actions

ACTION	PURPOSE	COMMON FAILURE
Save	Stores the configuration	Missing required field

ACTION	PURPOSE	COMMON FAILURE
Validate	Checks configuration consistency	Invalid endpoint, model or authentication
Test	Runs a functional check	Provider unreachable or deployment missing
Activate	Makes the configuration active	Validation or test not completed

Configure the provider

1. Open **Platform Administration**.
2. Select **AI Provider Settings**.
3. Choose the provider type.
4. Enter the fields requested by the UI.
5. Select **Save**.
6. Run **Validate**.
7. Run **Test**.
8. Select **Activate** when checks pass.
9. Run a simple agent request and verify the effective provider in AI Log.

Important distinction

SITUATION	MEANING
Marketplace deployment	Initial provider family choice
AI Provider Settings	Current saved and activated configuration
AI Log	Evidence of the provider actually used by a run

Secrets must stay in secured administration tools and must not be copied into user documentation or support tickets.

Platform integrations

Home · Platform integrations

Goal

This page explains the difference between platform integration, project binding, governance policy and real usage by users.

Four levels to check

LEVEL	QUESTION
Platform	Is the connector or ingestion provider defined and validated?
Project	Is it bound to the active project?
Governance	Does policy allow, deny or require approval?
Execution	Are health, live mode and user rights sufficient?

Define and validate an integration

1. Open **Platform Administration**.
2. Select **Platform Integrations**.
3. Choose **Execution Connectors** or **Ingestion Providers**.
4. Create or open the integration definition.
5. Fill in the fields requested by the UI.
6. Save the definition.
7. Run validation.
8. Review validation history.
9. Disable the integration if it should no longer be offered.

Bind to a project

1. Open the project.
2. Open project integrations.
3. Select the validated platform integration.

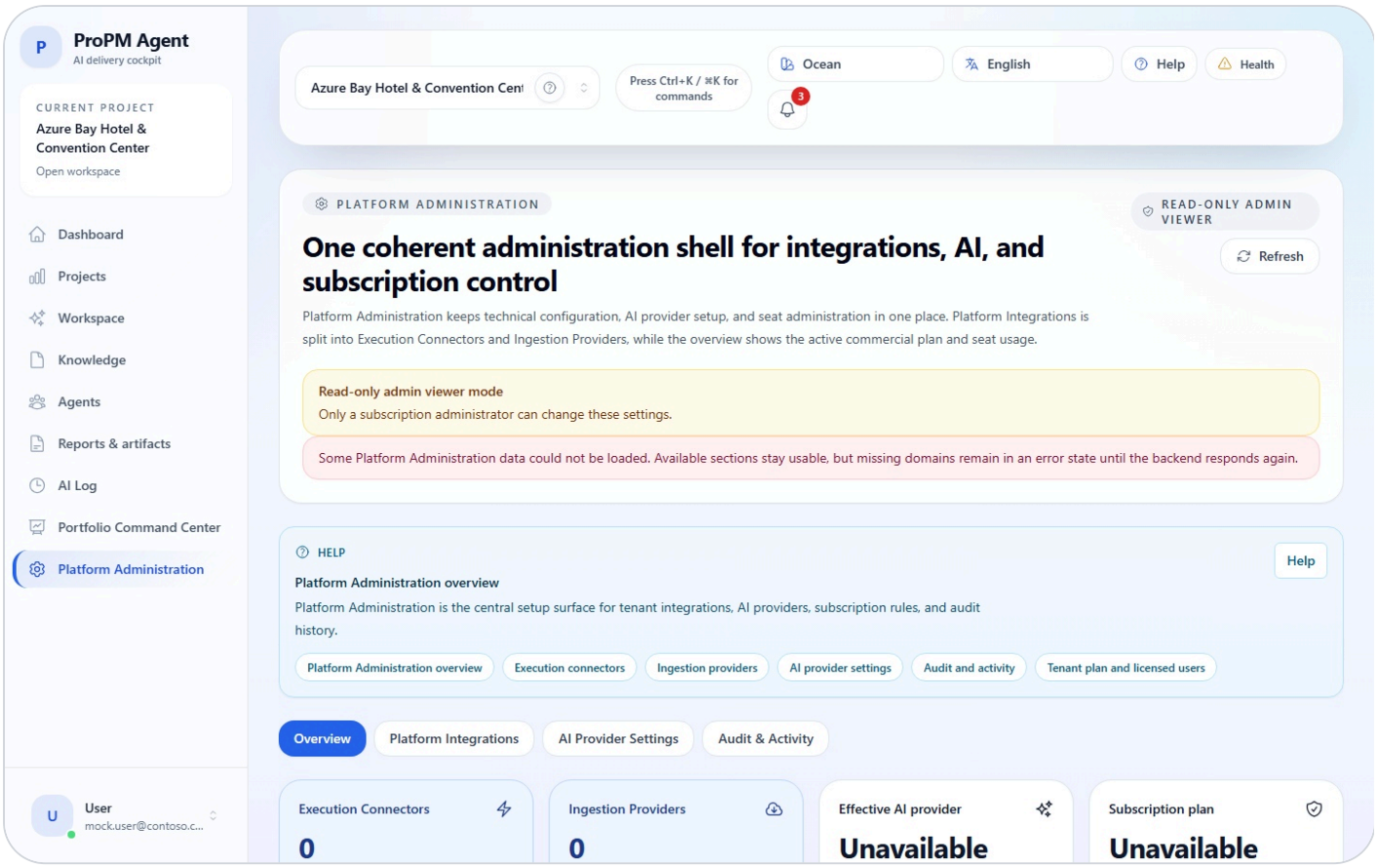
4. Choose **Bind to project** or the equivalent label.
5. Validate the binding.
6. Check readiness, health and available actions.

Readiness

STATE	MEANING
Ready	Prepared for intended use
Healthy	Health check is positive
Not configured	Platform definition is incomplete
Not bound	Project does not use the integration yet
Blocked	Role, policy, health or live mode blocks usage

Licenses, plans and updates

Home · Licenses, plans and updates



Goal

This page explains administrator operations for Marketplace plan, user licenses, plan changes, existing data attachment and application updates.

For a simple in-place application image update, use **Deployment & Updates** when available. For a plan change or major update that must keep existing data, create a new Azure Marketplace deployment and select **Attach existing ProPM data resources**.

Counters

COUNTER	MEANING
Ordered seats	Marketplace ordered capacity
Included seats	Licenses included in the plan

COUNTER	MEANING
Supplemental seats	Additional purchased licenses
Used seats	Users currently counted
Remaining seats	Licenses still available

Manage licensed users

1. Open **Platform Administration**.
2. Stay in **Overview**.
3. Open **Tenant plan and licensed users**.
4. Review licensed or connected users.
5. Identify a user to release if policy allows it.
6. Use **Release user** only when the user no longer needs access.
7. Verify remaining seats.

Refresh Marketplace plan

1. Open **Tenant plan and licensed users**.
2. Select **Refresh Marketplace plan**.
3. Wait for synchronization.
4. Verify counters.

Refresh Marketplace plan synchronizes known information. It does not buy seats, change plan or modify billing.

Change plan or major update with existing data

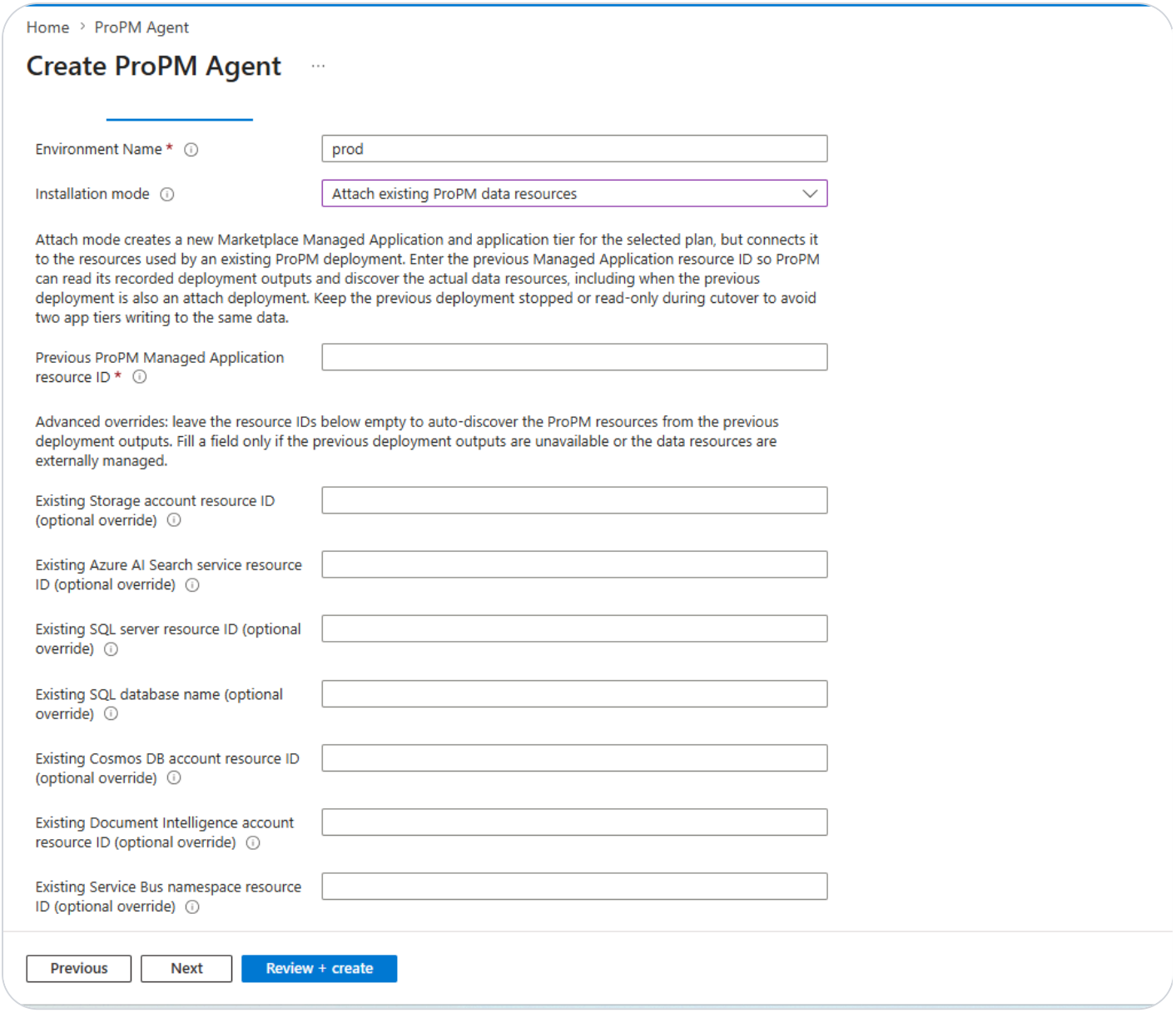
Use this flow when you need a new Marketplace deployment but must keep existing ProPM data, such as a commercial plan change or a major update that cannot be applied in place.
This is not a manual database migration. The new application tier connects to the data resources used by the previous ProPM deployment.

High-level flow

1. Open Azure Marketplace.
2. Select the target ProPM Agent plan.
3. Create a new ProPM Agent deployment.
4. In **Application Settings**, select **Attach existing ProPM data resources**.
5. Enter the previous ProPM Managed Application resource ID.
6. Leave optional override fields empty unless support or your architecture requires them.
7. Configure cutover, administration, CORS, monitoring, password and VNet settings.
8. Select **Review + create**.
9. Validate the new instance before deleting or disabling the previous one.

Update screen 1 - Attach existing data

This part of **Application Settings** identifies the previous deployment and the existing data resources to reuse.



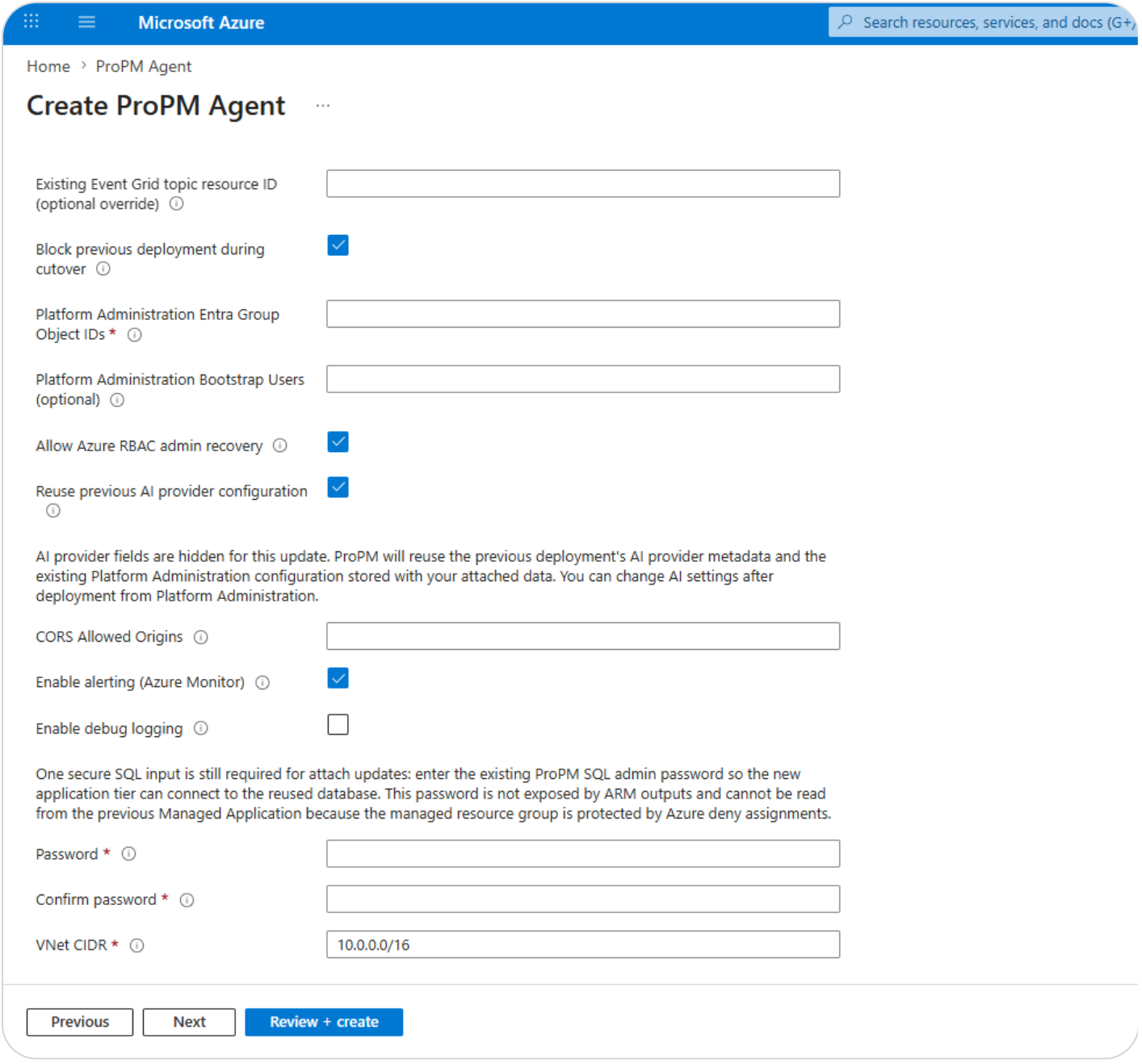
Attach existing data fields

FIELD	REQUIRED	WHAT TO ENTER	GUIDANCE
Environment Name	Yes	Short environment name, for example `prod`, `uat` or `test`.	Use a stable, non-secret value that identifies the new application tier. Keep it aligned with your naming convention.
Installation mode	Yes	**Attach existing ProPM data resources**.	This mode creates a new Marketplace Managed Application and application tier, then connects it to existing ProPM data resources. Use **New installation** only for a new empty environment.

FIELD	REQUIRED	WHAT TO ENTER	GUIDANCE
Previous ProPM Managed Application resource ID	Yes	Full Azure resource ID of the previous ProPM Managed Application.	Copy the full **Id** from the previous Managed Application **Properties** page. Do not enter only the application name, resource group name or managed resource group name.
Existing Storage account resource ID (optional override)	No	Existing storage account resource ID, only when auto-discovery cannot be used.	Leave empty in the standard case. Fill only when previous deployment outputs are unavailable, resources are externally managed, or ProPM support asks for it.
Existing Azure AI Search service resource ID (optional override)	No	Existing Azure AI Search service resource ID.	Leave empty unless the standard discovery from the previous Managed Application cannot identify the search service.
Existing SQL server resource ID (optional override)	No	Existing SQL server resource ID.	Use only for known custom or externally managed SQL topology. This field identifies the SQL server, not the database name.
Existing SQL database name (optional override)	No	Existing SQL database name.	Use with care and only when the database name cannot be discovered from previous deployment outputs.
Existing Cosmos DB account resource ID (optional override)	No	Existing Cosmos DB account resource ID.	Leave empty unless you intentionally need to point the new app tier to a specific existing Cosmos DB account.
Existing Document Intelligence account resource ID (optional override)	No	Existing Document Intelligence account resource ID.	Use only if the previous deployment uses an externally managed or custom Document Intelligence resource.
Existing Service Bus namespace resource ID (optional override)	No	Existing Service Bus namespace resource ID.	Leave empty in the standard case so the deployment can reuse the discovered namespace.
Previous	No	Returns to the previous wizard step.	No deployment is started by this action.
Next	No	Moves to the next wizard step.	Use it after required fields are complete.
Review + create	No	Runs Azure validation before creation.	Use it only after the remaining settings below are complete.

Update screen 2 - Cutover and platform settings

This part controls cutover safety, platform administration, AI provider reuse, CORS, monitoring, SQL password confirmation and network range.



Cutover and platform fields

FIELD	REQUIRED	WHAT TO ENTER	GUIDANCE
Existing Event Grid topic resource ID (optional override)	No	Existing Event Grid topic resource ID.	Leave empty unless auto-discovery cannot find the topic or support instructs you to provide it.
Block previous deployment during cutover	No, recommended for cutover	Check the box when you want to prevent changes in the previous deployment during validation.	This helps avoid two application tiers writing to the same data. If you do not use this checkbox, keep the previous deployment stopped or read-only by another controlled method.
Platform Administration Entra Group Object IDs	Yes	Object IDs of Entra groups that can administer the new platform instance.	Enter group Object IDs, not display names. Confirm the groups are the intended administrators before deployment.
Platform Administration Bootstrap Users (optional)	No	Bootstrap or recovery users, if your access procedure requires them.	Keep this list minimal. Use it only for first access or controlled recovery.
Allow Azure RBAC admin recovery	No	Checkbox allowing administrator recovery through Azure RBAC.	Keep enabled if your operating model allows authorized Azure administrators to recover access. Disable only if governance requires strict separation.
Reuse previous AI provider configuration	Recommended	Check the box when the new deployment should reuse the previous AI provider metadata and platform administration configuration.	When enabled, AI provider fields are hidden for this update. You can change AI settings later from **Platform Administration**.
CORS Allowed Origins	Scenario-dependent	Additional allowed web origins, for example `https://portal.contoso.com`.	Leave empty if no additional origin is required. Avoid broad wildcards.
Enable alerting (Azure Monitor)	No	Enable or disable Azure Monitor alerting.	Recommended for production. Verify recipients and action rules after deployment.

FIELD	REQUIRED	WHAT TO ENTER	GUIDANCE
Enable debug logging	No	Enable more detailed logs.	Keep disabled except during controlled diagnosis because it can increase log volume and expose more technical detail to administrators.
Password	Yes	Existing ProPM SQL admin password.	Attach updates still require this secure SQL input so the new application tier can connect to the reused database. Store and enter it through approved secret-handling procedures.
Confirm password	Yes	Same value as **Password**.	Azure validates that both values match. Re-enter both fields from the secure source if validation fails.
VNet CIDR	Yes	Private network range for the new deployment, for example `10.0.0.0/16`.	Validate with the network team. Avoid overlap with existing networks, peerings, VPNs or planned ranges.

Before Review + create

CHECK	EXPECTED RESULT
Previous Managed Application ID	Full Azure resource ID from the previous ProPM Managed Application
Override fields	Empty unless discovery is impossible or support requested overrides
Previous deployment	Blocked, stopped or read-only during cutover
Admin groups	Correct Entra Group Object IDs for the new instance
AI provider reuse	Enabled when previous configuration should be carried forward
SQL password	Existing ProPM SQL admin password entered through secure handling
VNet CIDR	Validated and non-overlapping
Review + create	Azure validation passes before creation starts

After the new deployment

1. Wait for Azure provisioning to finish.
2. Open the new ProPM Agent URL.
3. Verify administrator sign-in.
4. Verify users, projects, documents, knowledge, reports, agents and integrations.
5. Verify AI provider configuration and run a test if needed.
6. Verify Marketplace plan and license counters.
7. Move users to the new deployment only after validation.
8. Keep the previous deployment temporarily if your cutover plan requires it.
9. Do not delete reused data resources while the new deployment depends on them.

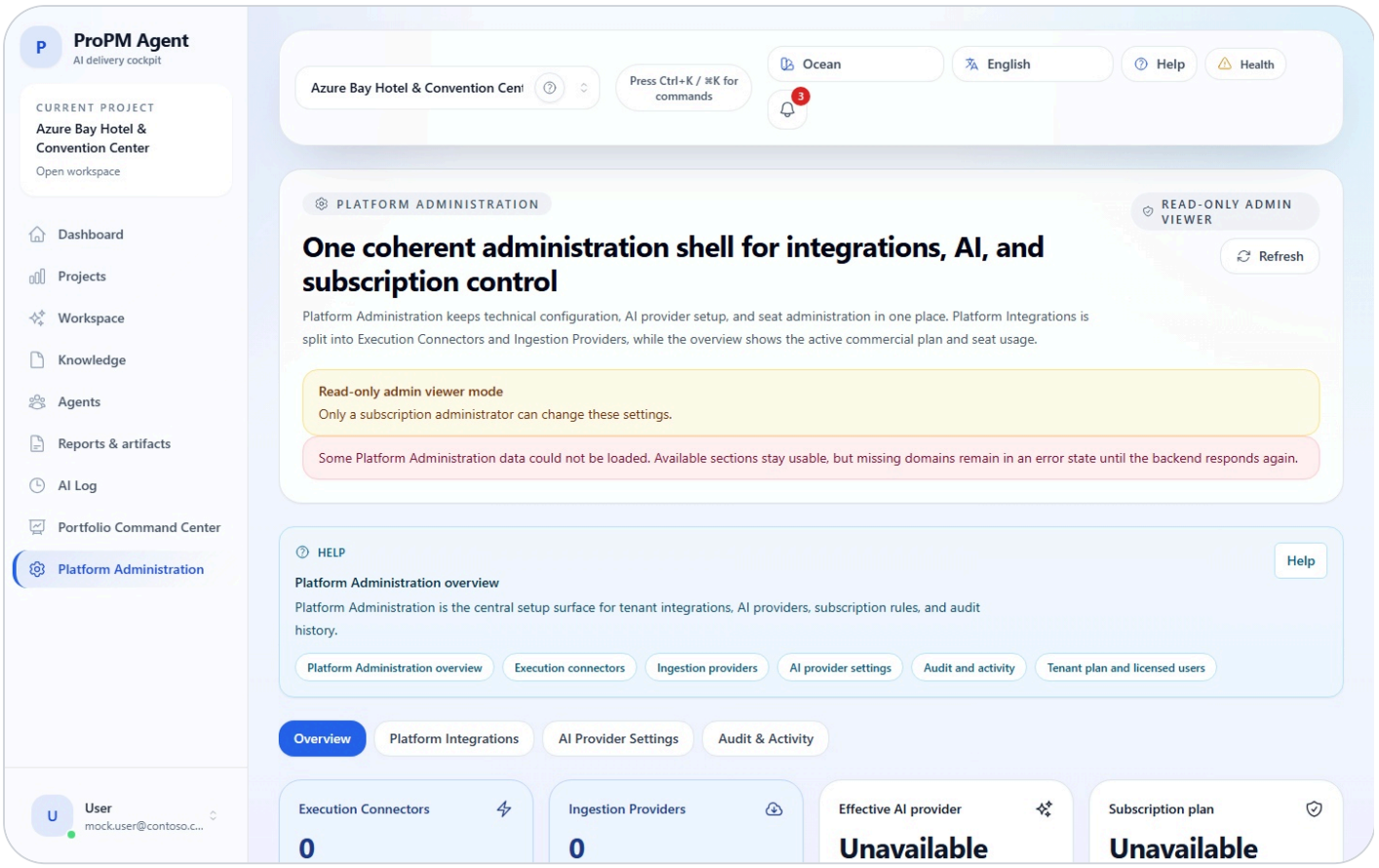
Application updates

Use **Deployment & Updates** to check and apply in-place application image updates when the surface is available. Run updates during a controlled window and keep support information if rollback is needed.

Do not use **Deployment & Updates** for commercial plan changes, new Marketplace purchases, major updates requiring a new Managed Application, or cutovers to existing data resources.

AI providers and platform integrations

Home · AI providers and platform integrations



This page has been split

Platform technical preparation is now documented in two dedicated pages to avoid mixing AI provider setup, connectors and project usage.

Where to go now

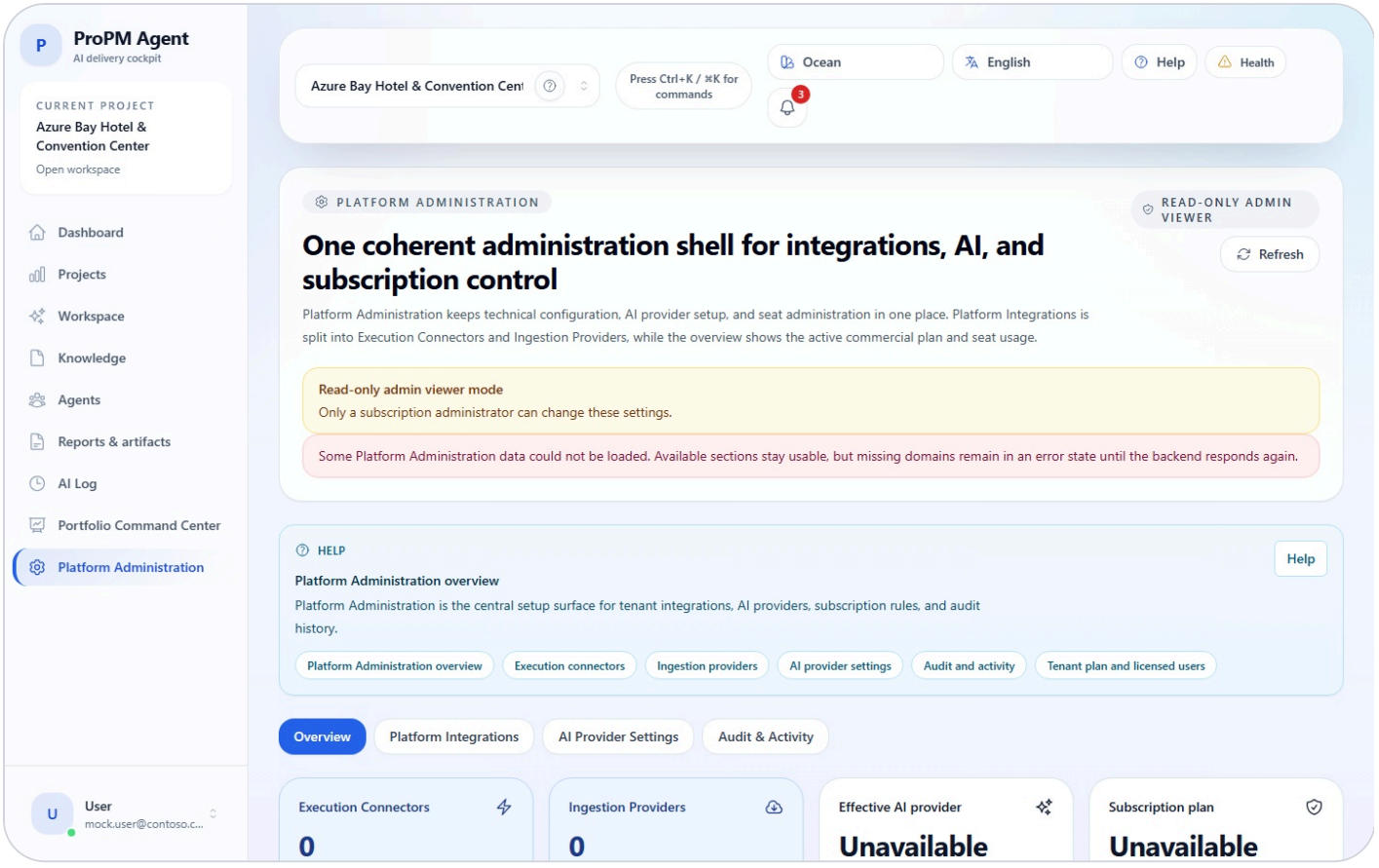
NEED	RECOMMENDED PAGE
Configure AI provider Save / Validate / Test / Activate	AI provider
Verify the provider actually used	AI Log
Define and validate connectors or ingestion providers	Platform integrations
Understand business connector usage	Connectors and integrations
Diagnose a blocker	Support, audit and diagnostics

Rule to remember

The platform decides what is technically available. The project decides what is bound. Governance decides who can act. AI Log proves the provider actually used by a run.

Tenant plan, licensed users, and app updates

Home · Tenant plan, licensed users, and app updates



This page has been split

Tenant plan, licensed users and application updates are now centralized in Licenses, plans and updates.

Where to go now

NEED	RECOMMENDED PAGE
Understand ordered, included, supplemental, used and remaining seats	Licenses, plans and updates
Release a licensed user	Licenses, plans and updates
Refresh the Marketplace plan	Licenses, plans and updates
Change plan or attach existing data	Licenses, plans and updates

NEED	RECOMMENDED PAGE
Check application updates	Licenses, plans and updates

Important rules

- Azure Marketplace remains the official source for plan, price and billing.
- **Refresh Marketplace plan** synchronizes known information but buys nothing.
- Infrastructure details belong to advanced administration or support, not the end-user guide.

Security and compliance

Home · Security and compliance

Goal

This page explains security and compliance principles for users, Project Owners, tenant administrators and support.

Identity and access

- Sign-in relies on Microsoft Entra ID in Azure environments.
- Project access depends on membership and assigned role.
- A signed-in user may see no project if no project access has been granted.
- Project roles control reading, agents, PM Documents, members, governance, actions and audit.

Tenant and project isolation

The tenant defines the organization perimeter. The active project limits visible documents, runs, signals, actions and PM Documents. **All projects** does not bypass permissions.

Data and privacy

Do not upload documents that are not authorized for the project. Mask sensitive data in support screenshots. Never copy secrets, keys, tokens or passwords into notes, prompts, actions or tickets.

Traceability

ITEM	WHERE	USE
Run	AI Log	Find an AI execution
Effective AI provider	AI Log	Confirm actual provider used
PM Document	PM Documents	Review version, diff and lineage

ITEM	WHERE	USE
Governed action	Actions and approvals	Check proposal, approval and execution
Trace ID	Support diagnostics	Correlate an incident

Responsible AI

Review AI outputs before decisions or publication. Check evidence, freshness, contradictions and missing information. Confidence is not automatic approval.

Known limitations

Home · Known limitations

Goal

This page clarifies known limitations to avoid incorrect expectations.

User limitations

TOPIC	LIMITATION	GOOD PRACTICE
Chat history	May be browser-local	Use PM Documents, artifacts and AI Log for shared traceability
Active project	Project pages depend on selected project	Check the top bar before acting
All projects	Does not grant access to every tenant project	Verify actual project access
Agent answer	Not automatically a final deliverable	Review and use PM Documents when needed
AI confidence	Not an approval	Check evidence and freshness
`unavailable` source	Not exploitable evidence	Reopen source or reimport

Integration limitations

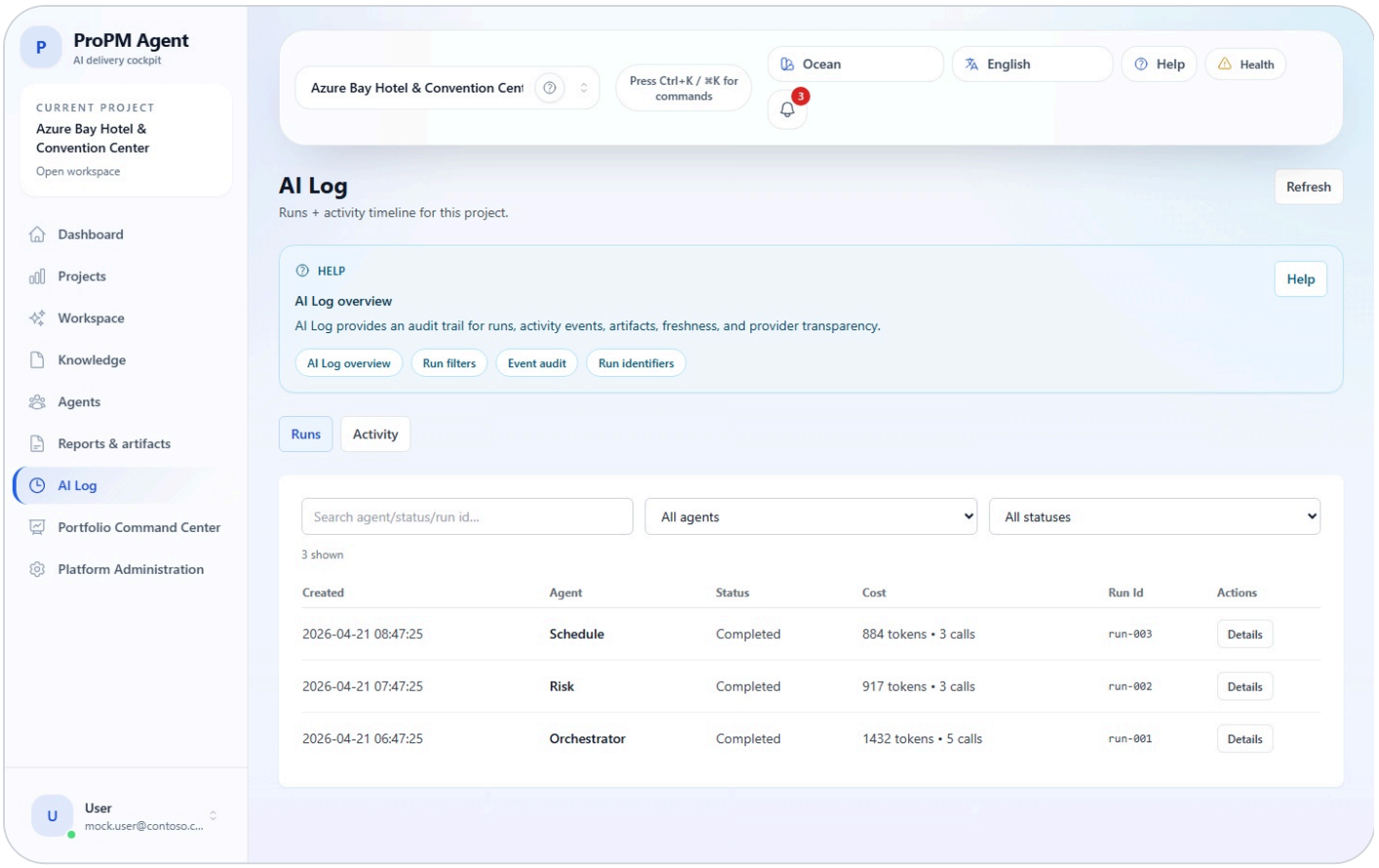
Visible connectors may not be usable until platform, project binding, policy, role and health are ready. Approved actions may still not be executed. Live mode is required for real external object creation when applicable.

Azure administration limitations

The provider selected during deployment does not prove which provider a run used. **Refresh Marketplace plan** synchronizes information but does not buy seats or change billing.

Maintenance, support and FAQ

Home · Maintenance, support and FAQ



Goal

This FAQ helps recognize common symptoms and open the right page without exposing internal or sensitive details.

NovaBiz Contact

For NovaBiz support:

- Email: support@navabiz.pro
- Address: NovaBiz, 131 Continental Dr, Suite 305, Newark, DE 19713, United States

Symptom to first action

SYMPTOM	FIRST ACTION	USEFUL PAGE
Sign-in fails	Check account, tenant, consent and URL	Getting started

SYMPTOM	FIRST ACTION	USEFUL PAGE
No project visible	Check project membership and role	Active project
Empty page	Check active project, filters and role	Interface
Document not indexed	Check `Indexed`, `Ingesting` or `Failed`	Knowledge
Agent has no answer	Check whether a run exists in AI Log	AI Log
Incomplete answer	Check evidence, freshness and missing information	Structured outputs
AI provider not operational	Check effective provider and admin settings	AI provider
Connector blocked	Check platform, project, policy and health	Connectors
Action pending	Check approval, policy and execution	Actions and approvals
License unavailable	Check remaining seats and licensed users	Licenses, plans and updates

Frequent answers

A successful Microsoft sign-in with no project usually means the account is not yet a project member. A visible page with disabled buttons usually means read-only access. Agent answers without evidence must be treated as exploratory before publication or external decision.

PM Documents and actions

Download creates a local copy. **Publish** sends to a governed destination. **Add to knowledge** adds a reviewed document back to project knowledge. **Approved** means authorized; **executed** means the action actually ran.

Advanced support and audit

For support, provide URL, active project, time, attempted action, visible message, Run ID, Trace ID or Action ID when visible. Never share secrets, keys, tokens or sensitive payloads. Open Support, audit and diagnostics for details.

Support, audit and diagnostics

Home · Support, audit and diagnostics

Goal

This page centralizes the information to collect for access, project, knowledge, agents, AI provider, connectors, actions, licenses or update issues.

NovaBiz Contact

For NovaBiz support:

- Email: support@navabiz.pro
- Address: NovaBiz, 131 Continental Dr, Suite 305, Newark, DE 19713, United States

What to provide

INFORMATION	USEFUL EXAMPLE
Page URL	Page visible when the issue occurred
Active project	Project name
User	Account or group, never password
Time	Date, time and timezone
Action attempted	Upload, run agent, publish, approve
Visible message	Exact text or masked screenshot
Visible IDs	Run ID, Trace ID, Artifact ID, Action ID when present

Never provide

- Passwords, secrets, API keys or tokens.
- Complete payloads with sensitive customer data.
- Unmasked screenshots containing confidential documents.
- Detailed infrastructure configuration unless support explicitly asks.

Quick diagnosis

SYMPTOM	FIRST CHECK	USEFUL PAGE
Sign-in fails	Tenant, Entra account, consent, license	Getting started
No project visible	Active project, membership, role	Active project
Document not indexed	Knowledge status, format, import history	Knowledge
Agent has no answer	Run in AI Log, effective provider	AI Log
Action pending	Approval, policy, connector, live mode	Actions and approvals
License denied	Remaining seats, licensed user	Licenses and updates

Support IDs

ID	WHERE TO FIND IT	USE
Run ID	AI Log	Identify an agent execution
Trace ID	Error or support detail	Correlate a technical error
Artifact ID	PM Document or artifact	Identify a deliverable
Action ID	Actions & approvals	Identify a governed action
Context snapshot ID	Run detail when shown	Understand the execution context

Glossary

Home · Glossary

Label convention

This glossary uses the business term first and keeps the exact UI label when it avoids ambiguity. Terms marked **support/admin** are not required for daily end-user work.

Key terms

TERM	DEFINITION
Active project	Current context for Dashboard, Knowledge, Agents, PM Documents, Signals, Actions and AI Log
Add to knowledge	Adds a reviewed PM Document back into project knowledge
Agent	Specialized assistant working in active project context
All projects	Scope limited to projects the same account can already access
Artifact	Governed object from a run or review
Cohort	Saved Portfolio comparison configuration
Confidence	Signal of output solidity, read with evidence and freshness
Context snapshot	**Support/admin** context captured for a run
Diff	Comparison between artifact or PM Document versions
Document PM / PM Doc	Reviewed, versioned and governed project deliverable
Download	Creates a local copy
Effective AI Provider	Provider actually used by a run, visible in AI Log
Entitlement	**Support/admin** operational blocker, not a Marketplace feature tier
Evidence gap	Missing or insufficient evidence

TERM	DEFINITION
Freshness	Evidence state: `fresh`, `aging`, `stale`, `conflicting`, `unavailable`
Lineage	Traceability chain from run to PM Document and evidence
Portfolio	Multi-project comparison surface
Project binding	Link that makes a platform integration usable in a project
Publish	Sends a deliverable to a governed destination
Refresh Marketplace plan	Admin sync action; it does not buy licenses or change plan
Run	Recorded AI or agent execution
Seat	User license unit
Structured output	Formatted run result with summary, evidence and actions
Trace ID	**Support/admin** correlation identifier
Validate	Admin consistency check before test or activation

Knowledge and Agents

This page has been split

The former **Knowledge and Agents** page is now split into shorter task-oriented pages for document ingestion, agent usage, evidence reading and deliverable production.

Where to go now

NEED	RECOMMENDED PAGE
Upload, import or search documents	Knowledge, documents and imports
Choose an agent, run it or create a custom agent	Agents
Read citations, freshness and confidence	Structured outputs, evidence and freshness
Turn a result into a governed deliverable	PM Documents and artifacts
Audit an execution or verify effective provider	AI Log

Important reminder

An indexed document can feed project searches and agents. An agent answer still requires review and only becomes a PM Document through the review and governance flow.

Legacy portfolio and administration page

This page is kept for compatibility

The former **Portfolio and technical administration** page has been replaced by specialized pages to separate end-user guidance, project administration, Azure administration and support diagnostics.

Where to go now

NEED	RECOMMENDED PAGE
Compare projects, cohorts, signal profiles and outliers	Portfolio
Deploy from Azure Marketplace and finalize Entra	Azure and Entra administration
Configure Save / Validate / Test / Activate for the AI provider	AI provider
Define, validate or bind connectors	Platform integrations
Manage seats, Marketplace plan, refresh and updates	Licenses, plans and updates
Diagnose with Run ID, Trace ID or AI Log	Support, audit and diagnostics

Marketplace plan rule

Marketplace plans alone do not explain whether an action is available or blocked. When something is blocked, check remaining licenses, role, project access, binding, policy, health and effective AI provider.

Related pages

- Audiences, roles and scopes
- AI Log
- Maintenance, support and FAQ

Reports, AI Log, and Traceability

Home · Reports, AI Log, and Traceability

This page has been split

The former **Reports, AI Log and traceability** content is now separated to clarify documentation review workflows and audit diagnostics.

Where to go now

NEED	RECOMMENDED PAGE
Find, review, download or publish a PM Document	PM Documents and artifacts
Understand diff, versions and lineage	PM Documents and artifacts
Find a run, activity or effective AI provider	AI Log
Prepare a support request with Run ID or Trace ID	Support, audit and diagnostics

Important reminder

Traceability is read in two complementary places: **PM Documents and artifacts** for deliverable lifecycle, and **AI Log** for execution details.