BUILDWITHHQ · AI AGENTS

Build a governed AI team

Native management walkthrough · 13 November 2026

Give each agent a narrow job, explicitly selected context, and a human approval boundary. The current native product supports agent identity, instructions, security settings, exact-version Skills, and deactivation.

> **Current qualification:** Native create, edit, and deactivate were exercised through the real rendered development product, with desktop and mobile page checks. Microsoft Agent Framework planning is read-only. Deterministic execution workers remain disabled; this walkthrough does not activate them.

1. Open the persisted AI Agents page

Sign in to the intended tenant as its authorized human account owner. Open the app's published AI Agents page or its secured native review surface. Page identity comes from persisted app/page associations. A page or menu entry never grants record authority.

Existing saved pages retain their identities through a narrow renderer compatibility mapping. New pages use `module.ai-agents`. The public HTTP migration is a hard cut and has no old-route aliases.

2. Create one specialized operator

1. Choose **Add agent**. Enter a name, identity and role instructions, and operating instructions.
2. Choose the execution model and a bounded daily quota. Keep human approval required for material actions.
3. Explicitly select the permitted DataRole context. The server rechecks the actor and selected permissions; a role is never inferred from a page or a sole membership.
4. Select authorized locations or explicitly choose **All permitted locations**. An empty choice does not mean All.
5. Save once and wait for the acknowledged result. Inspect the created agent and its saved context.

Record profiles are optional. If the profile module is not configured, native management remains available; the application does not invent a profile module, owner, record scope, or DataRole. When a profile is configured, its new record requires an explicit Personal or shared record scope, with the normal secured record envelope.

3. Update and retire deliberately

Select an agent, choose **Edit agent**, and review its instructions, quota, approval requirement, DataRole and location settings. Saving uses the native identity and security contracts. Leaving the identity prompt blank during an edit preserves it.

Deactivate agent requires confirmation. Deactivation stops new work acceptance while retaining identity, profile and audit evidence under the existing contract. It is not deletion of execution history.

If a save is interrupted, the editor retains the draft and blocks resubmission until a successful team review. It does not automatically deactivate an acknowledged new agent or replay an uncertain profile write.

4. Publish exact-version Skills

Create a focused instruction set with a stable Skill key, semantic version, description and instructions. Choose the available personal or organization scope explicitly. Assign an exact version to an eligible agent; retain the returned version and digest as evidence.

Skills guide preparation. They do not grant tools, record access, location access, or approval authority. Revoking a Skill retains immutable history and stops its future resolution under the governing contract.

5. Understand the planning boundary

Secured context is assembled only after tenant, DataRole, location and `CanAiReadRecords` checks. Microsoft Agent Framework runs the routed model through `ChatClientAgent` and an `IChatClient` adapter. No executable tools are registered with the model.

The planner returns a structured proposal, which is validated before canonical suggestion batches and suggestions are written. SQL approval and execution contracts remain authoritative. The Inbox `ActivityVersion` fence is preserved through preparation, approval and execution.

Deterministic dispatch, quotas, action-specific permission and audit remain separate from model planning. Internal SQL names are retained until the separately reviewed Tier 2 migration; a public route rename does not authorize a database rewrite.

Deployment and solution status

The native workflow is development-verified. The larger Build This solution blueprint and optional record families are separate from installing or publishing a Template Library template. Source examples are not evidence that every tenant has those modules provisioned.