Recast

Connect Copilot to Application Workspace Reporting

Last Modified on 2026-11-27

Copilot Integration Overview

With this integration, you can query Application Workspace reporting data using natural language from Microsoft 365 Copilot and Microsoft Teams.

To set up the integration, you'll need to complete the following tasks:

- Create Application Workspace API Credentials
- Create the Copilot Studio Agent
- Create the Power Automate Agent Flow
- Configure the OData HTTP calls
- Publish the Agent Flow as an Agent Tool

For step-by-step instructions, see Set Up the Copilot Integration for Application Workspace Reporting.

In this scenario, the Copilot Studio agent does not treat the OData feed as static knowledge. Instead, it invokes an agent flow at runtime. The flow authenticates to Application Workspace, retrieves reporting datasets, and returns the results to the agent.

Runtime Sequence

The integration uses the following runtime sequence:

1. The user asks the Application Workspace agent a question in Microsoft 365 Copilot.

2. Copilot Studio decides that the Application Workspace reporting tool is needed and calls the published Power Automate agent flow.

3. The flow sends an authenticated GET request to `https://<zone>/api/reports/v4/` to enumerate available datasets.

4. The flow parses the value array and loops through each dataset.

5. For each dataset, the flow sends another GET request, appends the response to an AllData string, and labels the block with the dataset name.

6. The flow returns AllData to the agent as text. The agent uses that output to answer the user.

> The Application Workspace client-secret credential belongs to the flow, not to the person chatting with the agent. Unless you add another authorization layer, every authorized agent user can potentially receive data available to that shared credential.

Copilot Integration Prerequisites

- **Application Workspace**: An active zone with API access available and network reachability from Power Automate.

- **API credential:** A client secret and the generated client-secret username from **Manage** > **System** > **API Access**.
- **Power Platform:** A Microsoft Dataverse environment (not Dataverse for Teams) and permission to create/publish an agent flow.
- **Copilot Studio:** Permission to create, configure, test, publish, and share agents.
- **End users:** Appropriate Microsoft Copilot access and permission to use the shared agent.
- **Governance:** Approval for the HTTP endpoint, connector combination, data loss prevention policies, and intended audience.

Security Considerations Before Starting

- Decide on the minimum set of datasets the agent is allowed to return.
- Choose whether the agent is for administrators only or for a broader end-user audience.
- Store the secret only in the secure authentication fields of the HTTP action or in an approved secret-management pattern. Never paste it into agent instructions, prompts, descriptions, variables, or responses.
- Set an expiration date that matches your credential-rotation policy. The Application Workspace secret value is displayed only once.
- Confirm that agent transcripts, flow run history, and returned payloads meet your organization's data handling requirements.

Set Up Copilot Integration for Application Workspace Reporting

To set up the integration, you'll need to complete the following tasks described below:

- Create Application Workspace API Credentials
- Create the Copilot Studio Agent
- Create the Power Automate Agent Flow
- Configure the OData HTTP calls
- Publish the Agent Flow as an Agent Tool

Create Application Workspace API Credentials

To create required Application Workspace API credentials:

1. In Application Workspace, go to **Manage** > **System** > **API Access**.

2. Select **Create**.

3. Enter a description that identifies the agent flow and set an expiration date.

4. Finish the wizard. Copy the generated secret immediately and store it in an approved secure location; it cannot be displayed again.

5. Open the newly created client secret and copy the client-secret username. The username is static and unique per zone.

CREDENTIAL MAPPING
Power Automate Basic authentication username = Application Workspace client-secret username
Password = the secret value shown when the client secret is created

Recast

Create the Copilot Studio Agent

To create the Copilot Studio agent:

1. Open Copilot Studio in the intended Power Platform environment.

2. Create a custom agent. Give it a clear name, such as Application Workspace assistant.

3. Add a concise description that explains the data the agent can retrieve and the users it is intended for.

4. Add instructions that tell the agent when to use the Application Workspace reporting tool, to avoid guessing, and to say when the returned dataset does not support the request.

5. Save the agent and test a simple conversational response before adding the flow.

> **Suggested instructions for the Copilot Studio Agent**
>
> - Use the Application Workspace reporting tool for questions about applications, deployments, launches, installations, and related health data.
> - Base answers only on data returned by the tool.
> - If a requested field is unavailable or ambiguous, say so and ask for the minimum clarification needed.
> - Do not reveal credentials or raw secrets.

Create the Power Automate Agent Flow

From the agent's Tools page, select **Add tool** > **Agent flow**. Copilot Studio creates a starter flow with the required 'When an agent calls the flow' trigger and 'Respond to the agent' action.

1. Keep the agent trigger. The current sandbox does not require an input parameter; a production design can accept a dataset name, user identifier, application name, date range, or another constrained parameter.

2. Initialize a string variable named AllData with an empty value.

3. Add an HTTP action named HTTP call to locate all available datasets.

4. Add Parse JSON for the first HTTP response.

5. Add Apply to each and iterate over body('Parse JSON')?['value'].

6. Inside the loop, call the selected dataset and append its name and response body to AllData.

7. Return AllData as a text output named alldata in Respond to the agent.

8. In the Respond to the agent action settings, open Networking and keep Asynchronous response turned off so the flow returns in real time.

9. Publish the flow.

Configure the OData HTTP calls

Discovery request

```
Method: GET
URI: https://<zone-fqdn>/api/reports/v4/
Authentication type: Basic
Username: <client-secret username>
Password: <client secret>
```

The response is an OData document with a value array. The exported sandbox definition expects each entry to include name, kind, and url.

```
Minimal Parse JSON shape
{
"type": "object",
"properties": {
"value": {
"type": "array",
"items": {
"type": "object",
"properties": {
"name": { "type": "string" },
"kind": { "type": "string" },
"url": { "type": "string" }
},
"required": ["name", "kind", "url"]
}
}
}
}
```

Dataset request inside the loop

```
Method: GET
URI expression:
concat('https://<zone-fqdn>/api/reports/v4/', items('Apply_to_each')?['name'])
Authentication: same Basic credential
```

Append each result to AllData with a recognizable delimiter:

```
concat(
'=== ', items('Apply_to_each')?['name'], ' ===', '\n',
string(body('Individual_Dataset_HTTP_Call')), '\n\n'
)
```

PRODUCTION RECOMMENDATIONS

- Do not retrieve every dataset for every question unless the payload is small and the run consistently completes within the agent-flow action limit.
- Prefer an allowlist and route the request to only the needed dataset.
- Where the selected dataset supports them, use OData query options such as $select, $filter, and $top to reduce data volume, and validate each option against your Application Workspace version.

Publish the Agent Flow as an Agent Tool

To publish the Agent flow:

1. Open the agent and go to Tools.

2. Select Add a tool and Flow.

3. Choose the published Application Workspace flow and select Add and configure.

4. Give the tool a name and description. The description should tell the orchestrator when the tool applies and what its output contains.

5. Configure completion behavior, save the tool, and confirm it appears in the agent's tool list.

Publish and share the agent

To publish the agent:

1. Test the agent in the Copilot Studio test pane.

2. Select Publish and confirm. Republish after every change that users need to receive.

3. Open Channels and select the Microsoft 365 and Microsoft Teams channel.

4. Turn on Make agent available in Microsoft Copilot if the agent should be available from Microsoft 365 Copilot, and then add the channel.

5. Install and test the published version yourself before expanding access.

Choose an availability option

Audience	Method	Notes
Pilot user	Install for yourself / see the agent in Teams.	Best first validation of the published channel experience.
Small approved group	Share the agent and use the installation link or Show to my teammates and shared users.	Users still need permission to the agent; possession of a link alone is not sufficient.
Organisation	Show to everyone in my org and submit to the organisation catalog.	Requires administrator approval. Use only after security and data-scope review.

In Microsoft 365 Copilot, users can invoke the installed agent by typing @ and selecting it, or by selecting it from the available agents. A newly published version appears in new sessions. Use start over or begin a new session when validating an update.

BEFORE BROAD DISTRIBUTION
Verify the shared Application Workspace credential exposes only data appropriate for every person who can access the agent. Microsoft 365 does not automatically translate the user's Application Workspace permissions to the sharedOData connection.

Example Prompts and Response Patterns

The exact answer depends on the datasets available in your zone. The examples below show safe response formats rather than real environmentvalues.

Prompt	Recommended response pattern
Which applications are deployed to user@example.com?	Confirm the user matched; list application name and deploymentstate; state which dataset supplied the answer; flag duplicates orambiguous users.
How many installs of 7-Zip do we have?	State the application match, count records by relevant status, andidentify any date or device scope used. If "install" is ambiguous,distinguish successful, pending, and failed records.
Show failed deployments in the last 24 hours.	Filter by the supported timestamp and failure status, list the mostuseful identifying fields, and state the time zone. If the dataset hasno suitable timestamp, say so.
What is the install health of Visual Studio Code?	Summarize success/failure totals and rates only if the returnedfields support the calculation; show the denominator and daterange.
Which applications were launched by this user this week?	Use launch/event data only if present. List application and mostrecent launch; avoid inferring a launch from availability or deploymentalone.

Prompt design guidance

- Include the smallest useful identifier: user principal name,device name, or exact application name.
- Add a date range for event, health, or count questions.
- Ask for a specific output shape, such as a total, a short table,or failed items only.
- Treat natural-language wording as a request to analyze returnedrecords, not permission to infer missing facts.

Troubleshoot Copilot Integration for Application Workspace Reporting

Symptom	Likely cause	Resolution
HTTP 401 Unauthorized	Username/secret is incorrect, expired, or mapped to the wrong Basic auth field.	Use the client-secret username as Username and the one-time secret as Password. Check expiration and create a replacement secret if needed.
HTTP 403 Forbidden	The credential or network path is not allowed to access the endpoint.	Verify API access, zone permissions, URL, network policy, and Power Platform data policies.
HTTP 404 Not Found	Incorrect zone FQDN, path, or dataset name.	Test the base discovery URL first. Use the exact name returned by /api/reports/v4/.
Parse JSON fails	The schema does not match the current discovery payload.	Inspect the successful HTTP body and regenerate or relax the schema while keeping name, kind, and url.
Agent cannot find the flow	The flow lacks the required trigger/response, is unpublished, or is in another environment/solution.	Use When an agent calls the flow and Respond to the agent, publish it, and verify the current environment.
Agent reports a timeout	The all-dataset loop or payload is too large to return within the action limit.	Retrieve only relevant datasets; filter/select fields; reduce rows; parallelize only after validating endpoint behavior.

Recast

Symptom	Likely cause	Resolution
Agent does not call the tool	The tool name/description or agent instructions are too vague.	Clarify when the tool applies; add representative prompts; start a new test session and retest.
Users can install but cannot chat	The agent is not shared with the user or group.	Share the agent with the user/security group, then retest with that identity.
Published answer seems stale	An old conversation is using earlier published content or a cached answer.	Start over or create a new session; confirm the latest agent and flow versions are published.