

Microsoft Copilot Studio documentation

Discover how to build AI-driven agents easily with Microsoft Copilot Studio with online training courses, docs, and videos. Learn how to quickly and simply integrate chat into your website.



GET STARTED
[What is Microsoft Copilot Studio?](#)



WHAT'S NEW
[What's new in Microsoft Copilot Studio? ↗](#)



TRAINING
[Microsoft Copilot Studio training](#)



CONCEPT
[Troubleshoot](#)

Create

Try/Buy

[Sign up for free](#)
[Buy license](#)

Create agents with generative AI

Use generative AI to build agents fast
Tell your agent what it should do
Build an agent with generative AI from the ground up

Publish and deploy

[Publish to live or demo sites](#)
[Connect and configure for Microsoft Teams](#)
[Publish to Facebook](#)

Enhance and improve your agents

[Use entities and slot filling](#)
[Use end-user authentication](#)
[Use Microsoft Bot Framework Skills](#)

Extend agents with tools

Build and manage agent flows

[Extend the capabilities of your agent](#)
[Call a Power Platform connector](#)
[Add tools to custom agents](#)

[Build an agent flow with natural language](#)
[Add an agent flow to an agent](#)
[Edit and manage your agent flow in the designer](#)

Best practices guidance

[Topics best practices](#)
[Measure and improve agent engagement](#)
[Plan a conversational AI project](#)

Administer and analyze

Administer Microsoft Copilot Studio
[Assign licenses and manage access](#)
[Work with environments](#)
[Configure authentication](#)

Analyze, monitor, and diagnose
[Use analytics to improve your agent](#)
[Analyze conversational agent performance](#)
[Analyze autonomous agent health](#)

Related content

Power Platform
One-stop learning resource for Power Platform applications and features including admin, connectors, developer, and guidance documentation.

Power Apps
Quickly build low-code apps that modernize processes and solve tough business challenges in your organization using Power Apps.

Power Automate
Create automated workflows between your favorite apps and services to synchronize files, get notifications, collect data, and more.

Power BI
Turn your unrelated sources of data into coherent, visually immersive, and interactive insights.

Power Pages
Design, host, and administer secure, modern, and low-code business websites.

Microsoft 365 Agents SDK
Build agents deployable to channels of your choice, with scaffolding to handle the required communication.

[Power Platform & Copilot Studio Architecture Center](#)

Browse reference architectures and solution ideas for Power Platform and Copilot Studio.

[Dynamics 365](#)

Discover the next generation of CRM and ERP applications.

[Microsoft 365 Copilot extensibility](#)

Customize Microsoft 365 Copilot with agents and use actions and connectors to extend Copilot's skills and knowledge.

[Microsoft Azure](#)

Explore the ever-expanding set of cloud computing services to help your organization meet its business challenges.

Build your skills

[Mastering Copilot Studio](#)

Your go-to guide for building and customizing agents in Microsoft Copilot Studio.

[Copilot learning hub](#)

Discover tools, guidance, and support for adopting, securing, and managing custom and Microsoft Copilot experiences in your organization.

[Agent in a Day events](#)

Learn to build low-code AI agents with Copilot Studio in a free, hands-on, instructor-led workshop.

[Copilot Studio Adoption](#)

Check out adoption resources for Copilot Studio.

[AI learning hub](#)

Build practical skills to unlock the potential of AI and prepare for AI innovation using the Microsoft Cloud.

[Copilot Studio Blog](#)

Keep updated with the Copilot Studio Blog, your source for AI-powered agent news.

[Copilot Studio scenarios](#)

Explore agent scenarios to jumpstart your projects with downloadable resources.

Copilot Studio overview

07/15/2025

Copilot Studio is a graphical, low-code tool for building agents and agent flows.

The screenshot shows the Copilot Studio interface. At the top, there's a navigation bar with icons for Home, Create, Agents, Flows, Library, and three dots. The main title is "Copilot Studio" with a gear icon. To the right are buttons for "Environment", a gear, and a question mark. The central area has a heading "Describe your agent to create it" with three categories: "Helpdesk", "Expense tracking", and "HR and benefits". Below this is a text input field with placeholder text "Use everyday words to describe what your agent should do" and a "Next" button. A note at the bottom says "Features labeled as 'preview' are subject to supplemental terms. [See terms](#)".

Explore agents

- Website Q&A** Agent template: Instantly answer user questions using the content of your website or other knowledge.
- Financial Insights** Agent template: Help financial services professionals get quick and concise info from their org's financial documents and other available resources.
- Safe Travels** Agent template: Provides answers to common travel questions and related health and safety guidelines.
- Benefits** Agent template: Benefits Agent provides personalized information on various benefits offered by the employer that are tailored to employee's unique circumstances.
- Citizen Services** Agent template: Enable Public Sector organizations to build an agent with their publicly available websites to assist citizens navigate services and information.
- IT Helpdesk** Agent template: Empowers employees to resolve issues and effortlessly create/view support tickets.

Learning resources

- Free Copilot Studio Workshop**
- Quick start: Create and deploy an agent**
- Documentation**
- Security and Governance**
- Responsible AI FAQs**
- Quick start: Use Generative AI in an agent**
- Support community**
- Coming soon: See release plans**
- What's new: See release notes**

One of the standout features of Copilot Studio is its ability to connect to other data sources using either prebuilt or custom plugins. This flexibility enables users to create and orchestrate

sophisticated logic, ensuring that their agent experiences are both powerful and intuitive.

The platform's low-code experience puts the power of AI at the user's fingertips, making it accessible even to people without extensive technical backgrounds.

What is an agent?

An agent is a powerful AI companion that can handle a range of interactions and tasks, from resolving issues requiring complex conversations to autonomously determining the best action to take based on its instructions and context. Agents coordinate a collection of language models, along with instructions, context, knowledge sources, topics, actions, inputs, and trigger to accomplish your desired goals.

Agents can engage with customers and employees in multiple languages across websites, mobile apps, Facebook, Microsoft Teams, or any channel supported by the Azure Bot Service. They can also improve productivity by performing tasks to assist users and organizations.

You can easily create agents in Copilot Studio without the need for data scientists or developers. Some of the ways you might use agents include:

- Sales help and support issues.
- Opening hours and store information.
- Employee health and vacation benefits.
- Public health tracking information.
- Common employee questions for businesses.

Agents can be used on their own or to extend Microsoft 365 Copilot with enterprise data and scenarios.

What is an agent flow?

Agent flows offer a powerful way to automate repetitive tasks and integrate your apps and services. Agent flows can be triggered manually, by other automated events or agents, or based on a schedule.

With Copilot Studio, you can create agent flows using either natural language or a visual editor.

Agent flows can be run as standalone automations. You can also configure agent flows to trigger from an agent and return results to the same agent, so that the agent can use the agent flow as a tool.

How does an agent conversation work?

Copilot Studio agents use customized NLU model and AI capabilities to understand what a user types or says, then respond with the best topic. A topic is a portion of a conversational thread between a user and the agent. For more information, see [Create and edit topics](#).

For example, you might create an agent for your customers to ask common questions about your business. Your agent reduces support overhead by deflecting support calls. In the agent, you can create a topic about your store's opening hours and name the topic **Store hours**.

When a customer asks a question such as "When do you open?" or "What are your opening hours?", the agent uses natural language understanding (NLU) to understand the *intent* behind the question. The agent matches that intent to the best topic, the **Store hours** topic.

The agent follows the *conversation flow*—which is a group of connected nodes—that you define in the **Store hours** topic. Some nodes can ask questions, others use conditions (if/else) to determine which store the customer wants. The final output of the topic shows the hours and contact information for that specific store.

However you can't anticipate all the types of questions your customers ask. To help mitigate this, Copilot Studio incorporates powerful AI-powered capabilities that use the latest advancements in NLU models. With your agent linked to knowledge sources, it can automatically generate responses. These responses are conversational, plain language, and don't depend on the agent maker to create topics for every eventuality.

As well, when AI general knowledge is enabled, your agent can access information not in its knowledge sources.

Copilot Studio also uses AI powered by the Azure OpenAI GPT model, also used in Bing, to create topics from a simple description of your needs. Similarly, you can modify and update any topic in your agent by describing the changes you want to make.

Access Copilot Studio

Copilot Studio is available as both a standalone web app, and as a discrete app within Teams. Most of the functionality between the two is the same. However, there might be different reasons to choose one version or the other based on the ways you want to use Copilot Studio.

[] Expand table

Copilot Studio version	Use cases	More information
Web app at https://copilotstudio.microsoft.com	<ul style="list-style-type: none">- An IT admin who wants to create agents to perform tasks or interact with customers.- You're familiar with agent services	Explore the Copilot Studio demo

Copilot Studio version	Use cases	More information
	and want to trial or test Copilot Studio. - an agent user who wants to explore advanced agent concepts, such as entities and variables, and create complex agents.	
Teams app	- An employee or member of an organization who wants to use agents to answer common employee questions. - You want to use advanced concepts, such as entities and variables, and have an internally available agent in Teams. - You want to create and distribute an agent in the shortest time possible.	Open or add the Copilot Studio app in Teams

Plan your agent

Consider the following points when planning your agent.

Extend Microsoft 365 Copilot with an agent

Consider extending Microsoft 365 Copilot with an agent if:

- You want to craft your own agent by declaring instructions, actions, and knowledge to customize Microsoft 365 Copilot for specific tasks and domain knowledge.
- You wish to utilize the existing Copilot orchestrator.
- You want a standalone custom version of the Microsoft 365 Copilot chat experience.

Create an agent

Copilot Studio makes creating agents easy. You only have to describe the agent you want in plain language to create it. Tell Copilot Studio what specific instructions, topic triggers, knowledge sources, and actions you want for your agent. Then test your agent before you deploy. Publish your agent when you're ready across multiple channels.

Consider creating an agent if:

- You want an agent that can:
 - Integrate company data and documents
 - Retrieve real-time data from external APIs

- Take actions in response to external events
- Be embedded in company applications
- You require a customized end-to-end solution for your web or mobile app or automation workflow that meets specific business needs and allows for complete control over product branding.
- You want to surface your agent to other agents as their supported agent extension.
- You're a proficient developer looking to create a customized end-to-end solution to cater to your business needs, and want:
 - Full control on product branding
 - Choice of language models and orchestration

Or, if you're building products like:

- A customer service chatbot for your e-commerce site
- A virtual assistant to schedule appointments for your healthcare service
- Gaming experiences that incorporate generative AI

Accessibility

The agent authoring canvas is built for accessibility in accordance with [Microsoft accessibility guidelines](#) and supports standard navigational patterns.

Important information

Important

Microsoft Copilot Studio (1) is not intended or made available as a medical device for the diagnosis of disease or other conditions, or in the cure, mitigation, treatment or prevention of disease, or otherwise to be used as a component of any clinical offering or product, and no license or right is granted to use Microsoft Copilot Studio for such purposes, (2) is not designed or intended to be a substitute for professional medical advice, diagnosis, treatment, or judgment and should not be used as a substitute for, or to replace, professional medical advice, diagnosis, treatment, or judgment, and (3) should not be used for emergencies and does not support emergency calls. Any agent you create using Microsoft Copilot Studio is your own product or service, separate and apart from Microsoft Copilot Studio. You are solely responsible for the design, development, and implementation of your agent (including incorporation of it into any product or service intended for medical or clinical use) and for explicitly providing end users with appropriate warnings and disclaimers pertaining to use of your agent. You are solely responsible for

any personal injury or death that may occur as a result of your agent or your use of Microsoft Copilot Studio in connection with your agent, including (without limitation) any such injuries to end users.

Related content

- [AI-based agent authoring overview](#).
- [Create and delete agents](#).
- [Create and edit topics](#).
- [Key concepts - Publish and deploy your agent](#).
- [Key concepts – Analytics](#).
- [Agent flows overview](#)

Choose the right Copilot Studio experience

10/20/2025

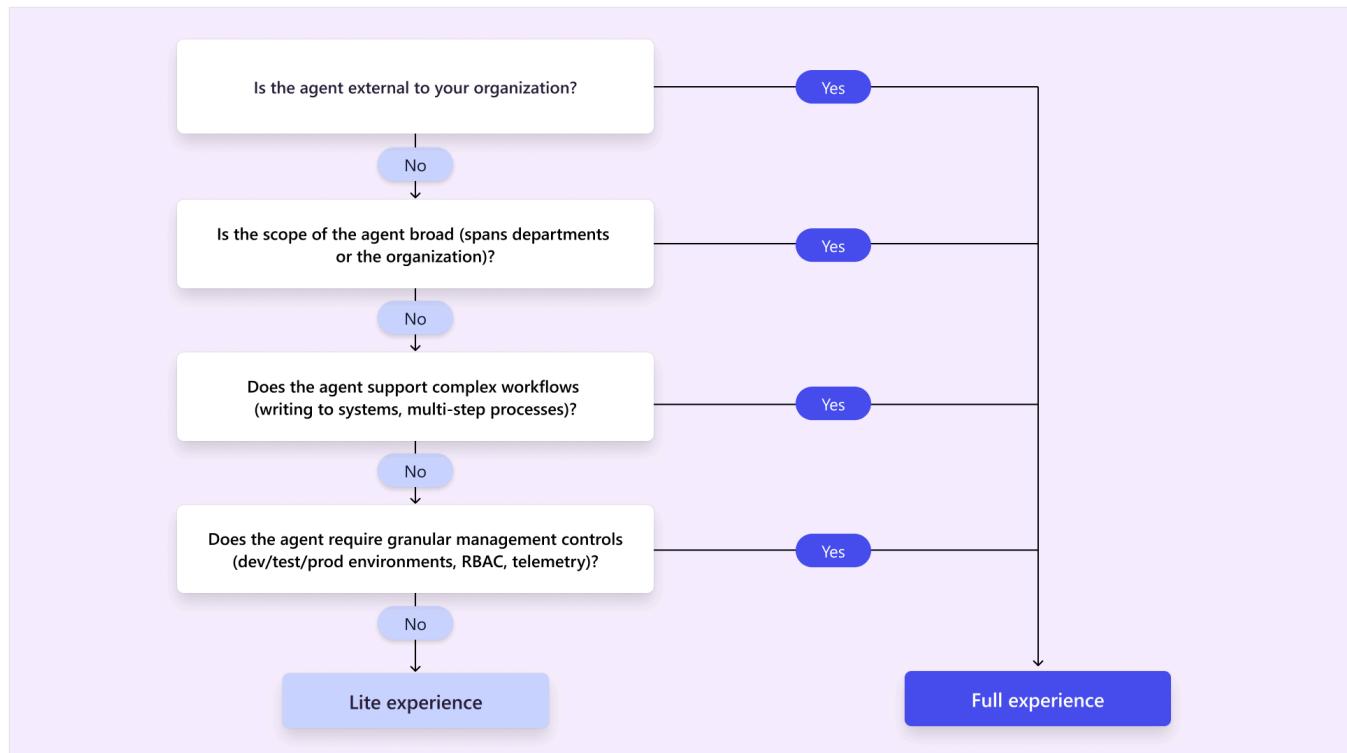
Microsoft Copilot Studio is a powerful platform for building secure, scalable, and intelligent agents that work across Microsoft 365 and line-of-business systems. Copilot Studio offers two ways to build AI agents: a lite experience (integrated into the Microsoft 365 Copilot app) and a full experience (a standalone Copilot Studio application). Both experiences enable you to create powerful agents, but they serve different needs. This article describes the differences between the lite and full experiences to help you choose the right Copilot Studio experience for your scenario.

Compare Copilot Studio experiences

To choose between the lite and full experiences of Copilot Studio to build your agent, consider the following factors:

- **Audience** – Who will use the agent?
- **Deployment scope** – How widely do you plan to share the agent?
- **Functionality** – What tasks will the agent perform?
- **Governance needs** – Does your agent require granular application lifecycle management?

The following decision tree helps you map your scenario to the right experience.



In summary:

- **Choose the lite experience** (formerly agent builder) if you want to quickly create an agent for yourself or a small team, using natural language and existing content (for example, a bot that answers questions from your team's SharePoint files or emails). The lite version is simple, accessible, and integrated with the Microsoft 365 Copilot experience, so you can build agents in context without any code.
- **Choose the full experience** if you need an agent for a broader audience (such as your whole department, organization, or external customers) or if the agent requires advanced capabilities like multi-step workflows or custom integrations, or you need more control over deployment and management. The full version of Copilot Studio is a standalone web portal with a rich set of tools for complex or scalable solutions.

 **Note**

If you choose to use the Copilot Studio lite experience to create your agent and you later want to take advantage of the features available in the full experience, you can [copy your agent to the Copilot Studio full experience](#).

The following table provides a more detailed feature comparison.

 Expand table

Feature	Lite experience	Full experience
Access point	Microsoft 365 Copilot app	Copilot Studio
User type	Information workers	Makers and developers
Agent target audience	Individuals or small teams.	Department, organization, or external customers.
Agent type	Lightweight Q&A agents with organizational knowledge.	Agents with complex scenarios like multi-step workflows or business system integration, and that require enterprise governance and robust controls.
Key capabilities	<ul style="list-style-type: none"> • Natural language authoring • Content-focused Q&A scenarios based on organization context from Microsoft Graph • Respects user permissions to Microsoft 365 data • Uses the Microsoft 365 Copilot orchestrator, foundation models, and services 	<ul style="list-style-type: none"> • Broad and external publishing • Supports multistep logic, approvals, and branching workflows • Supports advanced AI models and integration with Azure AI services • Provides access to prebuilt and custom connectors to connect with data sources beyond Microsoft 365 • Autonomous capabilities

Feature	Lite experience	Full experience
		<ul style="list-style-type: none"> Lifecycle management tools including versioning; development, test, and production environments; role-based access controls; and telemetry and analytics.
Use cases	<p>Use the lite experience to build:</p> <ul style="list-style-type: none"> Project FAQ bots that answers common questions based on project documentation. Product documentation assistants that help employees find information from internal product manuals or wikis. Onboarding agents that help new team members get answers from internal knowledge bases. 	<p>Use the full experience to build:</p> <ul style="list-style-type: none"> Customer support agents that create support tickets and escalates issues to a human. IT help desk triage agents that handle employee IT requests and routes them to the right support team. Sales assistants for CRM that retrieve sales data, makes notes, or kicks off an approval workflow.
Management and governance	Managed primarily through the Microsoft 365 admin center.	Managed through the Power Platform admin center with finer-grained controls for enterprise scenarios.

Copy agents from lite to full experience

You can copy an agent created in the Copilot Studio lite experience to the full experience when you need advanced capabilities or broader integration options. This process ensures that work done in the lite experience is not lost and can be extended in the full experience without a need to start over.

Transitioning to the full experience unlocks additional features such as richer customization, expanded connectors, and governance controls. When you copy your agent, the agent's core configuration and instructions are preserved, and you can enhance them with the advanced settings available only in the full experience.

Consider copying an agent to the full experience when:

- You need enterprise-grade deployment options.
- You want to integrate with more data sources or apply advanced security policies.

For more information, see [Copy an agent to the Copilot Studio full experience](#).

Licensing requirements

Both Copilot Studio experiences are included with a Microsoft 365 Copilot add-on license for authenticated users. If you don't have a Copilot license, you can use Copilot Credits or a pay-as-you-go plan to access either experience.

You can also use the lite experience for free to build agents grounded on web knowledge only. For more information, see [Using agents in Microsoft 365 Copilot Chat](#).

Lite experience governance principles

The lite experience, embedded within Microsoft 365 Copilot allows users to create agents that act as reusable templates. These agents help retrieve insights from Microsoft Graph by packaging repeatable prompts and content connections. They operate within existing enterprise boundaries and respect Microsoft 365 controls.

The lite experience applies the following key governance principles:

- **No new privileges** - Agents respect existing Microsoft 365 permissions. If a user doesn't have access to a SharePoint site, Teams channel, or Outlook mailbox, the agent doesn't surface content from those sources.
- **Built-in visibility and auditing capabilities** - Agents are surfaced within Microsoft 365. Standard audit logs, activity reports, and DLP/retention policies apply.

IT administrators manage agent visibility, sharing, and lifecycle policies in the Microsoft 365 admin center via the **Copilot > Agents** page. Admins can:

- View agent inventory and agent metadata.
- Enable, disable, assign, block, or remove agents to align with organizational policies.
- Manage agent sharing controls.
- Configure pay-as-you-go billing and review agent usage and consumption.
- Enforce compliance using Microsoft Purview (sensitivity labels, audit logs).

Full experience governance principles

The full experience supports the creation of more sophisticated agents, often by makers or developers. These agents can integrate external data sources, call APIs, orchestrate complex workflows, and connect to systems beyond Microsoft 365—ideal for departmental or enterprise-wide solutions.

The full experience applies the following key governance principles:

- **Structured development** - Application Lifecycle Management (ALM) enables development across dev, test, and production environments.
- **Connector governance** - Admins control which systems agents can connect to, reducing risk of unauthorized access.
- **Environment-level policies** - Data loss prevention (DLP), role-based access, and auditing are enforced at the environment level.
- **Flexible deployment** - Agents can be published across Teams, websites, and custom endpoints with granular access controls.
- **Secure collaboration** - Agents support view/edit rights for cross-functional teamwork with oversight.
- **Development and publishing oversight** - Application Lifecycle Management (ALM) supports dev/test/prod environments, and publishing to an organization's app catalog requires admin approval. This ensures visibility and control over what becomes broadly available.

IT administrators use the Power Platform admin center to manage:

- Agent environments and connectors.
- Lifecycle policies and publishing workflows.
- Compliance via Microsoft Purview (sensitivity labels, audit logs, retention).
- Telemetry and usage analytics to monitor agent behavior and ensure policy alignment.

Related content

- [Choose the right tool to build your declarative agent](#)
- [Overview of the Copilot Studio lite experience](#)
- [Use the Copilot Studio full experience to build agents](#)

AI-based agent authoring overview

Article • 04/14/2025

Copilot Studio offers generative AI features to reduce manual authoring and dramatically expand the scope of an agent's knowledge and its ability to interact with users.

Generative AI is an artificial intelligence technology that uses language models to generate original content and provide natural language understanding and responses. Learn more about [Generative AI](#) in the Artificial Intelligence (AI) playbook.

In Copilot Studio, you can use the following generative AI features to retrieve and create content, either individually or all together.

- **Create an instantly useful agent.** With no manual authoring of topics required, an *empty* agent can generate answers based on knowledge sources you specify such as websites and files. See [Generative answers](#) and the [Quickstart](#).
- **Harness AI general knowledge.** When this option is enabled, the agent can answer general questions unrelated to your specific knowledge sources or topics. See [AI general knowledge](#).
- **Author topics using natural language.** Describe what you want your topic to do, and Copilot Studio creates it for you. Your agent includes conversational responses and multiple types of nodes. Use the suggested default topic or as a starting point for further development. See [Create and edit topics with Copilot](#).
- **Turn on generative orchestration.** Let the agent select the most appropriate topics, actions (formerly known as plugins), and knowledge sources at runtime. See [Orchestrate agent behavior with generative AI](#).

Using generative AI in Copilot Studio transforms how you build agents, significantly reducing manual work and configuration.

Generative answers

Generative answers in Copilot Studio allow your agent to find and present information from multiple sources, internal or external, without created topics. Generative answers can be used as primary information sources or as a fallback source when authored topics can't answer a user's query. As a result, you can quickly create and deploy a functional agent. You don't need to manually author multiple topics that might not address all customer questions.

What changed?

Traditionally, when an agent can't determine a user's intent, it asks the user to rephrase their question. If after two prompts, the agent still can't determine the user's intent, the agent escalates to a live agent, using the [Escalate](#) system topic.

Today, before involving a live agent, the agent uses natural language processing (NLP) to:

- Parse what a user types to determine what they're asking.
- Find, collate, and parse relevant information from a specified source. This source can be your company's website, or from multiple sources, including Sharepoint.
- Summarize search results into plain language delivered to the agent user.

Your workflow might look like this:

1. You create an agent and enable the **Generative** option in the **Generative AI** page of Settings. You test the agent thoroughly.
2. After testing, you publish your agent to instantly provide answers, help, and guidance to your agent users.
3. You create individual topics for frequently asked questions. These topics might develop from [analytics from previous agents](#) or existing support issues.

AI general knowledge

In addition to generative answers, you can use AI general knowledge to allow your agent to find and present information in response to your customer's questions. General knowledge saves you from needing to manually author multiple topics, which might not address all your customer's questions. It can also help when a user's intent can't be addressed by existing agent topics.

What is AI general knowledge?

AI general knowledge applies the capabilities of AI to access and provide information, insights, and assistance across a wide range of topics.

 Knowledge
Add data, files, and other resources to inform and improve AI-generated responses.

Allow the AI to use its own general knowledge (preview). [Learn more](#)  Enabled

Why use it?

- **Accessibility:** The agent can instantly access a vast repository of information and expertise across a wide range of subjects.
- **Versatility:** It's capable of addressing diverse topics and tasks, making it a versatile resource for various needs.

(!) Note

While AI general knowledge can provide valuable information and assistance, it's essential to critically evaluate the information it provides and consider consulting other sources for verification or further clarification when necessary.

Prerequisites

- An account for Copilot Studio. If you don't have an account, follow the instruction in [Sign up for a Copilot Studio trial](#).
- The current version of Copilot Studio. The agent type must not be Classic. Classic agents have (classic) added to their name, for example "Contoso store hours (classic)."
- Review AI response generation training, model, and usage in the [FAQ for generative answers](#) and [Learn more about Azure OpenAI](#).

What's supported?

AI-based authoring might be subject to usage limits or capacity throttling.

Quotas

Quotas are default constraints applied to agents that limit how often messages can be sent to an agent. The purpose of quotas is to throttle the client's service load, which protects a service from being overloaded and the client from unexpected resource usage.

Agents with generative answers enabled have a limit on the number of queries they can make derive answers from the URL you specified. Normal conversations that use agent topics follow the [usual quotas and limitations](#).

Languages

See [Chat-based agent language support](#) for the list of supported languages.

Related content

- Get up and running with [Quickstart: Create and deploy an agent](#).
- Add [knowledge sources](#) to your agent.
- Have a conversation to [author topics using natural language](#).
- Use [generative orchestration](#) to call your actions automatically at runtime.

AI features for classic chatbots

06/16/2025

!**Note**

This legacy article only applies to classic chatbots created in the Microsoft Copilot Studio app for Microsoft Teams. Copilot Studio agents as such include next-generation AI features that supersede or replace the features listed in this article.

With classic chatbots, you can turn on extra features that improve the core AI.

You can use these AI features for backward compatibility with older chatbots and support some scenarios. In most cases, you should [convert your chatbots into agents](#) to get the best implementation of multiple AI technologies.

!**Important**

[Generative AI features](#), such as generative answers and natural language understanding, are only available if you create or [convert a chatbot](#) in the Copilot Studio web app.

Prerequisites

You can use the AI features described in this article if:

- Your chatbot is marked **Classic**.
- You created your chatbot in the Teams app.
- You created your chatbot before May 23, 2023.

AI models in Copilot Studio

Copilot Studio employs a language understanding model that uses an example-based approach, powered by a deep neural model. With the Copilot Studio model, you provide a few examples when you craft trigger phrases for a topic. The examples for a single topic are usually 5 to 10 phrases. Shorter trigger phrases are better, and you should aim for 2 to 10 words per phrase. Trigger phrases should be semantically different. For example, changing a single verb or noun could be enough to expand a topic's coverage.

Other changes or additions between phrases can be:

- Articles such as *the*, *a*, or *an*

- Capitalization
- Contractions such as *you're* or *don't*

Plurals don't improve the triggering because contractions are already accounted for in the AI model.

Entities used in related topics automatically identify in user intents when matched with their trigger phrases. For example, the user intent "I want to book a ticket to Boston" matches with the trigger phrase "I want to book a ticket to Paris."

Topic overlap detection

Topic overlap detection helps improve topic triggering accuracy by finding overlaps between topics. Resolving topic overlaps reduces the chatbot's need to ask clarifying questions before triggering a topic.

After you turn on [Topic overlap detection](#), you can view a list of overlapping topics. Go to the [Analytics](#) page for your agent and select [Topic triggering](#).

The **Overlapping topics** list shows a similarity score for each overlapping topic. A higher score means a topic has one or more trigger phrases close to another topic's trigger phrases. You can sort the list by similarity score, topic name, or number of trigger phrase overlaps.

Select a topic in the list to see the overlapping trigger phrases—phrases that have similar sentence structure, words, and grammar.

Using semantically similar trigger phrases for two different topics can lead to confusion. The chatbot might need to ask follow-up questions to the user.

Identifying semantically similar trigger phrases can also help you consolidate topics if they're too similar, or edit topics to make them more distinct.

You can go directly to a topic from the **Topic overlap details** panel by selecting the corresponding link.

After you revise trigger phrases for a topic and save your changes, the overlapping status automatically refreshes. To manually refresh the topic overlap status, select the refresh icon in the **Overlapping topics** section.

Overlapping topics

Trigger phrases that are too similar blur the triggering, review overlapping topics and tri

 Updated 6 minutes ago

Topic suggestion from chat transcripts (preview)

This feature analyzes sessions between your chatbot and users and gives suggestions based on unmatched user input.

After you turn on [Topic suggestions from chat transcripts](#), go to the [Analytics](#) page for your agent and select **Topic triggering (preview)**. A list of potential topics appears with the number of times a query about this topic was made by users of the chatbot. The top 200 suggestions are shown.



Topic suggestions from chat transcripts (preview)

Each topic suggestion represents related queries about the same issue. Your bot was unable to match these to an existing topic - consider adding a new topic to expand coverage.

Updated 17 hours ago

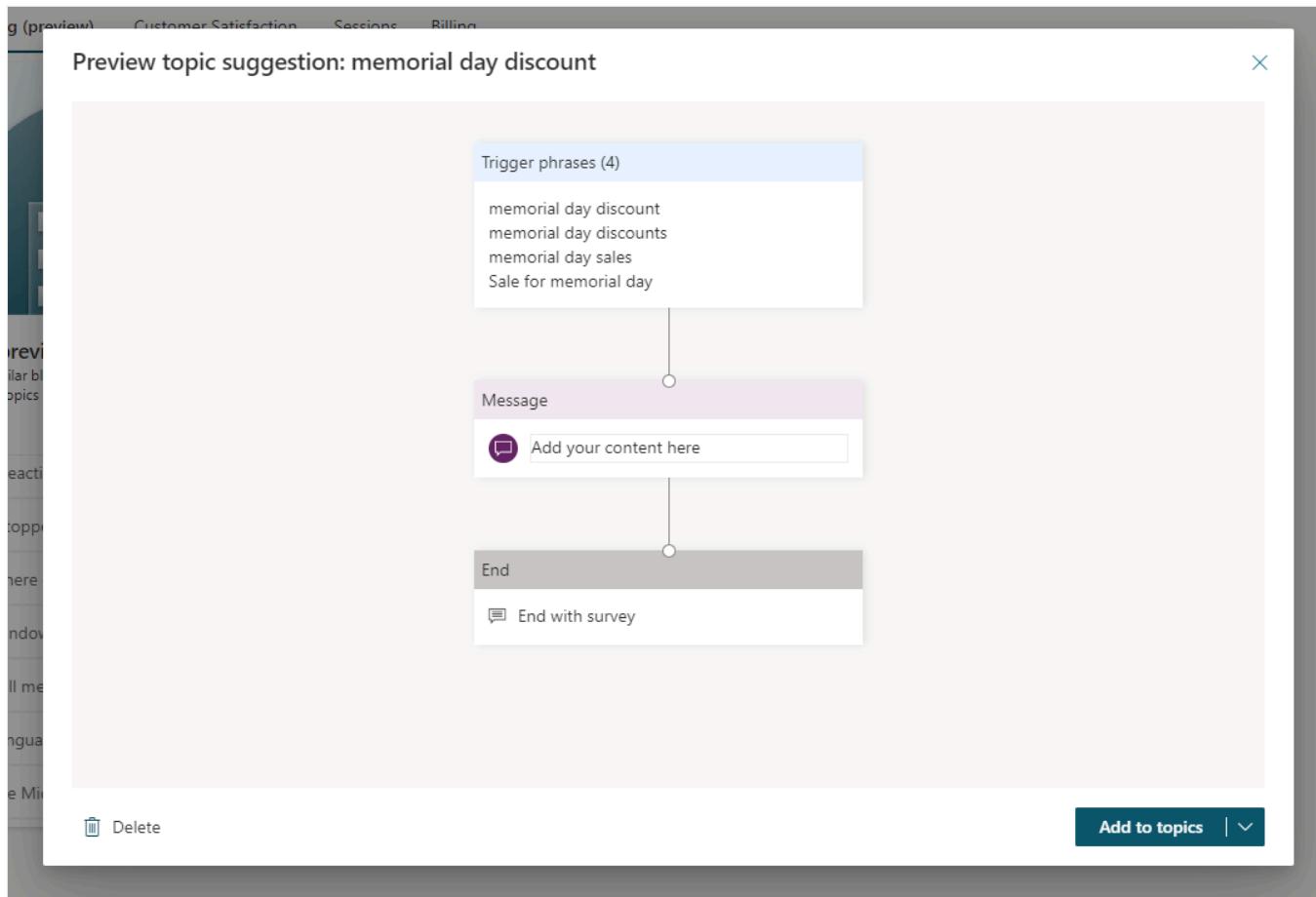
Topic suggestions (4)	Sessions ↓
memorial day discount	4
holiday deals	3
Surface laptop 4	3
Hololens	2

The topic suggestion analyzer automatically runs once every one to two hours. The analyzer scans all new queries made since the analyzer last ran, and groups queries that don't match an existing topic. Your chatbot needs at least 100 new conversations from the last time any

suggestion was generated to trigger for the process. Suggestions with more than three user sessions are shown.

When you select an item in the suggestion list, a topic suggestion window appears, showing the topic with some suggested trigger phrases. The suggested trigger phrases are based on queries made by users that aren't matched to an existing topic.

You can choose to delete the entire suggested topic, for example if the topic is irrelevant to the chatbot, or add it to your list of topics by selecting **Add to topics**.



Turn AI capabilities on or off

1. Open your chatbot and go to **Settings**. In the web app, the chatbot must be marked with **Chatbot (classic)**.
2. In **Settings**, select **AI capabilities**.
3. Turn the features on or off, as desired.
4. Select **Save** at the top.

What's new in Copilot Studio

10/24/2025

This article provides resources to learn about new features and features planned for upcoming releases of Copilot Studio.

Released versions

For information about the new features, fixes, and improvements released in the past few weeks, see [Released versions of Microsoft Copilot Studio](#).

 Note

Releases are rolled out over several days. New or updated functionality might not appear immediately.

Notable changes

The following sections list features released in the past months, with links to related information.

October 2025 - Announcement

- GPT-4o Retirement for customers using Generative Orchestration. Starting **October 27, 2025**, GPT-4o will be retired in Copilot Studio, except for GCC customers who will continue using GPT-4o. The new default model is GPT-4.1, which delivers improved performance, reliability, and consistency across experiences. GPT-4o remains available until **November 26, 2025** if you enable the "[Continue using retired models](#)" option. We recommend transitioning to GPT-4.1 or another supported model to maintain access to advanced AI capabilities.

September 2025

- (Preview) Automate tasks in desktop applications on Windows using [Computer-Using Agents \(CUA\)](#), which combines vision and reasoning to interact with interfaces—even when APIs aren't available.
- Embed Copilot agents into Android, iOS, and Windows apps using the [Client SDK](#) to provide rich, multimodal conversations within native experiences.

- (Preview) Upload Excel, CSV, and PDF files for your agent to analyze using Python code, powered by the [code interpreter in chat](#).

August 2025

- (General availability) Use [code interpreter](#) to generate Python code-based actions from natural language in both the prompt builder and agent workflows.
- (General availability) Enhance agentic response accuracy in Copilot Studio agents by using [file groups](#) to organize local files to be uploaded as a single knowledge source and apply variable-based instructions.
- Allow users to [upload files and images](#) that your Copilot Studio agent can analyze and use to generate responses, then pass those files to downstream systems using Agent Flows, Power Automate, connectors, tools, and topics for seamless integration.
- (General availability) Track and analyze unanswered queries and response generative AI quality with the [generated answer rate and quality](#) section in the Analytics page to improve your agent's performance.
- Connect to an existing [MCP server](#) directly within Copilot Studio using a guided experience.

July 2025

- Use [advanced NLU customization](#) to define topics and entities using your own data for higher accuracy and improved containment, especially for Dynamics 365 scenarios.
- [Search across your agent's knowledge](#), topics, tools, skills, and entities instantly using a new in-app search experience accessible via keyboard shortcut or top-level search.
- Estimate time and cost savings based on successful runs or actions and customizable to your organization's metrics with [ROI analytics](#) for agents with autonomous capabilities.
- View user comments submitted with [thumbs up/down reactions](#) in analytics, offering deeper insight into customer feedback on agent responses.
- (Preview) Display [Microsoft Information Protection \(MIP\)](#) labels across connectors, test chat, Teams, and Microsoft 365 Copilot to prevent oversharing and support secure, compliant AI experiences. With new integrations between Copilot Studio, Dataverse, and Microsoft Purview, you can automatically classify sensitive data and ensure agents respect Purview sensitivity labels.

- Publish agents directly to a [WhatsApp](#) phone number, making it easier to reach customers.
- (Preview) Streamline authentication for Microsoft Entra ID-backed actions and connectors with the SSO Consent Card by allowing users to grant consent directly within the chat with no redirects and no interruptions.

June 2025

- Improved experience for tools:
 - Grouping and filtering for easier search and discovery of tools.
 - Support for IntelliSense automatic completion and input widgets such as calendar control, file picker, and timezone picker, when configuring tools.
 - Improved tool invocation experience for customers with more affordances for complex inputs and clearer error messaging.
 - Automatic detection of SSO for connectors.
- (Preview) [Support for Microsoft 365 Copilot Tuning](#) to train models on your own enterprise data for domain-specific tasks and integrate these models into Microsoft 365 experiences like Copilot in Teams, Word, and Chat. You can also connect your fine-tuned models to custom agents.
- Actionable insights for questions that generative AI left unanswered, grouped by themes, in the **Answer rate and quality** section of the **Analytics** page.
- Knowledge sources analysis for autonomous agents.
- Ability to insert Power Fx formulas directly in the embedded [AI Builder prompt editor](#).
- Simplified text validation and extraction with regular expression support for [Power Fx formulas](#) that use IsMatch, Match, or MatchAll functions.
- (Preview) Support for [file groups](#) as knowledge sources.
- (Preview) Generative orchestration available for all [supported languages](#).
- Redesigned **Channels** page.
- (US-only preview) Ability to select the GPT-4.1 mini [experimental response model](#) for generative answers.

May 2025

- (Preview) Ability for makers to [connect an agent to other agents](#) to complete tasks and respond to users.
- Integrated [Adaptive Card designer](#) with built-in localization support.
- Ability to deploy Copilot Studio agents to [SharePoint](#).
- [New controls](#) in Generative AI agents to better shape agent responses with knowledge and feedback.
- General availability of [Microsoft 365 Agents SDK](#) and [Microsoft 365 Agents Toolkit](#).
- Edit agents with the [Copilot Studio extension](#) for Visual Studio Code.
- (Preview) New autonomous agent template: [Document Processor](#).
- General availability of agents ability to [use web search](#).
- (Preview) Bring Your Own Models from Azure AI Foundry to your custom engine agents.
- General availability of [Model Context Protocol support](#) with enhanced tracing and analytics.
- (Preview) Ability to build agents with [Dataverse and Dynamics 365 Model Context Protocol \(MCP\) servers](#).
- General availability of new and improved knowledge creation experiences with knowledge recommendations, previews, and connection creation.
- (Preview) Support for [unstructured knowledge](#) from ServiceNow, Salesforce, Confluence, and Zendesk, enabled through Power Platform connectors.
- (Preview) Enhanced file upload experience with support for adding files from OneDrive and SharePoint.
- General availability of [Azure AI Search knowledge](#) with support for hybrid search index, virtual network support, and better citations.
- General availability of [tabular data knowledge](#) from systems of records like Dataverse, Salesforce, ServiceNow, Azure SQL Server.
- General availability of connectors.
- Support for [Federated Identity Credentials \(FIC\)](#).

April 2025

- (Preview) Support for [Customer Managed Keys](#).
- Support for generative and autonomous agents in Analytics.
- Inclusion and visibility of agent data and ROI analysis in Viva Insights.
- [Nine new Graph connectors](#): *Guru, GitLab* (Issues, Merge Requests, Knowledge), *Asana, 15Five, Miro, Trello, Zendesk Ticket, SmartSheet, and Seismic*.

March 2025

- Enhancement of conversation transcripts to include node-level data.
- General availability of [autonomous agents](#).
- Simplified integration with AI apps and agents using [Model Context Protocol](#).
- Worldwide general availability of [generative orchestration](#) in Copilot Studio.
- Preview release of [deep reasoning models](#) for agents.
- Ability to [build agent flows](#) directly in Copilot Studio.

February 2025

- [Hebrew \(he-IL\) language support](#) for interactions between agents and users, including support for generative answers, for text only (voice isn't supported).
- A new topic trigger, **AI response generated**, lets the agent override, modify, or log responses generated by the orchestrator before sending them to the conversation.
- (Preview) Ability to publish custom agents created in Copilot Studio directly to [Microsoft 365 Copilot Chat](#).
- (Preview) Support for more enterprise data sources using [Microsoft Copilot connectors](#).
- Security: Improved mitigation of cross-prompt injection attacks (XPIA).
- [Data policy enforcement](#) applies by default for agents in all tenants.
- Microsoft 365 Agents [Software Development Kit for JavaScript](#).

January 2025

- Microsoft Purview can access and manage [user interaction audit logs](#) from Copilot Studio agents.

- Chinese (Traditional) (zh-TW) [language support](#) for interactions between agents and users, including support for generative answers.
- Default data policy enforcement for agents is set to [Soft-enabled](#) for all tenants.

December 2024

- Arabic (ar-SA) language support for interactions between agents and users, including support for generative answers. For more information, see [Language support](#).
- New usage-based billing model. For more information, see [Manage message capacity](#).
- Ability for administrators to limit or prevent [sharing of agents](#).
- Agents can analyze [uploaded images](#).

November 2024

- (Preview) Ability to build [autonomous agents](#) that respond to events and can complete workflows in the background.
- Ability to add [Azure AI Search indexes](#) as knowledge sources.
- Real-time knowledge connectors for Salesforce, ServiceNow Knowledge, and ZenDesk, adding to an [extensive library of connectors](#).
- Improved [agent effectiveness analytics](#), with knowledge source usage details.
- Improved overall AI quality, with better generative answers attribution, and contextual understanding of Microsoft Copilot connector and SharePoint knowledge sources.
- Increased maximum supported file size for [SharePoint knowledge sources](#) to 7 MB.
- More intuitive and accessible UI for adding knowledge sources.
- Ability to create [multilingual voice-enabled agents](#).
- Automatic [security scan](#) in Copilot Studio.
- Ability to [configure a welcome message](#), to inform makers about important privacy and compliance requirements.
- Autonomous agents governance with [data policies](#).

October 2024

- When using generative AI, [voice-enabled agents](#) return responses and citations optimized for voice.
- Copilot agents built in Copilot Studio can be published [directly to Microsoft 365 Copilot](#), surfacing right in the user's flow of work.
- New [Copilot Studio experience](#) that empowers users of any skill level to create Copilot agents from within SharePoint and Microsoft 365 Copilot Business Chat (BizChat).
- Manage Copilot agents with [centralized controls](#).
- Integrated [solution explorer](#) within Copilot Studio.
- (Preview) Ability to [create reusable component collections](#).
- [GPT-4o model](#) updated for [answers](#).
- Updates and improvements to [generative orchestration](#).

September 2024

Generative answers can cite non-text elements in [uploaded files for generative answers](#).

August 2024

- Improved UI for citations and AI-generated labels in Teams chat using generative answers.
- Generative answers support [uploaded files up to 512 MB in size](#).
- Copilot Studio is available in Germany and the United Arab Emirates: [Geographic data residency in Copilot Studio](#).
- Generative orchestration is available in GCC High: [Orchestrate agent behavior with generative AI](#).
- Improved Web Chat performance.
- Improved authoring canvas performance.
- [GPT-4o model](#) updated for [orchestration](#).
- [GPT-4o model](#) updated for [creating a new copilot](#).

Release plans

For information about new features being released over the next few months that you can use for planning, see [Release Planner](#).

Get access to Copilot Studio

10/03/2025

This article includes information about Copilot Studio licensing and the free trial.

ⓘ Important

For the most up-to-date Copilot Studio licensing and billing information, refer to the the [Microsoft Copilot Studio Licensing Guide](#).

If you already have licenses or you're an administrator, see the [Assign licenses and manage access to Copilot Studio](#) article.

Copilot Studio is [available in the US Government Community Cloud \(GCC\) plan](#).

First, [sign up for Copilot Studio](#). For more information and to request assistance, visit the [Microsoft Copilot Studio Community](#).

Sign up for a Copilot Studio trial

You can sign up for Copilot Studio as an individual. After you finish the sign-up process, your free trial for Copilot Studio starts. You see notifications and receive emails to inform you about the trial expiry. When the trial expires, you can extend it by 30 days.

! Note

Your agent continues to work for up to 90 days after your trial expires, so you don't have to worry about extending at the exact time of expiry.

1. Go to the [sign-up page](#).
2. Enter your email address and select **Next**.
3. Follow the instructions. After you complete the process, you can use Copilot Studio to create and publish agents.

Troubleshooting the trial sign-up process

If you have trouble signing up for the trial, check the following common issues:

- Rejected email address: This issue might happen if you used a personal email address for the trial. Instead, use a work or school account.
- Received a message that your sign-up couldn't be completed: This issue likely means your organization's IT administrator disabled the self-service sign-up for Copilot Studio. To finish signing up, contact your IT administrator and ask them to follow the instructions to enable sign-up.

Standalone Copilot Studio subscription

The standalone Copilot Studio subscription allows you to build agents on any supported channel and connect to any data using premium connectors.

You can obtain a standalone Copilot Studio subscription from the Microsoft 365 admin center. For more information, see [Assign licenses and manage access to Copilot Studio](#).

Copilot Studio for Microsoft Teams plans

Copilot Studio for Teams enables customers to build conversational interfaces within Teams. The agents can use data stored in Microsoft Dataverse for Teams or many other sources, using the supplied standard connectors.

Capabilities available in the Copilot Studio app in Teams are available as part of select Microsoft 365 Copilot subscriptions with Microsoft Power Platform and Teams capabilities. This plan excludes plans for US government environments (GCC, GCC High, and DoD), EDU A1, and SUB SKUs.

This table compares key capabilities in the Copilot Studio for Teams plan, which is available in select Microsoft 365 Copilot subscriptions, against the standalone Copilot Studio subscription. For a full, comparative list, see the [Microsoft Power Platform Licensing Guide](#).

Also see the [Quotas and limits](#) article for other capacity considerations.

 Expand table

Capability	Select Microsoft 365 subscriptions	Copilot Studio subscription
Gen AI: AI-enabled conversations	Not available	Orchestrate agent topics and actions with generative AI
Deploy agents to channels	Teams	Any channel supported by Copilot Studio

Capability	Select Microsoft 365 subscriptions	Copilot Studio subscription
Power Platform connectors	Standard connector actions in Copilot Studio	Premium connector actions in Copilot Studio
Power Automate flows (Automated, instant, and scheduled flows) within the context of Copilot Studio creations	Not available	Create a flow
Web security	Secure access enabled by default, can't generate secrets to enable secure access	Can generate secrets and turn on or off secure access by the agent author
Create (edit) with Copilot	Not available	Can create and iterate on topics by describing what you want, then AI builds it
Use Microsoft Bot Framework skills	Not available	Can extend Copilot Studio copilots with Microsoft Bot Framework skills
Use a Copilot Studio copilot as a Bot Framework skill	Not available	Use a classic chatbot as a skill in a Bot Framework bot
Hand off agent conversation to a live representative	Not available	Hand off to a live agent

Upgrade your license

To access the full range of Copilot Studio capabilities, upgrade your plan to a standalone Copilot Studio subscription.

After you upgrade your license, you can continue using the same agent in the same environment. Capabilities that were previously only available in a standalone license are now available. These capabilities might include [billed sessions that require Copilot Studio capacity](#).

You can upgrade within the Copilot Studio app for Teams from **Settings > Channels**. Select the apps you want to publish to and you're prompted to upgrade.

You can [start a 60 day free trial of Copilot Studio](#) to try out all the capabilities. You're prompted to sign up for a trial if you try to create an agent without a license.

 **Note**

The option to create a free trial might be disabled. For more information, see the [Self-service sign-up disabled](#) section.

Other subscriptions that include Copilot Studio

Entitlements for Copilot Studio are included in Digital Messaging and Chat add-ons for Dynamics 365 Customer Service. For more information, see the [Dynamics 365 Licensing Guide](#).

Quickstart: Create and deploy an agent

08/14/2025

⚠ Note

This article applies to the Copilot Studio online portal. To create a chatbot in the Microsoft Copilot Studio app for Microsoft Teams, see [Quickstart: Create a chatbot in Teams and deploy it](#).

Copilot Studio empowers teams to quickly and easily create powerful agents using a guided, no-code graphical experience without needing data scientists or developers.

This quickstart shows you how to create an agent for the first time. Learn how to:

- Add knowledge to your agent.
- Test content changes in real time.
- Publish your agent to a demo website you can share with others.

Create an agent

When you create an agent, you can describe everything you want your agent to be and do in your own words, or configure those details as separate pieces of information.

The examples in this quickstart create an agent called *Friendly Tutor* to help users learn about Copilot Studio using knowledge from the official Copilot Studio documentation. However, you can create your own agent using any publicly available website as a knowledge source.

1. Sign in to Copilot Studio, or [sign up](#). You land on the **Home** page.
2. Enter a brief description of what you want your agent to be—for example, "Help users learn how to create agents with Copilot Studio." The description can have up to 1,000 characters.

Describe your agent to create it

Helpdesk

Expense tracking

HR and benefits

Help users learn how to create agents with Copilot Studio.



Features labeled as 'preview' are subject to supplemental terms. [See terms.](#)

A page appears with two panes where you can set up your agent on the left-hand side, and test it on the right-hand side. The left-hand pane has two tabs: **Describe** and **Configure**. You can alternate between these two experiences, and also iterate between setting up your agent and testing it.

Start building your agent

Create

...

Describe

Configure

Jul 9, 2025, 6:48 PM



Hello, I'm here to help you start building an agent. Just describe what it should do using everyday words.

If you prefer to start building manually, you can switch to the 'Configure' view at any time. You may also skip this step by selecting 'Continue'.

Help users learn how to create agents with Copilot Studio.



Great! Now let's determine a name for your agent. How about "Copilot Studio Guide"? Does that work for you?



Agent

This agent helps users learn how to create agents with Copilot Studio.

Getting Started

How do I start creating an agent in Copilot Studio?

Configuration

What are the steps to configure an agent?

Best Practices

Can you share some best practices for creating agents?

Troubleshooting

I'm having trouble with my agent setup. Can you help?

Advanced Features

What advanced features can I add to my agent?

Examples

Can you provide examples of well-configured agents?

Type your message



Type your message

0/2000



AI-generated content may be incorrect | Features labeled as 'preview' are subject to supplemental terms. [See terms](#).

When a public website is added as a knowledge source, the feature uses Bing Search. Your data will flow outside your organization's compliance and geo boundaries. Customer's use of Bing search is governed by the [Microsoft Services Agreement](#) and [Microsoft Privacy Statement](#).

⚠ Note

The conversational agent creation experience uses your browser language. If your browser language isn't in the list of [supported languages](#) for the conversational agent creation experience, only the **Configure** tab with the more discrete configuration experience is available.

3. Chat with Copilot in the left pane to further define your agent. Copilot asks questions to guide you, and uses your responses to fill in the details, such as the name, description, instructions, and knowledge, that define your agent.

- a. For this example scenario, tell Copilot you want to name your agent "Friendly Tutor." Notice the new name appears in the test pane. The maximum length for the name is 30 characters.
- b. If desired, refine the instructions Copilot generated for your agent but keep them simple for now. Just make sure you include information about what your agent helps users do. Notice the suggested prompts in the test pane are automatically updated, following your changes. The instructions can have up to 8,000 characters.
- c. Specify the desired conversation style and tone your agent should use, for example, "Friendly Tutor should talk to users like a kind, patient teacher."
- d. Add knowledge to your agent, if desired. For Friendly Tutor, tell Copilot you want to use <https://learn.microsoft.com/microsoft-copilot-studio> as a knowledge source.

 **Note**

During the creation experience, your agent might need a few moments before it can start using its knowledge sources.

Once you have a name, description, and instructions for your agent, you can already start testing it!

4. Chat with your agent in the test pane. Enter your own question or try one of the suggested prompts. Go back and forth between the test pane and the creation pane as needed, to further refine your agent. If you want to see the suggested prompts again, select **New chat** at the top of the test pane.
5. Change the primary language your agent uses to communicate, if desired.

 **Important**

You can't change the primary language of an agent after the agent's creation. However, you can [change the region](#) for the agent's primary language, if available. You can also [add secondary languages](#).

- a. Select **Edit language** in the right-hand pane.
- b. Select the desired language and select **Save**.
6. Select **Create**. The **Overview** page for your agent appears. Now you can start [testing and improving your agent](#).

Tip

Instructions determine the goals, tone, and limitations of your agent. When writing instructions, make sure you:

- Use conversational language with a full description of how you want your agent to behave. Avoid single-word responses, because they don't give the agent enough information or context.
- Provide details about the specific goal you want your agent to help users achieve. For example, *Create an agent for topics related to Human Resources and employee benefits.*
- Describe the tone as how you would want a person to respond in a conversation. You can use words and phrases like *casual, formal, and understandable for children in grade three.*

Improve your agent

Now that you have an agent, you can test how changing your agent's instructions affects its behavior and improve it.

To open your agent, select **Agents** in the left sidebar, then select your agent.

Test changes to your agent

The best way to improve your agent? Test it. Make some changes. Test it again. Repeat.

In this section, you test how changing your agent's knowledge sources affects how your agent responds to users.

1. Start by testing how your agent currently responds in the test chat. Ask your agent a question. For example, ask: "How do I add a knowledge source?"

The screenshot shows a user interface for 'Copilot Studio'. At the top, there's a navigation bar with icons for settings, file operations, and a refresh button. Below this, a blue header bar contains the text 'How do I add a knowledge source?'. Underneath the header, the text 'A minute ago' is displayed. A checkbox labeled 'Based on official sources' is checked. The main content area provides instructions for adding a knowledge source:

To add a knowledge source in Copilot Studio, follow these steps:

- 1. Navigate to the Knowledge Sources**
Section: Open Copilot Studio and go to the section where you can manage knowledge sources.
- 2. Add a New Source:** Look for an option to add a new knowledge source. This might be a button labeled "Add Source" or something similar.
- 3. Select the Type of Source:** Choose the type of knowledge source you want to add. This could be a document, a database, or another type of data repository.
- 4. Provide Necessary Details:** Enter the

Below these steps is a text input field with the placeholder 'Ask a question or describe what you need' and a character limit of '0/2000'. To the right of the input field are two icons: a paperclip and a send arrow.

At the bottom of the main content area, a note reads: 'Make sure AI-generated content is accurate and appropriate before using. [See terms](#)'.

In this example, the agent's instructions are to *talk to users like a kind, patient teacher*. What if you give your agent different instructions?

2. Go to the **Overview** page, and update the instructions for your agent to use a different tone, like *talk to users like Jane Austen*.
3. Test your agent's new instructions with another question. How has the response changed?

Change your agent's introduction

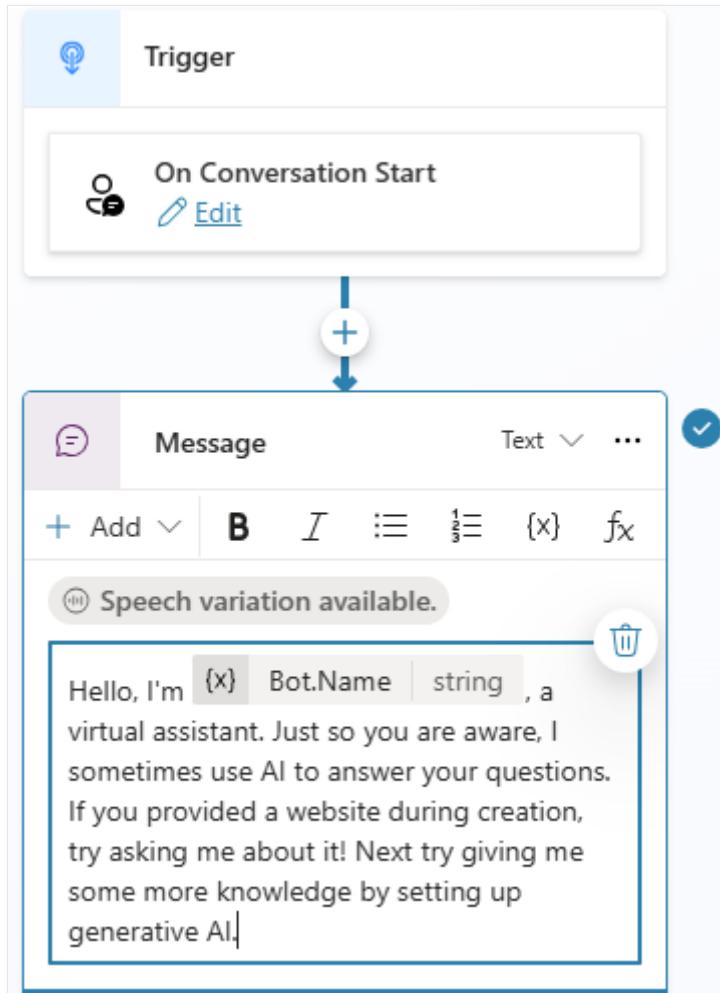
Help your agent make a great first impression with a new introductory message. This message lets users know what your agent does and encourages them to interact with your agent.

1. In the **Test your agent** chat, select your agent's introductory message. The **Conversation Start** topic opens, and the **Message** node for your introductory message is in focus.

💡 Tip

If you can't see the introductory message in the test panel, select the Refresh icon at the top of the panel to restart the conversation.

2. In the **Message** box, select the text of the message.



3. Replace the default message with your own. In the introductory message, your agent should greet users, tell them what it does, and how to start interacting with it. You can also give users an example question or prompt.

For Friendly Tutor, update the introductory message to say:

Hello, I'm here to help you learn how to use Microsoft Copilot Studio. You can ask me all about agents: "What is an agent?" "How do I make an agent?" "How do agents work?"

4. Select **Save**.

5. To test this change, select the **Refresh** icon in the **Test your agent** chat panel.

Suggest ways of starting conversations

If your agent is meant to be used in Teams or Microsoft 365, you can configure up to six suggested prompts customers can choose from to start a conversation with the agent. When you use the conversational agent creation experience, Copilot automatically generates suggested prompts based on information in the description and instructions for the agent.

To add or update suggested prompts:

1. On the **Overview** page, select the **Edit** icon  at the top of the **Suggested prompts** section.
2. Revise or add titles and prompts, as desired, and select **Save** when you're done.

Note

Suggested prompts are meant for Teams and Microsoft 365 Copilot Chat. They appear on the agent's welcome page, before you start a new chat. You can't see or use them when you test your agent in Copilot Studio, except in the initial creation experience. Learn more about [configuring suggested prompts](#).

Edit your agent's basics

You can change your agent's name, description, instructions, and knowledge sources after creating it. Remember to test your changes as you go!

To update your agent's name and description:

1. On the **Overview** page, select **Edit**, at the top of the **Details** section.
2. Make the desired changes.
3. Select **Save**.

To update the instructions:

1. On the **Overview** page, select **Edit**, at the top of the **Instructions** section.
2. Make the desired changes.
3. Select **Save**. Remember, editing the instructions changes how your agent engages with users. Make sure to test your changes.

To add a knowledge source:

1. On the **Overview** page, go to the **Knowledge** section, and select **Add knowledge**.
2. Select the desired type of knowledge. This quickstart uses **Public websites** sources.
3. Enter the URL for the website and select **Add**.
4. Name and describe the knowledge source so you can keep track of all your agent's sources.
5. Select **Add**.

To change an existing knowledge source:

1. On the **Overview** page, go to the **Knowledge** section, and select the three dots (...) for the knowledge source.
2. Select **Edit** to change the knowledge source, or **Delete** to remove it from your agent's sources.

Make your agent unique with a distinctive icon

1. Select the agent icon in the top bar.
2. Select **Change icon**.
3. Choose an image from your device. The image file must be in PNG format and less than 30 KB in size, with a maximum resolution of 192 × 192 pixels.
4. Select **Save**.

Publish your agent

Once you're happy with your agent, you can publish it to a live or demo website. Since this step depends on the platforms and channels you have access to, go to [Key concepts - Publish and deploy your agent](#) and [Publish an agent to a live or demo website](#) for more information.

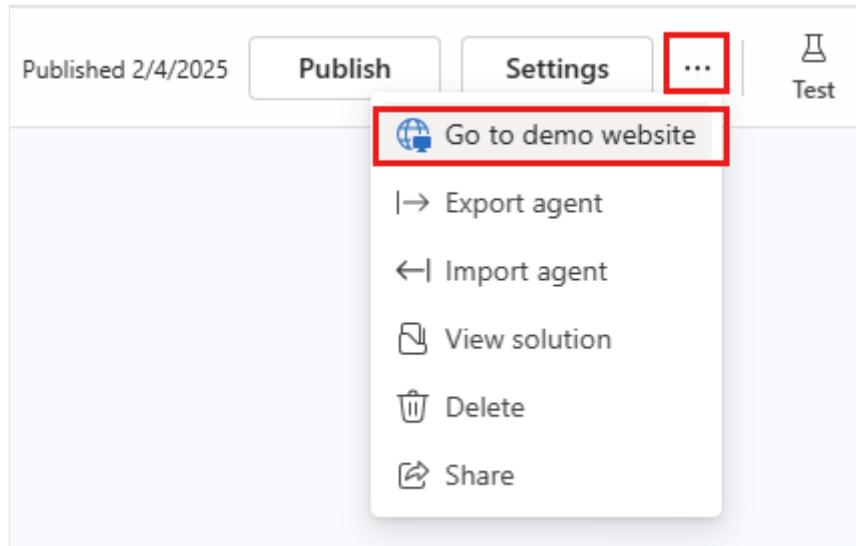
(!) Note

You might need to change the authentication for your demo website, depending on who you want to access your agent. Go to [Key concepts - Publish and deploy your agent](#) for information on authentication levels and how to change them.

1. At the top of the page, select **Publish**, and then select **Publish** again in the **Publish this agent** confirmation message. If the publish is successful, you see a green banner on the

top of the page.

2. At the top of the page, select the three dots (...) and select **Go to demo website**.



3. Send the URL to others to demonstrate it.

For more information about publishing your agent to other channels, see [Key concepts - Publish and deploy your agent](#).

What's next?

You created an agent, tested it out, and published it to a demo site. Congratulations! Your agent has many more capabilities, so try it out and play with the advanced features.

For questions not covered in the documentation or for feature ideas, [visit our community](#) and post questions.

We'd love to hear your ideas on Copilot Studio. [Visit our Ideas board](#) and post your ideas.

Quickstart: Create and deploy a classic chatbot

05/30/2025

ⓘ Note

This article only applies to the Microsoft Copilot Studio app for Microsoft Teams. To create your first chatbot in Copilot Studio, see [Quickstart: Create and deploy an agent](#).

Copilot Studio empowers anyone in the organization to build chatbots in Microsoft Teams. The chatbot has built-in natural language processing (NLP) and a no-code graphical interface. With our latest Teams integration, you can create, author, test, and publish your chatbot to Teams within the Microsoft Teams interface.

When you create a chatbot, consider:

- The types of questions someone might ask
- The different ways a question is asked
- The expected answers
- Whether or not a user has follow-up questions or concerns

To help you get started making human resource (HR) chatbots, this quickstart shows you how to:

- Map your scenario and needs to align with what a chatbot can do
- Create a chatbot in the Copilot Studio app in Teams
- Build and enhance topics that answer common HR-style questions
- Test, publish, and share the chatbot with your organization

HR Support Bot is a friendly question and answer chatbot that brings a support professional expert from the HR team, when HR personnel are unable to help. An employee can ask the chatbot a question and the chatbot responds with an answer if the chatbot has content for it.

If the chatbot doesn't have an answer, the employee can submit a query. The query is sent to a predetermined team of experts who respond to query notifications.

Scenario

You work in human resources (HR) at Contoso and need to build an HR support chatbot in Microsoft Teams. The chatbot answers common HR questions, such as employee time-off, with

relevant information from the company's internal knowledge base. The chatbot contacts HR experts and notifies them that an employee needs assistance for uncommon scenarios.

Once the chatbot is published, Contoso employees can access HR information in Teams without needing to search the company's intranet.

Install the Copilot Studio app in Microsoft Teams

Start by launching the Copilot Studio app in Teams:

1. Go to the Teams app store and search for "Microsoft Copilot Studio."
2. Select **Add or Open**, if you already added it, for the Microsoft Copilot Studio app.

The app opens to the Copilot Studio home page. Teams users can build chatbots here without leaving Teams.

💡 Tip

Select the three dots (...) in the side bar of Teams to see your apps. Select and hold (or right-click) the Microsoft Copilot Studio icon and select **Pin**. The Microsoft Copilot Studio app appears in your side bar. You can now return to the app more easily.

Create a new chatbot in a team

1. Select **Start now** on the Home page.
2. Pick which team manages the chatbot, and select **Continue**. You can [create a new team ↗](#) if necessary, before you add a chatbot to it.
3. Select the desired language. For more information about choosing a language, see [Language support](#).
4. Enter a name for your chatbot—for example, "HR Support Bot."
5. Replace the default icon with a more representative one, if desired.
6. Add suggested prompts, if desired.
7. Select **Create**.

Build an employee time-off topic for common time-off queries

A chatbot comes with a collection of topics that defines how it interacts with users. Let's build a topic to answer employee questions about taking time off.

1. Select **Topics** in the left pane.
2. Select **New topic** > **From blank**.
3. Select the **Trigger Phrases** box to add phrases.
4. Add trigger phrases. For example, enter the following phrases:
 - Need information on time off
 - I need help with time off
 - How many days of paid vacation do I have
 - What are the national holidays
 - I need extended leave

Trigger phrases for each topic define the ways users ask questions about that topic. These phrases are how we expect employees to ask about time off.

Copilot Studio comes with built-in natural language capabilities. You only need to define a few trigger phrases about time-off and the chatbot can accurately trigger the topic based on the employee's input.

5. Close the **Trigger phrases** panel when you're done. The **Trigger Phrases** node shows the phrases.
6. Select **Details** on the toolbar.
7. Name the topic "Employee time off" and enter an appropriate description.
8. Close the **Details** panel.
9. [Define how the chatbot interacts with employees](#).
10. Select **Save**. Your new topic becomes available.

Define how the chatbot interacts

1. Enter the following text in the existing message node: "I can help with questions related to time-off."

As an HR employee, you know the most common time-off questions are about paid vacation time and national holidays.

2. Select the **Add node (+)** icon below the message node, then select **Ask a question** to add a question node to the topic.
3. Enter "What information are you looking for?" in the **Ask a question** text box. The employee might ask this question.
4. Under **Options for user**, add "Paid vacation" and "National Holidays" as two options.

The screenshot shows the Power Virtual Agents (preview) interface. On the left is a sidebar with icons for Home, Chatbots, and About. The main area has a breadcrumb trail: Back / Employee time off. A green banner at the top says 'Dialog saved successfully'. The main content area shows a dialog flow:

- Trigger Phrases (5):** need extended time off, what are the national holidays, how many days of paid vacation do i have, I need help with time off, need information on time off.
- Message:** I can help with questions related to time off.
- Question:** What type of information are you looking for?
- Identify:** Multiple choice options > Paid vacation, National Holidays.
- Options for user:** Paid vacation, National Holidays.

A toolbar on the right includes icons for search, refresh, and delete. A bottom bar has icons for gear, magnifying glass, and a plus sign.

User choices are stored in a variable and the topic branches off, based on the option the user chooses. You can rename the variable to track it better in the topic.

5. On the variable, under **Save response as**, select the pencil icon to edit the variable properties.
6. The **Variable properties** panel opens. Rename the variable `TimeOffType`. Close the **Variable properties** panel and you see the changes reflected on the authoring canvas.

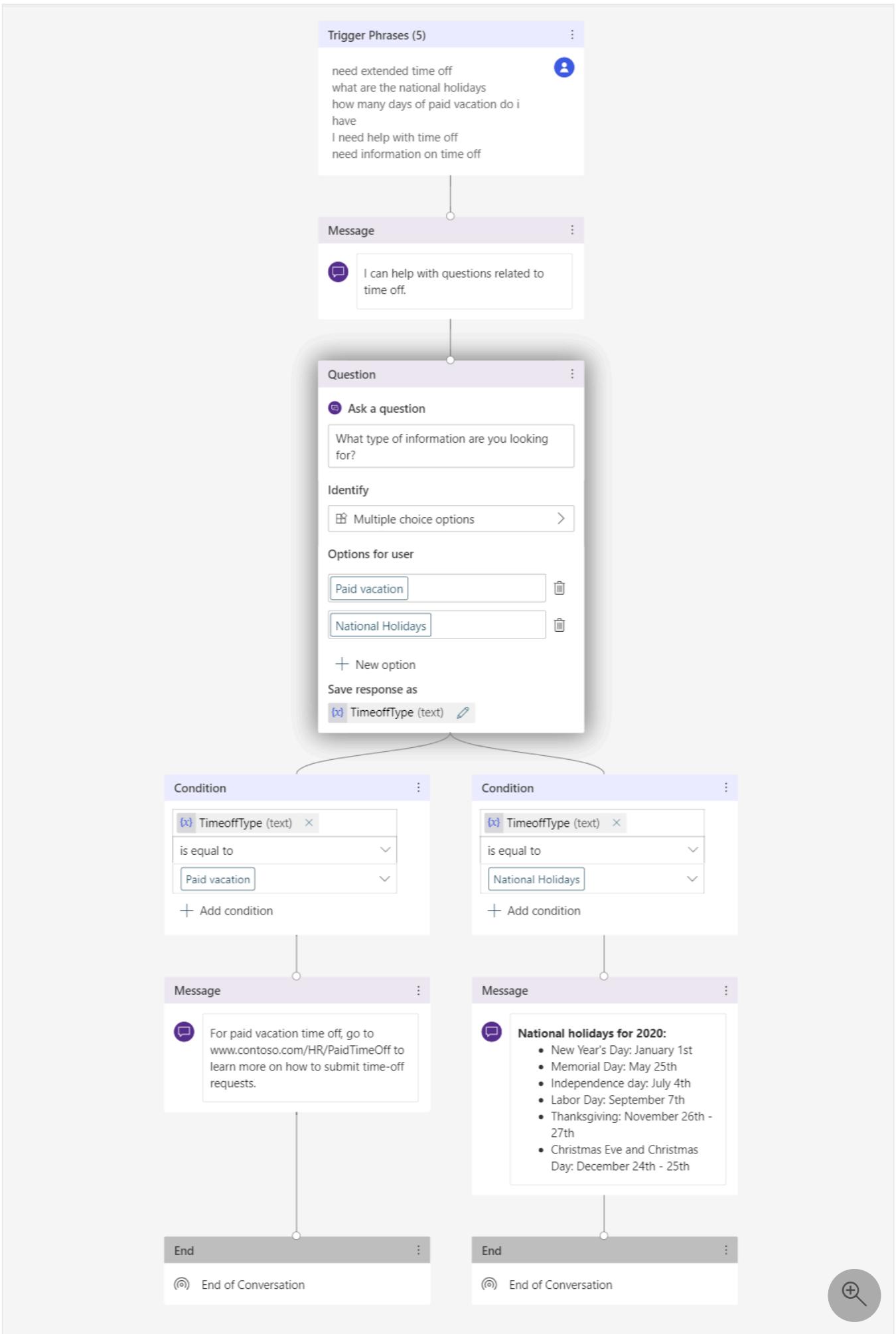
The screenshot shows the Microsoft Bot Framework Composer interface. On the left, a 'Question' node is open, displaying options to 'Ask a question' (with a placeholder 'What type of information are you looking for?'), 'Identify' (with a 'Multiple choice options' dropdown), 'Options for user' (listing 'Paid vacation' and 'National Holidays' with edit icons), and 'Save response as' (set to 'TimeoffType (text)'). Two parallel condition nodes branch from this question node. The first condition node has a single condition: 'TimeoffType (text) is equal to Paid vacation'. The second condition node also has a single condition: 'TimeoffType (text) is equal to National Holidays'. To the right, the 'Variable Properties' pane is open for the 'TimeoffType' variable, showing it is of type 'Text' and has a source 'Go to Source'. It also indicates 'Topic (limited scope)' usage and no external sources can set values. A search icon is at the bottom.

You can add different responses for each option. For *Paid vacation*, the chatbot can point employees to an internal HR website to look up paid time-off policies.

7. Add a message node for the *Paid vacation* branch with this message to the user: "For paid vacation time-off, go to www.contoso.com/HR/PaidTimeOff to submit time-off requests."
8. Add a node by selecting the **Add node** (+) icon to end the conversation with a survey. Select **End the conversation**, then **End with survey**. This survey is the [customer satisfaction survey](#) prebuilt in the chatbot for use in topics.
9. In the *National Holidays* path, add a message node with the following text:

```
text
National holidays for 2020:
- New Year's Day: January 1st
- Memorial Day: May 25th
- Independence day: July 4th
- Labor Day: September 7th
- Thanksgiving: November 26th - 27th
- Christmas Eve and Christmas Day: December 24th - 25th
```

10. End this path in the topic with a survey as well. Add a node by selecting the **Add node** (+) icon then **End the conversation** and **End with survey**.



Test your chatbot

Copilot Studio comes with a built-in test pane that lets you test the chatbot in real time.

1. To open the test panel, select **Test bot** on the toolbar.
2. Enter "I need time off information."

Even though *I need time off information* doesn't match any trigger phrases exactly, Copilot Studio processes the user's input and triggers the employee time off topic.

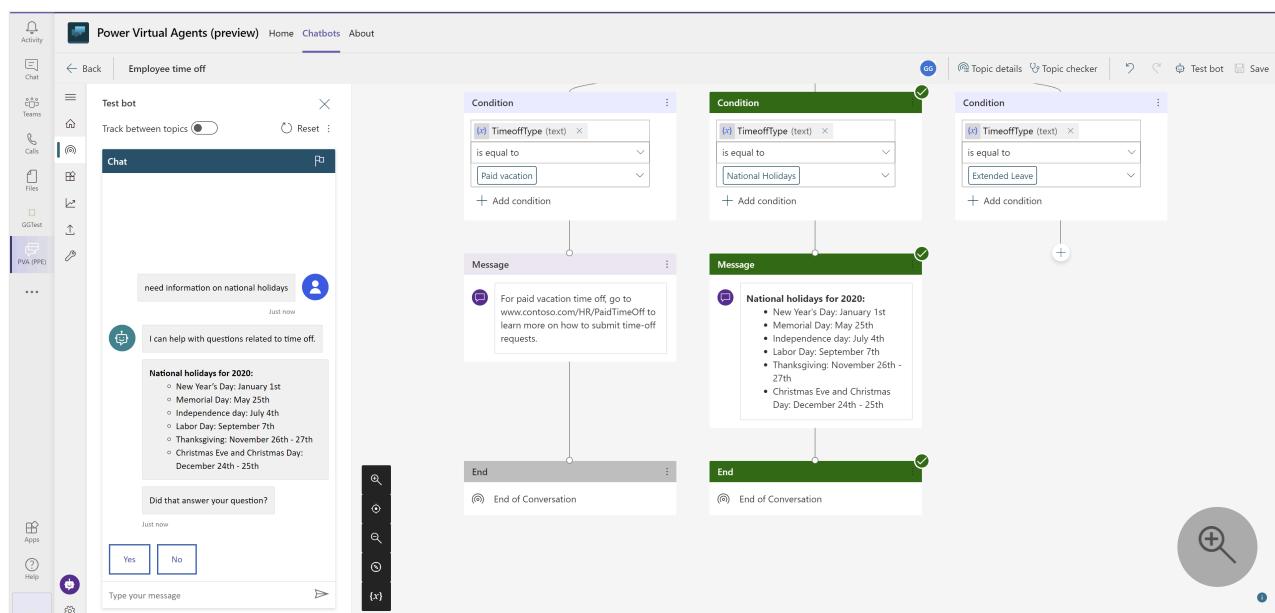
As you chat with the chatbot, you see where it is in the conversation in real time by following the green highlights.

3. Select **Paid vacation**.

The topic branches off based on the user's response. You can open the **TimeOffType** variable to see the value of **Paid time-off** by the user. However, what happens if the user is looking for national holidays in their inquiry? The user might feel annoyed if the chatbot asks them to repeat themselves. Let's give it a try.

4. At the top of the test pane, select the **Reset** icon.
5. Enter "Need information on national holidays."

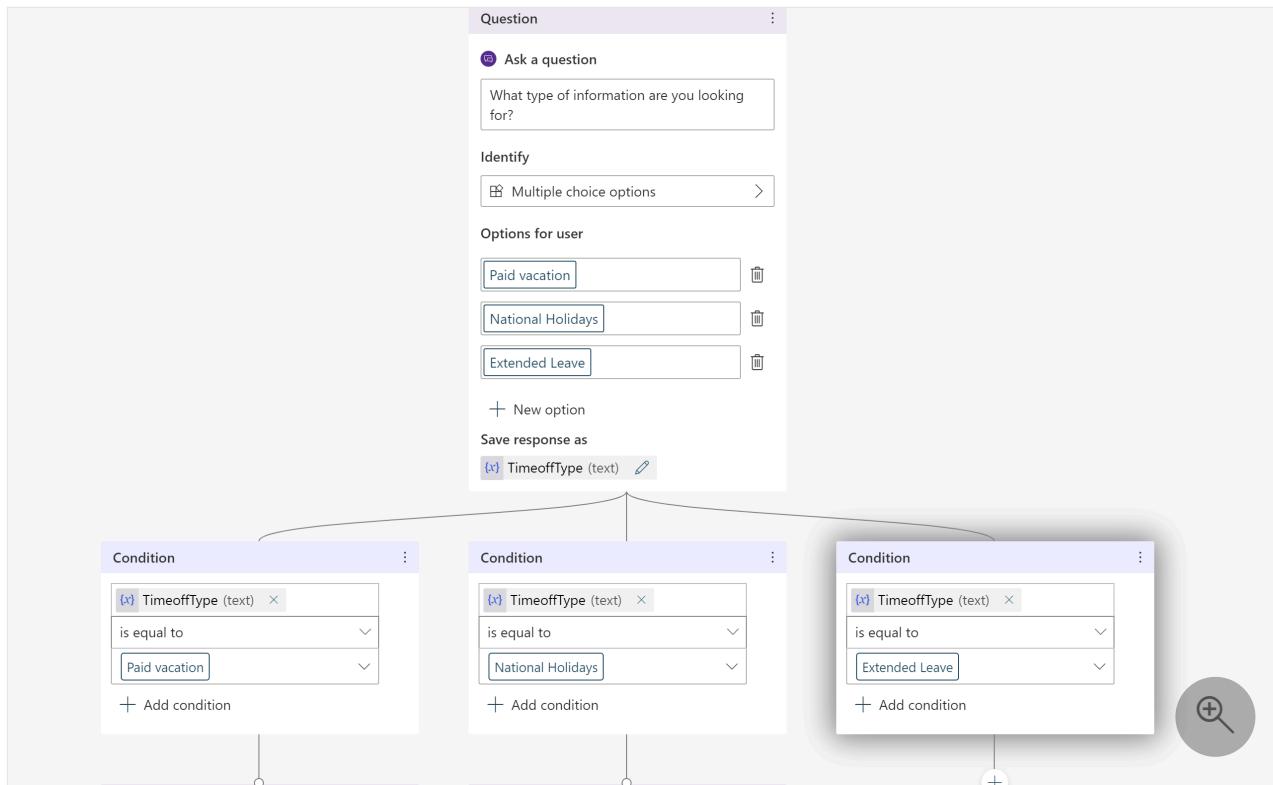
The chatbot bypasses its response question on vacation type and directly provides information on national holidays. This behavior happens because Copilot Studio supports **entity extraction** and can determine that the user asks about national holidays.



Enhance the topic to handle complex queries by escalating to HR experts

Now that you performed the basics, you can add a more complex "Extended leave" option to the topic.

1. Return to the topic you created.
2. In the **Question** node, add an option named "Extended leave."



Since each employee's personal situation is different, questions about extended leave require a discussion with an HR expert. You can build the chatbot to connect HR experts with your employees.

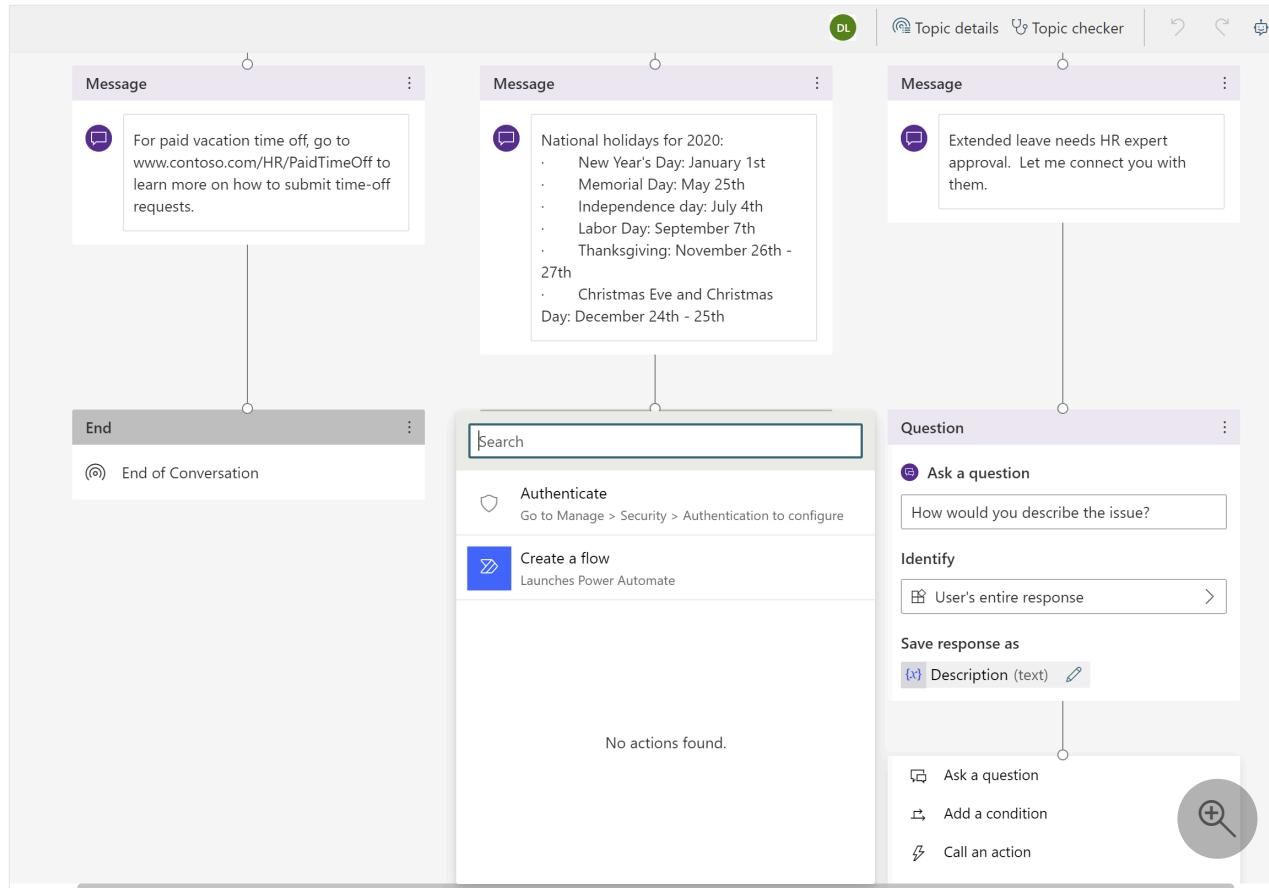
3. Add a message node that says "Extended leave needs HR expert approval. Let me connect you with them."

Copilot Studio chatbots can take actions by [adding actions into a topic](#). In this example, you add a previous basic action that notifies an HR expert. This basic action sends an adaptive card to the HR expert channel in Teams to let experts know a user needs them to reach out.

You can create the basic action to send a message to a Teams channel and then call it with the appropriate input.

4. Add a question node asking for a description for the issue and add the text "How would you describe the issue?"

5. Save the description in a variable named **Description**.



6. Add a node under the question and select **Call an action**.

7. Select **Basic action** and then **create new flow**, which launches Power Automate.

8. Choose the template **Send a message to a Teams channel**, then select **Continue**.

The flow template takes in two inputs from the chatbot, the **Subject** (title for the message) and **Message text** (actual message) to post to the channel.

a. Under the **Post a message (v3)** flow definition, select the **Team ID** and the **Team channel ID** where the message should be posted.

b. (Optional) Add text to the message, in addition to the description entered by the user, and complete the **Message** field.

c. Give the flow a name and save it.

d. Close the flow authoring canvas.

You return to the authoring canvas. You can now add an action and call this newly created flow.

9. Pass in the variable you created earlier called *TimeOffType* for the **Subject** and the **Description** gathered in the chat. When the flow posts a message on the Expert channel, it shows **Extended leave** as the title and the message defined in the flow.

Tip

Copilot Studio is directly integrated with Teams, so the chatbot knows which user from Teams it's chatting with. This feature allows chatbot authors to build personal and customized chatbot responses.

To use the name of the person chatting with the chatbot, there's a predefined variable named *bot.UserDisplayName* that can be used [like any other variable](#).

You can let the employee know their request notified the experts, who will reach out to them soon.

10. Add a message node with the message *We notified the expert. They'll reach out shortly.*

11. End the conversation with a survey.

12. Save the topic.

You can add more topics so that HR Support Bot can answer more questions from Contoso employees.

Tip

Always test your chatbot to ensure it's working as expected for the best employee experience. You see a message posted to the Teams channel when the flow executes in the test canvas.

Publish your chatbot

Publishing the chatbot makes the latest updates available to users.

In the navigation menu, select **Publish**. Select **Publish** again.

You need to do a few more actions to make the chatbot available to employees.

Tip

If you're already in a conversation with the chatbot, you can type "start over" after publishing your changes. This restarts the conversation with the latest content.

Make your chatbot available to other employees

1. On the **Publish** page, select the option to open your chatbot. You can test your chatbot before sharing it.
2. Select the option to make your chatbot available to others. The configuration panel for the **Teams + Microsoft 365** channels appears.
 - a. [Edit details](#) for your chatbot.
 - b. [Share your chatbot with your team](#).

Viewing the chatbot in the app store is a workaround to test it, since admin approval isn't required and avoids any spam to the admin.

- c. Once you're fully satisfied, [share the chatbot with your organization](#), which requires admin approval.

Employees can now find your published chatbot in the Teams app store.

Edit chatbot details

In the customization pane, you can edit the chatbot icon that displays in Teams and change the short and long descriptions for your chatbot. These settings are prepopulated with starter content, but can be customized.

Share your chatbot with your team

1. Select **Add to <your team>**. A panel appears where you can configure how your chatbot greets your team members.
2. Select **Add**. A message indicates that the chatbot is available to the team. You can close the configuration panel.

Share your chatbot with your organization

Sharing your chatbot with the entire organization makes it appear [in the Built for your org section of the Teams app store](#). Access requires admin approval.

1. Select **Submit for admin approval**.
2. Confirm the chatbot icon, short description, and long description are correct.
3. Confirm you want to share the chatbot with everyone in the organization.
4. Submit for admin approval. This part of the process can't be completed right away and requires follow up with the admin for approval.

Find your chatbot in the Teams app store

An employee named Melissa can find the chatbot you built in the Teams app store.

Depending on how you shared your chatbot, Melissa can find it either in the **Copilot extensions** section or the **Built for your org** section of the Teams app store. For a chatbot to appear in the **Built for your org** section, it must be approved by an admin first.

Melissa selects the chatbot app and selects **Add for me**. The chatbot appears in the Teams side bar.

The **Greeting** topic automatically welcomes Melissa.

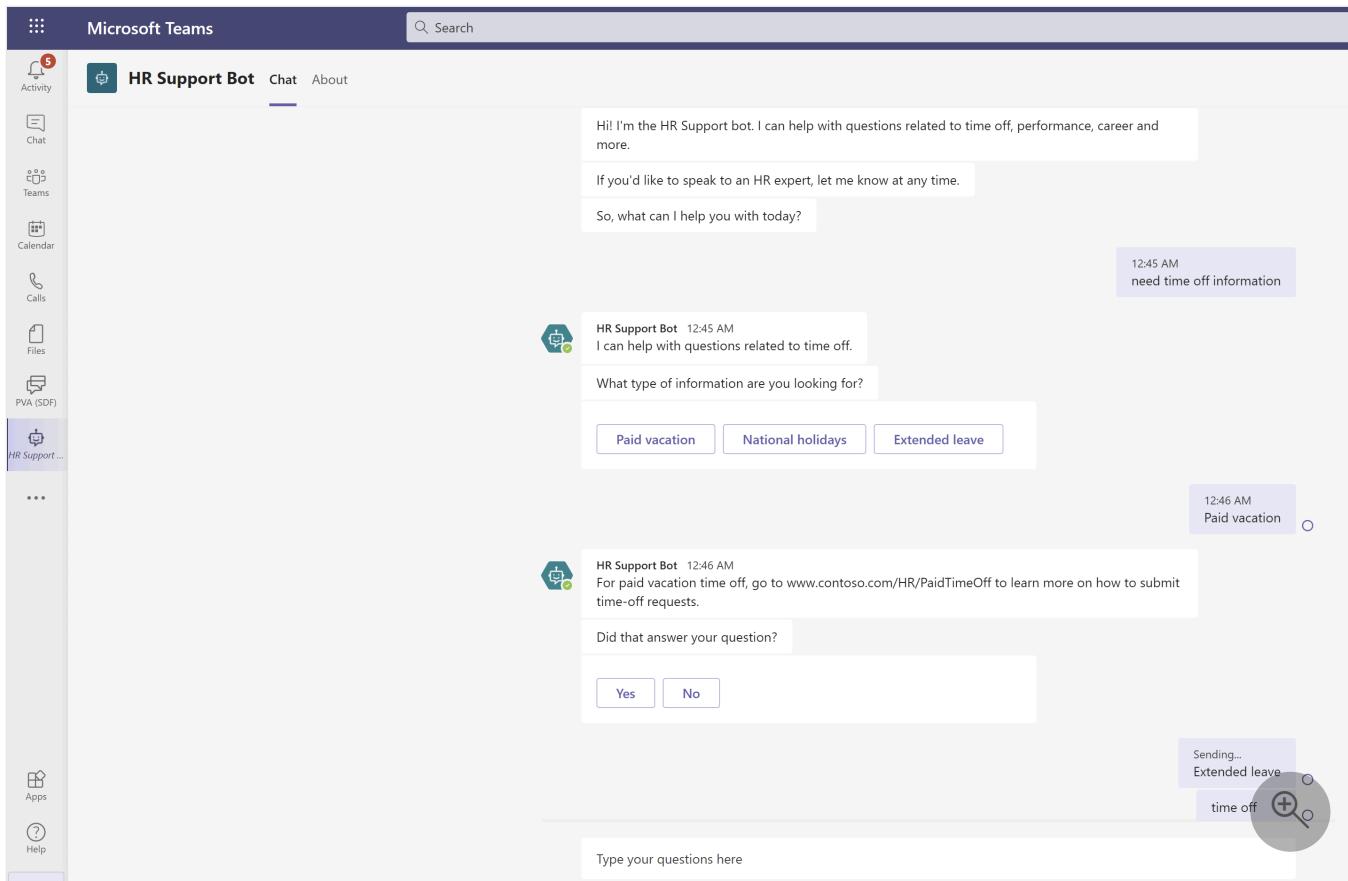
Tip

You can [edit the Greeting topic](#) to have a custom welcome message.

Employee chats with the chatbot to ask about time-off

Melissa needs time-off in July and August for personal reasons and enters *Need time-off information*. Melissa selects **Paid vacation** and a website appears where she can learn more about paid time-off.

The chatbot then gathers customer satisfaction data and ends the conversation.



Melissa reviews the HR site for paid time-off and understands she needs an extended leave. Melissa returns to the chatbot and types *I need extended leave*.

The chatbot lets Melissa know that an HR expert is needed for the extended leave request and asks Melissa for a description of the issue.

The chatbot notifies an expert, who will reach out to Melissa shortly.

Return to the author's view and go to the HR team's experts channel.

You see the chatbot ran the flow you added earlier and posted Melissa's request to the channel.

Microsoft Teams

Activity

Chat

Teams

Calendar

Calls

Files

PVA (SDF)

HR Support...

...

Search

HR

Your teams

HE HR Experts

General

Take Microsoft Teams with you on the go. Check out our mobile apps.

General Posts Files Wiki +

Welcome to the team!

Here are some things to get going...

Add more people Create more channels Open the

David Longmuir 12:50 AM
Extended leave
Get approval for 6 month leave
Reply

This screenshot shows the Microsoft Teams interface. On the left, there's a sidebar with various icons for Activity, Chat, Teams, Calendar, Calls, Files, PVA (SDF), HR Support, and an ellipsis. The main area is titled 'HR' and shows 'Your teams' with 'HE HR Experts' and 'General' listed. A purple bar at the top right says 'Take Microsoft Teams with you on the go. Check out our mobile apps.' Below it, the 'General' channel is selected, showing a welcome message and three circular icons representing different team topics. A post from 'David Longmuir' at 12:50 AM is visible, asking for approval for a 6-month leave. There are buttons to 'Add more people', 'Create more channels', and 'Open the'. The bottom right corner has a search icon.

Explore lesson topics in classic chatbots

07/07/2025

⚠ Note

This legacy article only applies to classic chatbots created in the Microsoft Copilot Studio app for Microsoft Teams.

The Copilot Studio app for Microsoft Teams includes sample topics with every new agent. These samples range from simple to complex scenarios that use [conditional branching](#), [variables](#), and [custom entities](#).

The topics are functional but aren't designed for production use.

Available sample topics

Each lesson topic is designed to teach you how to use the authoring canvas to create basic and advanced bot conversations.

You should start with [Lesson 1 - A simple topic](#) and work through each lesson in order.

Lesson 1 - A simple topic

This topic responds to questions about store hours.

Lesson 2 - A simple topic with a condition and variable

This topic shows you how to create a question that asks customers which store they're interested in and keeps the response in a [variable](#). In this example, `pva_StoreLocation` is the variable that stores the customer's response when you ask for their preferred store location. The condition uses this variable to determine which store's hours to return.

Lesson 3 - A topic with a condition, variables, and a prebuilt entity

This topic asks customers which state they would like their order to be shipped. The chatbot uses the prebuilt [State entity](#) to recognize the name of a US state in the customer's response, and stores it in the `pva_State` variable.

A condition uses the **pva_State** variable to determine which message to send to the customer. Another condition uses the customer's response to a multiple-choice question, stored in the variable **pva_Item**, to determine which product to order.

When you use entities in your topic, the bot can identify key information from what the user types and automatically stores that information in your variables.

For example, if you type "I want to buy a red car," the bot doesn't need to ask which color car, because the bot recognizes the **Color** entity in what you typed. The bot then skips the question where you asked for color.

Lesson 4 - A topic with a condition, variables, and a custom entity

This topic has a conditional branch, a variable, and a [custom entity](#).

You can see the chatbot bypass a follow-up question. For example, try testing with the phrase "I want a business laptop" in the testing pane.

Use sample topics to understand how topics work

1. Go to the **Topics** page for your classic chatbot.
2. Select **Lesson 1 - A simple topic**.
3. Select **Details**, and review the title and description that appear on the **Details** panel.

The screenshot shows the Microsoft Bot Framework Composer interface. At the top, there are tabs: Details (which is highlighted with a red box), Trigger phrases, Variables, and Analytics. Below the tabs, the title 'Lesson 1 - A simple topic' is displayed. On the left, under 'Trigger Phrases (4)', there are four entries: 'When are you closed', 'When are you open', 'Store hours', and 'Daily open hours'. A vertical line connects these trigger phrases to a 'Message' block below. The 'Message' block contains the text 'I'm happy to help with store hours.' On the right, a modal window titled 'Details' is open, also highlighted with a red box. It contains three fields: 'Name *' with the value 'Lesson 1 - A simple topic', 'Display name' with the placeholder 'Enter a display name', and 'Description' with the text 'Lesson 1 - This lesson will show you how you can create a simple topic with one conditional branch.'

4. Select **Trigger phrases**, and review the trigger phrases for this topic.

The screenshot shows the Microsoft Bot Framework Composer interface. At the top, there are tabs: Details, Trigger phrases (which is highlighted with a red box), Variables, and Analytics. Below the tabs, the title "Lesson 1 - A simple topic" is displayed. On the left, there are two vertical panels: "Trigger" and "Message". The "Trigger" panel lists four trigger phrases: "When", "When", "Store hours", and "Daily open hours". The "Message" panel shows a message icon and a text input field with placeholder text "Enter text". To the right of these panels, under the heading "Trigger phrases (4)", is a descriptive text: "Trigger phrases teach the bot language understanding helping learning, the bot needs 5-10 seconds to learn each phrase". Below this is a "Show writing tips" link and an "Add phrases" section with an "Enter text" input field. A note says "To add phrases in bulk, paste in line-separated text". The four trigger phrases listed are highlighted with a red box.

Trigger phrases (4)

Trigger phrases teach the bot language understanding helping learning, the bot needs 5-10 seconds to learn each phrase

Show writing tips

Add phrases

Enter text

To add phrases in bulk, paste in line-separated text

Trigger phrases (4)

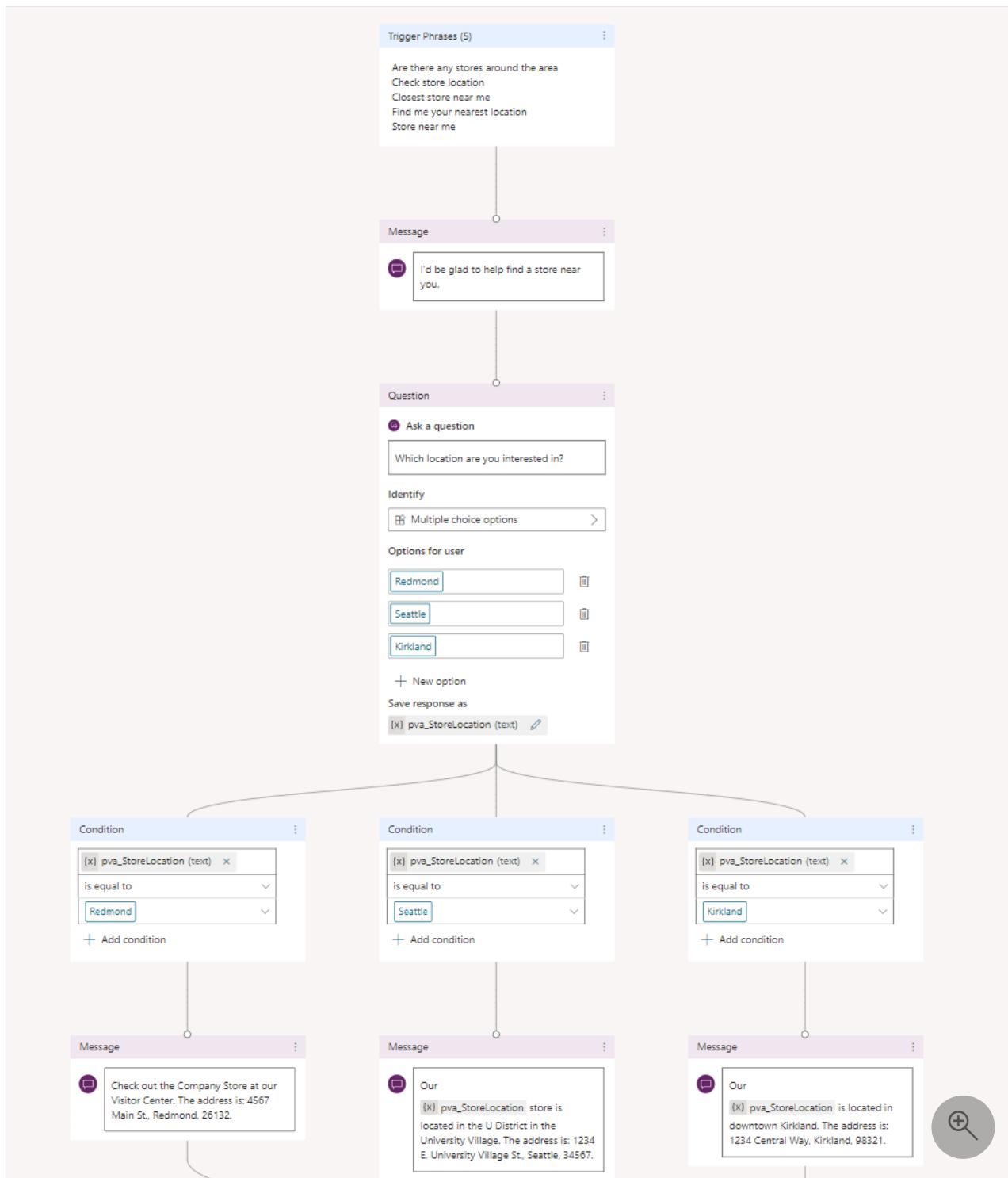
When are you closed

When are you open

Store hours

Daily open hours

5. To review the conversation flow for this classic chatbot, select each topic. The flow includes expected user responses, decision points, and entity references. The following sample is from **Lesson 2 - A simple topic with a condition and variable**.



Upgrade to Copilot Studio unified authoring

Article • 05/21/2025

The latest Copilot Studio release is now here and generally available (GA). This release brings a major update to the Copilot Studio product. The release introduces a significant number of updates and improvements, empowering organizations to build conversational experiences that are valuable and relevant to their customers and internal users.

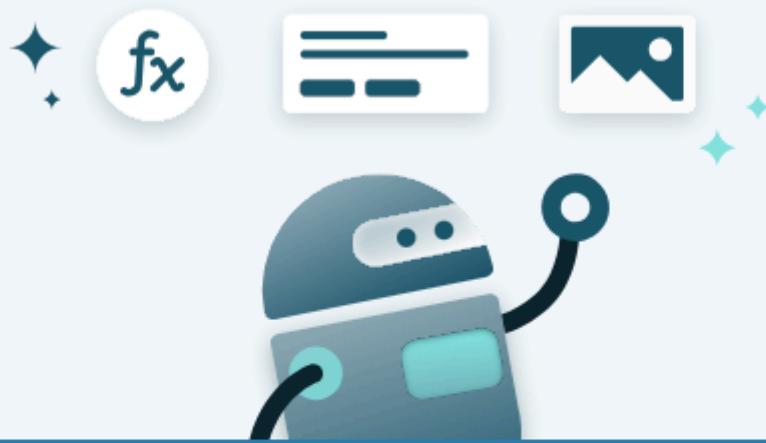
With the release of our latest version of Copilot Studio, now is the time to consider migrating from your classic Power Virtual Agents experience to the latest version of Copilot Studio. This guide aims to help readers understand some of the significant improvements within this release, and areas to consider when it comes to your organization migrating your classic Power Virtual Agents experience.

Upgrading to Copilot Studio unified authoring

If you already created one or more classic bots using Power Virtual Agents, you might be wondering what the process is for upgrading to the latest release of Copilot Studio.

Cloning an existing Power Virtual Agents classic bot to a Copilot Studio agent

To help the upgrade experience, at launch, a clone feature is available for classic bots built using Power Virtual Agents.



New features in Microsoft Copilot Studio

Explore the new features in Microsoft Copilot Studio by converting a copy of this chatbot. Add images and video in your messages, insert logic using equations, and build quickly with intelligent authoring and answers created with generative AI. You'll be able to convert a copy of Copilot 1.

 Copy this chatbot

 Learn more

This feature clones the selected Power Virtual Agents classic bot, and creates a new agent that uses the latest unified authoring version. There are some important considerations to be aware of in this cloning process, such as the following considerations:

- The cloning process is only available in the same language from the source classic bot to target the agent experience.
- The cloning process only clones topics built using the Copilot Studio web canvas.
- The cloning process clones entities and associated synonyms and custom entities.
- An agent administrator needs to reconfigure authorization, channels, and security settings in the cloned unified authoring agent.
- Any Bot Framework Skills from the Power Virtual Agents classic bot would need to be reconnected with the cloned unified authoring agent.
- The connection to Power Automate Flows, built using the Copilot Studio web canvas, are cloned and associated within the topic within the cloned unified authoring topic. However, we strongly recommend that the connections are tested.
- Consideration should be given for any custom canvas and other custom architecture components you might have previously connected to your Power Virtual Agents classic bot. These custom components would need to be reconnected or reconfigured to your new cloned unified authoring agent and tested. In addition, wider consideration should be given to your entire architecture. This consideration includes any effect within that architecture that might have been updated since the updating of your classic bot (and its bot ID).

- Any automated deployment pipelines would need to be updated if utilizing the Power Virtual Agents classic bot ID or any reference to the classic bot.
- Any custom reporting to your new agent deployment should be updated. Examples include if the custom data pipeline solution is being used or alternative reporting solutions.

 **Important**

This list isn't exhaustive. Ensure you complete the required testing for your agent before deploying it to production, even if you're using the clone feature to migrate your agent between versions.

By migrating your classic bot from Power Virtual Agents to Copilot Studio unified authoring, you can access the latest features within Copilot Studio unified authoring. These Copilot Studio features include generative AI-enabled features, variable management and Power Fx, code view, events, and so much more. You can take the opportunity to add extra value to your agent and, as a result, an improved experience for your customers.

 **Note**

In the future there might be other tools available to assist with migration of classic bots and we would appreciate your feedback on the topic of feature migration. [Please submit feature requests ↗](#).

To migrate your classic bot, go to [Migrating your Power Virtual Agents classic bots](#).

What's new or updated in the latest Copilot Studio

The following overview isn't an exhaustive list of new features or updates in Copilot Studio unified authoring. It aims to highlight the significant new capabilities now available within the authoring experience of Copilot Studio, or significant changes that should be brought to the attention of readers.

 **Important**

To review more detail about the core capabilities and features of Copilot Studio, see [Quickstart: Create and deploy an agent](#)

Creating a new agent

When creating an agent, the new creation process lets you name the agent, specify the agent's language, and also get started with boosted conversations.

Additionally, you can configure the icon, and select the default solution and default schema name for new topics. When you select the schema name, it directly utilizes the prefix from the selected solution.

User interface changes

Many user interface updates were made with the aim to make the agent maker and administration experience easier and more intuitive. The updates begin with the agent creation experience, but don't end there. The following improvements also include:

- Topic creation: Copilot makers can create a new topic manually or use Copilot Studio.
- Trigger tags: You now see all types of triggers within the topic view, and different triggers due to the events feature, such as *Message Received*.
- Connectors within a topic: Previously the connectors between nodes were curved.
- Topic user interface: More nodes are now available on the creation menu within the authoring canvas, new productivity panel, and extended menu to include agent and variables.
- Variable watch window: Specific call out where the variable watch window is no longer in the mini map, and instead on the variable panel at the top area of the authoring canvas.
- Flyout menus: Copilot Studio now utilizes the full window within the authoring experience. Other properties, based on selections within the authoring canvas, were traditionally held within a flyout panel on the side of the screen.

System topics

If you're new and inexperienced to agent building with Copilot Studio, we don't recommend that [system topics](#) are changed. If you're familiar with Copilot Studio, you might notice that new and updated system topics are now available in the latest unified authoring release.

Conversation Start: Previously, classic bot makers had to customize the classic bot with extra code to automatically begin a conversation with Copilot Studio. This code is no longer required, and the Conversation Start system topic begins the conversation automatically and is turned on by default. You can see the conversation automatically begin working by using the **Test your agent** panel within Copilot Studio when your agent automatically starts using the Conversation Start topic data.

Multiple Topics Matched: For conversational experiences, when a user asks a question, there can be more than one topic matched with high confidence. The multiple matches cause a disambiguation experience to occur, normally referred to as a *Did You Mean* experience within

Copilot Studio. The disambiguation experience is when the agent asks the user, "Did you mean X, or did you mean Y?" This experience helps to disambiguate between two or more high ranking topics matched by the user's original question. In Copilot Studio, you can edit the disambiguation experience using the **Multiple Topics Matched** system topic, for example, by not displaying specific topics when disambiguation is triggered.

Pass entire objects back to Copilot Studio using Power Automate

Copilot Studio is seamlessly integrated with Power Automate to allow agent makers to connect to many different connectors to help retrieve and access data in other systems. In the previous version of Copilot Studio, only text, number, or Boolean (yes/no) type objects could be passed between Copilot Studio and Power Automate.

In the Copilot Studio unified authoring release, you can now send an entire object. Whether that object is a Dataverse record, or a collection of Dataverse records, the Object is sent back using the text output of the Copilot Studio return statement.

With these new authoring capabilities in Copilot Studio, an agent author/maker can then retrieve that object from Power Automate. Then, the author/maker uses the parse node feature and the new variable features to extract the information from that object. Once retrieved, they format it as required (for example, within a custom Adaptive Card).

Variables, variable watch window improvements, and testing

In the latest release, variables were updated. These updates include a selection of variable nodes to create new variables and update them, and utilize system data and variables. Also, agent makers can include Power Fx formulas to manipulate data and perform calculations within the runtime of Copilot Studio.

When testing, it's critical to be able to test variables within the topic process flow when creating conversational experiences. To test, the variable watch window was previously in the topic mini-map where an agent maker/author could use the testing panel within Copilot Studio to test topics and the data between them. Testing variables are now available in the variable panel, on the side panel within the authoring canvas, under **Test**. Together with the testing panel window, agent makers/authors can now watch the variables and their data within this panel for both topic and global level variables. Additionally, as this panel has a large amount of space, this testing experience is improved for topics that have a large number of variables within them.

For more information, see [Create expressions using Power Fx](#).

Write your agent in YAML

Fusion teams are enabled in Copilot Studio. These teams have the capabilities to author conversational experiences that use both the graphical user interface (GUI) authoring tool and code view. The GUI uses rich responses, adaptive cards, and more for use by everyone. Developers can switch to the code view within a topic, in real-time, to directly build or edit the YAML referenced by the interface.

Events

Events introduce extra capabilities within the Copilot Studio web interface to create and manage the conversational experience. Traditionally, trigger phrases are used to detect the intent from a spoken sentence or word and the best matched topic, if found. Events are an alternative way to manage events that are either sent to or received from the user or alternative systems.

Generative AI enabled features

Boosted conversations: Copilot makers/authors are empowered to create a useful agent in seconds by connecting it to a datastore, such as a public facing website, allowing the agent to be able to generate answers to questions using that datastore. You can use this feature at agent creation or from the **Knowledge** page for your agent.

Topic creation with Copilot: Creating agents with AI is simplified by using Copilot in Copilot Studio. In the latest release, agent makers can now utilize Copilot on the topic screen when creating a new topic, entering a name and an initial prompt to build the topic directly within Copilot Studio. Copilot makers/authors can also use Copilot, using the Copilot button within the topic, to open the Copilot panel on the side of the authoring screen to update the topic using natural language, including select specific nodes.

Application Insights integration by default

By default, Application Insights is now integrated within Copilot Studio. Application Insights allows administrators to monitor the performance of applications. Copilot Studio allows administrators to both proactively understand how the application is performing, and reactively to review root cause analysis and determine the cause of an incident. Within Copilot Studio, administrators can connect their agent to an instance of Application Insights within the settings area using a connection string. Then the administrator specifies whether to log incoming and outgoing messages, log sensitive activity properties, and node actions, by default. Additionally, agent authors can log custom events with properties using the new **Log custom telemetry event** node.

How to get started with structured learning for Copilot Studio unified authoring

Register for our *Copilot Studio in a Day* training on Microsoft Learn. The Copilot Studio in a Day training also contains specific material for instructors. Check out the [Microsoft Events page](#).

Migrating your Power Virtual Agents classic bot

You can turn an existing, classic chatbot into an agent to use the latest features of Copilot Studio.

Eligibility

Classic chatbots that use any of these features aren't eligible to be cloned:

- Languages other than English
- Handoff to Omnichannel for Customer Service
- Knowledge Management extensions

Clone a classic chatbot

1. Open your classic chatbot in Copilot Studio.

2. On the **Overview** page, select **Copy this chatbot**.

A **Chatbots** window opens, with these options: **Copy and convert this bot** or **Try out the new studio experience**.

3. Select **Copy and convert this bot** and modify the default name for your copy as desired.

4. Select **Create**.

Important

If your [bot isn't eligible for cloning](#), Copilot Studio prompts you to create a new agent instead.

5. Wait for Copilot Studio to create your new agent. When it's ready, select **Go to your bot** to open it in Copilot Studio.

Testing guidance in Copilot Studio unified authoring

With any application, it's vital that companies ensure there's enough time within the migration process to complete adequate testing. Adequate testing is dependent on the type of agent and conversational experience a company builds, together with any other or connected infrastructure and technology. As you consider migrating to unified authoring, ensure the testing framework and processes in your organization allow for agent testing, even for simple agents.

Testing and deployment practices

Check out the following resources to learn more about business application technology deployments, and best practices around conversational projects using Copilot Studio:

- [Success By Design](#)
- [PowerCAT Microsoft Copilot Studio Playbook ↗](#)
- [Microsoft Copilot Studio Copilot Testing Framework by PowerCAT ↗](#)

Warning

Ensure you completed all your testing, and complied with your organizational policies, before deploying your migrated classic bot to production.

Working with Microsoft Support

Raising any issues to Microsoft should include the snapshot of the agent, found in the testing pane, and the metadata of the agent, found in the **Copilot details** section of the agent management area, under **Advanced**, such as *Environment ID*, *Tenant ID*, *Bot App ID*.

Related content

- [Find community help and support](#)

Find support and give feedback for Copilot Studio

09/24/2025

You can get support for Copilot Studio on the community forums, and features released for general availability are eligible for support through Microsoft Support. You can also give feedback on your Copilot Studio experience.

For training and certification preparation, see the [Create agents with Microsoft Copilot Studio](#) learning path that can help prepare you for [Exam PL-200: Microsoft Power Platform Functional Consultant](#).

Microsoft Copilot Studio community forums

Check community forums to find user solutions that might help you.

- [Ask the community ↗](#)

Visit the Copilot Studio community to get answers and tips directly from other Copilot Studio users.

- [Submit an idea ↗](#)

Do you have a great idea that makes Copilot Studio even better? We'd love to hear from you!

Microsoft support

If you're an administrator of your Copilot Studio tenant, you can use the Microsoft Power Platform admin center to request support from Microsoft.

1. Sign in to the [Power Platform admin center ↗](#).
2. On the side navigation, select **Support**, and then select **Support requests**. The **Support requests** page opens.
3. Select **Get support**.

For more information about help and support in the Power Platform admin center, see [Get Help + Support](#) at the Microsoft Power Platform admin documentation library.

Provide feedback

Find a problem, have an idea, or just want to give some kudos? You can submit feedback directly in Copilot Studio.

1. Select the **Help** icon  in the top navigation bar, then **Send feedback**.
2. Select the type of feedback you want to provide.
3. Add a description or explanation. You can also include a screenshot.

When you submit feedback, you might be contacted by Microsoft, so we can learn more. [Learn about Microsoft feedback for your organization](#).

Key concepts - Copilot Studio security and governance

08/28/2025

Copilot Studio follows a number of security and governance controls and processes, including geographic data residency, data loss prevention (DLP), multiple standards certifications, regulatory compliance, [environment routing](#), and regional customization. See the [Geographic data residency in Copilot Studio](#) article for information and details on how data is handled in Copilot Studio.

This article provides an overview of the security practices followed by Copilot Studio, a list of security and governance controls and features, and examples and suggestions for employing safety and security within Copilot Studio for your agent makers and users.

Security and governance controls

 Expand table

Control	Core scenario	Related content
Agent runtime protection status	Makers can see the security status of their agents from the Agents page.	Agent runtime protection status
Data policy controls	Admins can use data policies in the Power Platform admin center to govern the use and availability of Copilot Studio features and agent capabilities, including: <ul style="list-style-type: none">• Maker and user authentication• Knowledge sources• Actions, connectors, and skills• HTTP requests• Publication to channels• AppInsights• Triggers	Configure data policies for copilots
Makers audit logs in Microsoft Purview for admins	Admins have full visibility into maker audit logs in Microsoft Purview.	View audit logs
Audit logs in Microsoft Sentinel for admins	Admins can monitor and receive alerts on agent activities through Microsoft Sentinel.	View audit logs

Control	Core scenario	Related content
Run tools with user credentials	Agent makers can configure tools to use the user's credentials by default.	Use actions with custom copilots
Sensitivity label for Knowledge with SharePoint	Agent makers and users can see the highest sensitivity label applied to sources used in the agent's response and individual reference labels in the chat.	View sensitivity labels for Sharepoint data sources
User authentication with certificates	Admins and makers can configure agents to use Entra ID manual authentication with certificate provider.	Configure user authentication
Maker security warning	Makers can see security alerts for their agent before publishing it when security and governance default configurations are modified.	Automatic security scan in Copilot Studio
Environment routing	Admins can configure environment routing to provide their makers a safe space to build agents.	Work with Power Platform environments
Maker welcome message	Admins can configure a maker welcome message to inform makers about important privacy and compliance requirements.	Work with Power Platform environments
Autonomous agents governance with data policies	Admins can manage agent capabilities with triggers using data policies, ensuring protection against data exfiltration and other risks.	Data policy example - Block event triggers in agents
CMK	Admins can enable customer-managed encryption keys (CMK) for their Copilot Studio environments.	Configure customer-managed encryption keys

Security Development Lifecycle

Copilot Studio follows the Security Development Lifecycle (SDL). The SDL is a set of strict practices that support security assurance and compliance requirements. Learn more at [Microsoft Security Development Lifecycle Practices](#).

Data processing and license agreements

The Copilot Studio service is governed by your commercial license agreements, including the [Microsoft Product Terms](#) and the [Data Protection Addendum](#). For the location of data processing, refer to the [geographical availability documentation](#).

Compliance with standards and practices

The [Microsoft Trust Center](#) is the primary resource for Power Platform compliance information.

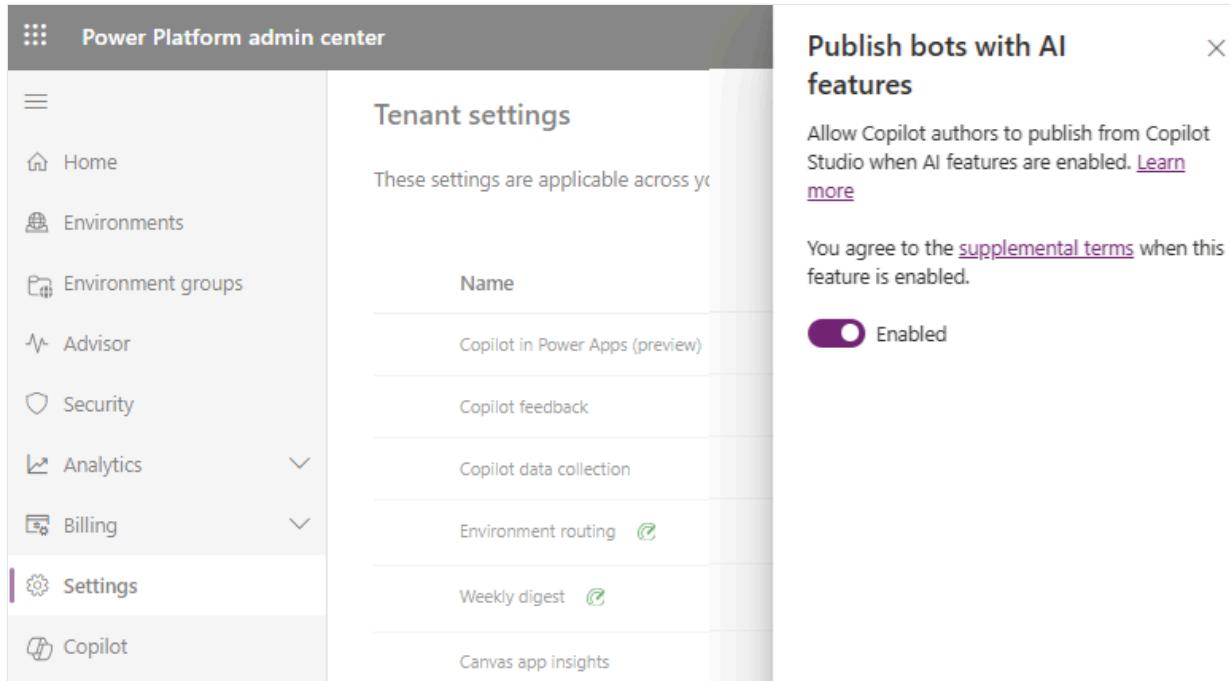
Learn more at [Copilot Studio compliance offerings](#).

Data loss prevention and governance

Copilot Studio has an extensive set of [Data Loss Prevention features](#) to help you manage the security of your data, along with [Power Platform policies](#).

Additionally, to further govern and secure Copilot Studio using generative AI features in your organization, you can:

- Disable agent publishing:
 - Your admin can use the Power Platform admin center to turn off the ability to publish agents that use generative AI features for your tenant.



The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with options like Home, Environments, Environment groups, Advisor, Security, Analytics, Billing, Settings, and Copilot. The Copilot option is currently selected. In the main area, under 'Tenant settings', it says 'These settings are applicable across your organization'. To the right, a modal window titled 'Publish bots with AI features' is open. It contains a description: 'Allow Copilot authors to publish from Copilot Studio when AI features are enabled.' with a link to 'Learn more'. Below this is a note: 'You agree to the [supplemental terms](#) when this feature is enabled.' A toggle switch is shown in the 'Enabled' position. The modal has a close button in the top right corner.

- [Disable data movement across geographic locations](#) for Copilot Studio generative AI features outside the United States.
- [Use the Microsoft 365 admin center to govern the conversational and AI actions and agents](#) that show in Microsoft 365 Copilot.

Finally, Copilot Studio supports securely accessing customer data using [Customer Lockbox](#).

Security FAQs for Copilot Studio

09/17/2025

These frequently asked questions (FAQs) are focused on security to help you get answers to accelerate your adoption and use of Copilot Studio in your organization.

Copilot Studio creates a single-tenant Microsoft Entra ID app registration on new agent creation. Existing agents continue to have multitenant Microsoft Entra ID app registration. Does a multitenant Microsoft Entra ID app registration for existing agents pose any security risk?

No, multitenant Microsoft Entra ID app registration doesn't pose any security risk.

Copilot Studio creates one app registration per custom agent to identify it and enable secure communication with the channels and skills it might use. This app registration doesn't access or expose any customer data, resources, or any agent information. Copilot Studio stores and manages the app registration securely and compliantly.

The app registration is for validating and securing calls from Copilot Studio to our Azure Bot Service resource. Copilot Studio creates and manages the registration for the customer application. This functionality exists in [Bot Framework and Azure Bot Service](#) since 2016.

- Any new agents created have single-tenant Microsoft Entra ID app registration. We're investigating moving existing agents to a single-tenant Microsoft Entra ID app registration in the future. If you'd like to be notified about this feature, [submit your vote for this feature here ↗](#).

Microsoft Power Platform has a rich ecosystem of connectors based on Microsoft Entra ID that allow authorized Microsoft Entra ID users to build compelling apps and flows establishing connections to the business data available through these data stores. Tenant isolation makes it easy

for administrators to ensure that these connectors can be harnessed in a safe and secure way within the tenant, while minimizing the risk of data exfiltration outside the tenant. Does Copilot studio support tenant isolation?

No, Copilot Studio doesn't support tenant isolation.

The default configuration in Power Platform with tenant isolation *Off* is to allow cross-tenant connections to be established seamlessly, if the user from tenant A establishing the connection to tenant B presents appropriate Microsoft Entra ID credentials.

If admins want to allow only a select set of tenants to establish connections to or from their tenant, they can turn tenant isolation *On*.

- Learn more about [cross tenant restrictions](#)

Copilot Studio creates service principal and certificates in the customer's Microsoft Entra ID tenant every time a custom copilot is created. What is the purpose of the service principal and certificates, and how are they governed?

To let custom copilots communicate with your data sources and services, Copilot Studio creates an application in your Microsoft Entra ID tenant, along with an associated service principal.

A service principal is an identity that represents an application and allows it to access resources in your tenant. For security and compliance reasons, Copilot Studio uses federated identity.

Can I disable Microsoft Copilot Studio agent creation in my organization?

You can't disable agent creation. Our guidance is to use data policies to disable anyone from chatting with that agent.

- For more information, see [Data policy example - Block channels to disable agent publish](#)
- If you need granular governance controls, [submit your feature request here](#)

I'm a Responsible AI Champion or Compliance Lead in my organization. Where can I find information on agent security and privacy related to data used by agents, data used by underlying models in agents, data protection practices, and moderation of content before a response is generated from Copilot?

Microsoft runs on trust. We're committed to security, privacy, and compliance in everything we do, and our approach to AI is no different.

- For more information about how we're protecting your data using industry-leading compliance, security, and privacy practices, see the [FAQ for Copilot data security and privacy for Dynamics 365 and Power Platform](#).

What auditing capabilities does Copilot Studio offer out of the box? How do I request other capabilities if needed?

You're an IT administrator in a Fortune 1000 organization. You want to govern custom copilots built by makers within your organization. To truly democratize Copilot Studio in your organization, you need detailed auditing capabilities. For example, ask the following questions:

- Who built a custom copilot?
- Are there any co-owners?
- Are public endpoints for generative answers available?
- Who configured or changed a copilot?

As an administrator, you can use out-of-the-box auditing capabilities of Copilot Studio to secure and govern your environments. Sign in to [Microsoft Purview compliance portal](#) and use filters to identify specific [audited events and activities](#).

If you're looking for more auditing events or fields, [submit your product idea](#).

How can I control generative AI capabilities in Copilot Studio?

You're a Power Platform admin in your organization. As part of your role, you need to selectively allow access to generative AI capabilities in Copilot Studio to makers across environments.

Copilot Studio provides granular and tenant-level governance controls for custom agents in your organization. Use the Power Platform admin center to:

- [Make publishing of custom agents available or unavailable](#), at the tenant level.
- [Control whether custom agents can use public URLs as knowledge sources](#), at the environment level.
- [Allow data movement across geographic locations for generative AI features](#), at the environment level.

If you're looking for more granular controls to govern custom agents, [submit your product idea](#).

How do I enforce access to knowledge sources across my environments?

As an environment administrator for Power Platform in your organization, you need to control which knowledge sources are available to makers when they build custom agents. For example, you want makers in your default environment to only upload files or use specific sites as knowledge sources when building custom agents.

Copilot Studio offers granular control to enable or disable specific knowledge sources using data policies in Power Platform admin center. You can configure data policies to control usage of SharePoint, public websites, or documents as knowledge sources. You can then apply these policies at the environment level or tenant level.

Does Copilot Studio offer data at rest encryption?

Your conversations might contain sensitive information and you would like to protect such information by encrypting it using customer-managed keys (CMK).

Copilot Studio lets you [turn on CMK](#). For more information, see [Manage your customer-managed encryption key](#). When CMK is turned on for the Copilot Studio environment, all Copilot Studio data is encrypted using the customer's key. The customer can cycle keys or turn off CMK as needed.

How does Copilot Studio ensure responses from confidential sites aren't visible to people who aren't allowed to see such information?

You can configure custom agents to access multiple internal sites, some of which contain confidential information that only certain authenticated users can access.

Copilot Studio is secure by default. The system tailors its responses based on who is speaking to it, and the permissions they have. Copilot Studio supports sensitivity labels to prevent oversharing. It also supports data loss prevention endpoint filtering for SharePoint knowledge sources.

Agent runtime protection status

06/05/2025

On the **Agents** page of Copilot Studio, makers can determine in a single agent-level status if published agents comply with security and governance standards and verify that an agent's threat detection is active. For more information, see [Key concepts - Copilot Studio security and governance](#).

The security status of published agents appears in the **Protection status** column on the **Agents** page. By displaying the protection status for each agent as one of **Protected**, **Needs review**, or **Unknown**, makers can determine quickly and with confidence when threat detection is active and their agents are protected.

The screenshot shows the 'Agents' page in Copilot Studio. On the left is a sidebar with icons for Home, Create, Agents (which is selected and highlighted with a red box), Flows, Tools, and an ellipsis. The main area has a header with 'Agents', 'New agent', 'Import agent', and a search bar. Below is a table with columns: Name, Type, Last modified, Last published, Owner, Protection status, and Engaged sessions. Four agents are listed:

Name	Type	Last modified	Last published	Owner	Protection status	Engaged sessions
Safe Travels 1	Agent	# Microsoft Copilo... 4 days ago		[REDACTED]	Protected	0
Agent 1	Agent	# Microsoft Copilo... Never		[REDACTED]	--	--
CONNECTS-wordsmith	Agent	# Microsoft Copilo... 27 days ago		[REDACTED]	Protected	0
Weather caster	Agent	# Microsoft Copilo... 28 days ago		[REDACTED]	Protected	0

A circular button with a plus sign and a magnifying glass is located in the bottom right corner of the table area.

The possible values a published agent might have under **Protection status** are **Protected**, **Needs review**, and **Unknown**.

When the agent-level status of an agent is **Protected**, no immediate action is required (based on detected signals). When the status is **Protected**, a green shield is displayed to visually emphasize that this agent is protected (**Protected**).

When the agent-level status is **Needs review**, either the agent's policies are violated or authentication is inadequate. Regardless of the status, makers can drill down to see an agent-level summary dialog with more information about your agent's security broken down into categories.

Select the status of your agent in the **Protection status** column to see an agent-level summary dialog of your agent's protection status.

Protection status summary

On the protection status summary dialog, the protection profile of your agent is broken into three categories: *Authentication*, *Policies*, and *Content moderation*. Each of these categories has a possible status of **Protected**, **Needs review**, or **Unknown**, as with the agent-level summary status on the **Agents** page.

Additionally, this dialog displays the number of blocked messages due to potential threats, policy violations, and violations of content moderation settings. Potential threats are either direct or indirect. For example, a potential threat might be in the form of a user overriding security settings with admin access. Alternatively, a potential threat might be a referenced knowledge source. All published agents automatically have threat detection enabled and display the **Active** label.

As with agent-level protection status, if either **Authentication** or **Policies** are in violation of accepted security standards, as defined by your agent's security setup, a **Needs review** label appears next to the category in the summary dialog. If either of these categories has a **Needs review** label, this label rolls up to the single agent-level protection status on the **Agents** page.

The screenshot shows the protection status summary dialog for an agent. At the top, there is a section titled "Needs review" with a shield icon, followed by the text "Contact your admin to take action". Below this, it says "3,684 messages blocked during the past 7 days, out of 5,959 sessions scanned." with a "Learn more" link. A "See details" button is highlighted with a red box. The main content area is divided into four sections: "Threat detection" (Active, 1,137 potential threats blocked), "Authentication" (Microsoft authentication required), "Policies" (Needs review, 1,344 messages blocked by org's policies), and "Content moderation" (8,460 messages blocked due to content moderation settings).

Category	Status	Details
Needs review	Active	Contact your admin to take action
Threat detection	Active	1,137 potential threats were blocked.
Authentication	Active	Microsoft authentication is required to interact with this agent.
Policies	Needs review	1,344 messages blocked by your org's policies.
Content moderation	Active	8,460 messages from users were blocked due to your content moderation settings.

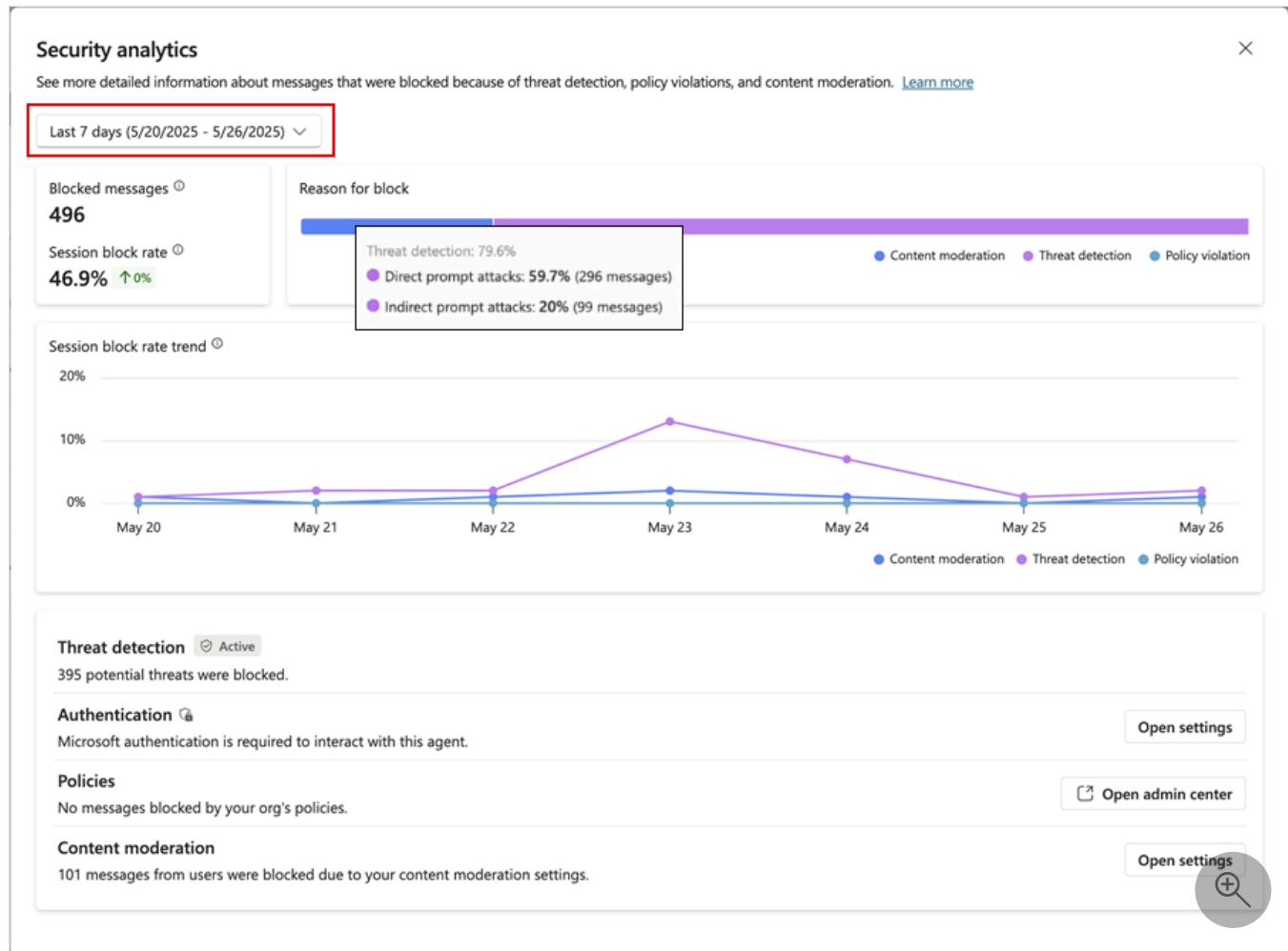
From the agent-level summary protection dialog, if you want to see more detail, select **See details** to open the **Security analytics** dialog.

Security analytics

The Security analytics dialog displays statistics and trends about blocked messages and the agent's status in the categories of authentication, policies, and content moderation.

Select the date range of security analytics data you would like to see. If you don't have data that falls within one or more of the following ranges, those ranges don't appear as selectable options in the date range picker. Select from the following range options:

- Last 7 days
- Last 14 days
- Last 30 days



The **Reason for block** stacked bar chart breaks down the sum of all blocked messages in the configured time frame into color-coded bar segments whose individual lengths signify the proportional share for each blocking reason. For example, in the graphic, there are 496 blocked messages in the last seven days, of which 79.6% are blocked as potential threats (pink bar segment) with about 60% of the total number of blocked messages being direct prompt attacks. A quick glance at the **Session block rate trend** graph shows the rate of message blocking due to the detection of potential threats peaks on May 23rd.

The **Session block rate trend** graph charts the share of the total number of sessions in which a prompt was blocked as a trend line over time. Each of the three categories of protection is individually represented as separate, color-coded trend lines, using the same color-coding as the **Reason for block** bar chart. For more information about conversational sessions, see [Conversational sessions](#).

 [Expand table](#)

Threat detection or protection category	Description	Action
Threat detection	This component shows statistics on blocked prompt attacks, which are actively blocked by default. It provides makers with insights into the number of attacks and trends, helping them understand the security landscape of their agents.	
Authentication	This component indicates whether the agent requires end-user authentication or is public. If an autonomous agent is public, it might expose sensitive data to potential attackers or unauthorized individuals. In such cases, the status reflects a possible threat vector that might require the maker's attention.	Select Open settings to navigate to your agent's Settings > Security > Authentication page, where you can select an authentication method.
Policies	This component reflects policy violations set by admins in the Power Platform admin center (PPAC). Agents might violate these policies, and makers need to be aware of these violations and make necessary adjustments. For example, an agent might use a connector that is blocked by organizational data loss prevention (DLP) policy.	Make the required changes in your agent so that it complies with your org's policies. To review errors due to policies, select the Review errors link under Policies . If, instead, it's necessary to change your org's policies, and you have access, select Open admin center to navigate to the Power Platform admin center so you can view or edit your org's policies.
Content moderation	This component doesn't affect the protection status directly but is part of the statistics and trends available to the maker. It helps ensure that the content generated by the agent adheres to the desired sensitivity levels.	Select Open settings to navigate to your agent's Settings > Generative AI page, and under Moderation > Content moderation level , adjust the slider to the desired content moderation level.

Automatic security scan in Copilot Studio

Article • 11/19/2024

By default, agents are secure. However, you can modify the default security settings for valid scenarios without knowing the risk. Copilot Studio automatically runs a security scan and warns makers before publishing.

Makers see risks when the following secure default settings are updated:

- Set the authentication mode for an agent to **No authentication** to allow anyone who has the link to interact with the agent. The default agent authentication mode is **Authenticate with Microsoft** but makers can select **No authentication** instead. For more information, see [Choose an authentication option](#).
- The maker selects **Author authentication** option under **User authentication** for connectors and flows. The default option for connectors and flows is **User authentication**, and the security scan provides a warning if the maker changes it to **Author authentication**. For more information, see [Use connectors with agent author's credentials](#).
- The maker shares an agent with everyone in the organization. The default agent is shared with no one and makers can then share it with everyone in the organization. For more information, see [Share an agent with everyone in the organization](#).

Related content

- [Configure user authentication](#)
- [Use connectors with agent author's credentials](#)
- [Share a cloud flow with run-only permissions](#)
- [Share an agent with everyone in the organization](#)

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

Data locations in Copilot Studio

09/18/2025

Data management in Copilot Studio is comprehensive, offering [flexibility and control to organizations](#) while [supporting a wide range of international standards and requirements](#).

This allows businesses to operate globally without compromising on compliance or user experience.

With Copilot Studio, you can deploy your agent to any of the Microsoft Azure datacenters around the world.

You can also configure your [environment](#) to suit your development or production needs. Learn more in [Data storage and governance in Power Platform](#). For example, you can [import and export agents as solutions in the Power Apps portal](#) to support application lifecycle management (ALM) workflows.

Data locations

No matter where customer data is stored, Microsoft doesn't control or limit the locations from which customers or their users can access customer data.

Microsoft might replicate customer data to other regions available within the same geography for data durability.

[Learn more about Copilot availability by geography.](#) ↗

 **Note**

If a tenant's location is not listed in the data locations table, data is stored in the United States.

 [Expand table](#)

Azure geographic locations	Azure datacenters (regions)
Asia	East Asia (Hong Kong Special Administrative Region)
Australia	Australia East (New South Wales), Australia Southeast (Victoria)
Canada	Canada Central (Toronto), Canada East (Quebec City)

Azure geographic locations	Azure datacenters (regions)
Europe	West Europe (Netherlands), North Europe (Ireland)
France	France Central (Paris), France South (Marseille)
Germany	Germany North (Berlin), Germany West Central (Frankfurt)
India	Central India (Pune), South India (Chennai)
Japan	Japan East (Tokyo, Saitama), Japan West (Osaka)
Norway	Norway East (Oslo), Norway West (Stavanger)
Singapore	Southeast Asia
South Africa	South Africa North (Johannesburg), South Africa West (Cape Town)
South America	Brazil South (Sao Paulo State) (As there is only one region in Brazil, customer data in Brazil South might be replicated to South Central US (Texas) for disaster recovery purposes)
South Korea	South Korea Central (Seoul), South Korea South (Busan)
Sweden	Sweden Central (Gävle)
Switzerland	Switzerland North (Zurich), Switzerland West (Geneva)
United Arab Emirates	UAE North (Dubai)
United Kingdom	UK South (London), UK West (Cardiff, Durham)
United States	East US (Virginia), South Central US (Texas), West US 2 (Washington)

Customer data

Microsoft ensures that customer data in Copilot Studio remains within the chosen Azure geographic location, with some exceptions:

- **South America:** Data may be replicated to South Central US (Texas) for disaster recovery purposes.
- **Support and compliance:** Data might be transferred to provide customer support, troubleshoot issues, or comply with legal requirements.
- **Global services:** Certain services inherently require global operation and may store data globally. Examples include:

- **Email Marketing:** Configured by customers to send messages globally.
 - **Dynamics 365 Home Page:** Stores application details globally for performance.
 - **Microsoft Entra ID:** Might store data globally.
 - **Microsoft Entra multifactor authentication:** Data might be stored globally.
 - **Microsoft 365 admin center:** Collects customer data during onboarding.
 - **Routing Services:** These services, such as Azure DNS, provide global routing without processing or storing customer data.
 - Additionally, certain types of customer data (specifically the application name, application description, and application logo) are stored globally, rather than in the primary storage geographic location.
- **External services:** Customers can configure Copilot Studio to use external services, which may transfer data outside the selected geographic location. Examples include:
 - **Customer Service hand-off:** Transfers escalations to human representatives.
 - **Multi-channel configurations:** Integrations with platforms like [Facebook](#) and [Microsoft Teams](#).
 - **Dynamics 365 Customer Service Insights:** Topic suggestions may be processed in a different geographic region.
 - **Microsoft 365-powered features:** Certain features powered by Microsoft 365 services transfer data outside of the Azure compliance boundary and store data in Microsoft 365 according to [Microsoft 365 terms and data residency commitments](#), including:
 - **Historical activity data for agents:** [Activity](#) data, including conversation logs, is stored in the geographic region of the end user's Exchange mailbox.

Related content

- [Language support](#)
- [Geographic data residency](#)
- [Security and geographic data residency](#)

Geographic data residency in Copilot Studio

09/17/2025

Geographic data residency refers to the policies and practices that govern where data is stored, processed, and managed geographically. This concept is crucial for organizations that need to comply with various regulatory requirements, ensure data sovereignty, and optimize data access and performance.

Microsoft Copilot Studio addresses the needs of geographic data residency by ensuring that data is stored and processed in compliance with regional regulations and organizational policies.

This involves focusing on key aspects such as security, privacy, the [General Data Protection Regulation \(GDPR\)](#), data location, and compliance. By adhering to these principles, Copilot Studio helps organizations manage their data effectively across different regions, ensuring that they meet regulatory requirements and maintain data sovereignty.

This approach not only optimizes data access and performance but also provides a robust framework for data management in a globally distributed environment. Copilot Studio can be configured to access generative AI features in other areas, including when capacity is constrained. Learn more about directing data outside a geography in [Move data across regions for Copilots and generative AI features](#).

Security

Security is paramount when dealing with geographic data residency. Copilot Studio employs robust security measures to protect data at rest and in transit. Data is encrypted using industry-standard protocols, ensuring that unauthorized access is prevented. Additionally, Microsoft continuously monitors and updates its security infrastructure to defend against emerging threats.

Learn more about [security and geographic data residency](#) in Copilot Studio.

Privacy ↗

Privacy is a cornerstone of data handling practices in Copilot Studio. The platform adheres to strict privacy policies to ensure that user data is not only protected but also used responsibly. Microsoft Copilot Studio provides transparency about data collection, usage, and storage, allowing users to make informed decisions about their data.

Learn more about [privacy](#) in Copilot Studio.

General Data Protection Regulation (GDPR)

The GDPR imposes stringent requirements on how personal data is handled. Microsoft Copilot Studio is designed to comply with GDPR by ensuring that data is stored within the designated geographic boundaries and that data subjects' rights are respected. This includes the ability to handle Data Subject Requests (DSRs) and perform Data Protection Impact Assessments (DPIAs).

Learn more about [GDPR](#) in Copilot Studio.

Data locations

Microsoft Copilot Studio allows organizations to choose where their data is stored, providing flexibility to meet regional data residency requirements. Data can be stored in various Azure datacenters across the globe, ensuring that it remains within the specified geographic boundaries. This capability is crucial for organizations with specific data localization needs.

Learn more about [Data Locations](#) in Copilot Studio.

EU Data Boundary (EUDB) compliance

Copilot Studio offers EU Data Boundary compliance. For more information, see [details of the EU Data Boundary commitment](#).

For Copilot Studio, if a customer provisions a tenant with a billing address in the EU or EFTA, that tenant will be in-scope for the EU Data Boundary if the customer also creates all of its environments within a [geographic region](#) inside the EU Data Boundary.

For more information, see the [Microsoft Privacy & Security Terms](#).

Compliance

Compliance with regional and international regulations is a key focus for Copilot Studio. The platform supports compliance with various data protection laws, including GDPR, CCPA, and others. By providing tools and features that facilitate compliance, Microsoft Copilot Studio helps organizations mitigate legal risks and maintain trust with their users.

Learn more about [Compliance](#) in Copilot Studio.

Data Flows Using Connectors

Copilot Studio and Power Platform utilize connectors to facilitate seamless data flows between various systems and services. These connectors act as proxies or "wrappers" around APIs, enabling communication between Microsoft services (like SharePoint, Dataverse, and Microsoft Graph) and external systems (such as Salesforce and other third-party APIs).

Data that is transmitted as part of a connector for a Microsoft service follows this process:

1. **Initiation:** A user action or an automated trigger initiates the data flow.
2. **Connector invocation:** The appropriate connector is invoked to handle the data transfer.
For example, an agent can invoke a Power Automate flow to use the SharePoint connector to move data from a form submission to a SharePoint list.
3. **Data transfer:** Data is securely transferred between systems. Connectors ensure that data is encrypted during transit and adhere to the security protocols of both the source and destination systems.
4. **Processing and storage:** Once the data reaches its destination within the Microsoft cloud, it is processed and stored according to the predefined rules and configurations. For instance, data sent to Dataverse can be used to trigger further workflows or analytics.
5. **Compliance and monitoring:** Throughout the data flow, compliance with regional regulations and organizational policies is maintained. Microsoft provides tools to monitor and audit these data flows, ensuring transparency and accountability.

When using connectors to send and retrieve data from external systems (example Salesforce), the responsibility to maintain the measures described in this article depends on whether the connection is to Microsoft services or external services:

- For connectors that send and retrieve data from external, non-Microsoft systems (such as Salesforce), the responsibility belongs to the agent maker.
- For connectors communicating inside the Microsoft cloud, these responsibilities are handled by Microsoft.

For more information on using connectors in Copilot Studio and Power Platform, see the [Use Power Platform connectors in Copilot Studio](#) article.

Security and geographic data residency in Copilot Studio

Article • 05/07/2025

Geographic data residency in Microsoft Copilot Studio provides a robust framework for ensuring data security and compliance with local regulations.

Copilot Studio leverages the robust infrastructure of Azure to provide secure and compliant data residency options, in addition to enhanced security features within Copilot Studio itself.

By leveraging the advanced security features in Azure and Copilot Studio, and by adhering to best practices, your organization can confidently deploy chatbots that meet important data protection requirements.

Data residency and security in Copilot Studio

Copilot Studio allows organizations to create agents that can operate within [specific geographic locations](#). This capability ensures that data generated and processed by these agents adheres to local data residency requirements. The primary benefits include:

- **Compliance with local regulations:** By storing data within the specified geography, organizations can comply with local data protection laws and regulations.
- **Enhanced data security:** Localized data storage reduces the risk of data breaches and unauthorized access, as data is confined within a controlled environment.

Azure data security measures

Microsoft employs several security measures to ensure the integrity and confidentiality of data within its Azure datacenters:

- **Encryption:** All data at rest and in transit is encrypted using industry-standard protocols. This includes data stored in Azure SQL Database, Azure Storage, and other services used by Copilot Studio.
- **Access controls:** Strict access controls are implemented to ensure that only authorized personnel can access sensitive data. This includes multifactor authentication (MFA) and role-based access control (RBAC).
- **Data replication and durability:** Data is replicated across multiple datacenters within the same geographic region to ensure durability and availability. This replication is done in a manner that complies with data residency requirements.

- **Compliance certifications:** Azure datacenters comply with various international standards and certifications, such as ISO/IEC 27001, SOC 1/2/3, and General Data Protection Regulation (GDPR), ensuring that data handling practices meet stringent security and privacy standards. Learn more about [compliance](#) and [personal data requests](#) in Copilot Studio.

Additional data security measures in Copilot Studio

In addition to the general security measures provided by Azure, Copilot Studio incorporates several enhanced security features:

- **Bot authentication and authorization:** Copilot Studio supports various authentication methods, including OAuth 2.0, to ensure that only authorized users can interact with the bots. This prevents unauthorized access and ensures that user interactions are secure.
- **Data masking:** Sensitive information, such as personal identifiers and financial data, can be masked within bot interactions to prevent exposure. This is crucial for maintaining privacy and compliance with data protection regulations.
- **Audit logs and monitoring:** Copilot Studio provides detailed audit logs and monitoring capabilities to track bot activities and user interactions. This helps in identifying and responding to potential security incidents promptly.
- **Secure integration with other services:** Copilot Studio can securely integrate with other Microsoft services, such as Dynamics 365 and Microsoft Teams, using secure APIs and connectors. This ensures that data remains protected during inter-service communication.

Multi-geographic capabilities

For organizations operating in multiple regions, Copilot Studio supports multi-geographic deployments. This enables organizations to deploy environments and agents to different environments in different geographic locations.

Best practices for implementing geographic data residency

To maximize the benefits of geographic data residency in Copilot Studio, organizations should follow these best practices:

- **Understand local regulations:** Stay informed about local data protection laws and ensure that your data residency strategy complies with these regulations.

- **Regular audits and assessments:** Conduct regular security audits and assessments to identify and mitigate potential vulnerabilities.
- **Employee training:** Train employees on data protection best practices and the importance of adhering to data residency requirements.
- **Leverage Azure security tools:** Utilize Azure's built-in security tools, such as Azure Security Center and Azure Policy, to monitor and enforce security policies.

Learn more about [Security](#) in Copilot Studio.

Ensure compliance with Copilot Studio

Article • 12/20/2024

In today's digital landscape, compliance is more critical than ever. Organizations must adhere to various regulations and standards to protect sensitive data, maintain customer trust, and avoid legal repercussions. One key aspect of compliance is ensuring data residency, which involves storing and processing data within specific geographic boundaries. Microsoft Copilot Studio offers robust features to help organizations meet critical compliance requirements, [particularly in terms of geographical data residency](#).

Why compliance is important

- Legal requirements: Many countries have stringent data protection laws that mandate where data can be stored and processed. Non-compliance can result in hefty fines and legal actions.
- Customer trust: Adhering to compliance standards demonstrates a commitment to data security, which can enhance customer trust and loyalty.
- Risk management: Compliance helps in identifying and mitigating risks associated with data breaches and unauthorized access.
- Operational efficiency: Following compliance guidelines can streamline processes and improve overall operational efficiency.

Copilot Studio is designed with compliance at its core and is an **Online Service** as defined in the [Online Services Terms \(OST\)](#). It is compliant with or covered by:

- Health Insurance Portability and Accountability Act (HIPAA) coverage
- Health Information Trust Alliance (HITRUST) Common Security Framework (CSF)
- Federal Risk and Authorization Management Program (FedRAMP)
- System and Organization Controls (SOC)
- Various International Organization for Standardization (ISO) certifications
- Payment Card Industry (PCI) Data Security Standard (DSS)
- The Cloud Security Alliance (CSA) Security Trust Assurance and Risk (STAR)
- United Kingdom Government Cloud (G-Cloud)
- Outsourced Service Provider's Audit Report (OSPAR)
- Korea-Information Security Management System (K-ISMS)
- Singapore Multi-Tier Cloud Security (MTCS) Level 3
- Spain Esquema Nacional de Seguridad (ENS) High-Level Security Measures

Health Insurance Portability and Accountability Act (HIPAA) coverage

HIPAA [↗](#) is a United States healthcare law that establishes requirements for the use, disclosure, and safeguarding of individually identifiable health information. It applies to covered entities—doctors' offices, hospitals, health insurers, and other healthcare companies—that have access to patients' protected health information (PHI), in addition to business associates—such as cloud service and IT providers—that process PHI on their behalf.

Microsoft Copilot Studio is covered under the Health Insurance Portability and Accountability Act (HIPAA) Business Associate Agreement (BAA).

You can create agents that handle protected health information when your organization is bound by HIPAA, as in the following scenarios where the agent can:

- Ask individuals to provide their health information (blood pressure, weight, and so on).
- Capture health information and personally identifying information, such as the customer's IP address or email address.

Note

Although Copilot Studio is covered under HIPAA, it still isn't intended for use as a medical device. See the disclaimer on [the intended use of Copilot Studio and medical devices](#).

[Learn more about HIPAA.](#)

Health Information Trust Alliance (HITRUST)

HITRUST is an organization governed by representatives from the healthcare industry.

HITRUST created and maintains the Common Security Framework (CSF), a certifiable framework to help healthcare organizations and their providers demonstrate their security and compliance consistently.

The CSF builds on HIPAA and the HITECH Act, which are US healthcare laws that have established requirements for the use, disclosure, and safeguarding of individually identifiable health information and enforce non-compliance.

HITRUST provides a benchmark—a standardized compliance framework, assessment, and certification process—against which cloud service providers and covered health entities can measure compliance.

[Learn more about HITRUST.](#)

Federal Risk and Authorization Management Program (FedRAMP)

FedRAMP was established to provide a standardized approach for assessing, monitoring, and authorizing cloud computing products and services under the Federal Information Security Management Act (FISMA) and to accelerate the adoption of secure cloud solutions by federal agencies.

Microsoft's government cloud services meet the requirements of FedRAMP.

By deploying protected services including Azure Government, Office 365 US Government, and Dynamics 365 Government, federal and defense agencies can use a rich array of compliant services.

[Learn more about FedRAMP.](#)

SOC compliance

SOC is a method for assuring control regulation within a service. Microsoft Copilot Studio has been audited to be compliant with SOC.

SOC audit reports are available from the [Microsoft Service Trust Portal](#).

[Learn more about SOC.](#)

ISO compliance

Microsoft Copilot Studio is compliant with the ISO standards listed in the following table. Audit reports for each are available from the [Microsoft Service Trust Portal](#).

expand Expand table

Standard	Name of report and certificate	Link to standard (www.iso.org)
ISO	Microsoft Azure, Dynamics 365, and Other Online Service	ISO 9001:2015

Standard	Name of report and certificate	Link to standard (www.iso.org)
9001:2015	- ISO9001 Certificate and Assessment Report ↗	
ISO 20000-1:2011	Microsoft Azure, Dynamics 365, and Other Online Service - ISO20000-1 Certificate and Assessment Report ↗	ISO/IEC 20000-1:2011 ↗
ISO 22301:2012	Microsoft Azure, Dynamics 365, and Other Online Service - ISO20000-1 Certificate and Assessment Report ↗	ISO/IEC 22301:2012 ↗
ISO 27001:2013	Microsoft Azure, Dynamics 365, and Other Online Service - ISO27001 and 27701 Certificate ↗ and Microsoft Azure, Dynamics 365, and Other Online Service - ISO27001, 27018, 27017, 27701 Assessment Report ↗	ISO/IEC 27001:2013 ↗
ISO 27017:2015	Microsoft Azure, Dynamics 365, and Other Online Service - ISO27017 Certificate ↗ and Microsoft Azure, Dynamics 365, and Other Online Service - ISO27001, 27018, 27017, 27701 Assessment Report ↗	ISO/IEC 27017:2015 ↗
ISO 27018:2019	Microsoft Azure, Dynamics 365, and Other Online Service - ISO27018 Certificate ↗ and Microsoft Azure, Dynamics 365, and Other Online Service - ISO27001, 27018, 27017, 27701 Assessment Report ↗	ISO/IEC 27018:2019 ↗
ISO 27701:2019	Microsoft Azure, Dynamics 365, and Other Online Service - ISO27701 Certificate ↗ and Microsoft Azure, Dynamics 365, and Other Online Service - ISO27001, 27018, 27017, 27701 Assessment Report ↗	ISO/IEC 27701:2019 ↗

Payment Card Industry (PCI) Data Security Standard (DSS)

The Payment Card Industry (PCI) Data Security Standards (DSS) form a global information security standard designed to prevent fraud through increased control of credit card data.

Organizations of all sizes must follow PCI DSS standards if they accept payment cards from the five major credit card brands:

- Visa
- MasterCard
- American Express
- Discover
- Japan Credit Bureau (JCB).

Compliance with PCI DSS is required for any organization that stores, processes, or transmits payment and card-holder data.

[Learn more about PCI DSS.](#)

The Cloud Security Alliance (CSA) Security Trust Assurance and Risk (STAR)

From the [CSA STAR website](#):

- The Security Trust Assurance and Risk (STAR) Program encompasses key principles of transparency, rigorous auditing, and harmonization of standards. Companies who use STAR indicate best practices and validate the security posture of their cloud offerings.

The STAR registry documents the security and privacy controls provided by popular cloud computing offerings. This publicly accessible registry allows cloud customers to assess their security providers in order to make the best procurement decisions.

Microsoft Copilot Studio has been audited to be compliant with CSA STAR.

[Learn more about CSA STAR.](#)

United Kingdom Government Cloud (G-Cloud)

Government Cloud (G-Cloud) is a UK government initiative to ease procurement of cloud services by government departments and promote government-wide adoption of cloud computing.

G-Cloud comprises a series of framework agreements with cloud services suppliers (such as Microsoft), and a listing of their services in an online store, the Digital Marketplace. These enable public-sector organizations to compare and procure those services without having to do their own full review process.

Inclusion in the Digital Marketplace requires a self-attestation of compliance, followed by a verification performed by the Government Digital Service (GDS) branch at its discretion.

[Learn more about G-Cloud.](#)

Outsourced Service Provider's Audit Report (OSPAR)

The OSPAR framework was established by the Association of Banks in Singapore (ABS), which formulated IT security guidelines for outsourced service providers (OSPs) that seek to provide services to Singapore's financial institutions. The ABS Guidelines are intended to assist financial institutions in understanding approaches to due diligence, vendor management, and key technical and organizational controls that should be implemented in cloud outsourcing arrangements, particularly for material workloads.

Microsoft Copilot Studio has OSPAR attestation.

[Learn more about the ABS OSPR.](#)

Korea-Information Security Management System (K-ISMS)

[K-ISMS](#) is a country/region-specific ISMS framework that defines a stringent set of control requirements designed to help ensure that organizations in Korea consistently and securely protect their information assets.

[Learn more about ISMS \(Korea\).](#)

Singapore Multi-Tier Cloud Security (MTCS) Level 3

The MTCS Standard for Singapore was prepared under the direction of the Information Technology Standards Committee (ITSC) of the Infocomm Development Authority of Singapore (IDA).

The ITSC promotes and facilitates national programs to standardize IT and communications, and Singapore's participation in international standardization activities.

[Learn more about MTCS.](#)

Spain Esquema Nacional de Seguridad (ENS) High-Level Security Measures

In 2007, the Spanish government enacted Law 11/2007, which established a legal framework to give citizens electronic access to government and public services. This law is the basis for Esquema Nacional de Seguridad (National Security Framework), which is governed by Royal Decree (RD) 3/2010.

The goal of the framework is to build trust in the provision of electronic services, and ensure the access, integrity, availability, authenticity, confidentiality, traceability, and preservation of data, information, and services.

[Learn more about ENS.](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Data, privacy, and security for web search

08/28/2025

Copilot Studio agents have an optional feature that allows them to reference web content when responding to user prompts. Allowing Copilot Studio agents to reference web content improves the quality of agent responses by grounding them in the latest information from the web.

Copilot Studio web search functionality

Copilot Studio accesses the web via Bing APIs through three different mechanisms: specific URLs, open web search, or custom search. These capabilities can be used individually or in combination. Specific URLs as [individual knowledge sources](#). Open web search via the [Web Search setting](#). Or a custom search in [a generative answers node](#). All or a subset of public websites indexed by Bing are searched, per the maker's configuration, using the [Grounding with Bing Search](#) or [Grounding with Bing Custom Search APIs](#).

How web search works

When web search is enabled and a user asks a question, Copilot Studio generates a search query. This is a brief, focused keyword query derived from the user's question. Copilot Studio then sends this query to the Bing search service. This generated query isn't the full user prompt or conversation, but a distilled set of terms the system determines will retrieve relevant info. For example, if the user asks, "What are the latest trends in electric vehicle sales worldwide?" Copilot Studio might send a query such as "latest global electric vehicle sales statistics 2025" to Bing.

Bing then processes this query and returns relevant search results (web page titles, snippets, citations). Copilot Studio uses the information from these results to compose a summarized final answer given to the user. It integrates the web content with any other active knowledge sources or actions. The user sees a unified answer combining all relevant information and the cited websites (including links to those websites) that informed the generation of the summary output.

Web search query logging

Search queries that Copilot Studio sends to Bing aren't associated with the customer's tenant ID. There are also no cookies or tracking tokens sent with the query that could link it to a user

session. The Bing service treats the query as an anonymous request from Copilot Studio, preventing any customer-specific profiling.

The content of the query is logged to Bing's systems and used according to the [Microsoft Privacy Statement](#) including to improve the search service – for example, to refine Bing's ranking algorithms or results quality. Because the queries lack identifiers, this logging can't single out an organization or users. Bing uses the data in aggregate to tune search results but can't attribute any query to any individual or organization.

Terms governing Bing web search in Copilot Studio

Copilot Studio's web search capabilities are governed by the [Terms of Use for Grounding with Bing Search and Grounding with Bing Custom Search](#) and the [Microsoft Privacy Statement](#). These terms specify that the Microsoft Privacy Statement applies to any data sent to Bing and that the Microsoft Products and Services Data Protection Addendum (DPA) doesn't apply to that data. Copilot Studio service's own terms (which include Microsoft's Online Services Terms for enterprise) carve out Bing queries from the usual enterprise customer data protections. This is because the data is leaving the controlled enterprise boundary and entering Bing's consumer service. Microsoft is committed to compliance with all applicable laws as a data controller for this information, including GDPR and other privacy regulations.

Controls available to manage web search

Admins can enable or disable Bing search in Power Platform Admin Center. Calls to the Grounding with Bing Search and Grounding with Bing Custom Search APIs introduce cross-geo data flows.

Extra resources

- [Terms of Use for Grounding with Bing Search and Grounding with Bing Custom Search](#)
- [Microsoft Privacy Statement](#)
- [Add a public website as a knowledge source](#)
- [Enable Web Search for your agent](#)
- [Search public data or use a Bing Custom Search for generative answers nodes](#)

Use entities and slot filling in agents

10/03/2025

A significant part of agent conversations in Copilot Studio is natural language understanding, which is the ability for the AI to understand a user's intent. For example, a customer might say "I tried to use my gift card but it doesn't work." The agent knows to route the customer to the topic related to gift cards not working, even if that exact phrase isn't listed as a trigger phrase.

One fundamental aspect of natural language understanding is to identify *entities* in a user dialog.

An entity can be thought of as a unit of information that represents a certain type of a real-world subject. For example, a phone number, postal code, city, or even a person's name. With the knowledge granted by entities, an agent can smartly recognize the relevant information from a user input and save it for later use.

Prebuilt entities

Entities in agents let you store information in similar groups.

Copilot Studio supports a set of [prebuilt entities](#), which represent the most commonly used information types in real-world dialogs, such as age, colors, numbers, and names.

With the knowledge granted by entities, an agent can smartly recognize the relevant information from user input and save it for later use.

To help understand that notion, let's use the **Money** entity as an example.

1. Open your agent in Copilot Studio and select **Settings** at the top of the page and then **Entities** in the menu.

A list of all available entities appears.

2. Select the **Money** entity. The details panel for the entity opens.

Money

X

Description

Monetary amounts, extracted as a number

Method

Prebuilt

Usage

User input	Entity	Saved value
It costs 1000 dollars	1000 dollars	1000.00
3 items for \$1K	\$1K	1000.00
It costs a thousand five hundred	A thousand five hundred	1500.00

Here you can see an explanation of this entity and ways it can be used to extract information related to money or currency from user input.

For example, when a user inputs "It costs 1000 dollars," using this money entity the agent knows that "1000 dollars" represents the *money* type of information. When the agent extracts this information and saves it to a variable, it saves "1000" as a number even though the surrounding information was text.

Custom entities

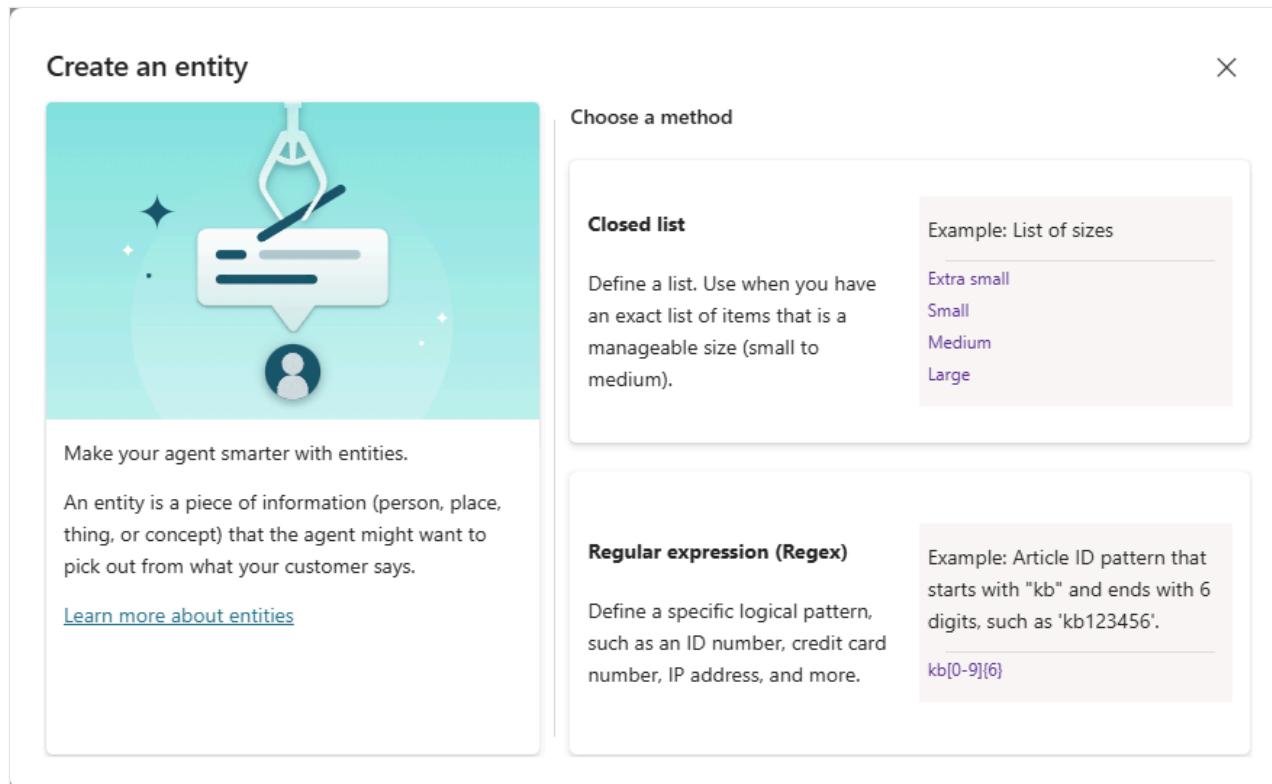
The prebuilt entities cover commonly used information types. Occasionally, such as when building an agent that serves a specific purpose, you might need to teach the agent's language understanding model domain-specific knowledge.

For instance, let's say you want to build an agent for an outdoor store. In this case, you need to teach the agent to acknowledge the "outdoor gears product" category in a conversation.

First, create a custom entity. In this case, you can create an entity that gives the agent the knowledge of all outdoor product categories.

1. Go to the **Settings** page for your agent and select **Entities**.
2. Select **Add an entity > New entity**.

3. Select the desired type of entity: either a [closed list](#) entity or a [regular expression \(regex\)](#) entity.



Closed list entities

Closed list entities let you define a list of items. These entities are best used for small lists that are easy to manage and that have simple item labels.

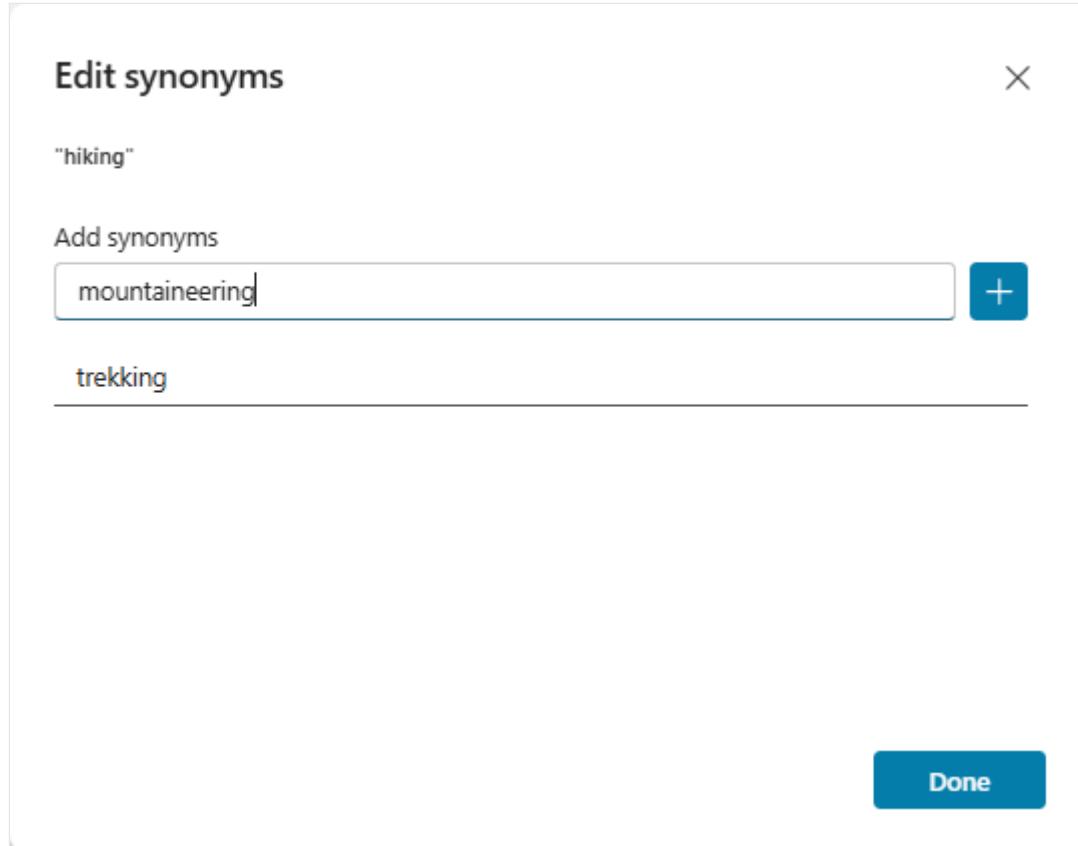
1. In **Create an entity**, select **Closed list**.
2. In the panel that opens, enter a name for your new entity—for example, *Camping activity*.
3. Add a description, if desired.
4. Under **List items**, enter values you want to associate with this entity. For example, *hiking* and *yoga*.

Alternatively, you can [upload a file](#) listing all the desired values you want to add for this entity.

When you enter items, you can:

- Select an item and change its name.
- Select the trash can icon to delete the item.
- Select **Synonyms** (or select the listed synonyms if they're already added) to open the **Edit synonyms** panel.

You can add synonyms to expand the matching logic for each value in the list. For example, for *hiking*, you can add *trekking* and *mountaineering* as synonyms.



5. If desired, turn on **Smart matching** for this entity.

Smart matching is part of the intelligence supported by the agent's language understanding model. When smart matching is active, the agent interprets user input using fuzzy logic against the values listed for the entity.

In particular, the agent can autocorrect misspellings and expand its matching logic semantically. For example, the agent can automatically match "softball" to "baseball."

6. When you're finished creating or editing your entity, select **Save**. Select **Close** to discard your changes.

Camping activity

Name *
Camping activity

Description
Activities that customers can ask about.

Method
List
The agent will try to match an item on the list based on what the customer says.

Smart matching
 on
The Smart matching option enables the agent's understanding of natural language. This can help match misspellings, grammar variations, and words with similar meanings.
If the agent isn't matching enough related words, enhance the agent's understanding further by adding synonyms to your list items.
[Learn more about entities](#)

List items

Item	Synonyms
hiking	mountaineering, trekking
yoga	+ Synonyms

Save **Close**

Upload values for a closed list entity

From the details panel for a closed list entity, you can add or replace the set of values and synonyms associated with the entity.

1. Prepare a text file (maximum 3 MB) listing all values you want to add for your entity, including synonyms, if any. Use this format: one value per line, and use the pipe (|) as a delimiter for synonyms. For example:

```
text

hiking|trekking
hiking|mountaineering
yoga
cycling|bicycling
cooking
```

2. Under **List items**, select **upload a file**.

3. In **Upload a file**, select the desired option:

- If you want to add new values and synonyms for your entity, select **Append**.
- To replace all existing values and synonyms with the list from your file, select **Replace**.

4. Drop your file onto the window. Alternatively, select **click to browse**, navigate to your file, and select it.

5. Review the values and synonyms that appear, and select **Add**.

Download values for a closed list entity

From the details panel for a closed list entity, you can download a text file listing the set of values and synonyms associated with the entity.

1. Under **List items**, select **download a file**.

2. Save the resulting text file, or open it in a text editor to review its content. Each entity value appears on a separate line. If a value has multiple synonyms, they appear on separate lines, like this:

```
text  
  
hiking|trekking  
hiking|mountaineering
```

Regular expression (regex) entities

Regular expression (regex) entities let you define logical patterns that you can use to match and extract information from an input. Regex entities are great for complex pattern matching against user input. They're also useful if you need to allow for specific variations in how a user might format or enter their input in a conversation.

For example, use regex entities to identify items such as a tracking ID, a license number, a credit card number, or an IP address.

1. In **Create an entity**, select **Regular expression (regex)**.

2. In the panel that opens, enter a name for your new entity.

Untitled

X

Name *

Enter entity name

Description

Enter description (optional)

Pattern *

Enter regular expression pattern

Method

Regular expression (Regex)

The agent will extract any text that matches the specific pattern.

[Learn how to use entities](#)

Examples

Knowledge base article ID that starts with "kb" and ends with 6 digits, "kb123456" for example.

kb[0-9]{6}

Credit card number that has 13 or 16 digits and starts with a 4.

4[0-9]{12}(?:[0-9]{3})?

Tracking number that starts with 9 digits and ends with one letter (upper or lower case), "100456789A" for example.

(?i)\d{9}[A-Z]{1}

Save

Close

3. Add a description, if desired.
4. Enter the regular expression pattern that should be used to match and extract entity items from the input.

Copilot Studio [NLU and CLU](#) language understanding use the [.NET regular expressions syntax](#). [NLU+](#) uses the JavaScript regular expressions syntax.

By default, pattern matching is case sensitive. To make it case insensitive, use the `(?i)` regular expression option in your pattern string. Some basic examples are included in the regex entity pane. For more information on the syntax, and for more examples, see [Regular Expression Language - Quick Reference](#).

If you need the regular expression to accommodate multiple patterns, you can use the alternation operator `|` to concatenate regular expressions. For example, you could combine two regex patterns to look for both the old and new formats of a product ID code.

5. When you're finished creating or editing your entity, select **Save**. Select **Close** to discard your changes.

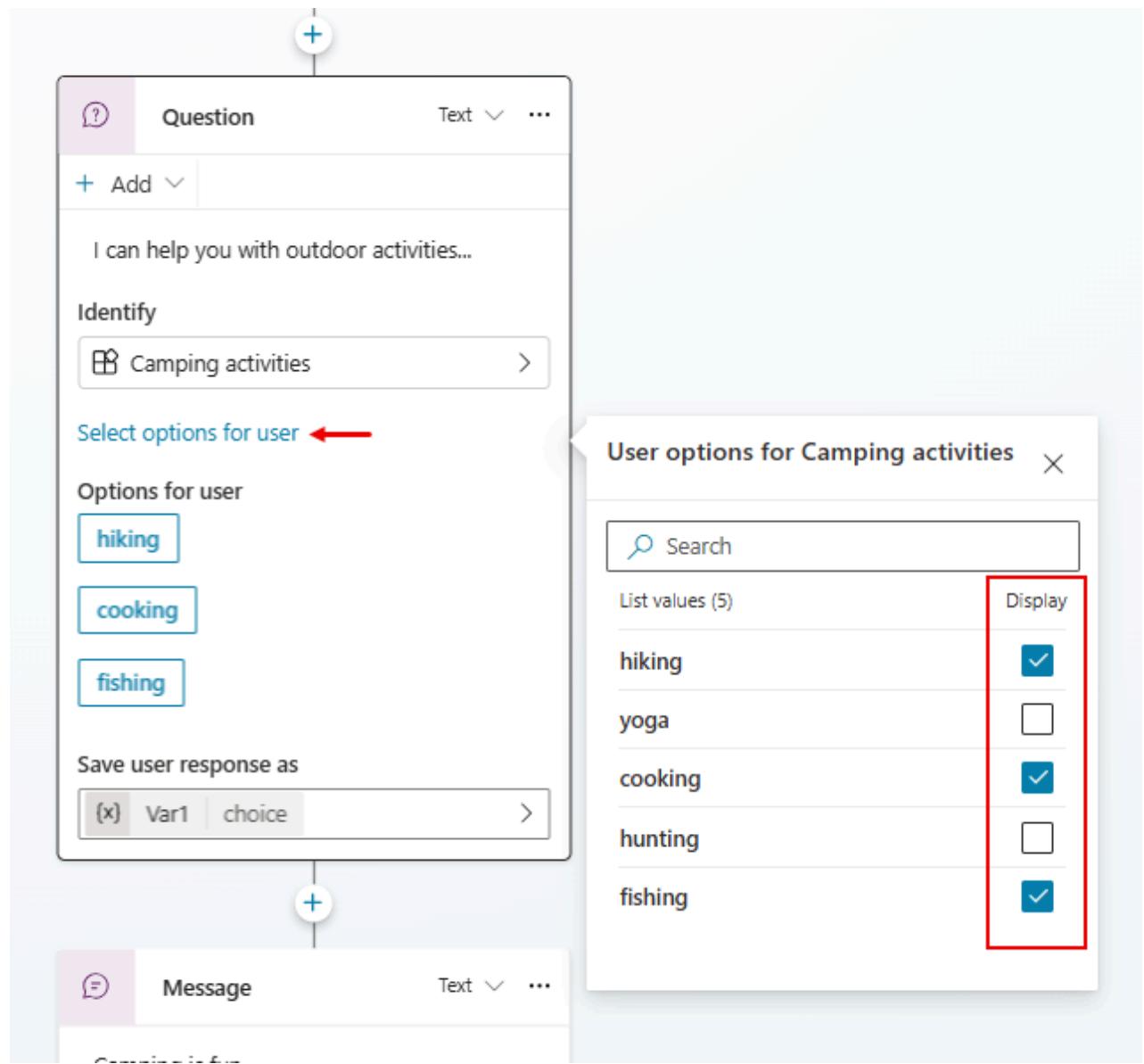
Use entities in a conversation

Now that your agent knows about outdoor gear based on the [Camping activity](#) entity and other custom entities you created, you can start to use them in your agent.

1. Go to the [Topics](#) page for your agent.
2. Open the topic for an entity you want to collect.
3. Select the **Add node** icon  and then select [Ask a question](#).
4. Under **Identify**, select the entity you created in [Custom entities](#).

The screenshot shows the Microsoft Bot Framework designer interface. On the left, there's a list of entities: 'Question' (Text), 'Add', and 'I can help you with outdoor activities...'. Below this is the 'Identify' section, which contains a button labeled 'Camping activities' with a red arrow pointing to it. Under 'Select options for user', there's a dropdown menu set to '(x) Var1 choice'. On the right, a modal window titled 'Choose information to identify' is open. It contains a search bar, a 'Create an Entity' option, and a 'Multiple choice options' section. A red box highlights the 'Camping activities' section, which is described as 'These are activities that users can ask about.' At the bottom of the modal, it says 'User's entire response' and 'No entity extraction; saved as is'.

5. You can also optionally select items to show as buttons. For example, if you'd like to show some categories as buttons for users to conveniently choose from as their input, you can choose **Select options for user** and then pick them from the list that contains the items you added when you created the custom entity.



6. Name the variable for the output of the user's response, if necessary.

Slot filling

Slot filling is a natural language understanding concept that means saving an extracted entity to an object. In Copilot Studio, slot filling means placing the extracted entity value into a variable. For more information about the variable base type associated with each prebuilt entity, see [Entities](#).

Let's continue to use the camping activities topic, which is triggered by typing "I want something to do" in the test chat, as an example.

The topic is successfully triggered and the agent asks for the type of activity, also showing the button choices specified when authoring the **Question** node. If you turn on the option to track between topics, the dialog tree also shows the agent is running to this **Question** node.

The screenshot shows a conversational AI authoring interface. On the left, a 'Trigger' card is open, displaying a list of phrases: 'I want to do an outdoor activity', 'I'm bored.', 'I want something to do.', 'What activities can I do outdoors?', and 'What can I do when camping?'. A blue arrow points down to a 'Question' card. The 'Question' card contains the text 'I can help you with outdoor activities...'. Below this, under 'Identify', there is a button labeled 'Camping activities'. Under 'Select options for user', there are three buttons: 'hiking', 'cooking', and 'fishing'. At the bottom, 'Save user response as' is set to '(x) trig choice'. On the right, a conversation window shows a message from 'Just now': 'Hello, I'm Clear Skies Weather Buddy, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI.' Below it, another message from 'Just now' says 'I'm bored.' with a timestamp 'Just now'. The AI's response is 'I can help you with outdoor activities...' with a timestamp 'Just now'. Below the AI's response, there are three buttons: 'hiking', 'cooking', and 'fishing'. At the bottom of the conversation window, there is a note: 'Ask a question or describe what you need' and '0/2000' characters used. A small note at the bottom right says 'Make sure AI-generated content is accurate and appropriate before using. See terms.'

A user can use the predefined choices by selecting one of the buttons. Alternatively, they can also type in something like "trekking" and see that it maps to "hiking," because those words were defined as synonyms.

The tracking view shows that the dialog is correctly routed to the path in which the product category value is "Hiking." You can inspect the variable value from the variable watch window at the bottom of the authoring canvas. The watch window shows that the variable value is "Hiking."

Essentially, slot filling happens by inserting the extracted entity "Hiking" into the variable `VarCampType`.

Variables

Browse Test

Topic (1)

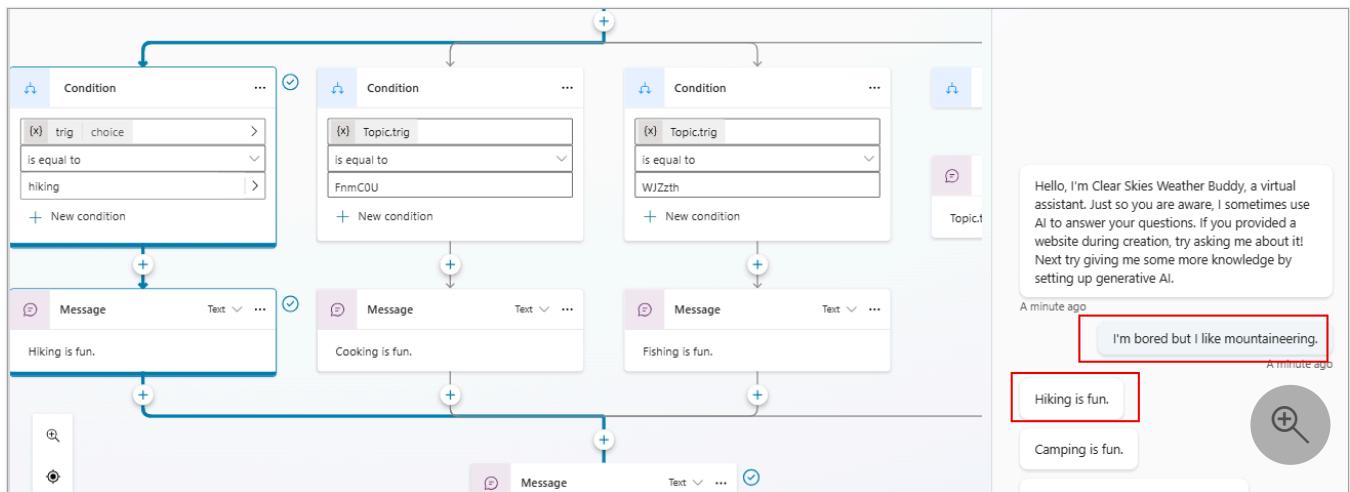
VarCampType choice
hiking

Global (0)

Environment (0)

You can also use what is known as "proactive slot filling" where the user can specify multiple pieces of information that map to multiple entities. The agent is able to understand what information belongs to which entity automatically. In cases where it's unsure of the intended mapping, it prompts the user to be more specific by providing choices.

In this example, the user wrote "I'm bored but I like mountaineering." This message includes the trigger phrase that the user wants help with outdoor activities and a second piece of information, "mountaineering," which is a synonym for hiking. In this case, the agent fills in both the entity for choosing an outdoor activity, and for the type of activity.



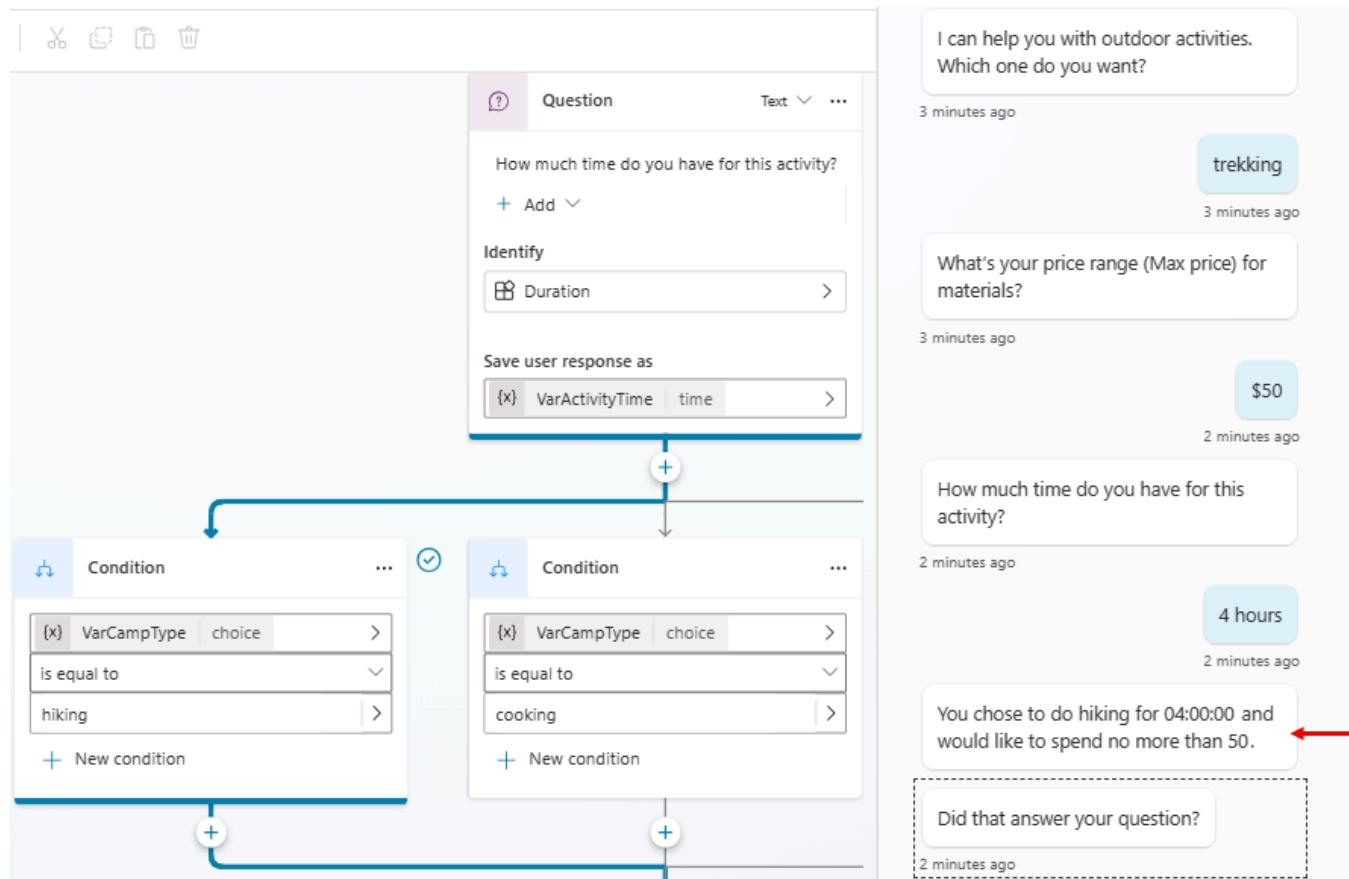
The tracking view shows that the agent takes in this user input, and intelligently skips the **Question** node asking for the type of activity.

The agent is always actively listening to the user input, remembering information upfront so it can skip unnecessary steps as appropriate.

Let's restart the testing again and try another case. In this round, you can add a couple more **Question** nodes asking for things like how much time you have for the activity (using the **Duration** entity) and the price range (using the **Money** entity).

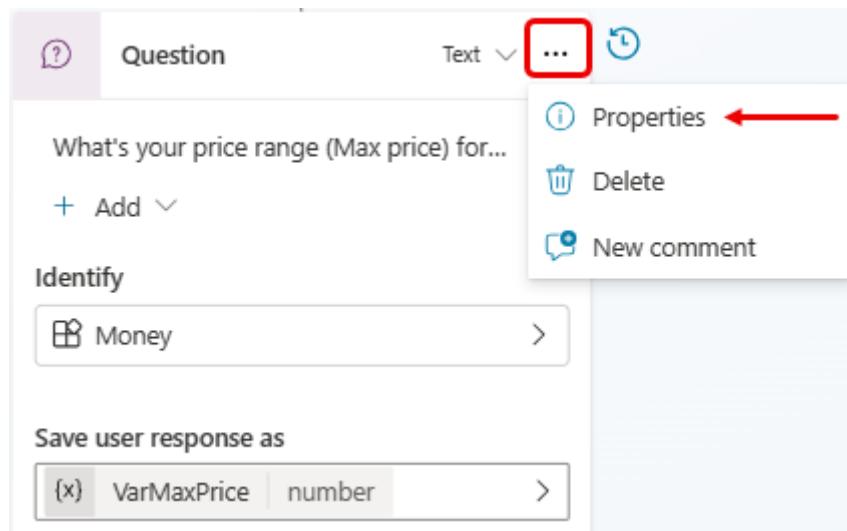
This time when the product category question is presented, instead of telling the agent only the product category, the user can say "I want to buy a pair of hiking boots under \$100." In this

example, the agent isn't only able to route to the correct hiking product category path, but also actively fill the slots asking for the type of hiking gear and the target price range information.



Proactive slot filling can be manually controlled at the node level. If you'd like to always prompt for the question within a specific node, regardless of whether the slot is already filled from previous user responses, you can disable the **Skip question** option for that **Question** node. Perform the following steps to disable the **Skip question** option.

1. Select the **More** icon (...) of the **Question** node, and then select **Properties**. The **Question properties** panel appears.



2. On the **Question properties** panel, select **Question behavior**.

3. On the **Question behavior** panel, under **Skip question**, select **Ask every time**, and then select **Save** at the top of the page.

Question behavior

Skip behavior

Decide if the question should be skipped if the variable already has a value

Skip question ⓘ Manual input ▾

Allow question to be skipped
 Ask every time ←

Accept one of multiple entities at a conversation turn

In some situations, an agent can accept one of multiple valid answers at a given conversation turn. For example, your agent might prompt a customer to provide either their account number or their phone number. You might also want your agent to let a customer state that they don't have the information, and route the customer accordingly instead of immediately escalating to a customer service representative.

1. Add a [Question node](#).
2. Under **Identify**, select **One of multiple entities**.
3. For each entity you want to accept at this node, select **New entity** and select the desired entity. A **Question** node can support up to five different entities.
4. Select the default variable name under **Save user response as** to open the **Variable properties** panel and change the name to something meaningful—for example, **Identifier**.

The variable that stores the recognized entity value is of type *record*, with one element for each supported entity at this node—for example **Identifier.account**, **Identifier.phone**, **Identifier.unknown**.

5. Add conditions to route the conversation according to the information provided by the customer. It's a good practice to set up your conditions with the **is not Blank** operator (or **is Blank**).

Recognition behavior for "One of multiple entities"

This type of recognition is intended to identify *a single entity* out of a set of possible entities at a conversation turn. If a customer enters a statement that contains two or more of the entities configured for identification at the corresponding **Question** node, the agent identifies *only* the first entity in the list. Therefore, make sure to craft the questions and the conditional logic that follows such **Question** nodes accordingly.

For example, let's say a **Question** node is configured to recognize one of the following: a library card number, a phone number, or "I don't know," in this order. If a customer says "My phone number is 777 555-1212 and my card number is 123456789," the recognized entity value is the card number because it's the first entity in the list of possible entities at the corresponding **Question** node.

If the agent can't identify any of the entities a **Question** node is configured to recognize, it applies the [reprompt behavior](#) specified for this node.

Limitations

The following are known limitations for **Question** nodes configured to collect one of multiple entities at a conversation turn:

- Support limited to a maximum of five entities.
- No support for external entities.
- Support for only one entity of any given type. For example, one **Question** node can't recognize two entities of type **Date**.

Create and delete agents

09/24/2025

With Copilot Studio, you can easily create agents by using a conversational creation experience.

(!) Note

Agents created in Copilot Studio and in Microsoft Teams are automatically configured for Microsoft Entra ID authentication.

You can also configure single sign-on (SSO) so your users don't need to sign in manually. For more information, see [Configure single sign-on with Microsoft Entra ID](#).

To create your first agent, see [Quickstart: Create and deploy an agent](#).

Create an agent

Web app

1. Sign in to [Copilot Studio](#).
2. Switch to the desired environment, if needed.
3. On the **Home** page, enter a brief description of what you want your agent to be.

Alternatively:

- Go to the **Create** page and select **New agent**.
- Go to the **Agents** page and select **New agent**.

A page appears with two panes where you can set up your agent on the left-hand side, and test it on the right-hand side. The left-hand pane has two tabs: **Describe** and **Configure**. You can alternate between these two experiences, and also iterate between setting up your agent and testing it.

4. Chat with Copilot, in the **Describe** tab, to describe your agent. Copilot asks questions to guide you, and uses your responses to fill in the details, such as the name, description, instructions, and knowledge, that define your agent.

Alternatively, switch to the **Configure** tab, and fill in the form.

 **Note**

- The conversational agent creation experience uses your browser language. If your browser language isn't in the list of [supported languages](#) for the conversational agent creation experience, only the **Configure** tab with the more discrete configuration experience is available.
- The maximum length for the name is 30 characters.
- The description can have up to 1,000 characters.
- The instructions can have up to 8,000 characters.

Once you have a name, description, and instructions for your agent, you can already start testing it!

5. Chat with your draft agent in the test pane. Enter your own question or try one of the suggested prompts. Go back and forth between the test pane and the creation pane as needed, to further refine your agent. If you want to see the suggested prompts again, select **New chat** at the top of the test pane.

 **Note**

During the creation experience, your agent might need a few moments before it can start using its knowledge sources.

6. If desired, [change the primary language](#) your agent uses to communicate.
7. Optionally, [select a solution and set the schema name](#) for your agent.
8. When you're satisfied with your tests, select **Create**. The **Overview** page for your agent appears.

Make your agent unique with a distinctive icon

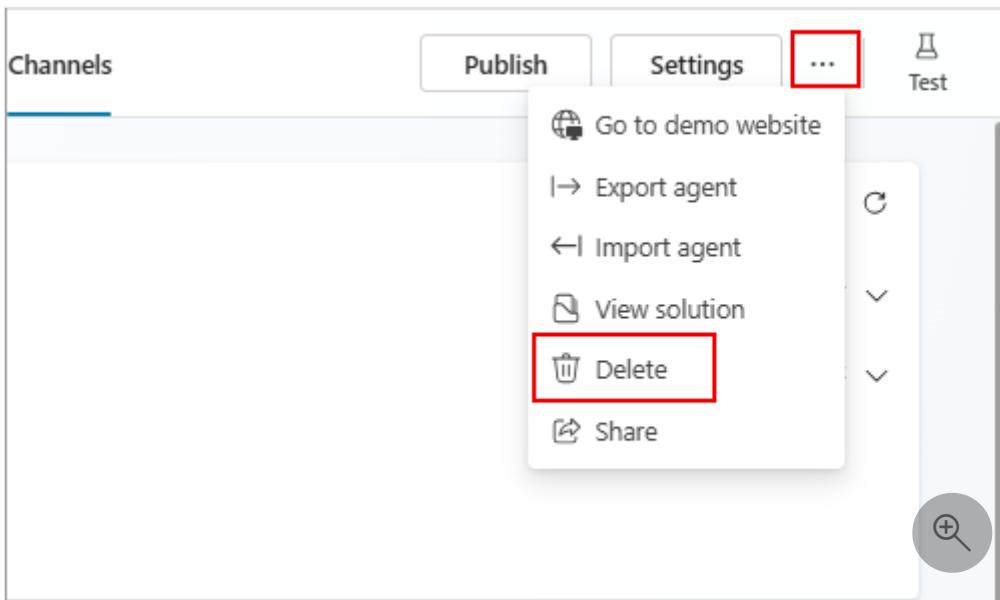
1. Select the agent icon in the top bar.
2. Select **Change icon**.
3. Choose an image from your device. The image file must be in PNG format and less than 30 KB in size, with a maximum resolution of 192 × 192 pixels.
4. Select **Save**.

Delete an agent

Web app

You can delete agents to remove them from your environment.

1. Go to the **Agents** page.
2. Select the agent you want to delete.
3. On the top menu bar, select the **More** icon (...), and select **Delete**.



Copilot Studio prompts you to confirm your intention.

4. Enter your agent's name and select **Delete agent**. After a few minutes, all data associated with the agent is deleted permanently.

Note

If your license is expired, you can delete your agents by selecting **Permanently delete your agents**.

Known issues with creating an agent

When you're creating your agent, you might encounter these issues.

Insufficient permissions for the selected environment

You might see this error: "You don't have permissions to any environments, Get access from an administrator." To resolve the error, [create a new environment](#). Use that environment to create your agent again.

For more information, see [Understand error codes](#).

Write agent instructions

Article • 05/19/2025

Instructions are the central directions and parameters an agent follows. Agents depend on instructions to:

- Decide what tool or knowledge might need to be called to address a user query or autonomous trigger.
- Fill inputs for any tool based on the available context.
- Generate a response to the end user.

Agents can't act on instructions to use tools, knowledge sources, or topics the agent doesn't have. You must first configure your agent with the appropriate tools and knowledge sources. For example, if you give an instruction for your agent to search a website's FAQ, the agent can't follow that instruction unless you add the website FAQ as a knowledge source. Ensure that any instructions you give your agent are grounded in actions and knowledge you configured for your agent.

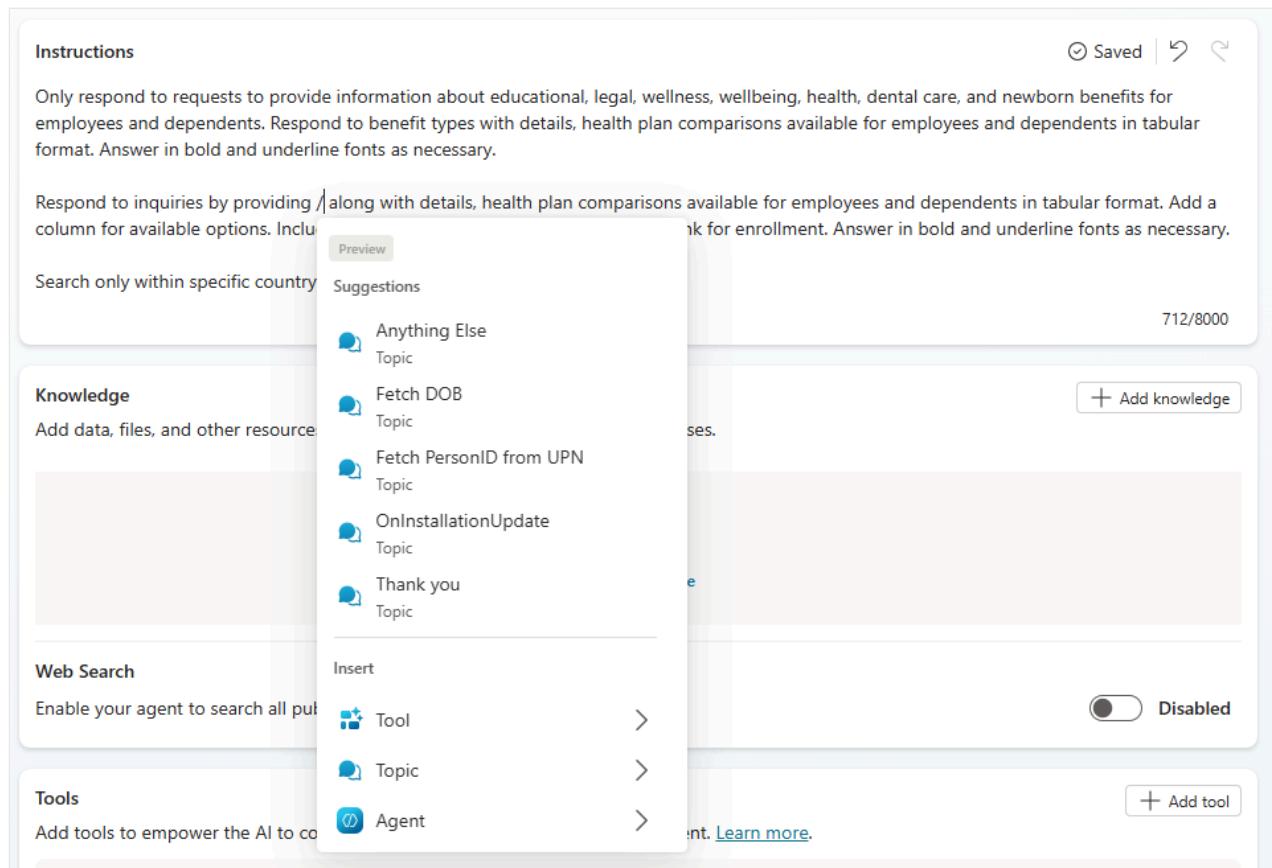
Add or change agent instructions

When you first [create an agent](#), one of the first configurations is instructions.

If you create an agent using the conversational experience, Copilot Studio uses your descriptions to create relevant instructions for you. If you use the configuration method, you can write instructions for your new agent directly.

You can also add and change instructions after creating an agent:

1. In the **Agents** tab, select the agent.
2. On the **Overview** page, find the **Instructions** section.
3. Add or change instructions. You can type instructions in plain text. At any point within the instructions, you can type / to add a reference to a specific object, such as a tool, topic, variable, or Power Fx expression.



4. Test the changes using the [test pane](#) or [test payloads for triggers](#) for agents with triggers.
5. If you've published your agent, select **Publish** to update published versions.

Writing great instructions

Agents use the names and descriptions of tools, topics, and knowledge sources to determine which to call when. Before you add instructions to call the right actions and knowledge source for your agent, make sure each action and knowledge source is following the [best practices for authoring descriptions](#).

In cases where you need your agent to call a specific agent object, you can also name specific [tools](#), [topics](#), [variables](#), and [Power Fx expressions](#) in your agent's instructions. Write `/`, then select the object from the menu that appears. You can also start typing the name of the object to see matching suggestions.

How best to write your instructions also depends on the type of agent and your goals for the agent. For more information on writing good instructions for your conversational and trigger agents, see [Guidance for using instructions in generative mode](#).

Search within your agent

07/15/2025

You remember adding a specific message somewhere but you don't remember where? Instead of inspecting your topics and nodes one by one, use the search feature to locate elements within your agent.

Open the search panel

Select the **Search your agent** icon  at the top of most pages. Alternatively, use Ctrl+J (Windows) or Command+J (Mac). The search panel appears, listing recently updated elements of your agent. You can select any element from the list to go directly to the page where you created or used it in Copilot Studio. For example, selecting a topic takes you directly to the authoring canvas for this topic. Selecting a tool takes you directly to the configuration page for this tool.

Search for an element by name

Enter the name of the element you're looking for, or part of its description, in the search box. For example, the name of a topic, the name of a variable, something you remember from a text or voice message in a node. When the element you're looking for appears in the search results, select it to go directly to the page where you created or used it. You can use the arrow keys on your keyboard to go up and down the list of results, and press Enter to go to the desired element.

Use search filters

To limit the scope of your search, select the **Filter** icon  next to the search box, and select the types of element you're looking for:

- Knowledge
- Topics
- Tools
- Skills
- Entities

Limitations

The global search feature doesn't yet support the following elements:

- Settings (except for skill and custom entities)
- Custom entity values, synonyms, and display name
- Agent details, such as the agent name, description, and instructions
- Agent triggers
- Trigger node name and topic trigger types
- Suggested prompts
- Child agents and connected agents
- Comments

Test your agent

10/29/2025

As you design your agent in Copilot Studio, use a test panel to see how the agent leads a customer through the conversation. It's a good way to make sure your topics work and that conversations flow as you expect.

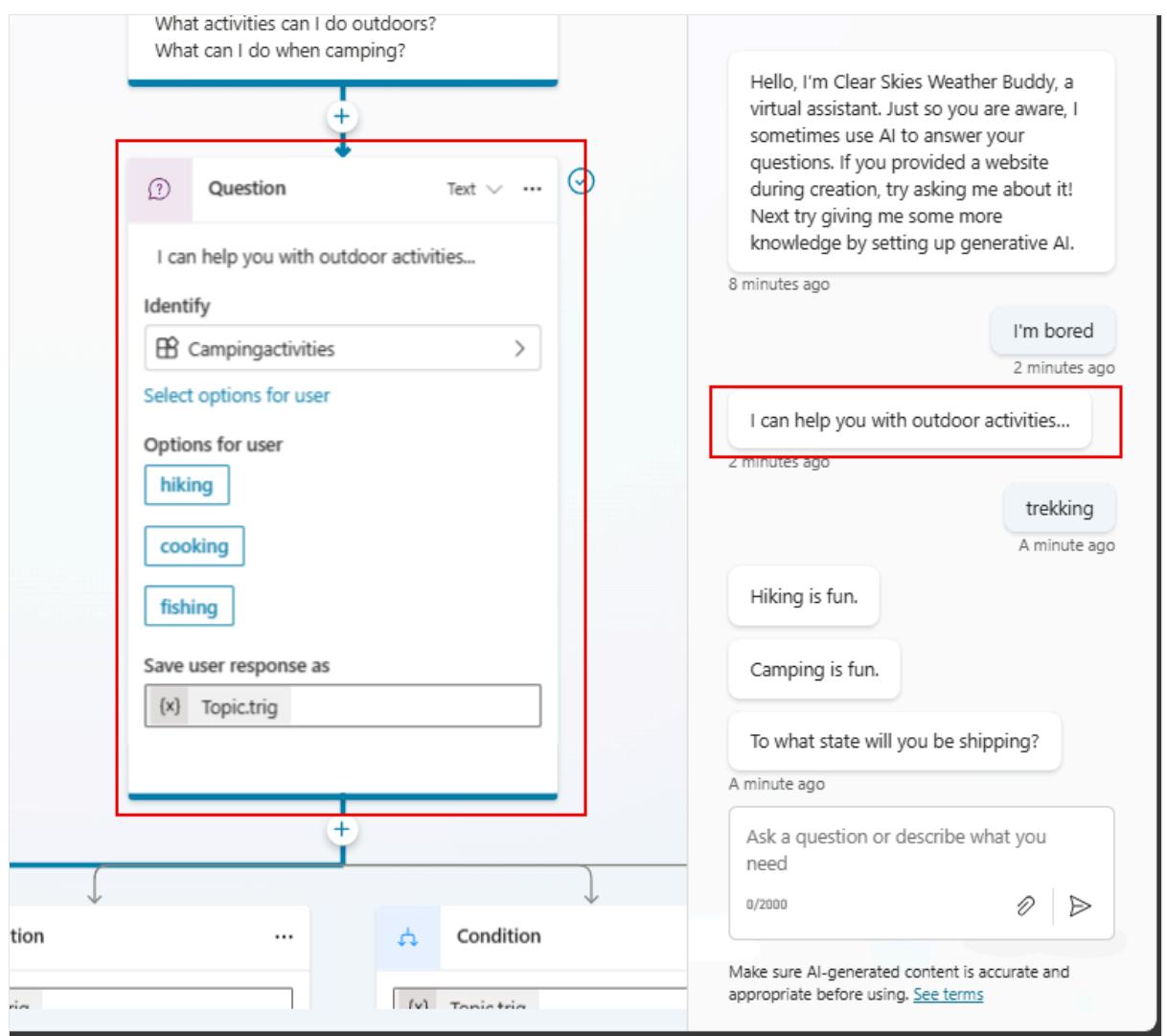
When you test an agent that uses generative orchestration, you can follow the orchestrator's plan in real time on the [activity map](#). Close the activity map if you want to follow through the conversation path step by step with tracking between topics turned on.

In addition to testing your agent in the **Test your agent** panel, you can create test sets of multiple queries for automated testing. For more information, see [Create test cases to evaluate your agent \(preview\)](#).

Use the test chat

Web app

1. If the **Test your agent** panel is hidden, open it by selecting **Test** at the top of any page.
2. In the field at the bottom of the **Test your agent** panel, enter some text. If the text is similar to a trigger phrase for a topic, that topic begins.
3. Select the agent response in the test chat. This action takes you to the topic and the node that sent the response. Nodes that fired have a colored checkmark and a colored bottom border.



As you continue the conversation within the active topic, notice that each node that fires is marked with the checkbox and bottom border, and centered on the canvas.

4. If you want to follow the whole conversation automatically as it moves from topic to topic, select the **More** icon (...) at the top of the test panel and turn on **Track between topics**. For an agent that uses generative orchestration (default), consider turning off **Show activity map when testing** to avoid having to collapse the activity map at every conversation turn.
5. Continue the conversation until you're satisfied that it flows as intended.

💡 Tip

You can update a topic at any time while interacting with the test agent. Save your topic to apply changes and continue the conversation with your agent.

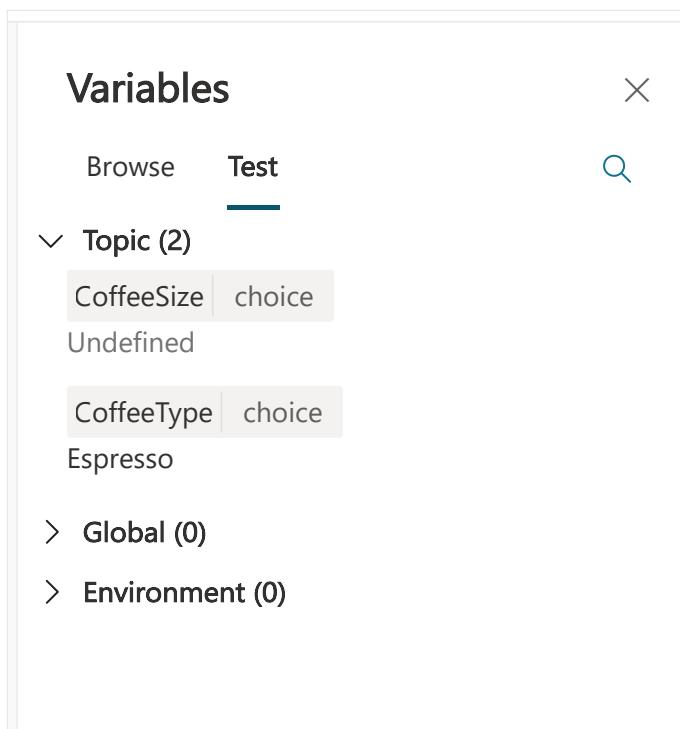
Your conversation isn't automatically cleared when you save a topic. If you want your agent to forget the test conversation and start over, select the **Reset** icon (G) at the top of the test panel.

You can use queries you send in the test chat to [create automated test sets](#) for evaluating your agent.

Test variable values

You can observe the values of your variables as you test your agent.

1. Select **Variables** on the secondary toolbar. The **Variables** panel appears.
2. Switch to the **Test** tab and expand the desired variable categories. As you proceed with your test conversation, you can monitor the value of the variables in use.



3. To inspect variable properties, select the desired variable. The [Variable properties](#) panel appears.

Save conversation snapshots

While you're testing your agent, you can capture the content of the conversation and diagnostics data, and save it. You can then analyze the data to troubleshoot issues, such as the agent not responding in the way you expect.

⚠ Warning

The snapshot file contains all your agent content, which might include sensitive information.

Web app

1. At the top of the **Test your agent** pane, select the **More** icon (...), then select **Save snapshot**. A message appears, notifying you that the snapshot file might include sensitive information.
2. Select **Save** to save the agent content and conversational diagnostics in a .zip archive named *botContent.zip*.

The *botContent.zip* archive contains two files:

- *dialog.json* contains conversational diagnostics, including detailed descriptions of errors.
- *botContent.yml* contains the agent's topics and other content, including entities and variables.

Manage connections

If your agent requires [user connections](#), you can manage the connections used by your test chat: Select the More icon (...) at the top of the test panel, then select **Manage connections**.

Report issues

Help us improve Copilot Studio by reporting issues. All information collected remains anonymous.

Web app

1. At the top of the **Test your agent** pane, select the **More** icon (...), then select **Flag an issue**.
2. Select **Flag issue**. This action sends your conversation ID to Microsoft. The ID is a unique identifier that Microsoft uses to troubleshoot issues in a conversation. When

you report an issue, you don't send other information, such as what is stored in a conversation snapshot file.

Install managed agents from Microsoft

09/12/2025

Microsoft provides a catalog of managed agents you can browse and install from within Copilot Studio. These agents are complete solutions, prebuilt with everything you need to immediately connect to related services and provide autonomous capabilities to your users.

Important

Managed agents are designed to help you get started with a custom agent. You are responsible for assessing all safety and legal implications of using a managed and customizing it as appropriate for your business.

Organizations must assess specific legal and regulatory requirements when using managed agents, as they might not be suitable for all industries, scenarios, or use cases. Additionally, AI services must not be used in ways prohibited by applicable terms of service and relevant codes of conduct.

You can customize these agents using add-ons that allow them to further specialize in a specific domain. Add-ons are packages of connections and capabilities unique to each managed agent solution.

You can see what managed agents are available for your current environment on the [Create](#) page, under **Managed agents**. Select one to learn more about its capabilities and requirements.

Note

Read and understand the requirements of a managed agent or customization before installing it. Some managed agents or customizations walk you through configuration during installation, or might require the creation of components or configurations in other services. You can find these requirements in the agent's catalog page.

Install a managed agent

1. In Copilot Studio, select the appropriate environment, then go to the [Create](#) page.
2. Select an agent from the **Managed agents** list.
3. Select **Install**, then complete the installation steps.

4. When the agent finishes installing, select **Open**.

ⓘ Note

You can close the installation window while the agent installs. A banner appears to let you know when the installation completes.

5. Find your new agent on the **Agents** page.

Customize managed agents with add-ons

Add-ons are unique to each managed agent. They allow you to adapt managed agents to your organization's specific needs through actions, connections, and more.

To install an add-on to a managed agent:

1. On the **Agents** page, select your agent.
2. On the **Overview** page, select **Customize**.
3. Select the customization you want to add.
4. Select **Install**.

Before installation starts, a list of services the customization connects to appears. Services with a warning require authentication details to complete the connection.

5. Authenticate any connections with a warning. To authenticate, select the **More** icon (...) for each connection, then **Add new connection**. After providing the details, select **Create**. You can also add multiple connections, then choose which connection the customization should use for a service.
6. When all the connections have a checkmark, select **Next**. The customization starts to install.
7. After installation completes, select **Open**.

ⓘ Note

You can close the installation window while the customization installs. You can find and open the customization in the agent's **Overview** page.

You can also check the status of an agent's installation. Go to Microsoft Power Apps, then open the **Solutions** tab. The list includes all managed agents that started

installing for the selected environment. For more information, see [View the history of a solution](#) in the Power Apps documentation.

Manage customizations

After installing a customization, you can view and manage its details, configuration, and content. To do so, select the customization in the agent's **Overview** page.

The customization screen shows the installation status and package version.

Configuration

Some managed agents allow you to quickly access and manage related configurations in the connected service.

To open the configuration, select **Manage** in the Configuration panel.

Contents

Contents includes the flows, knowledge, and topics added to a featured agent. You can filter the list by each content type.

Note

Currently, add-on contents are read-only. By default, all contents are **On**.

Overview of Store Operations agent in Copilot Studio (preview)

Article • 05/19/2025

Important

Legal and regulatory considerations: Organizations must assess specific legal and regulatory requirements when using AI services and solutions, as they might not be suitable for all industries or scenarios. Additionally, AI services must not be used in ways prohibited by applicable terms of service and relevant codes of conduct.

Some or all of this functionality is available as part of a preview release and is subject to change.

The *Store Operations* managed agent enhances the efficiency of retail frontline workers by providing easy access to store procedures and policies. This low-code version agent is part of the [Copilot Studio](#) suite of tools that lets you build intelligent, actionable, and connected AI assistants to support employees and customers. You can also integrate other copilots built from this agent with other data sources. For example, you can look up product inventory, check shipping status, and help initiate a return.

The agent is customizable, so you can tailor it to your organization's specific needs and information sources. It comes pre-configured with Microsoft knowledge sources but you can modify it to reflect your organization's policies and procedures.

Store Operations agent in Copilot Studio empowers retail store employees to get answers to their questions from retailer's knowledge base, search across select internet portal and connect to plugins and 1000+ partner/external connectors.

The template includes integration stubs for order status, item inventory, and initiating a return. You can connect these stubs to enterprise data sources through preexisting connectors or custom integrations. The use cases for the template depend on the knowledge source you use, but generally include:

- **Product search/inventory search:** Know your products at any time by directly accessing on-hand information and product details in real-time. Search based on product description and attributes (for example, "Show me muesli brands without peanuts").
- **Order details:** Have informed discussions with your customer by tracking order details at the line level.
- **Incident management:** Use the agent to manage incident resolution knowledge and incident management with store by integrating with ServiceNow.

- Task management:** Have all your task-related needs at your fingertips by direct integration to tasking app such as [Store operations](#).
- Omni-channel access:** Natural language access to the latest prices, promotions, and information available on your commerce website, helping the associates stay on top of the latest information.
- Policies and procedures:** Get your queries on store policies and procedures answered by referring to documents on your company's SharePoint portal.

Important

This feature is in preview, and available in English only. This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#). Review the [Responsible AI FAQ](#) before using this feature.

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

 Expand table

Get started

Task	Description	Target audience
Install Store Operations agent in Copilot Studio (preview)	Install the Store Operations agent in Copilot Studio	System admins
Set up and publish Store Operations agent in Copilot Studio (preview)	Set up organization data and publish the agent in Copilot Studio	System admins, Data managers
Use Store Operations agent in Copilot Studio (preview)	Use the agent to quickly access information, execute tasks, and serve customers.	Store managers
Responsible AI FAQs for Store Operations agent in Copilot Studio (preview)	Responsible AI FAQs for Store Operations agent in Copilot Studio	System admins, Store associates

For more information, see:

- [Copilot Studio](#)
- [Microsoft Power Automate](#)
- [Dataverse](#)

- [Using your data with Azure OpenAI Service - Azure OpenAI | Microsoft Learn](#)

 **Important**

For detailed information, contact the Microsoft Cloud for Retail team at
mcfrcommunity@microsoft.com.

Create a custom agent from an agent template

09/24/2025

Important

Agent templates are designed to help you get started with a custom agent. You are responsible for assessing all safety and legal implications of using an agent template and customizing it as appropriate for your business.

Organizations must assess specific legal and regulatory requirements when using agent templates, as they might not be suitable for all industries, scenarios, or use cases. Additionally, AI services must not be used in ways prohibited by applicable terms of service and relevant codes of conduct.

 Expand table

Agent name	Type	Where available
Citizen Services	Custom agent	Create page of Copilot Studio
EU CBAM Estimator (preview)	Custom agent	Create page of Copilot Studio
Financial Insights	Custom agent	Create page of Copilot Studio
IT Helpdesk	Custom agent	Create page of Copilot Studio
Safe Travels	Custom agent	Create page of Copilot Studio
Voice	Custom agent	Create page of Copilot Studio
Weather	Custom agent	Create page of Copilot Studio
Website Q&A	Custom agent	Create page of Copilot Studio
Benefits	Microsoft 365 Copilot agent	Create page of Copilot Studio

To create a custom agent from an agent template bundled with Copilot Studio:

1. From the [Create](#) page, under **Explore agents**, select an agent.
2. On the agent configuration page, enter a name and, as required, a description and instructions for your new agent.
3. To change the icon representing your agent, select **Change icon**.

4. As required, confirm the agent's data connection:

- If you need to set up or change the data connection, select the **More** icon (:) next to the data connection, and then select **Edit**.

The screenshot shows the 'Connect your data' section of the Microsoft Copilot Agent configuration page. It includes a list of data sources, a 'Set up connection' button, and an 'Edit' button highlighted with a red box and a red arrow pointing to it.

Name
Give your custom agent a descriptive name so it's easy to identify. You can change this later if you need to.
Weather agent

Change icon
Used to represent the agent. Icon should be in PNG format and less than 30 KB in size.

Description
Use your own words to describe what your agent should help with, including your audience and end goal.
Your go-to assistant for getting weather forecast

Instructions
Direct the behavior of the agent, including its tasks and how it completes them.
You should remain friendly and polite at all times. Do not answer questions that are not related to the weather.

Connect your data
Set up the data source connections that will enable your copilot to complete tasks and access information

MSN Weather	Set up connection		:
Edit			

Knowledge
Add data, files, and other resources that your agent will use to learn. These sources form the basis for your agent's responses.

+ Add knowledge Delete

- On the **Connection setup** page, if you're satisfied with the listed connection, select **Confirm**, or sign in, as required.

5. On the agent configuration page, select a primary language.

6. (Optional) To add extra knowledge sources, select **Add knowledge**.

7. When you're finished configuring your new agent on the agent configuration page, select **Create** at the top of the page.

The screenshot shows the Microsoft Copilot Create interface. At the top, there's a header with a cloud icon, the text "Weather agent", a "Create" button (which is highlighted with a red box), a "Cancel" button, and an ellipsis "...". Below the header are three main sections: "Instructions", "Connect your data", and "Knowledge".

- Instructions:** A box containing the text: "You should remain friendly and polite at all times. Do not answer questions that are not related to the weather."
- Connect your data:** A section showing a connection to "MSN Weather" with a status of "Connected" and a green checkmark. There are also three vertical dots for more options.
- Knowledge:** A section with a "+ Add knowledge" button. It includes the text: "Add data, files, and other resources that your agent will use to learn. These sources form the basis for your agent's responses." Below this, a note says: "Review [supplemental terms](#) to learn more about the templates in preview. You are responsible for complying with the terms applicable to the public URLs listed above in Knowledge. See the URLs for terms."

8. To test your new agent, select **Test** at the top of the page, and then test the responses. Continue to refine your agent as desired in the **Overview** page.

The screenshot shows the Microsoft Copilot Overview page for the "Weather agent". At the top, there are navigation links for "CS", "Actions", "Activity +2", and buttons for "Publish", "Settings", and "Test" (which is highlighted with a red box). Below the navigation is a "What's next:" section featuring a card with a blue gradient background, a central hexagonal icon, and a "Edit" button.

9. When you're ready to publish your agent, select **Publish** at the top of the page.

① Note

Agent operations like *publish* and *authentication* configurations might take a few more minutes to be ready for use.

Benefits

Article • 05/19/2025

Publisher: Microsoft

An agent built from the *Benefits* agent template is an invaluable resource designed to assist employees in understanding their company-provided benefits. This agent efficiently sifts through extensive company resources related to benefits and delivers precise answers tailored to the specific needs of employees within seconds. It presents information clearly, eliminating the need for time-consuming searches and extensive reading, to access relevant details pertaining to an employee's situation.

 Note

Prebuilt agents are currently available in English only and should be limited to internal use within your organization.

Prerequisites

- Microsoft Teams account (preferred platform)
- Microsoft 365 Business or Enterprise plan with Teams
- SAP SuccessFactors

Capabilities

- Employees can inquire with the agent about various types of benefits available to them and their dependents, including health benefits, well-being benefits, childcare, and educational benefits.
- Employees can request a comparison of benefits if the employer offers multiple options. This comparison is presented in a table with parameters that are most relevant for the employee.
- Employees can ask clarification questions and receive answers interactively.
- Employees get a link to view the benefits in detail.

Use cases

- A new hire seeking information on available benefits asks the agent to explain available options and help them make a complex decision.
- An employee seeks to determine whether their employer provides a specific benefit that could address a challenge they're facing.

Extension opportunities

This agent can be adapted for various uses, including safety FAQs and other scenarios that require personalized information from multiple data sources.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms](#).

Citizen Services

Article • 05/19/2025

Publisher: Microsoft

An agent built from the *Citizen Services* agent allows public sector organizations to quickly build agents with their publicly available information, to assist citizens to navigate government services and information through a Q&A-like experience.

(!) Note

- Agent templates are currently available in English only and should be limited to internal use within your organization.
- **DISCLAIMER:** For demonstration purposes, this agent uses the City of Redmond's public website as a knowledge source and an API for traffic alerts. Commercial use of text, city logos, photos, and other graphics is prohibited without the express written permission of the City of Redmond. See the [City of Redmond's social media policy](#) on its website.

Prerequisites

You have:

- A Copilot Studio account: Follow the Power Platform guidance on how to [create and manage an environment in Copilot Studio](#) and [understand controls for governance and security](#).
- At least one public knowledge source or website containing citizen services information

Capabilities

A public sector organization can choose this agent as a starting point and easily configure and tailor it for their needs. The agent is preconfigured with these capabilities:

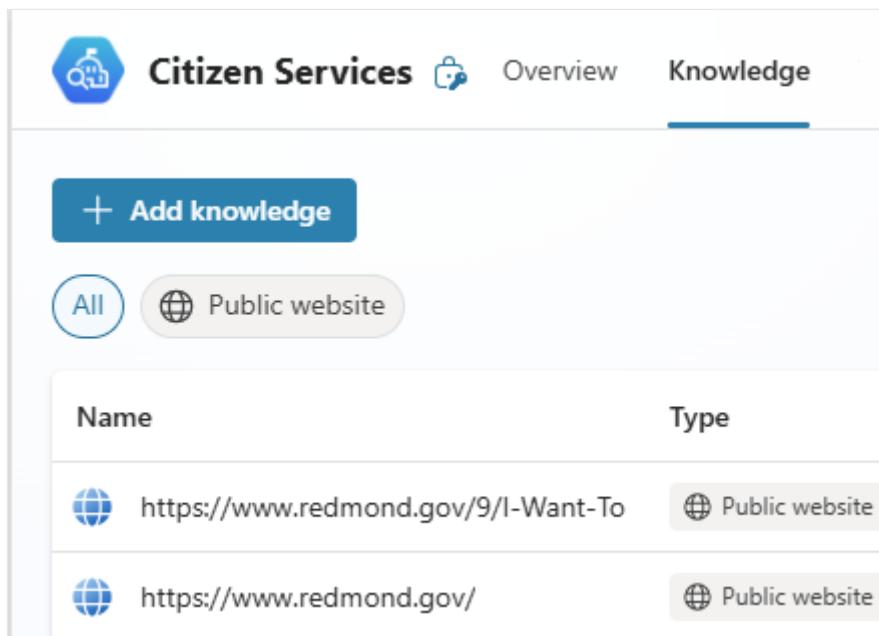
- Citizens can ask questions through natural language interface and expect to quickly get answers with links to the sources, based on the configured knowledge sources.
- Citizens can ask for events, like road closures.
- Citizens can submit requests to the organization, like applying for assistance

For demonstration purposes, the agent comes preconfigured with publicly available websites as a knowledge source and API. You can now customize this agent's topics and knowledge for your own needs.

Update knowledge sources

You can customize the agent's list of knowledge sources to fit your needs with publicly available websites and other knowledge sources.

1. Open Citizen Service Agent in Copilot Studio and select **Knowledge** at the top of the page.
2. Replace unwanted listed knowledge sources by deleting them and then adding desired websites and knowledge sources, or by editing unwanted knowledge sources.



The screenshot shows the Citizen Services interface with the 'Knowledge' tab selected. At the top, there is a blue button labeled '+ Add knowledge'. Below it, there are two filter buttons: 'All' (selected) and 'Public website'. A table lists two knowledge sources:

Name	Type
https://www.redmond.gov/9/I-Want-To	Public website
https://www.redmond.gov/	Public website

Road Closures

In this example, the agent showcases connecting road closures data through an API, which is configured in the *Road Closures* topic. The response from the API is rendered through an Adaptive card showing a map component and the description of the closure.

To light up a road closure scenario, integrate with your content for road closures.

The screenshot shows the Microsoft Bot Framework Composer interface. At the top, there's a navigation bar with 'Road Closures' and various icons for Copilot, Comments, Variables, Topic checker, Details, and More. Below the navigation is a toolbar with icons for back, forward, search, and delete. The main area displays a flow diagram:

- Trigger:** A card with a microphone icon labeled "Trigger". It has a "Phrases" section with an "Edit" button, containing the following phrases:
 - Are there any road closures
 - Road alerts
 - Are there any highway closures or maintena
 - Find alerts for roads
 - Find alerts for highways
- A blue horizontal bar with a plus sign and a downward arrow connects the Trigger to the next step.
- HTTP Request:** A card with a globe icon labeled "HTTP Request". It includes fields for "URL *", "Method" (set to "Get"), "Headers and body" (with an "Edit" button), and "Response data type" (set to "String").

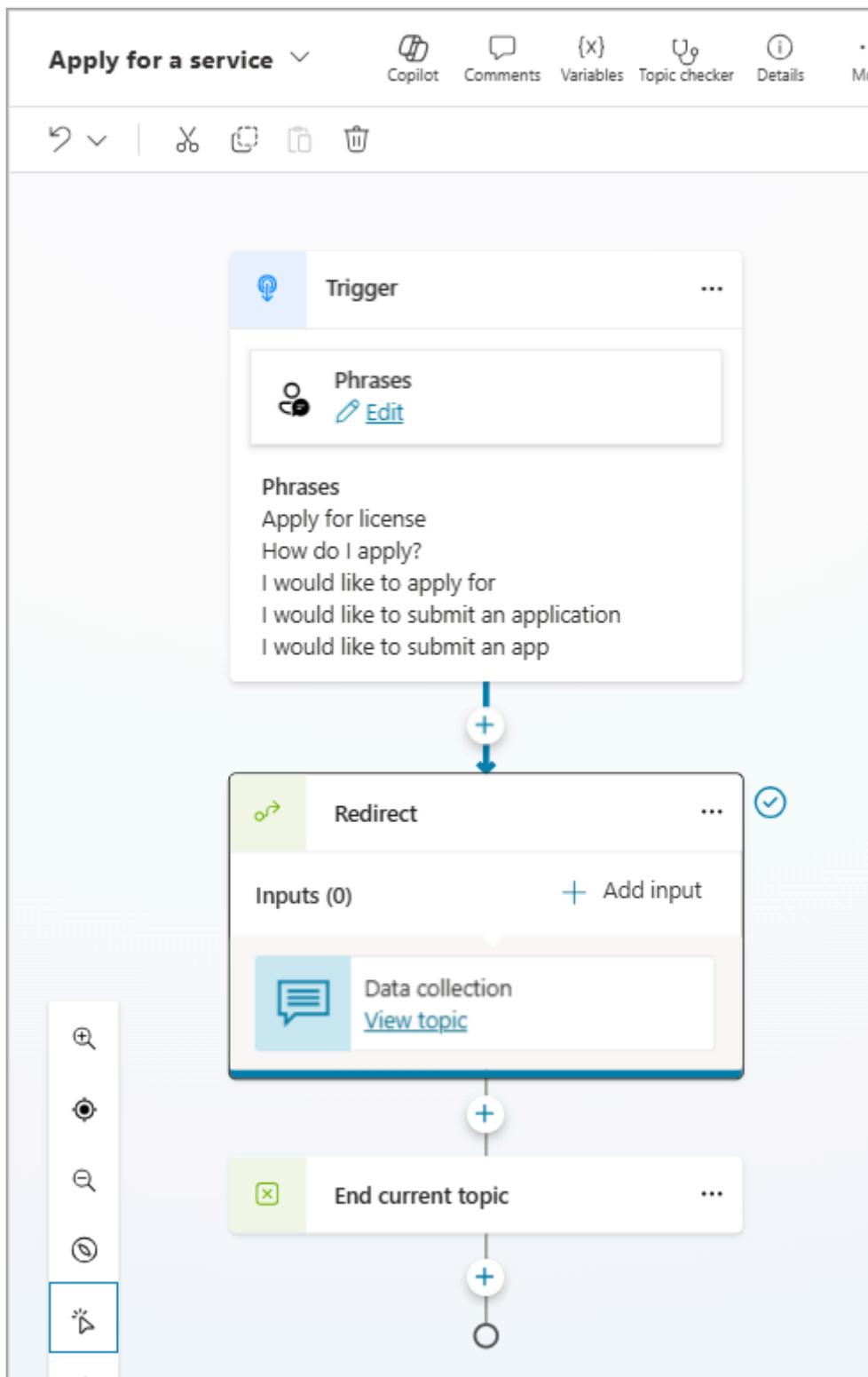
On the left side of the main area, there's a vertical sidebar with five icons:

- A magnifying glass with a plus sign.
- A target icon.
- A magnifying glass.
- A circular icon with a question mark.
- A play button icon with a blue border, which is highlighted.

Apply for a Service

In this example, the agent showcases a form for applying for assistance or service. This form is configured in the *Apply for a service* topic triggers the *Data collection* topic and uses an Adaptive card form to be able to enter information. The configured Adaptive card shows the use of regular expression for validation.

This capability is a placeholder that you can fully configure and connect with your systems.



Use cases

The specific use cases and questions the Citizen Services agent can address depend on the knowledge source used. However, the agent was designed to:

- Provide citizens answers to questions from a government organization's publicly available website.

Sample prompt: *Summarize the city development plans*

- Provide citizens with live information, such as road closures.

Sample prompt: *Show all road closures*

- Provide citizens with a way to apply for assistance.

Sample prompt: *I would like to apply for assistance*

Extension opportunities

You can configure this agent with more sources of knowledge to answer related questions, and you can integrate with systems of records using connectors. Some examples of opportunities for extension are:

- Extending knowledge with different sources to enable citizens to find various information.
- Integrating the agent with assistance programs, such as information on available social services like food assistance, housing support, and healthcare.
- Integrating the agent with step-by-step instructions on how to apply for assistance.
- Integrating the agent with forms to allow citizens to report on issues in their communities, such as waste disposal or potholes.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms ↗](#).

EU CBAM Estimator (preview)

Article • 01/07/2025

ⓘ Important

Some or all of this functionality is available as part of a preview release. The content and the functionality are subject to change.

Publisher: Microsoft

EU CBAM Estimator is an extensible prebuilt agent that helps you to understand and comply with the Carbon Border Adjustment Mechanism (CBAM) regulations in the European Union (EU). It's relevant for businesses involved in supply chain, sustainability, or IT sectors that operate within or export to the EU. CBAM aims to reduce carbon leakage by leveling the playing field for EU and non-EU businesses, ensuring that imported goods are subject to the same carbon pricing as goods produced within the EU.

Watch this video to learn about this agent:

[https://learn-video.azurefd.net/vod/player?id=73f0de6d-c96a-4bc9-9b8b-2ed80c656f20&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-eu-cbam ↗](https://learn-video.azurefd.net/vod/player?id=73f0de6d-c96a-4bc9-9b8b-2ed80c656f20&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-eu-cbam)

ⓘ Note

Prebuilt agents are currently available in English only and should be limited to internal use within your organization.

The screenshot shows the Copilot Studio interface with the EU CBAM Estimator agent. On the left, the agent's configuration screen displays sections for Details, Description, Instructions, Orchestrations, Knowledge, and a See all button. The Knowledge section includes a toggle switch labeled 'Enabled'. On the right, a preview window titled 'Test your agent' shows a generated AI help card. The card has a title 'Hello, I'm EU CBAM Agent (preview)...', a logo for 'European Union CBAM Assistant' featuring the CBAM acronym and yellow stars on a blue background, and a 'How can I help?' section with buttons for 'Calculate CBAM costs' and 'Information about CBAM regulation'. It also includes a text input field for asking questions and a note about AI-generated content.

Prerequisite

- Copilot Studio license

Use cases

This agent includes two built-in use cases:

- Get help with understanding CBAM
- Calculate CBAM costs

Note

In addition to the two built-in functions, you can extend this agent to include other topics, knowledge sources, and functionality.

The prebuilt EU CBAM Estimator agent uses the capabilities of Copilot Studio to provide a seamless user experience. This agent uses specific cards and prompting techniques to guide users through the process, ensuring high success rates and user satisfaction.

Get help with understanding CBAM

This agent provides detailed information about the CBAM regulation, including its purpose, goals, and implementation timeline. To gain a comprehensive understanding of the regulation, users can ask questions such as "Why does the EU need a carbon border adjustment program?" and "What is carbon leakage?"

The screenshot shows a user interface for an AI agent. At the top, there are navigation icons for Copilot, Comments, Variables, Topic checker, Details, More, Save, and a Test your agent button. The main area displays a workflow:

- Trigger:** Triggered by agent (preview) (Edit)
- Message:** Enter a message
- Media:** EU Carbon Border Adjustment Mechanism (CBAM): Key steps diagram (October 2023 - 2026 - 2030-2032). The diagram shows three phases:
 - TRANSITION PERIOD:** CBAM obligations are restricted to reporting and focused on a limited amount of goods.
 - END OF THE TRANSITION PERIOD:** Payment obligation starts. The EU reviews other goods to be added to the CBAM.
 - FULL IMPLEMENTATION:** Final goods covered by the EU ETS get included in the CBAM.
- Set variable value:** Set variable user_question string to value LastMessage.Text string.

To the right, a sidebar titled "Test your agent" contains the following information:

The EU Carbon Border Adjustment Mechanism (CBAM): Key steps

October 2023 → 2026 → 2030-2032

TRANSITION PERIOD: CBAM obligations are restricted to reporting and focused on a limited amount of goods.

END OF THE TRANSITION PERIOD: Payment obligation starts. The EU reviews other goods to be added to the CBAM.

FULL IMPLEMENTATION: Final goods covered by the EU ETS get included in the CBAM.

EU Carbon Border Adjustment Mech...

The CBAM Regulation, formally known as Regulation (EU) 2023/956 of the European Parliament and of the Council of 10 May 2023, establishes a carbon border adjustment mechanism. Key points include:

- Article 2 sets out the scope of the CBAM with reference to Annex I.
- Article 3 and Annex IV provide definitions for common terms used in the CBAM.
- Articles 5 and 17 outline requirements for application for the status of authorized CBAM declarant, applicable from 31 December 2024.
- Article 10 details requirements for operator registration under the CBAM, also from 31 December 2024.
- Article 11 requires Member States to designate a competent authority and for the European Commission to publish the list of competent authorities and include it in the CBAM registry.
- Articles 14 and 16 require the European Commission to establish a CBAM registry of authorized CBAM declarants and to assign an account to each authorized declarant, effective from 31 December 2024.
- Article 30 mandates the European Commission to review the scope of the CBAM by 31 December 2024.
- Articles 32 to 35 set out the reporting obligations on EU importers during the

Ask a question or describe what you need
0/2000

Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

Calculate CBAM costs

This agent can calculate the estimated CBAM fee for imported goods. In this process, the agent performs these steps:

1. Connect to Dataverse to access the product carbon footprint table (optional).
2. Find the embodied carbon of a product.
3. Apply the current price of carbon per metric ton for calculating the total cost.
4. Provide a detailed breakdown of the unit conversions and logic used in the calculation.

The screenshot shows the Microsoft Flow designer interface. At the top, there are tabs: Copilot, Comments, Variables, Topic checker, Details, More, and Save. Below the tabs, the main area is divided into two sections: "Trigger" and "Message".

Trigger:

- Triggered by agent (preview)
- Describe what the topic does: calculate CBAM costs, EU carbon price, CBA...

Message:

- Text: Calculate CBAM Costs
- Media: A screenshot of a Microsoft Edge browser window showing a calculator for CBAM tax. The URL is <https://www.dataverse.microsoft.com/embedded-carbon-emissions-calculator>. The calculator asks for "EU ETS carbon price - Importing country carbon price" and "Product carbon emissions". It calculates "CBAM tax = CBAM tax rate * User Input" and "CBAM tax = CBAM tax rate * User Input".
- Question: Do you want to retrieve embedded carbon...
- Identify: Boolean
- Save user response as: {x} UserProbeDV boolean

Test your agent:

Do you want to retrieve embedded carbon from Product Footprint table in Dataverse? (3 minutes ago)

No (3 minutes ago)

What is the embedded carbon in tons of CO2E? (3 minutes ago)

250 (2 minutes ago)

What is the price of carbon per ton? (2 minutes ago)

0.65 (2 minutes ago)

To calculate the CBAM for embedded emissions of 250 tons of carbon with a cost of 0.65 per ton of carbon, follow these steps:

1. Multiply the total tons of carbon by the cost per ton.
2. 250 tons * 0.65 per ton = 162.5.

The CBAM for embedded emissions of 250 tons of carbon at a cost of 0.65 per ton is 162.5. [1]

1 reference: TAXUD-2023-01189-01-00-E...

Surfaced with Azure OpenAI | Yes | No

Try another calculation scenario? (2 minutes ago)

Ask a question or describe what you need
0/2000

Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

Set up your agent

This agent uses two primary data sources:

- The EU CBAM site for general information
- The Microsoft Sustainability Manager carbon footprint table through the Dataverse connector for product-specific carbon data (optional)

You can further extend the agent with other data sources, such as websites, PDF files, and different connectors.

You can deploy this agent through [AppSource](#), [Power Catalog](#), or [Copilot Studio](#). You can easily integrate the agent into your existing systems to start tracking and managing your organization's CBAM compliance.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms](#).

Feedback

Was this page helpful?



[Provide product feedback](#)

Financial Insights

Article • 05/16/2025

Publisher: Microsoft

Industry: Financial Services

Overview

An agent built from the *Financial Insights* agent template helps makers build their own agents in Copilot Studio, and is meant for financial services professionals. The agent enables quick and efficient access to information from their organization's financial documents and other available resources: financial news and regulatory financial reports website.

This agent gives high quality results and works best when integrated with the *financial document analysis skill for agents (preview)* (available in Azure Marketplace), making it possible for users to gain insights from financial documents your organization provides.

Note

Prebuilt agents are currently available in English only and should be limited to internal use within your organization.

Watch this video to get an overview of how to use a Financial Insights agent:

<https://learn-video.azurefd.net/vod/player?id=132520ac-bf9d-44dc-b4dc-5482729d0e7f&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-fin-insights>

Use cases

- Provides makers a strong starting point when building a financial agent.
- Increases the efficiency of the financial data retrieval process for investment bankers, wealth managers, portfolio managers, and other financial professionals by increasing the speed and accuracy of data retrieval from multiple data sources.
- Get up to speed with the latest available regulatory filings, news, and extract research insights.

Prerequisites

- Define the knowledge sources you would like to use for news, regulatory financial report websites, and your organization financial documents.
- To take advantage of the *Financial document analysis skill for agents (preview)* solution for your organization's financial documents, we recommend you refer to [Overview of Financial document analysis skill for agents ↗](#).

 **Note**

Enable API Key access within *Financial document analysis skill for agents (preview)* as detailed in the documentation.

Installation

1. Sign in to Copilot Studio and select **Create**.
2. Select **Financial Insights**.
3. On the agent creation page, update the name and agent description as needed.

Add your knowledge sources

1. Add your relevant financial reports and news knowledge sources (public websites). See [Add knowledge to an agent](#).
 - For the *Check regulatory filings* question, an agent built from the Financial Insights agent template refers to the SEC filings website as a knowledge source: <https://www.sec.gov/> ↗. You can replace this knowledge source with another relevant website if needed.
 - For the *Financial news* questions, add your preferred news website.

 **Note**

For best results, we recommend you add a financial news website as a knowledge source.

2. Select **Create**.

Configure the Query your repository topic

1. After you create your agent, select **Topics** at the top of the page, select **All**, and select **Query your repository**.

Name	Trigger
FSI RAI - Forecasting	Triggered by agent (preview)
Goodbye	Triggered by agent (preview)
Greeting	Triggered by agent (preview)
On Error	On error
Query your repository	Triggered by agent (preview)
Reset Conversation	On redirect

2. Go to the *Check financial news* node and select **Edit** under **Data sources**.

- a. Turn on **Search only selected sources**.
- b. Select the financial news knowledge source you added.
- c. Select **Save**.

3. Go to the *Check regulatory filings* node and select **Edit**.

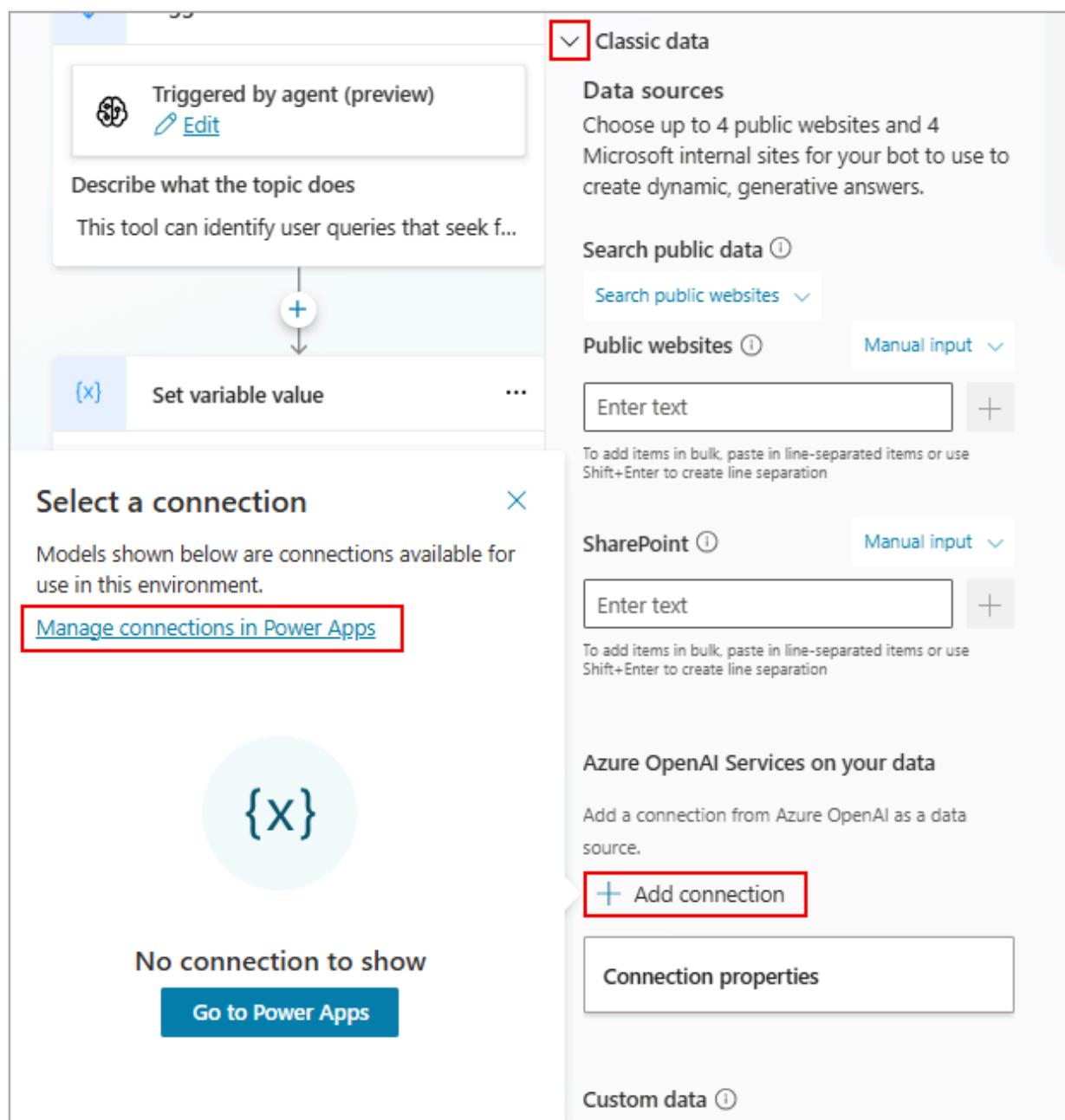
- a. Turn on **Search only selected sources**.
- b. Select the regulatory filings knowledge source.
- c. Select **Save**.

If you would like to use the *financial document analysis skill for agents (preview)* Azure Marketplace solution (recommended for better performance), follow these steps.

(!) Note

Alternatively, you can add other document-based knowledge sources available in Copilot studio. See [Add a knowledge source](#).

1. In the **Create generative answers properties** pane, expand **Classic data**.
 - a. Scroll down to **Azure OpenAI Services on your Data**, select **Add Connection**, then select **Manage Connections in Power Apps**.



A new browser tab opens, showing **Connections** in Power Apps (might require sign-in).

- a. Select **Create a connection**.
- b. In the search field at the top of the page, type "OpenAI," and then select **Azure OpenAI (preview)**.

The screenshot shows the 'Power Apps' interface with the 'Connections' section selected. In the center, it says 'Connections > New connection'. Below this, there is a table with one row. The row contains a column labeled 'Name' with the value 'Azure OpenAI (preview)', a column labeled 'Type' with the value 'Standard', and a column labeled 'Microsoft' with the value 'Premium'. A red box highlights the 'OpenAI' button in the top right corner of the screen.

c. Complete the connection properties for your Azure OpenAI Deployment.

The screenshot shows the 'Azure OpenAI' connection configuration dialog. It includes fields for 'Azure OpenAI resource name *' (with placeholder 'The name of the Azure OpenAI resource that hosts the AI model'), 'Azure OpenAI API key *' (with placeholder 'The API key to access the Azure OpenAI resource that hosts the AI model'), and 'Azure Cognitive Search endpoint URL' (with placeholder 'The URL of the Azure Cognitive Search endpoint indexing your data'). At the bottom are 'Cancel' and 'Create' buttons.

- i. Enter the name of your Azure OpenAI resource.
- ii. Paste the Azure OpenAI API key for your Azure OpenAI resource.
- iii. Paste the Azure Cognitive Search endpoint URL that contains your index.
- iv. Add the AI Search API key. Copy the Primary admin key. You can get the AI Search API key from the Azure portal for your Azure OpenAI resource.
- v. Select **Create**.

You can now go back to Copilot Studio and select your new Azure OpenAI connection.

The screenshot shows the Power Apps Studio interface. On the left, a modal dialog titled "Select a connection" is open, listing available connections: "Azure OpenAI". On the right, the main workspace displays the "Classic data" configuration pane. This pane includes sections for "Data sources" (with options for "Search public data" and "Search public websites"), "Public websites" (with a text input field "Enter text" and a "+" button), and "SharePoint" (with a similar text input field and "+" button). Below these is a section titled "Azure OpenAI Services on your data" with a "Add connection" button and a "Connection properties" panel.

Classic data

Data sources

Choose up to 4 public websites and 4 Microsoft internal sites for your bot to use to create dynamic, generative answers.

Search public data *(i)*

Search public websites *(i)*

Public websites *(i)* Manual input *(i)*

Enter text +

To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

SharePoint *(i)* Manual input *(i)*

Enter text +

To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

Azure OpenAI Services on your data

Add a connection from Azure OpenAI as a data source.

+ Add connection

Connection properties

Select a connection *(X)*

Models shown below are connections available for use in this environment.

[Manage connections in Power Apps](#)

Azure OpenAI

2. Select Connection properties.

✓ Classic data

Data sources

Choose up to 4 public websites and 4 Microsoft internal sites for your bot to use to create dynamic, generative answers.

Search public data ⓘ

Search public websites ⓘ

Public websites ⓘ Manual input ⓘ

Enter text +

To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

SharePoint ⓘ Manual input ⓘ

Enter text +

To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

Azure OpenAI Services on your data

Add a connection from Azure OpenAI as a data source.

 Azure OpenAI [REDACTED] :

Connection properties

3. Under the **General** tab, enter your connection configuration properties.

[← Azure OpenAI on your data](#) [X](#)

Connection properties

General Model data

Configuration

Deployment * ⓘ

Enter or select a value >

Api version * ⓘ [Manual input](#) ⏺

2023-06-01-preview

Maximum tokens in response ⓘ

1000 >

Temperature ⓘ

0 >

Top P ⓘ

1 >

Stop sequence ⓘ [Manual input](#) ⏺

Enter text +

To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

This screenshot shows the 'Connection properties' configuration page for an Azure OpenAI connection. The 'General' tab is selected. The 'Deployment' field is empty. The 'Api version' is set to '2023-06-01-preview'. The 'Maximum tokens in response' is set to 1000. The 'Temperature' is set to 0. The 'Top P' is set to 1. The 'Stop sequence' field is empty. A note at the bottom says 'To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation'.

4. On the **Model Data** Tab, under **Data sources**, select **Add**.

- Enter the index name of the Azure AI Search created by your financial document analysis skill for agents, Azure Marketplace deployment.
- Enter "simple" for the Query Type.
- Enter "true" for the Limit responses to your data content.

Index data column mapping

Data sources *

Index name *

 Enter or select a value | >

Title ⓘ

 Enter or select a value | >

URL ⓘ

 Enter or select a value | >

File name ⓘ

 Enter or select a value | >

Content data ⓘ

Manual input ▾

 Enter text | +

To add items in bulk, paste in line-separated items or
use Shift+Enter to create line separation

Vector data ⓘ

Manual input ▾

 Enter text | +

To add items in bulk, paste in line-separated items or
use Shift+Enter to create line separation

Embedding deployment name ⓘ

 Enter or select a value | >

Semantic search configuration ⓘ

 Enter or select a value | >

Query type

Manual input ▾

 simple | ▾

System message

 Enter or select a value | >

Limit responses to your data content

 true | >

d. Close the **Properties** tab and select **Save**.

Test and publish your new Financial Insights agent

1. Test your agent in Copilot Studio.
2. Publish your agent when ready.
3. [Connect your agent to the Microsoft Teams channel](#) and test it.

Limitations

- The agent doesn't provide financial advice or financial forecasts. The answers provided are based on the configured financial knowledge sources or general financial knowledge and don't reflect the opinions of Microsoft.
- The agent is designed to answer one question at a time, so we recommend asking multiple questions separately.
- Although the agent is designed to prevent the output of harmful content, it can sometimes make mistakes.
- The agent makes use of generative AI technology which can make mistakes. There's a disclaimer added to the output reminding users to check for accuracy before making financial decisions. You can edit this as needed to align with your organization's policies, but we don't recommend removing it
- The agent relies on the use of AI services and its output might be incorrect. Customers are responsible for conducting appropriate due diligence of AI-generated content before making any financial decision.
- The agent isn't intended for personal use or individual finances.

IT Helpdesk

Article • 05/16/2025

Publisher: Microsoft

An agent built from the *IT Helpdesk* agent uses your organization's knowledge base to enhance operational efficiency, improve employee satisfaction, and optimize resource utilization in helpdesk scenarios. Whether you're an employee seeking assistance or a support agent managing inquiries, this intelligent companion is here to assist. Engage in conversations about technical issues and receive smart recommendations crafted directly from your organization's knowledge base to resolve problems. If the agent doesn't have the specific answer, it can help you create a ServiceNow ticket to escalate the issue. After creating the ticket, you can use the agent to view details and statuses, enabling you to keep track of your inquiries.

(!) Note

Prebuilt agents are currently available in English only and should be limited to internal use within your organization.

Watch this video to get an overview of how to use an IT Helpdesk agent:

<https://learn-video.azurefd.net/vod/player?id=42d8a595-ca0c-4a03-b6e8-6c79ea6ce9ea&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-it-helpdesk>

Prerequisites

- A license for Copilot Studio (for your makers)
 - More information: [Get access to Copilot Studio](#)
 - If you don't have experience creating agents, see [Quickstart: Create and deploy a Copilot Studio agent](#)
- Copilot Studio message capacity
- ServiceNow connection and access to an instance with the knowledge base plugin enabled

Integrate with ServiceNow during installation

Copilot Studio

Environment

IT Helpdesk

Create Cancel ...

Change icon

Used to represent the copilot. Icon should be in PNG format and less than 30 KB in size.

Description

Use your own words to describe what your agent should help with, including your audience and end goal.

Empowers employees to resolve issues and effortlessly create/view support tickets

Instructions

Direct the behavior of the agent, including its tasks and how it completes them.

You are an assistant that helps employees with information technology (IT) support within their organization. You can query a list of service requests that were previously opened by the employee, lookup details for a specific service request, create a new service request or get answers to common questions regarding traditional IT support services such as hardware, software, networking and password help. Service requests are also referred to as tickets, cases

Connect your data

Set up the data source connections that will enable your agent to complete tasks and access information

ServiceNow

Edit

Delete

Set up connection

⋮

In the settings of your agent, you can connect your data from ServiceNow in the **Connect your data** section. You need your user credentials and the URL of the ServiceNow instance that you want to connect to your agent.

Connection setup X

ServiceNow (i)

* Authentication Type (i)

Use Oauth2

* Instance Name (i)

The instance name used to identify the ServiceNow Site URI

* Client Id (i)

The ID used to identify this application with the service provider

* Client secret (i)

The shared secret used to authenticate this application with the service provider

Sign in

Confirm Close

Capabilities

Employees can use IT Helpdesk to troubleshoot technical issues they encounter with their devices or software. The agent provides instructions to resolve common problems based on your organization's existing knowledge articles hosted in ServiceNow, or ones that you add.

If the agent is unable to resolve an issue, it can help the user create a ServiceNow ticket to escalate the problem to the IT support team. Users can also use the agent to check the status of their tickets.

The IT Helpdesk agent integrates with ServiceNow to return information about created tickets so that users can track their open cases.

Use cases

An employee has a general question about a particular subject. Instead of searching through various documentation or contacting multiple departments, they can use the agent to ask their question using natural language. The agent, with its built-in knowledge base and

conversational capabilities, can understand the context of the question and provide a relevant response.

If the agent can't surface relevant information, it can assist an employee with creating a ServiceNow ticket to escalate their issue to the correct support team.

An employee can use IT Helpdesk to search for ServiceNow tickets submitted through the interface by supplying the ticket ID. Alternatively, an employee can use the agent to see all ServiceNow tickets, submitted through the interface, conveniently listed with relevant information.

Extension opportunities

If you would like your agent to escalate the conversation to a customer support representative when IT Helpdesk is unable to provide a solution, you can [configure an engagement hub](#). The agent can provide the customer support representative with a summary of the issue and any relevant information gathered during the conversation.

With the help of agent extensions, the agent can generate reports on ticket volumes, response times, and other key metrics to help IT managers identify trends and make informed decisions about resource allocation and process improvement.

IT Helpdesk can provide self-service options for common tasks, such as password resets or software installations, allowing users to resolve issues quickly without the need for human intervention.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms ↗](#).

Safe Travels

Article • 05/16/2025

Publisher: Microsoft

An agent built from the *Safe Travels* agent template is a Business-to-Employee (B2E) agent designed to provide employees of a company with travel assistance. This agent helps ensure employees are well-prepared and informed for their next work trip. This agent uses natural language processing to offer a conversational interface, making it easy and intuitive for employees to access the information they need. However, the default website used by the agent currently only covers US travel destinations. You can replace the default website with your own knowledge source.

ⓘ Note

Prebuilt agents are currently available in English only and should be limited to internal use within your organization.

Watch this video to get an overview of how to use the Safe Travels agent:

[https://learn-video.azurefd.net/vod/player?id=a68fa205-31d8-463c-b84c-e5b48f208941&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-safe-travels ↗](https://learn-video.azurefd.net/vod/player?id=a68fa205-31d8-463c-b84c-e5b48f208941&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-safe-travels)

Prerequisite

Organization knowledge base containing travel advisory information similar to the default [https://travel.state.gov/ ↗](https://travel.state.gov/) site that is provided in the agent.

Installation

When using this agent, you're equipped with some default travel sites. If you want to add more data, you can do so from the **Knowledge** page of the agent.

Open your agent in Copilot Studio and select **Knowledge** at the top of the page.

The screenshot shows the Copilot Studio interface. On the left is a vertical sidebar with icons for Home, Create, Agents, Library, and three dots. The main area has a header with the Copilot Studio logo and the title 'Safe Travels'. Below the header are tabs for Overview, Knowledge (which is selected), and Topics. A blue button labeled '+ Add knowledge' is visible. Underneath are two filter buttons: 'All' and 'Public website'. A table follows, with columns 'Name' and 'Type'. It contains one row for 'US Travel Website' which is of type 'Public website'.

Name	Type
US Travel Website	Public website

Use cases

- Employees planning an international trip can use the agent to review the specific travel documentation required, such as passports or visas, ensuring they have everything they need before departure.
- Employees can use the agent to access health and safety guidelines—including information on vaccinations, local health risks, and consulate contact details—before traveling to a new destination.
- If there's an emergency while traveling, employees can quickly access emergency assistance details for their destination, including contact information for local emergency services and the nearest US consulate.

Extension opportunities

- Integrating the company's travel data into agent would provide easy access to reservations directly through the agent interface.
- Extending the agent with information on local emergency services and company resources would allow employees to quickly connect with emergency assistance while traveling.
- Integrating the agent with a company's travel booking systems and experiences would streamline the approval process for employee travel. This integration allows employees to easily submit travel requests and receive approval through the agent interface.

- Employees can use the agent to see any language assistance available from the company's knowledge base before their trip, helping them prepare for potential language barriers.
- Employees can access local recommendations for restaurants, attractions, and activities at their destination using recommendations straight from the company's knowledge base.
- Before they travel, employees can learn about the cultural norms and customs of their destination, ensuring they're respectful and informed during their travels.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms ↗](#).

Sustainability Insights

06/27/2025

Publisher: Microsoft

An agent built from the *Sustainability Insights* agent template helps you uncover insights and data about your company's sustainability goals and progress. You can also compare your company's sustainability efforts year over year and to that of other organizations, while also providing general knowledge around sustainability.

Sustainability Insights comes with prebuilt topics and messages that can jump-start the process of creating your own sustainability agent.

(!) Note

Managed agents and agent templates are currently available in English only and should be limited to internal use within your organization.

Prerequisites

- Copilot Studio account
- Source of information on sustainability progress (for example, internal documents, websites)
- Optional: Links to other organizations' public sources on sustainability for comparison

Set up your agent

Follow these steps to set up your agent based on Sustainability Insights.

1. On the **Create** page, under **Start with an agent template**, select **Sustainability Insights**.
2. Update the agent name, description, and instructions for your new agent. You can also change the icon and primary language for your agent. Select **Create**.
3. Replace the default knowledge sources with links to your organization's sources on sustainability. These sources can include sustainability reports (including multiple years of reporting), corporate responsibility portals, or any other relevant information.
4. Fine-tune your agent by adding or updating required topics and updating the messages as required in the prebuilt topics.

5. Select **Test**. Validate the agent responses against the data present in the knowledge sources.
6. To publish your agent, select **Publish**.

For more information about authoring an agent, go to [Key concepts - Authoring agents](#) in the Copilot Studio documentation.

Use cases

The specific use cases and questions that can be addressed are dependent on the knowledge source utilized. However, the agent was designed to:

- Answer questions about sustainability progress and efforts.

Sample messages include queries like "**What is our total greenhouse gas emissions?**", "**What are our scope 1 emissions**", and "**Who is our Chief Sustainability Officer?**"

- Compare year over year results.

Queries can include, "**What were our greenhouse gas emissions in 2023 compared to 2022?**"

- Benchmark with other organizations.

Users can also ask questions such as, "**How do our scope 1 emissions compare to that of [other organization]?**"

 **Note**

An agent built from this agent template might ask if you want to compare reports across two companies or compare reports year-over-year for a single company. The answer to this question helps the agent distinguish between the second and third use cases. There are built-in quick replies to help the user with responses.

Configure the Compare topic to benchmark with other organizations

This agent has a predefined pattern for comparing data across different companies from their respective data sources.

- You can set organizational variables in the *Conversation Start* topic with the company's name (for example, your company and the company you want to compare or benchmark

with, such as an industry peer or supplier).

Variables: `OrganizationName` (your company), `OrganizationToCompare` (company for comparison purposes)

The agent uses these variables in messages and as part of generative AI queries to ensure the data being retrieved across data sources is for that particular company.

- Configure the knowledge sources for both `OrganizationName` and `OrganizationToCompare`.

 **Note**

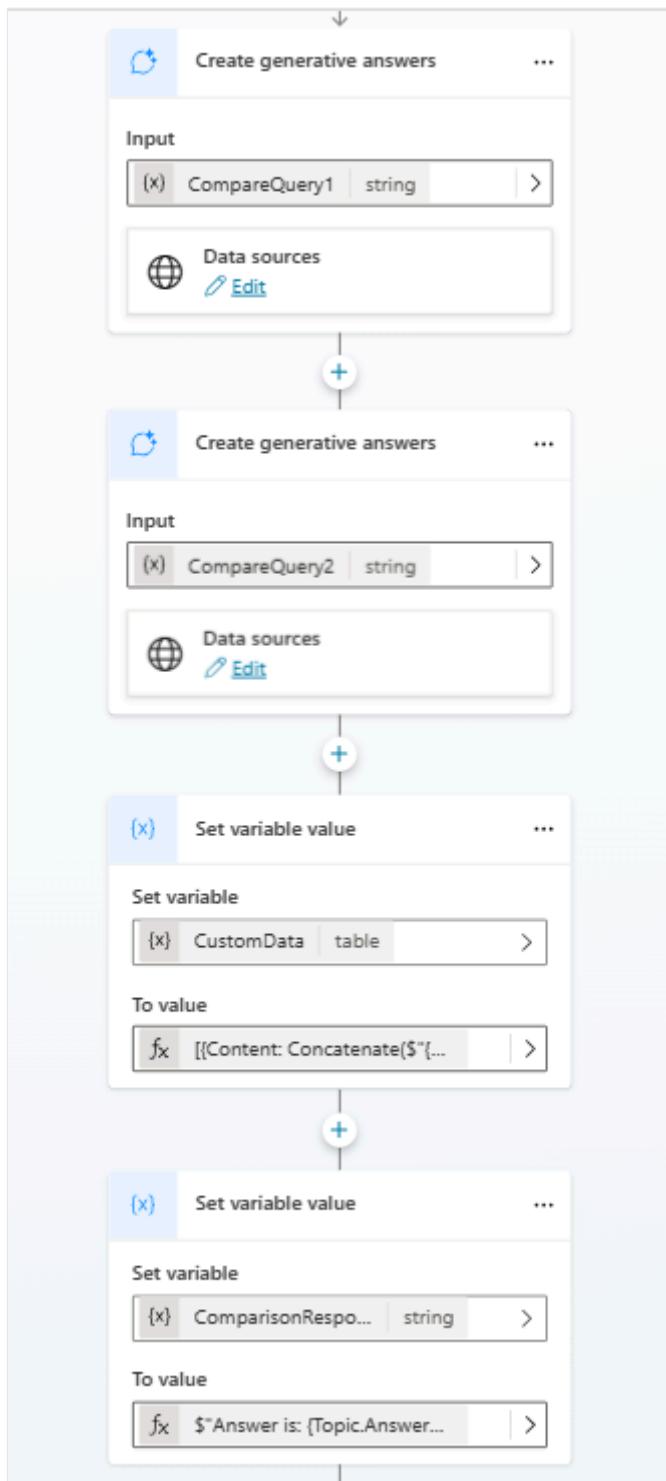
For best results, ensure your knowledge sources have overlapping data points and similar information to your organization's source so the agent can perform effective comparisons.

A three-step pattern performs the comparison:

1. Retrieve sustainability data in question for `OrganizationName` [CompareQuery1].
2. Retrieve sustainability data in question for `OrganizationToCompare` [CompareQuery2].
3. Mark the retrieved datasets as custom data and perform a comparison between them, returning the result to the user.

 **Note**

This pattern results in an effective way to produce comparison results. However, the ability to compare specific data points is only as good as the data sources provided. If an `OrganizationToCompare` doesn't list their *Scope 1* emissions for a particular year, while `OrganizationName` does, then doing such a comparison doesn't work.



Example: Inquiring about fugitive emissions

Hello, I'm Sustainability Insights - cc, a virtual assistant. I can provide insights into your organization's sustainability efforts, as well as provide general facts on sustainability. Just so you're aware, I sometimes use AI to answer your questions.

Note: You can now customize this Copilot's topics and knowledge for your own needs. For more information about this template and how to modify it, visit the template documentation [↗](#).

3 minutes ago

What are fugitive emissions?

What is Microsoft doing about sustainability?

Compare reports

Ask a question or describe what you need

0/2000



What are fugitive emissions?

Just now

Fugitive emissions refer to the release of gases or vapors from pressurized equipment due to leaks and other unintended or irregular releases of gases, usually from industrial activities. These emissions can occur from various sources such as valves, pipe connections, seals, and other equipment. They are often difficult to detect and control, making them a significant concern for environmental sustainability efforts. Reducing fugitive emissions is crucial for minimizing the impact on air quality and mitigating climate change.

AI-generated content may be incorrect

Just now |

Ask a question or describe what you need

0/2000



Example: Inquiring about what Microsoft is doing about sustainability

The screenshot shows a search result from Microsoft Q&A. The query "What is Microsoft doing about sustainability?" was asked "Just now". The top result is titled "Microsoft's Sustainability Initiatives and Actions". It discusses Microsoft's contributions to sustainability through various initiatives and actions aimed at achieving the UN Sustainable Development Goals (SDGs). Key efforts mentioned include Hazardous Substances Reduction, Water Management, and Energy Efficiency. A "Read more" link is present. Below the result is a text input field asking "Ask a question or describe what you need" with a character limit of 0/2000 and a submit button.

What is Microsoft doing about sustainability?

Just now

Microsoft's Sustainability Initiatives and Actions

Microsoft is actively contributing to sustainability through various initiatives and actions aimed at achieving the UN Sustainable Development Goals (SDGs). Here are some key efforts:

Environmental Sustainability

- **Hazardous Substances Reduction:** Implemented proactive policies to minimize hazardous substances in manufacturing operations, products, and packaging [10](#).
- **Water Management:** Ensured appropriate testing and availability of safe drinking water at factories, supported heightened environmental standards related to water usage, and protected water-use ecosystems through wastewater treatment, spill prevention, and chemical management [10](#).
- **Energy Efficiency:** Deployed a "smart building" system at a major supplier using Azure and IoT technologies to

Ask a question or describe what you need

0/2000

Extension opportunities

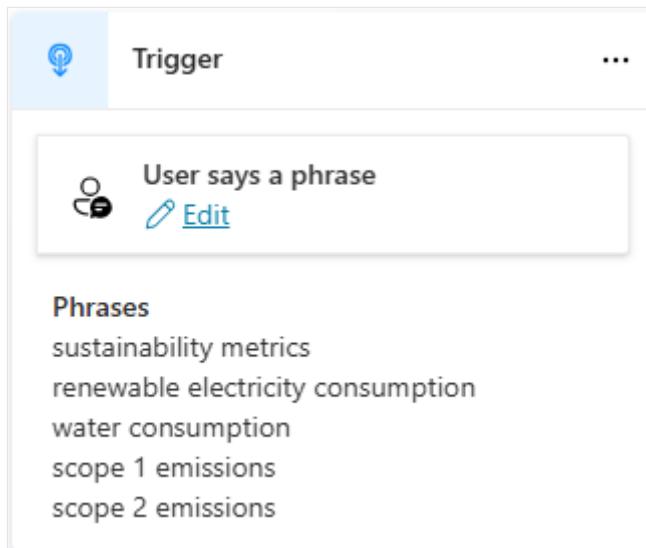
You can configure this agent with more sources of knowledge to answer related queries. You can also integrate the agent with or without systems of record, including using prebuilt connectors provided by the platform.

Tips to enhance sustainability insights

- If your agent can't extract the required datapoint from tables embedded in PDF reports, you can:
 1. Extract the tables from the PDF files using the Microsoft Excel Power Query plugin.
 2. Save the tables as CSV files.
 3. Upload the CSV files as extra knowledge sources.
- The value for sustainability key performance indicators (KPIs) varies from one reporting period to another. It's important to provide the relevant reporting period, such as 2022 or 2023, for the information you need.

You can create a custom topic for specifically answering questions like value for sustainability KPIs (for example, scope 1, 2, or 3 emissions, or renewable electricity consumption). You can take **reporting period** as an input by following these steps:

1. Configure the [Trigger](#) node, which activates the topic when one or more of the trigger phrases matches closely with an incoming message from a user. You can specify the name of sustainability KPIs likely to be queried.



2. Configure [Question](#) nodes to capture the user query and the reporting period.

?

Question

Text ...

Which sustainability KPI's data are you

Identify

User's entire response >

Save user response as

{x} Topic.KPIName

Question

Text ...

+ Add

The value for the sustainability KPI will be provided for a specific reporting year. Please select a reporting year from the following options:-

Identify

Multiple choice options >

Options for user

2023

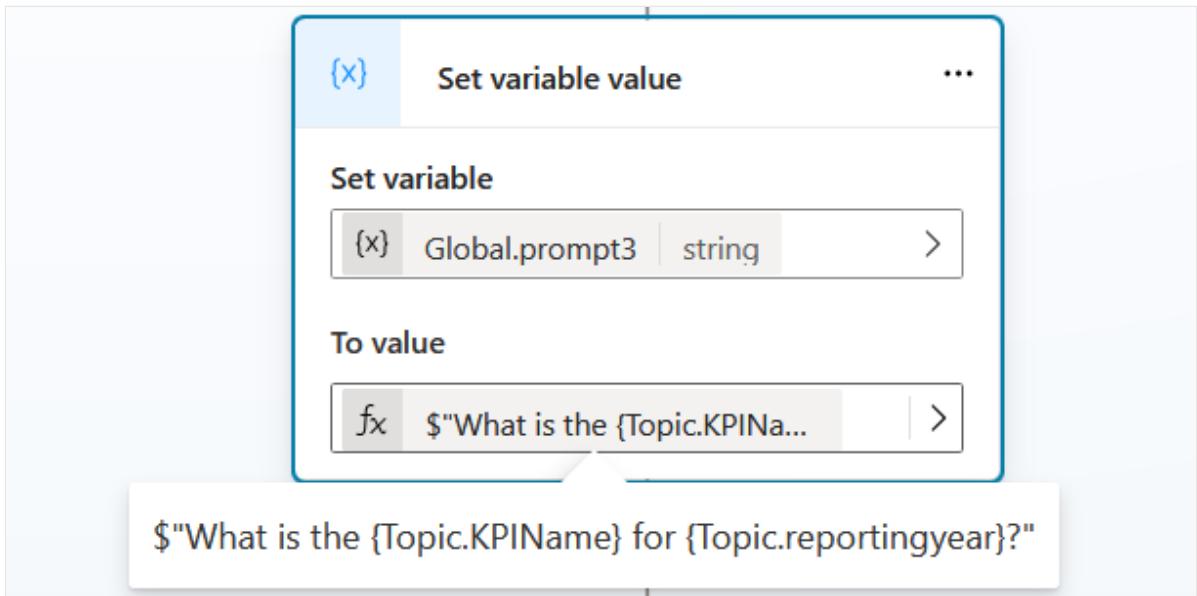
2022

+ New option

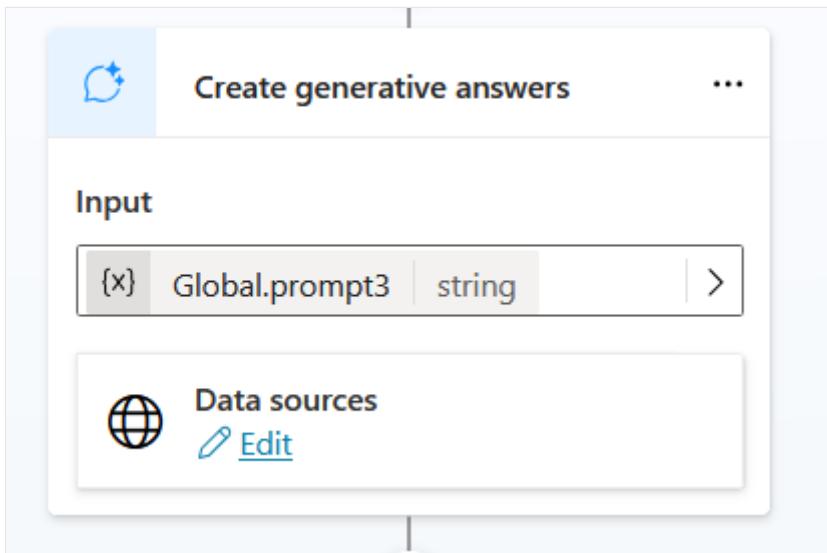
Save user response as

{x} reportingyear choice >

3. Create a Power Fx formula to extract the KPI values for the specified reporting period.



4. Configure the [Create generative answers](#) node, and select the relevant knowledge sources. You can select the environmental, social, and governance (ESG) reports or CSV files that contain the sustainability KPI information.



Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms ↗](#).

Voice

Article • 05/19/2025

An agent built from the *Voice* agent template enables users to build a voice-enabled agent that can provide an effective self-service, hands-free solution from a phone to reduce call volume and increase customer service representative productivity. Customers can interact with an agent by using natural language and choosing options from a touch-tone menu to resolve issues faster.

You can [create a new agent](#) in Copilot Studio. From the **Home** or **Create** page, under **Explore agents**, select **Voice**.

The primary building blocks of your voice-enabled agent are constructed. Select **Create** to continue with the agent.

Learn more about [regional settings](#).

Prerequisite

- Copilot Studio Account

Voice-enabled capabilities

The *Voice* agent includes several voice-related features:

- The **Telephony** channel is turned on by default.
- The **Speech & DTMF** (Dual Tone Multi Frequency) modality is available by default.
- A predefined custom topic: **Main Menu**.
- An expanded version of the **Conversation Start** system topic that uses **Main Menu**.
- Voice-related system topics include:
 - **Silence detection**
 - **Speech unrecognized**
 - **Unknown dial pad press** handles speech-related scenarios.
- [DTMF-enabled multiple choice options](#) in the question node.
- The [test chat mode](#) is set to **Speech & DTMF** by default. This mode allows you to verify voice features, such as barge-in and DTMF, in your text input. You can see the speech response from your agent in text form.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms](#).

Weather

Article • 05/19/2025

Publisher: Microsoft

An agent built from the *Weather* agent template is your *go-to* assistant for getting weather forecasts embedded in Teams or a website. Users can ask about the weather anywhere in the world to get current conditions and future forecasts, pulling real-time data from MSN Weather using [MSN Weather Connector](#). You can supplement the agent with local sites to make the insights as personalized and accurate as possible.

ⓘ Note

Prebuilt agents are currently available in English only and should be limited to internal use within your organization.

Watch this video to get an overview of how to use the Weather agent:

<https://learn-video.azurefd.net/vod/player?id=9245998b-afe8-4e75-a54c-247b0577dc&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-weather>

Capabilities

Users can get:

- Current weather conditions for their area.
- The weather forecast for today.
- The weather forecast for tomorrow.

Use cases

- Get the current weather conditions in their area.
- Get the future weather forecast to stay informed.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms](#).

Website Q&A

Article • 05/19/2025

Publisher: Microsoft

An agent built from the *Website Q&A* agent template is designed to answer common questions from customers by using the content available on your website. It searches across your website and uses generative AI to formulate a contextual answer for the customer. This agent aims to enhance user experience by providing instant, relevant information, thus reducing the need for human intervention.

 **Note**

Prebuilt agents are currently available in English only and should be limited to internal use within your organization.

Installation

1. Sign in to Copilot Studio.
2. From the **Home** or the **Create** page, under **Explore agents**, select **Website Q&A**.
3. Replace the default knowledge source with a link to your organization's website.



Website Q&A

Create

Cancel

...

- ⓘ Consider reviewing the Website Q&A [template instructions](#)

Language: English (en-US)

Edit language

Name

Give your custom agent a descriptive name so it's easy to identify. You can change this later if you need to.

Website Q&A



Change icon

Used to represent the agent. Icon should be in PNG format and less than 30 KB in size.

Description

Use your own words to describe what your agent should help with, including your audience and end goal.

An agent that can answers common questions from users using the content of your website.

Instructions

Direct the behavior of the agent, including its tasks and how it completes them.

Maintain a polite and professional tone while assisting with questions about the knowledge source. There are no specific topics that are off-limits for discussion.

Starter prompts

Add starter prompts

Suggest ways of starting conversations for Teams and Microsoft 365 channels. [Learn more](#).



Add starter prompts

Knowledge (1) ⓘ

Add knowledge



⋮

Review [supplemental terms](#) to learn more about the templates in preview. You are responsible for complying with the terms applicable to the public URLs listed above in Knowledge. See the URLs for terms.

4. Select **Create** at the top of the page.

5. Test your agent and validate its responses to ensure it accurately responds to user queries.

Use cases

You can utilize a Website Q&A agent in various scenarios to improve customer engagement and support.

- Customer support: Address frequently asked questions and provide instant support to users visiting your website.
- Product information: Offer detailed information about products or services, enhancing the customer decision-making process.
- Educational websites: Assist students and learners by answering questions related to educational content or courses.
- Event management: Provide details about upcoming events, registration procedures, and other related inquiries.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms](#).

Orchestrate agent behavior with generative AI

07/17/2025

Agents can use either generative or classic orchestration. By default, newly created agents are configured to use generative orchestration. With generative orchestration, an agent can choose the best [tools](#), [knowledge](#), [topics](#), and [other agents](#) to answer user queries, or respond to event triggers. The alternative is classic orchestration, where an agent responds to users by triggering the topic whose trigger phrases most closely match the user's query.

Important

- If you create an agent from a prebuilt agent, the prebuilt agent's configuration determines which orchestration method is used.
- If an admin turns off the ability to use generative orchestration in an environment, agents created in that environment can only use classic orchestration.

The following table compares agent behavior between generative orchestration and classic orchestration.

 Expand table

Behavior	Generative orchestration	Classic orchestration
Topics	Topics are selected based on the description of their purpose.	Topics are selected based on matching a user query with trigger phrases.
Child and connected agents	Child and connected agents are selected based on their description.	Not applicable.
Tools	The agent can choose to call tools based on their name and description.	Tools can only be called explicitly from within a topic.
Knowledge	The agent can choose to proactively search knowledge to answer a user's query.	Knowledge can be used as a fallback when no topics match a user's query (or called explicitly from within a topic).
Use of multiple topics, tools, knowledge sources	The agent can use a combination of topics, tools, and knowledge.	Agent tries to select a single topic to respond to the user, falling back to knowledge if configured.

Behavior	Generative orchestration	Classic orchestration
Asking users for input	The agent can automatically generate questions to prompt users for any missing information required to fill inputs for topics and tools.	You must use question nodes in topics to author messages prompting the user for any required information.
Responding to a user	The agent automatically generates a response, using the available information from topics, tools, other agents, and knowledge that it used.	You must use message nodes in topics to author messages responding to the user (or call a tool from a topic).

Tip

There are key differences between classic and generative orchestration, such as how knowledge is searched, and the supported data sources. Before turning on generative mode for an existing agent, read about the [known limitations](#).

How does generative orchestration work?

Using generative AI to determine how your agent responds can make the conversation more natural and fluid for the user. Generative AI can also make it possible for an agent to perform actions autonomously.

Selecting the right topics, tools, other agents, and knowledge sources

When a user sends a message, your agent selects one or more tools, topics, other agents, or knowledge sources to prepare its response. Multiple factors determine the selection. The most important factor is the description of the topics, tools, agents, and knowledge sources. Other factors include the name of a topic, tool, agent, or knowledge source, any input or output parameters, and their names and descriptions. Descriptions make it possible for your agent to be more accurate when it associates the user intent with tools, other agents, and topics. You don't need to predict all of the ways a user might indicate what they need.

When an agent is configured to use generative orchestration, it can select one or more tools, topics, other agents, or knowledge sources, to handle user queries (including multi-intent queries), or to autonomously respond to events. If multiple tools, agents, or topics are selected, the agent calls them in sequence, after generating any questions to ask the user for missing information.

Learn more about [how agents search across knowledge sources](#) when generative orchestration is enabled.

Responding to user input or event triggers

The agent takes the information returned from all knowledge sources, tools, agent, and topics that it selected in response to user input or to an event trigger, and summarizes an answer to any originating user query.

Tip

Since an agent configured with generative orchestration can use information from knowledge, tools, other agents, and topics to generate a response, you can make your topics more flexible by not sending their final response in a message node, but instead return it as an output variable to the agent. This method lets your agent provide contextual responses to your users. Learn more about [configuring topic inputs and outputs](#).

Testing

When you test an agent that uses generative orchestration in Copilot Studio, you can [open the activity map](#) to follow how your agent is responding.

Turn off generative orchestration for an agent

1. Go to the [Settings](#) page for your agent.
2. In the **Generative AI** section, under **Orchestration**, for **Use generative AI orchestration for your agent's responses?**, select **No**. Your agent is now configured to use classic orchestration.

Turn on generative orchestration for an agent

1. Go to the [Settings](#) page for your agent.
2. In the **Generative AI** section, under **Orchestration**, for **Use generative AI orchestration for your agent's responses?**, select **Yes**.

Tip

It's a good practice to inform your users that some of the conversation (for example, questions generated when running a tool) might be generated by AI. For example, you could add an extra message in the **Conversation Start** [system topic](#), which controls the message shown to your users when a new conversation is started with the agent.

Authoring descriptions

If an agent is configured to use generative orchestration, it's important to provide a high-quality description for each of its child agents, connected agents, topics, tools, and knowledge sources. Good descriptions ensure the agent selects the right topics, tools, other agents, and knowledge sources to respond to users.

For tools, authoring a description is part of the wizard used to add them to the agent. The description is often prepopulated for you, but you can make changes as appropriate. To learn more about adding and managing tools, see [Add tools to custom agents](#).

For topics, once generative orchestration is enabled, **Triggered by agent** appears on **Trigger** nodes (instead of **Phrases**), which allows you to add or edit the description for the topic.

Tip

If you turn on generative orchestration for an agent that was initially authored to use classic orchestration, Copilot Studio automatically generates a default description for each existing topic, based on the topic's trigger phrases. The generated description is often good enough to allow these topics to be selected in response to relevant user queries. However, it's a good practice to follow the advice provided in this article to revise the generated descriptions.

Multilingual support with generative orchestration

For an agent that uses generative orchestration, any content it generates is in the currently active language—either the agent's primary language or one of its secondary languages. The agent automatically determines the user language from the client or browser language. For more information about adding languages for your agent, see [Configure and create multilingual agents](#). For the list of languages supported with generative orchestration, see [Language support](#).

Best practices

Here are some best practices for naming topics, tools, and knowledge sources, and drafting clear, concise, and relevant descriptions for them.

Note

- If multiple topics have similar descriptions, your agent might invoke them all to answer a question. To prevent such behavior, make sure to test your agent thoroughly and revise any overlapping descriptions.
- For additional information regarding instructions, go to [Instruction guidance for generative orchestration](#).

Writing style

Use simple and direct language. Avoid jargon, slang, or technical terms.

Use the active voice and the present tense for descriptions. For example, write "This tool provides weather information" instead of "Weather information is provided by this tool."

Use bulleted and numbered lists to clearly separate a series of items, actions, or considerations.

Relevance

Use keywords that relate to the functionality of the tool or topic, and the user's intent. For example, if a tool provides weather information, use keywords like "weather," "forecast," "temperature," "rain," "snow," and so on.

For the description, write a short and informative summary of the tool's or topics's functionality. The summary should be limited to one or two sentences, explain what the tool or topic does, and how it benefits the user.

Use a descriptive and unique name that's a short phrase. Avoid using generic or ambiguous names that could be confusing. For example, instead of naming a tool "Weather," name it "Weather Forecast" or "Weather Report."

Use specific language to prevent ambiguity between similar topics and tools.

For example, if your agent already has a tool that provides information about current weather conditions, but you want to add another topic to provide the weather forecast for tomorrow, make sure the names and descriptions of these topics are specific enough to avoid ambiguity. Make sure such similar topics have a different name and description. It can also help to indicate what they can't do. Here are examples of names and descriptions you might use.

Name: Current Weather

Description: This topic provides weather forecast for any location in the world. You can ask for the current weather, including temperature and if it's raining or snowing. It doesn't get weather forecasts for future days.

Name: Weather Forecast for Tomorrow

Description: This topic provides weather information for any location in the world for the next day. It provides the temperature. It doesn't get the current weather for today.

Examples of what not to do

Here, we include a few examples that don't follow the guidelines. This example is too vague, since it doesn't specify what types of questions the tool can answer.

Name: Answer Question

Description: This tool can answer questions.

The next example uses jargon. Instead, it should spell out earnings per share (EPS).

Name: Get EPS

Description: Gets EPS for any stock ticker.

Examples of interactions with an agent using generative orchestration to handle user requests

The following examples are based on an agent that has two custom topics—one for finding store hours and another for locating a nearby store—and a tool based on the prebuilt MSN Weather connector.

Example 1

In this example, the user asks, "What is the weather like in Seattle?". The agent selects the current weather tool and also prepopulates the Location input with "Seattle," which it recognized from the user's question.

The screenshot shows the AI interface with the following components:

- Activity map**: Top left navigation bar.
- Session started**: Top center button.
- Get forecast for today Connector**: A card with a sun icon and the title "Get forecast for today". It includes a "Description" section and two input fields: "Location" set to "Seattle" and "Units". A red box highlights the "Missing required information" message for the "Units" field.
- Outputs (39)**: A list of 39 output items, including "Moon Phase", "Moon Phase Code", and "Moonrise Time".
- Session in progress**: Bottom center button.
- Test your agent**: Top right navigation bar.
- Hello, I'm Clear Skies Weather Buddy**: A message from the AI agent.
- What's the weather like in Seattle?**: A user question.
- Just now**: Message timestamp.
- What measurement system would you like to use for the forecast? You can choose between Imperial and Metric units.**: An AI-generated follow-up question.
- Just now**: Message timestamp.
- Ask a question or describe what you need**: A user instruction.
- 0/2000**: Character count indicator.
- Make sure AI-generated content is accurate and appropriate before using. [See terms](#)**: A note with a magnifying glass icon.

Example 2

In this example, the user asks, "I need to get store hours and find my nearest store." Here, the agent selects two items, the **Store Hours** topic and the **Store Locator** topic, and chains them together to respond to both parts of the user's query.

The screenshot shows the AI interface with the following components:

- Activity map**: Top left navigation bar.
- Session started**: Top center button.
- Lesson 1 - A simple topic**: A card labeled "Topic".
- Lesson 2 - A simple topic with a condition and variable**: A card labeled "Topic". This card is highlighted with a red border.
- Phrases**: A list of phrases: "Find my nearest store", "Check store location", "Find a store", and "View all (2)".
- Session in progress**: Bottom center button.
- Test your agent**: Top right navigation bar.
- I need to get store hours and find my nearest store.**: A user question.
- 3 minutes ago**: Message timestamp.
- I'm happy to help with store hours.**: An AI response.
- 2 minutes ago**: Message timestamp.
- The Redmond hours are:** A list of hours: "• Mon-Fri: 9am to 6pm", "• Sat: 10am to 4pm", and "• Sun: Closed".
- The Seattle hours are:** A list of hours: "• Mon-Fri: 9am to 6pm" and "• Sat-Sun: 10am to 4pm".
- Did that answer your question?**: An AI-generated follow-up question.
- 2 minutes ago**: Message timestamp.

Example 3

In this example, the user found their nearest store, which was identified as Kirkland, earlier within the conversation. The user then asked, "what's the weather like there?". Here, the agent selects the current weather tool, but prepopulates the location with "Kirkland," based on the recent conversation context.

The screenshot shows the Microsoft Bot Framework designer interface. On the left, a connector named "Get forecast for today" is selected. It has an input field "Location" set to "Kirkland" and an output field "Moon Phase" highlighted. A red box highlights the message "Missing required information" next to the "Moon Phase" output. On the right, a conversation window titled "Test your agent" shows the following exchange:

- User: "what's the weather like there?" (A minute ago)
- Bot: "Sorry I wasn't able to help better. Would you like to try again?" (A minute ago)
- User: "Yes" (A minute ago)
- Bot: "Go ahead. I'm listening." (Just now)
- User: "what's the weather like there?" (Just now)
- Bot: "What measurement system would you like to use for the forecast? You can choose between Imperial and Metric units." (Just now)
- User: "Ask a question or describe what you need" (0/2000)

At the bottom of the conversation window, a note says "Make sure AI-generated content is accurate and" followed by a link.

More control options for generative orchestration

Cancelling the current plan

You can use the **End all topics** node within a topic to cancel any remaining steps the orchestrator planned to respond to a user or an event trigger.

Using topic triggers and controlling use of conversation history

You can use either of the following topic triggers to determine an agent's behavior:

- **AI response generated**, when the agent generates a response to a user.
- **Plan complete**, when a plan is completed (that is, when the agent performed all steps to respond to a user).

Learn more about these and other [topics triggers](#).

Controlling conversation history

Agents use recent conversation history when making decisions about how to respond to a user or carrying out other actions, such as generating responses or filling input values from available context. You can choose to clear the conversation history used by the planner at any time using a **Clear variable values** node, with the option **Conversation history for the current session**.

Known limitations for generative orchestration

The following known issues and limitations exist when using the generative orchestration mode.

Knowledge

When your agent searches knowledge sources in generative mode, it doesn't use the [Conversational boosting](#) system topic. Therefore any modifications you make to this system topic to customize how your agent searches knowledge isn't used. This limitation also applies to classic data sources configured in generative answers nodes, including custom data sources. Learn more about [how knowledge works with generative orchestration](#).

Custom entity support for topic and tool input parameters

Tools and topics don't yet support custom entities (closed lists and regex entities) as input parameters. To collect information using a custom entity, you can use a **Question** node in a topic.

Disambiguation

An agent might fail to disambiguate between topics, when there's more than one topic that closely matches a user's intent. Normally, the agent asks the user to choose between one or more topics that match their intent via the [Multiple Topics Matched](#) system topic. However, agents that use generative orchestration currently don't call this topic. If you leave the **Multiple Topics Matched** system topic enabled, your agent will start disambiguating between topics automatically when this issue is resolved. If you wish to prevent your agent from automatically disambiguating between topics in the future, turn off the **Multiple Topics Matched** system topic. Turning off this topic allows you to test your agent. Optionally, you can opt back in to using disambiguation by turning the topic back on after testing your agent.

Previous conversation context

With generative orchestration, an agent uses the recent conversation between the user and the agent to provide context when making decisions about which tools to call or filling inputs with values. The amount of conversation history is currently limited, which means that sometimes the agent can't see or use the information in earlier parts of the conversation. In these cases it might be necessary to collect some information again from the user, or ensure that key information is included in the transcript at regular intervals.

Review agent activity

Creating an agent is an iterative process. It involves understanding how your agents work, making changes and seeing the results allow you to build better agents. Copilot Studio's activity tracking provides a visual mapping of your agents' sequence of inputs, decisions, and outputs within a session so you can find problems and opportunities for improvement.

An *activity map* is generated for every session, which begins when an agent starts a conversation or is triggered by an external event. A map represents each activity with a *node*.

There are two forms of activity tracking:

- [Real-time activity map](#) of your agent's performance during testing
- [Historical activity maps and transcripts](#) from past sessions

➊ Note

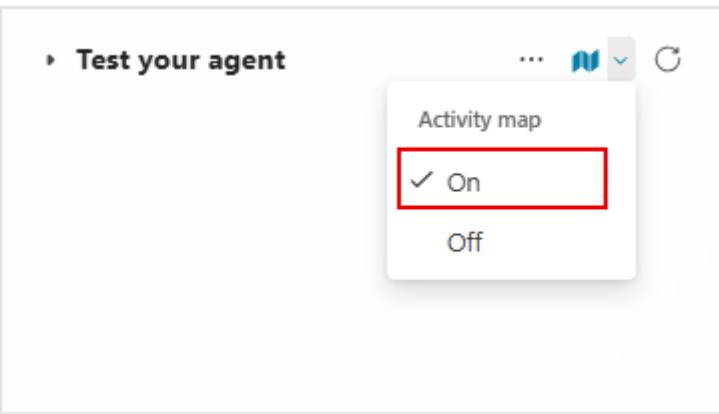
This feature is only available for agents with generative orchestration enabled.

Only your own interactions with the agent and where the agent used your credentials for authentication appear. Your agent must use [integrated Microsoft authentication](#) to identify your interactions.

Real-time activity map during testing

When you ask your agent a question in the test pane, the activity map shows a visual representation of the plan that was generated. The map highlights errors, such as missing or invalid input or output parameters for actions, and shows how long each step takes to execute. You can also verify that the information your agent is sending or receiving is correct.

- To open the activity map, select the map icon  at the top of the test pane.
- To make the activity map automatically appear when you send a query to your agent in the test pane, use the down arrow and select **On**.



Here, the user asked, "what are the current weather conditions?" You can see the action that was selected to respond, and the inputs and outputs.

Activity map

Get the forecast for tomorrow
Connector Action

Complete

Test your agent

Hello, I'm Weather Agent, a virtual assistant. I can answer general questions about the current weather as well as forecasts for today and tomorrow. Just so you are aware, I sometimes use AI to answer your questions.

Note: You can now customize this copilot's topics and knowledge for your own needs. For more information about this template and how to modify it, visit the [template documentation](#).

A minute ago: What's the weather going to be like in Bangalore tomorrow?

Just now: Tomorrow in Bangalore, you can expect heavy rain during the day with a high temperature of 76°F. At night, the skies will be mostly cloudy with a low temperature of 69°F. There is a 63% chance of rain during the day and a 30% chance at night.

Just now: Ask a question or describe what you need

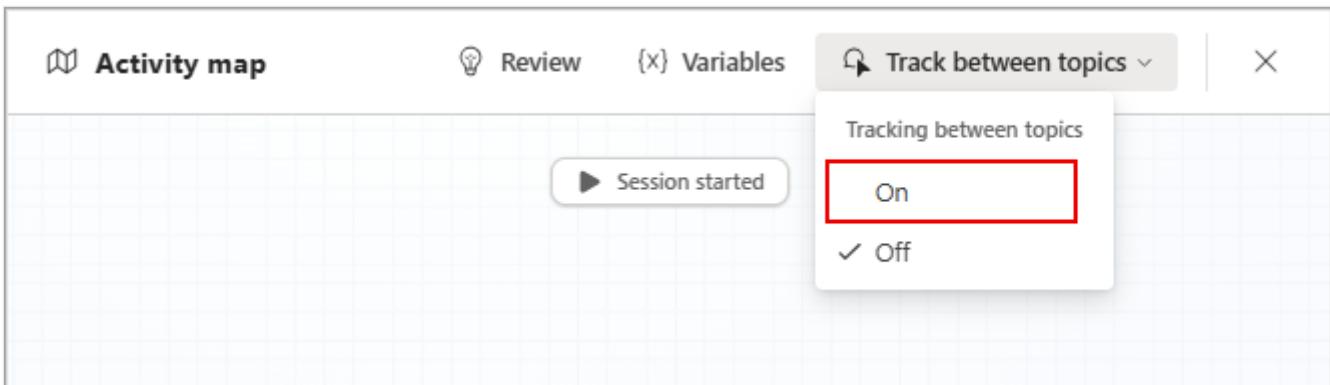
Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

Testing event triggers

When you [test event triggers](#), the trigger payload appears as a message in the test chat. This message is only visible within Copilot Studio, for testing purposes. It isn't visible to users of your agent. You can use the payload messages to understand when and what information and instructions a trigger sends to your agent.

Tracking between topics

When you use the activity map, the option to track between topics is available at the top of the activity map. With tracking enabled, when a topic is triggered as part of a plan, the nodes within the topic appear on the activity map as they're executed, allowing you to monitor the conversation flow.



(!) Note

Generative orchestration activities within a topic don't appear in the activity map.

Historical activity

Each time an agent begins a session, including test sessions initiated in Copilot Studio, the activity is recorded in the **Activity** page in real time. Go to the **Activity** page to:

- Review the interactions and the decisions that your agent made during a session
- Find where the agent's behavior doesn't match your goals
- See how long your agent takes to complete each activity
- Find error details

Historical agent activity is available for:

- Interactions in Copilot Studio [test chat](#)
- Agents published to the [Microsoft Teams channel](#)
- Sessions that started with an [autonomous trigger](#)

(!) Note

A maker must have a Microsoft Exchange license and an inbox to view historical agent activity. The data is stored using Microsoft 365 services, which is not governed by Azure data terms and commitments. Instead, this feature is powered by Microsoft 365 services and stored in the location of the end user's Exchange mailbox geographic region.

[Microsoft 365 terms and data residency commitments](#) governs historical activity data.

Admins should use the [Power Platform admin center](#) to turn off data being stored by Microsoft 365. Turning off this setting stops future data storage in Microsoft 365. Existing data stored in Microsoft 365 is deleted or purged according to the data retention policy set for Microsoft 365.

For more information, see [Manage activity data powered by Microsoft 365 services](#).

Go to the **Activity** page to review:

- The list of sessions and their details
- The step-by-step activity that happened within a session

Session list

A session is a unit of agent interaction involving conversation with a user, registering and responding to events, or a combination of these activities.

In the session list, you can see:

- User: The name of the user interacting with the agent. If the session didn't involve a user, such as when an agent performs actions without human prompting, the user appears as *Automated*.
- Channels: The channel over which the interaction took place. Sessions that happened in the agent test pane have the Copilot Studio icon.
- Date: When the first interaction of the session began.
- Completed steps: The numbers of steps the agent completed during the session.
- Last step: The final step completed in the session.

View session activity

In the **Activity** tab, you can view the activity within a session in either a visual activity map or as a conversation transcript. Select **Activity map** or **Transcript** to switch between these views.

To see a session's activity, select the session. To return to the session list, select the Back icon .

The **Transcript** view displays the conversation that took place with the agent during a session. The transcript records user input, trigger payloads, and agent responses.

You can use the transcripts to see how your agent responds to conversations with users. You can also see the trigger payload messages your agent receives from event triggers.

View all agent activity in the activity map

The **Activity map** is a visual representation of the flow of inputs, decisions, and reactions that happened during a session. Each activity your agent, a user, or a trigger performs with a *node*.

While viewing the activity map, select a node to see details on the inputs, decisions, and outputs for that activity.

The following image is an example of what you see when you select a knowledge node. It shows:

- The query used by the agent to search the knowledge sources. It can be different from the query written by a user, or from the input provided by a trigger.
- The response the agent created based on the knowledge sources.
- The sources the agent referenced.
- Other sources searched over, but not used to create the response. The agent searched these sources, but didn't find information relevant to the query.

The screenshot shows a modal window titled "Search sources" under the "Knowledge" category. The window has sections for "Query", "Response", "Referenced sources", and "Other sources searched over".

Query
Find Serena's coverage for an elbow X-ray (CPT code 73090)

Response
Serena is covered for an elbow X-ray (CPT code 73090) at 80% in-network and 60% out-of-network, after the deductible is met. The procedure may require pre-authorization for full reimbursement.

Referenced sources

- Group Insurance Policy 2025** ...
"Members under Group Insurance Plan 10000100 are covered for CPT code 73090 (elbow X-ray) at 80% of the negotiated rate after the deductible has been met. For out-of-network providers, coverage is 60%."
- Provider Billing Guide** ...
"The standard charge for CPT 73090 at Swedish Medical Seattle is \$504.00, with a discounted cash rate of \$241.92. Insurance rates may vary."

Other sources searched over

- Claim Coverage Summary** ...
- Claims Adjudication Manual** ...
- Eligibility & Benefits Overview** ...

Edit knowledge

In this example, you can see the details of an action node for **Get the forecast tomorrow** showing the inputs and outputs:

The screenshot shows the Microsoft Copilot interface. On the left, there's a sidebar with 'Create', 'Agents', 'Flows', and 'Tools'. The main area is titled 'Activity map' and shows a connector action 'Get the forecast for tomorrow' with a status of 'Complete'. Below it, there's a detailed view of the activity with fields for 'Inputs' (Units: String, Bangalore), 'Outputs' (Temperature High: Number, 76; Temperature Low: Number, 69; Temperature Units: String, °F), and a 'Daytime Summary' field. To the right, a 'Test your agent' section shows a conversation with an AI agent named 'Weather Agent'. The agent asks about the weather in Bangalore tomorrow and provides a detailed forecast. There's also a note about customizing the copilot's topics and knowledge.

Select **Edit** to view and change the configuration associated with the selected activity.

Rationale

Rationale provides an explanation for how an agent decided to call a particular tool. AI generates the rationale, on-demand, and is based on agent metadata and activity. **Rationale** is displayed for *Completed* status knowledge sources or connectors. Select **Show rationale** to display the rationale.

The screenshot shows the Microsoft Activity Manager interface. At the top, there's a navigation bar with 'Activity Manager', 'Overview', 'Knowledge', 'Tools', 'Agents', 'Topics', 'Activity' (which is underlined), 'Analytics', 'Channels', and buttons for 'Published 7/24/2025', 'Publish', 'Settings', and 'Test'. Below the navigation, there's a timeline entry for '07/24/25, 09:48 AM' with a status of 'Complete' for a 'List root level queries' connector action. This action has a duration of '0.70s'. A 'Complete' button is shown below it. To the right, there's a 'List root level queries' connector action with a 'Value (Table)' output. The table contains JSON-like data with fields like '_links', 'self', and href. Below the table, a red box highlights the 'Rationale' section, which contains a bulleted list: 'Identify the project "CCI" as the context for the query.', 'Locate and list the root level queries: "My Queries" and "Shared Queries".', and 'Provide the paths and URLs for each root level query.' To the right of the rationale, there's a large text box from the AI agent 'Activity Manager' providing context about the project and encouraging users to provide more knowledge. There's also a note about AI-generated content being inaccurate and a link to 'See terms'.

When reviewing agent activity, you can use **Rationale** to gain a better understanding of why an agent chose to call a particular tool or fill in parameters. It's helpful in troubleshooting agent behavior. However, since the rationale is generated with AI, there's a chance it's not accurate, so you must use your best judgement when reviewing the provided rationale.

Agent status

The following table provides information about the various states an agent might encounter during its operation.

 Expand table

State	Conditions	Applies to
Complete	No errors. The last message isn't a Manage Connections dialog. Chat is completed when the conversation initiator's defined plan of steps are in a completed state. A conversation might move in and out of a <i>complete</i> state.	Autonomous agents or conversational agents with an action
Incomplete	If there was an issue (one of the steps failed), or if all steps are in a terminal state, and not all of them are <i>complete</i> , then it's <i>incomplete</i> .	Autonomous agents or conversational agents with an action
Failed	If all of the activities have failed, then it's <i>failed</i> .	Autonomous agents or conversational agents with an action
In Progress	At least one of the defined steps isn't complete and still in progress.	Autonomous agents or conversational agents with an action
Waiting for User	The agent responded and is waiting for the user to continue the conversation or not. The last step requires human input.	Autonomous agents or conversational agents with an action
Created	Conversation just started.	Autonomous agents or conversational agents with an action
Canceled	Cancels any remaining dynamic plans and empties the dialog stack.	

Send feedback

If you have suggestions on how to improve your agent quality, send your feedback using the Thumbs Like Dislike  icon in the Activity tab.



Get the forecast for tomorrow...

Complete



Connector Action

Description

Get the forecast for tomorrow in the specified location.

Inputs

Units (String)

C

Submit feedback to Microsoft

Location

Toronto

Output

Moon Phase

First Quarter

Moon Position

FQtr

Moonrise

Moonset

Explain in more detail

Do not include any private or sensitive information.

Share prompt, generated response, relevant content samples, and additional log files? *

Yes No

Microsoft may contact you at VibhutiNair@pvabugbash.onmicrosoft.com about your feedback. [Learn more](#) about how this data is used and your rights. By pressing Submit, your feedback will be sent to Microsoft outside of the Azure compliance boundary and used to improve Microsoft products and services. [Privacy statement](#).

Submit

Cancel

Edit tool

Documentation



Last updated on 10/30/2025

Select a primary AI model for your agent

10/28/2025

AI capabilities evolve rapidly, and each generative model brings distinct strengths, whether it's faster responses, higher-quality outputs, or improved cost-efficiency. With Copilot Studio, you can choose the best model for your agent's orchestration using a simple dropdown menu.

Want to try out cutting-edge models before they're production-ready? Access the latest experimental models to evaluate them early. However, they may have limited testing, availability, and functionality.

Note

Between October 27 and 31, 2025, GPT-4o will be retired in Copilot Studio for agents using generative orchestration, except for GCC customers who will continue using GPT-4o. The new default model is GPT-4.1, which delivers improved performance, reliability, and consistency across experiences. GPT-4o remains available until November 26, 2025 if you enable the "[Continue using retired models](#)" option.

Important

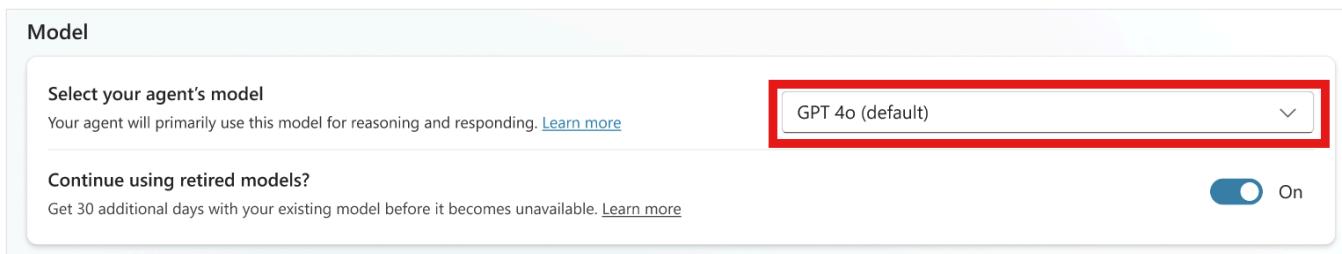
- Experimental models are available for exploration and testing but are not recommended for production use. Review [Limitations of experimental and preview models](#) before choosing an experimental or preview model for your agent.
- Data processed within an experimental model may be processed and stored outside of your organization's geographical boundaries. For more information, see [Admin controls for AI model selection](#).
- This article contains Copilot Studio documentation on model selection (including experimental model previews) and is subject to change.

Change your agent's AI model

Your agent starts with a default model optimized for most scenarios. To change your agent's model:

1. Go to the [Settings](#) page for your agent.

2. In the **Model** section, select the model to use as the primary model for your agent. You can switch between experimental and production models at any time.



The screenshot shows a user interface for selecting a model. At the top left, it says "Model". Below that, there's a section titled "Select your agent's model" with the sub-instruction "Your agent will primarily use this model for reasoning and responding." A dropdown menu is open, showing "GPT 4o (default)" as the selected option. To the right of the dropdown is a small downward arrow icon. Below the dropdown, there's another section titled "Continue using retired models?" with the sub-instruction "Get 30 additional days with your existing model before it becomes unavailable." To the right of this section is a toggle switch that is turned "On", indicated by a blue circle and the word "On".

Limitations of experimental and preview models

Experimental models are available for exploration and testing but are not recommended for production use:

- They might show variability in performance, response quality, latency, or message consumption, and may be prone to time-outs or limited availability.
- If you publish an agent with an experimental model and it's used by users, the usage will be billed at the [established rates](#).

You're welcome to experiment with these models to explore capabilities but proceed with caution when considering deployment in production environments.

Experimental models are subject to [preview terms](#). These models are available before an official release so that you can get early access and [provide feedback](#). If you're building a production-ready agent, see [Microsoft Copilot Studio overview](#).

Admin controls for AI model selection

Administrators can allow or disallow makers to add experimental AI models to agents using the followings settings:

- Administrators can choose to allow or disallow preview and experimental models in an environment. To use these models, [Preview and experimental AI models](#) must be turned on for your environment.
- Data processed within an experimental model may be processed and stored outside of your organization's geographical boundaries. To make experimental models available, your environment must have the [Move data across regions](#) setting turned on. This is an environment-level setting managed in the Power Platform admin center by the tenant administrator.

Continue using a retired AI model

10/28/2025

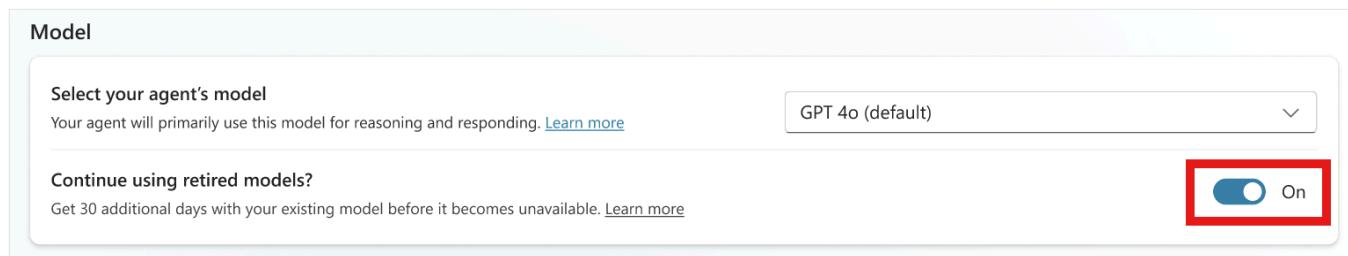
AI is advancing rapidly, and Microsoft Copilot Studio keeps your agent updated with the latest models. To balance innovation and stability, you can continue using retired models for 30 days after an automatic upgrade—helpful for compatibility, compliance, or specific business needs.

ⓘ Note

Between **October 27 and 31, 2025**, GPT-4o will be retired in Copilot Studio for agents using generative orchestration, except for GCC customers who will continue using GPT-4o. The new default model is GPT-4.1, which delivers improved performance, reliability, and consistency across experiences. GPT-4o remains available until **November 26, 2025** if you enable the “[Continue using retired models](#)” option.

To use this feature:

1. Go to your agent's **Settings** page.
2. In the **Model** section, turn on **Continue using retired models**.



Once **Continue using retired models** is turned on, you can [switch between retired and upgraded models](#) at any time during the 30-day period. Your preference applies to any future model upgrades until you turn off the setting.

Choose an external model as the primary AI model

09/24/2025

Want to try out cutting-edge AI models from outside Microsoft? You can add an external model to your non-production Copilot Studio agents as the primary AI model for reasoning or responding.

The external models available in Copilot Studio are [Anthropic](#) models.

Important

- These models shouldn't be used for production purposes. You can experience slowdowns or timeouts due to limited capacity and availability, and these models might not be supported in the future. Admins can [control access to this feature](#).
- Anthropic models are hosted outside Microsoft and are subject to Anthropic terms and data handling, which need to be reviewed and accepted before makers can use them. These models are available before an official release so that you can get early access and [provide feedback](#).
- If you're building a production-ready agent, see [Microsoft Copilot Studio overview](#).
- This article contains Microsoft Copilot Studio external model documentation and is subject to change.

Select an external model for your agent

1. Go to the **Settings** page for your agent.
2. In the **Model** section, select a model labeled **external**.

You can switch between external, [preview](#), and default models at any time.

If external models are visible but you can't select them, that means your admin might have [removed access](#) to external models. If an admin removes access to an external model after originally making it available, agents using that model automatically switch to the default model.

If you can access external models but the model you want to use isn't visible in the model picker, your admin might need to [allow Anthropic models](#) in the Microsoft 365 admin center.

Admin controls and requirements for external models

Admins can control whether makers can add external models to agents. To allow access to external models, [External models](#) must be turned on in Power Platform admin console for the environment or the environment group.

Admins must first allow access to Anthropic models in the Microsoft 365 admin center. Complete the steps in [Connect to Anthropic LLM](#) in the Microsoft 365 admin center.

Use deep reasoning models for complex tasks (preview)

09/01/2025

[This article is prerelease documentation and is subject to change.]

Your agents can use a deep reasoning model to better understand and complete complex tasks. This capability enhances an agent's decision making and returns more accurate responses.

You can add deep reasoning models to agents with generative orchestration and deep reasoning [turned on](#).

Note

Adding a deep reasoning model in Copilot Studio is currently only supported in the United States and the EU (excluding the United Kingdom). Both regions use the Azure OpenAI o3 model.

Deep reasoning models in Copilot Studio are in preview, and do not make data residency commitments. Your data may be transferred to, and stored and processed in, the United States or any other country in which Microsoft or its subprocessors operate.

For organizations using pay-as-you-go billing: Although deep reasoning is only available in some regions, processing for pay-as-you-go Copilot Credits may occur outside your agent's home region.

Once you turn on deep reasoning for your agent, the agent determines which tasks to apply a deep reasoning to, or you can add the *reason* keyword in the agent's instructions to make the agent apply deep reasoning for specific activity.

Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

How does deep reasoning work?

Deep reasoning uses an advanced AI model to complete tasks requiring logical reasoning, problem solving, and step-by-step analysis. This model allows your agent to complete complex tasks with greater accuracy, though with slower response times, than default models used by Copilot Studio.

Your agent determines which tasks or steps benefit from applying a deep reasoning model. You can also tell your agents to use deep reasoning to complete tasks by using the keyword *reason* in the agent instructions.

! Note

Use of deep reasoning consumes billed Copilot Credits. For information on billed Copilot Credits rates for deep reasoning and other agent activities, see [Billing rates and management](#).

Turn on access to deep reasoning models

In order to use reasoning with an agent, you must [turn on generative orchestration](#) and access to deep reasoning models for that agent:

1. Open the **Settings** for your agent.
2. Turn on **Deep reasoning (preview)**.

Use a deep reasoning model to complete tasks

Agents with access to deep reasoning models don't use them for all tasks or steps they perform. Instead, agents can use deep reasoning models to complete specific tasks.

Your agent determines which tasks or steps would benefit from a deep reasoning model.

You can also explicitly instruct the agent to use deep reasoning for specific tasks. Tell the agent to use *reason* in the agent instructions for completing a specific task or step.

For example, these instructions include the keyword *reason* in one of the steps. The agent uses the Azure OpenAI o3 model for that step only:

Instructions

As a Supplier Discovery Agent, parse the incoming requests for additional supplier analysis due to forecasted inventory surge. Perform market analysis based on the request.

1. Parse the request body to identify: Location (e.g., Arizona) and services requested (e.g. cleaning, landscaping, maintenance).
2. Use Get suppliers from ERP flow to retrieve internal supplier data.
3. Search your internal KB for information on suppliers capacity, location, and performance.
4. Run Search the web flow and extract Market Share, Revenue, Rank and other relevant information from all the suppliers.
5. Using the data, use **reason** to provide one top recommendation to the user request along with the relevant context.

1. Parse the request body to identify: Location (e.g., Arizona) and services requested (e.g. cleaning, landscaping, maintenance).
2. Use Get suppliers from ERP flow to retrieve internal supplier data.
3. Search your internal KB for information on suppliers capacity, location, and performance.
4. Run search the web flow and extract Market Share, Revenue, Rank and other relevant information from all the suppliers.
5. Using the data, use *reason* to provide one top recommendation to the user request along with the relevant context.

Note

An agent can use deep reasoning models for multiple tasks or steps, but each one must contain the keyword *reason*. Because deep reasoning models take longer to respond, using models for multiple steps can substantially slow down your agent's response times.

Reviewing reasoning model behavior

You can trace when and how your agent uses deep reasoning models in the [Activity](#) page.

The activity map shows a deep reasoning node where the agent used a deep reasoning model.

To see details, select the reasoning node to expand it. The expanded node explains the reasoning steps and data the model used, and the result the model outputted.

Related content

- FAQ for deep reasoning models
- Orchestrate agent behavior with generative AI
- Guidance for using instructions in generative orchestration
- Preview terms ↗

Choose a preview model for generative responses (preview)

09/02/2025

[This article is prerelease documentation and is subject to change.]

AI innovation moves at a rapid pace, and each generative model has different strengths and specialties. With preview models, you can evaluate the latest model for generative answer summarization.

Important

This feature is in public preview, and available only to makers in the US. This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. Preview features are subject to [preview terms](#). These features are available before an official release so that you can get early access and [provide feedback](#).

Administrators can choose to allow or disallow preview and experimental models in an environment. To use these models, [Preview and experimental AI models](#) must be turned on for your environment.

If you're building a production-ready agent, see [Microsoft Copilot Studio overview](#).

Your agent uses the model you select for [generative answers](#). Generative answers make your agent valuable out-of-the-box and increase the number of topics your agent is conversational in, without requiring any manual dialog tree authoring.

Preview models are experimental and haven't gone through the full testing process that the default models have. You can experience slowdowns or timeouts due to limited capacity and availability. The preview model might not be supported in the future and shouldn't be used for production purposes.

Data processed within the preview may be processed and stored outside of your organization's geographical boundary.

You can switch between preview and default models at any time.

1. Go to the **Settings** page for your agent.
2. In the **Responses** section, select the model you want to use for **Primary response model**.

 Note

In order for your agent to use information outside of your agent's knowledge sources, allow the model to use its own **general knowledge**.

Knowledge sources summary

10/10/2025

In Copilot Studio, knowledge sources work together with generative answers. When knowledge sources are added, agents can use enterprise data from Power Platform, Dynamics 365 data, websites, and external systems. Knowledge sources allow your agents to provide relevant information and insights for your customers.

Published agents that contain knowledge use the configured knowledge sources to ground the published agent. Knowledge can be incorporated at the agent level, in the **Knowledge** page, or at the topic level, with a [generative answers node](#) in an agent topic.

Knowledge sources can be incorporated into agents during their initial creation, added after the agent is created, or added to a generative answers topic node.

Add and manage knowledge for generative answers

Generative answers allow your agent to find and present information from multiple sources, internal or external, without having to create specific topics. Generative answers can be used as primary information sources or as a fallback source when authored topics can't answer a user's query. As a result, you can quickly create and deploy a functional agent. Makers don't need to manually author multiple topics, which might not address all customer questions.

By default, when you create an agent, Copilot Studio automatically creates the **Conversational boosting** system topic. This topic contains a generative answers node, which allows you to begin utilizing knowledge sources immediately. All knowledge sources that are added at the agent level are added to generative answers node in the **Conversational boosting** system topic.

For prerequisites and information on limitations, see [Generative answers](#).

Supported knowledge sources

[] [Expand table](#)

Name	Source	Description	Number of inputs supported in generative answers	Authentication
Public website	External	Searches the query input on Bing, only returns results from provided websites	Generative mode: 25 websites Classic mode: Four public URLs (for example, <i>microsoft.com</i>)	None
Documents	Internal	Searches documents uploaded to Dataverse, returns results from the document contents	Generative mode: All documents Classic mode: Limited by the Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	Generative mode: 25 URLs Classic mode: Four URLs per generative answers topic node	Agent user's Microsoft Entra ID authentication
Dataverse	Internal	Connects to the configured Dataverse environment and uses a retrieval-augmented generative technique in Dataverse to return results	Generative mode: Unlimited Classic mode: Two Dataverse knowledge sources (and up to 15 tables per knowledge source)	Agent user's Microsoft Entra ID authentication
Enterprise data using connectors	Internal	Connects to connectors where your organization data is indexed by Microsoft Search	Generative mode: Unlimited Classic mode: Two per custom agent	Agent user's Microsoft Entra ID authentication

(!) Note

- Agent user authentication for knowledge sources means that when a specific user asks a question of the agent, the agent only surfaces content that the specific user can access.
- Knowledge sources in generative answers nodes currently don't support Bing Custom Search, Azure OpenAI, or Custom Data. Instead, from the generative answers node properties, use the **Classic data** option for [Bing Custom Search](#), [Azure OpenAI](#), or [Custom Data](#) sources.

- For websites, you need to confirm which website(s) your organization owns that Bing will search through Copilot Studio.
- You can perform language-agnostic querying across all supported file types and languages.
- If you're using unstructured data, such as individual SharePoint files and folders, OneDrive files and folders, or connectors, there are different limits and limitations. For more information, go to [Limits and limitations](#).
- Currently, citations returned from a knowledge source can't be used as inputs to other tools or actions.

Knowledge search in classic and generative modes

How knowledge sources are searched depends on which [orchestration mode](#) the agent uses: *classic* or *generative*.

Classic orchestration

When an agent is configured to use classic orchestration, the following applies:

- In the [Conversational boosting](#) system topic, the number of knowledge sources the agent can search is limited, and depends on the type of knowledge source. Your agent can search any combination of knowledge sources, up to the maximum number indicated for each type in the following table:

[Expand table](#)

Type of knowledge source	Limit
Azure OpenAI Service connection	5
Bing Custom Search Custom Configuration IDs	2
Custom data sources	3
Dataverse knowledge sources	2 sources with up to 15 tables each
SharePoint URLs	4
Uploaded files	Unlimited

Type of knowledge source	Limit
Website URLs	4

- You can also embed a [generative answers node](#) in a topic, so that a search is performed for specific intents, and not only as a fallback. The preceding knowledge source limits apply.
- Classic orchestration supports [custom data sources](#), in addition to the other knowledge sources.

Generative orchestration

When an agent is configured to use generative orchestration, the following applies:

- The agent filters the knowledge sources using an internal GPT based on the description given to the knowledge source. For more information, see [Authoring descriptions](#).
- [All files uploaded](#) to the agent are searched.
- For all other knowledge sources, the agent selects the top four knowledge sources, regardless of type. Those four knowledge sources are searched, in addition to all of the uploaded files.

 **Note**

A single website URL counts as one knowledge source.

- Generative orchestration doesn't support [custom data](#) or [Bing Custom Search](#) as knowledge sources. To use those knowledge sources, you must embed them inside a [generative answers node](#) in a topic.

Enable Web Search for your agent

The **Web Search** setting on the **Generative AI** settings page lets your agent access broad, real-time, and up-to-date information beyond what is available in predefined or enterprise-specific knowledge bases. This setting requires that the agent has generative orchestration turned on.

When turned on, **Web Search** triggers when a user's question might benefit from information on the web. It searches all public websites indexed by Bing. This type of search happens in parallel with any searches of public websites you added as knowledge sources. Results from

Web Search are interleaved with results from your configured public website knowledge sources.

(!) Note

Web Search uses [Grounding with Bing Search](#) to return information from the web.

The screenshot shows the 'Knowledge' section of the Generative AI settings. It includes a 'Knowledge' heading, a note about adding data and files, a 'Add knowledge' button, and three specific knowledge sources listed: 'Business Unit, Account', 'https://learn.microsoft.com/ (verified)', and 'Surface-Pro-9-Fact-Sheet.pdf'. Below this is a 'See all' link. A red box highlights the 'Web Search' section, which contains a 'Web Search' heading, a note about enabling agent search, and an 'Enabled' toggle switch with a plus sign icon.

Allow the agent to use general knowledge

The **Allow the AI to use its own general knowledge** setting on the **Generative AI** settings page configures your agent to use [generative AI](#). This setting requires that the agent has generative orchestration turned on.

Generative AI includes general knowledge, which refers to the foundational knowledge that the generative AI is trained on. When this setting is turned on, it allows your agent to use this general knowledge in its answers. This general knowledge setting means that the agent answers questions unrelated to the domain of your agent. If you prefer that your agent is grounded with your specific knowledge sources only, turn off this setting.

The screenshot shows the 'Knowledge' section of the Generative AI settings. It includes a 'Knowledge' heading, a note about adding data and files, and a 'Add knowledge' button. A red box highlights the 'Allow the AI to use its own general knowledge' setting, which is described as enabling the agent to use generative AI, and features a 'Enabled' toggle switch with a plus sign icon.

Tenant graph grounding

The **Enhanced search results** setting on the **Generative AI** settings page determines whether your agent uses [semantic search](#) to improve search results. This setting requires that the agent has generative orchestration turned on.

This feature requires the agent to share a tenant with a Microsoft 365 Copilot license. It also requires that a semantic index is configured for use. To use a semantic index, the Microsoft 365 Copilot license must be assigned to at least one user in the enterprise.

 **Important**

The **Enhanced search results** feature requires that the agent's [user authentication](#) is set to **Authenticate with Microsoft**.

When the feature is turned on and the maker has a Microsoft 365 license in the same tenant, the agent supports SharePoint and connectors containing files up to 200 MB. The feature is turned on by default.

Premium features 

Enhanced search results

Can provide improved search performance for Microsoft 365 Copilot tenants. Availability varies by data source. [Learn more](#)



 **Note**

- If you don't have a Microsoft 365 Copilot license in the same tenant as your agent, or you experience lower response quality, turn off the feature.
- The agent maker doesn't need to have a Microsoft 365 Copilot license to create an agent with a semantic index.
- SharePoint and Microsoft Copilot connectors support files up to 512 MB if they have PDF, PPTX, or DOCX extensions. For more information on supported file types, see [Supported content types](#).
- The **Enhanced search results** feature is a separate feature from the [Dataverse search](#) feature. For more information about how Dataverse search works, see [Frequently asked questions about Dataverse search](#).

Source authentication

If you're using SharePoint, Dataverse, or enterprise data using Microsoft Copilot connectors, you need to incorporate authentication. For more information, see [Configure user authentication in Copilot Studio](#), and for individual generative answers nodes, see [Authentication](#).

In addition, you might need to account for [URL considerations](#) that require extra authentication for your sources.

Content moderation

The content moderation settings allow your agent to provide more answers. However, the increase in answers might affect the allowance of [harmful content](#) from the agent.

The following two areas allow you to configure the content moderation settings:

- The setting in the **Generative AI** settings page sets the moderation at the agent level
- The setting in the generative answers node sets the moderation at the topic level

At runtime, the setting at the topic level takes precedence. If content moderation isn't set at the topic level, it defaults to the **Generative AI** settings configuration.

To adjust the [content moderation settings at the agent level](#), change your agent's **Generative AI** option to **Generative**.

To adjust the [content moderation settings at the topic level](#), change the setting in your generative answers node.

1. Select the desired moderation level for your agent.

The moderation levels range from **Lowest** to **Highest**. The lowest level generates the most answers, but they might contain harmful content. The highest level of content moderation generates fewer answers, and applies a stricter filter to restrict harmful content. The default moderation level is **High**.

2. Select **Save**.

Official sources

When adding knowledge sources to your agent, you might not always control how the information evolves over time, or you might not fully trust this information. It's important to let

your users know that they should consider answers with caution, and they should verify them when appropriate.

However, when you know that information from a specific knowledge source goes through a strict verification process and is highly trusted, you can mark this knowledge source as an official source that can be used directly, without verification.

To mark a knowledge source as official, on the **Knowledge** page, select the **More actions** icon (:) for the knowledge source, point to **Official source** and select **Yes**.

 **Note**

- This feature isn't yet compatible with **generative orchestration**. If you want your agent to use official knowledge sources and mark them as such, turn off generative orchestration.
- When an agent uses authoritative knowledge sources, the response starts with a distinctive indication.

Add knowledge to an agent

10/10/2025

You can add knowledge sources to an agent when you first create it, or after you create it. These sources can be used at the agent level, or at the topic level.

Add knowledge to a new agent

You can add knowledge at the agent level while you're creating your agent. While responding to the prompts to create your agent, you can specify the following types of knowledge sources:

- A public website
- SharePoint

After you enter the URL to the knowledge source, your agent automatically adds the knowledge source. For more information, see [Create an agent](#).

Add knowledge to an existing agent

After the initial creation process, you can add knowledge sources at the agent level from the following areas in Copilot Studio:

- The [Overview](#) page
- The [Knowledge](#) page

You can add knowledge sources at the topic level from the following areas:

- The [Conversational boosting](#) system topic
- A topic containing a [generative answers node](#)

After you add one of the following knowledge source types, you're required to provide a name and description. The knowledge name for each source should be unique. The description should be as detailed as possible, especially if generative AI is enabled, as the description aids [generative orchestration](#).

- [Public website](#)
- [File upload](#)
- [SharePoint](#)
- [Dataverse](#)
- [Azure AI Search](#)
- [Real-time connectors](#)
- [Unstructured data](#)

Use knowledge source suggestions

To access a list of the top 10 suggested knowledge sources for your agent, select **See suggestions** on the **Add knowledge** dialog.

From here, you can discover, search, and add sources:

- Used in one of your previous agents
- Used in agents shared with you
- Used previously while working with Office products

Select **Add** to add the selected suggested knowledge sources to your agent. A copy of the knowledge source becomes part of your agent.

ⓘ Note

Knowledge suggestions are currently only provided for the following types of knowledge sources:

- Files
- SharePoint
- Public websites
- Dataverse (synonym and glossary configurations are preserved)

Add a public website as a knowledge source

08/27/2025

Public websites as knowledge sources at the agent level use publicly available information to improve the reach of your agent.

ⓘ Note

When you specify a URL, the knowledge source uses [Grounding with Bing Search](#) to return information from the web.

URL considerations

The URL used in your agent represents the scope of content for generating responses. There are requirements and restrictions on some URLs.

URL type and structure

- The URL can have up to two levels of depth—subpaths indicated by a forward slash `/`. A trailing forward slash, however, is allowed.

[Expand table](#)

Valid	Not valid
<code>www.contoso.com</code>	<code>www.fabrikam.com/engines/rotary/dual-shaft</code>
<code>www.fabrikam.com/engines/rotary</code>	
<code>www.fabrikam.com/engines/rotary/</code>	

- If the URL redirects to another top-level site, the content isn't included in results:

For example, if `www.fabrikam.com` redirects to `www.contoso.fabrikam.com`, your agent doesn't generate responses from content on either of those URLs.

- URLs that point to a website, requiring authentication or ones not indexed by Bing.

For example, wikis and SharePoint sites require authentication, therefore can't be used:

- `fabrikam.visualstudio.com/project/_wiki`
- `fabrikam.sharepoint.com`

URL domain structure

Any publicly viewable content in the URL you specify, including subdomains under a top-level domain, generate content for your agent.

Examples:

- If you use `www.fabrikam.com/engines/rotary`, the content on `www.fabrikam.com/engines/rotary/dual-shaft` is also used by the agent to generate responses.

Content on `www.fabrikam.com/tools` isn't used, since *tools* isn't a subdomain of *rotary*.
- If you use `www.fabrikam.com` (the *www* exists), the content on `news.fabrikam.com` (the *www* doesn't exist) isn't used, since *news.* is a subdomain under the top-level domain *fabrikam.com*.
- If you use *fabrikam.com*, then content on `www.fabrikam.com` and `news.fabrikam.com` is used, since they sit under the top-level domain *fabrikam.com*.

Social networks and forum URLs

Your agent might generate nonsensical, irrelevant, or inappropriate answers if you use a forum or social network site as your URL. Therefore, community content on social networks often increases the risk of more answers being rejected.

For more information, see the [FAQ for generative answers](#). AI is trained to avoid generating malicious and offensive responses.

Search engine URLs

Don't include URLs of search engines like *bing.com*, as they don't provide useful responses.

Add a public website as a knowledge source

To add a public website as a knowledge source, perform the following steps:

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. Select **Public websites**.

4. Provide the URL to the public website.
5. Add a name (by default, the website's URL is used).
6. Add a description. The description should be as detailed as possible, especially if generative AI is enabled, as the description aids [generative orchestration](#).
7. Select **Add** to finish adding the knowledge source.
8. [Test your agent](#) to see how well it responds to questions related to the content from your website. You might want to test edge case questions to decide if you need a lower moderation to be more inclusive.

 **Note**

- The **Classic data** options in the generative answers node properties allows you specify dynamically formed URLs using a Power Fx formula. For information, see [Use a custom data source for generative answers nodes](#).
- You can also specify [Bing Custom Search](#) in the **Classic data** options of the generative answers node properties. For information, see [Search public data or use a Bing Custom Search for generative answers](#).

Add SharePoint as a knowledge source

07/03/2025

SharePoint as a knowledge source for [generative answers](#) works by pairing your agent with a SharePoint URL or SharePoint lists. This option provides the full support of SharePoint in Copilot Studio.

!Note

- In the [Add knowledge](#) dialog, there are two SharePoint options. The [SharePoint option in the file upload section](#) is used to upload individual SharePoint files or folders, and enables file synchronization capabilities. This article coveres the SharePoint option that provides the full support of SharePoint in Copilot Studio.
- For a list of limits and supported SharePoint functionality, see [Copilot Studio web app SharePoint limits](#).
- To add individual SharePoint files or folders, see [Add SharePoint files and folders](#).
- Agent responses that use SharePoint as a knowledge source aren't included in [conversation transcripts](#).

When you publish your agent, the calls using generative answers are made on behalf of the user chatting with the agent, using the [authentication settings](#) configured in the agent. By default, agents created in Copilot Studio and in Microsoft Teams are configured with the **Authenticate with Microsoft** authentication option, which works in environments such as Microsoft Teams, Power Apps, and Microsoft 365 Copilot.

!Note

It's possible to use generative answers with SharePoint data in Microsoft Teams chats, and not require manual authentication. To use this method for a previously published agent, reconfigure the agent to use **Authenticate with Microsoft** and then publish it again to Microsoft Teams. It might take a few hours before this change takes effect. If a user is in the middle of a conversation and the change doesn't appear to have taken effect, they can type "start over" in the chat to force the conversation to restart with the latest version of the agent. These changes are now available for Teams 1:1 chats between the user and the agent. They're not yet available for group chats or channel messages.

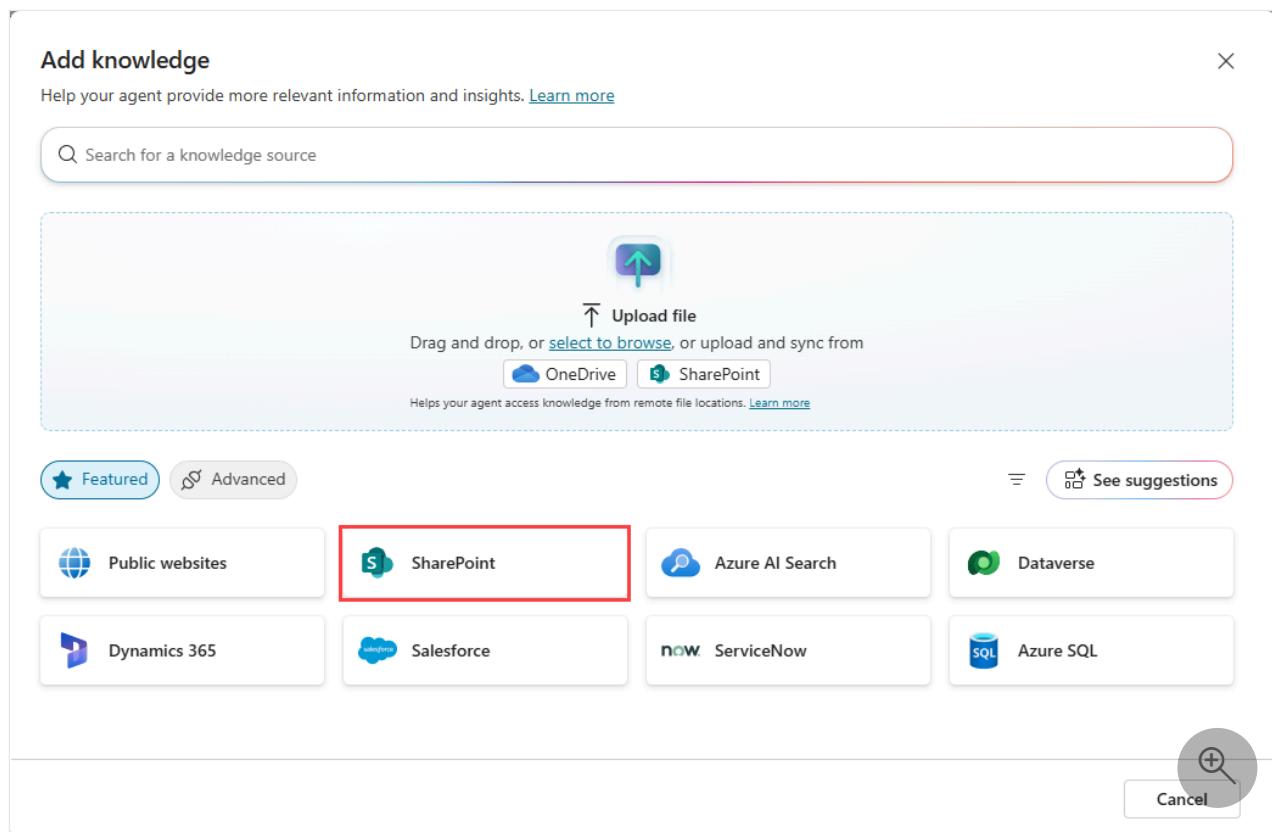
If you want to use manual authentication for your agent, see [Advanced authentication scenarios](#).

Add SharePoint sites as a knowledge source

When a user asks a question and the agent doesn't have a topic to use for an answer, the agent searches the URL and all subpaths. For example, a URL such as *contoso.sharepoint.com/sites* also includes subpaths like *contoso.sharepoint.com/sites/policies*. Generative answers summarize this content into a targeted response.

To add a SharePoint site as a knowledge source, perform the following steps:

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. In the **Featured** section, select **SharePoint**.



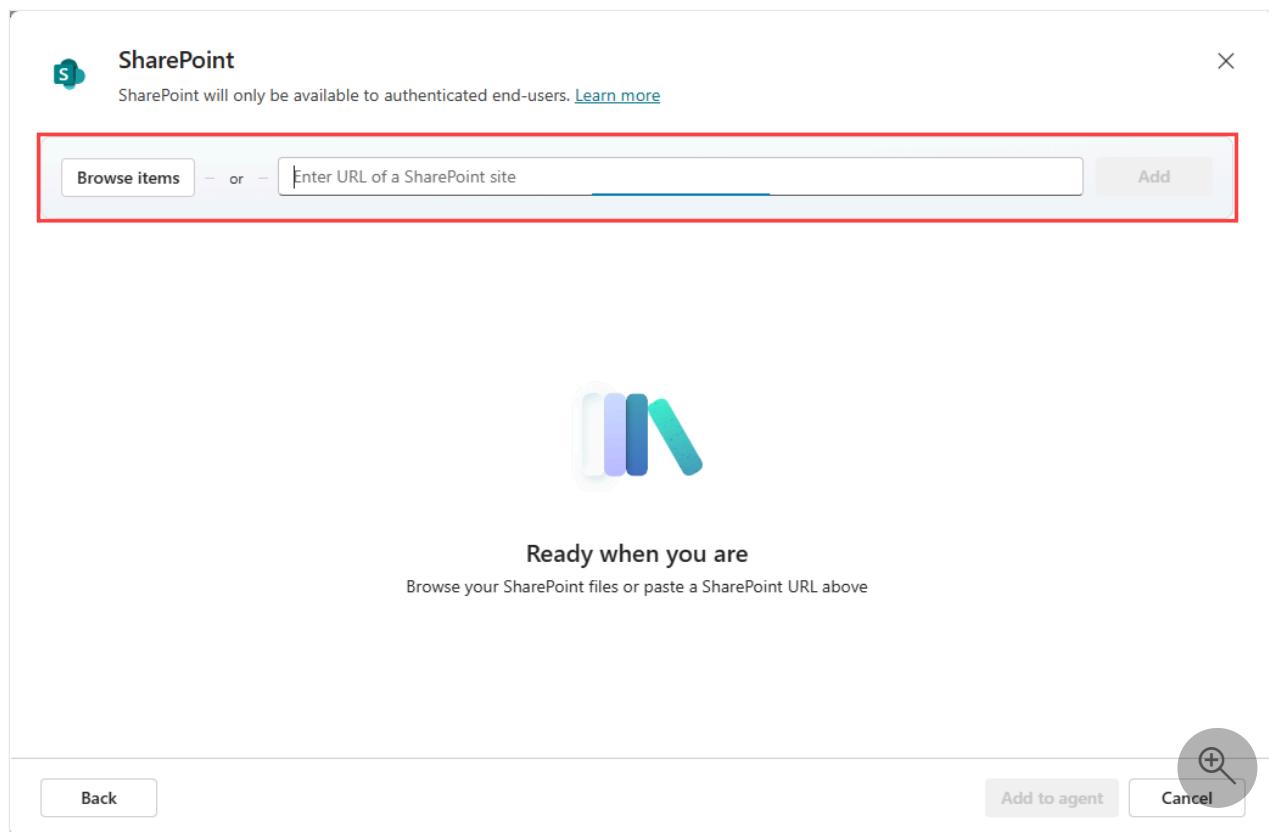
4. Provide the SharePoint URL. Separate multiple URLs with manual line breaks (use **Shift + Enter**).
5. Add a name and a description. The description should be as detailed as possible, especially if generative AI is enabled, as the description aids [generative orchestration](#).

6. Select **Add** to finish adding the knowledge source.

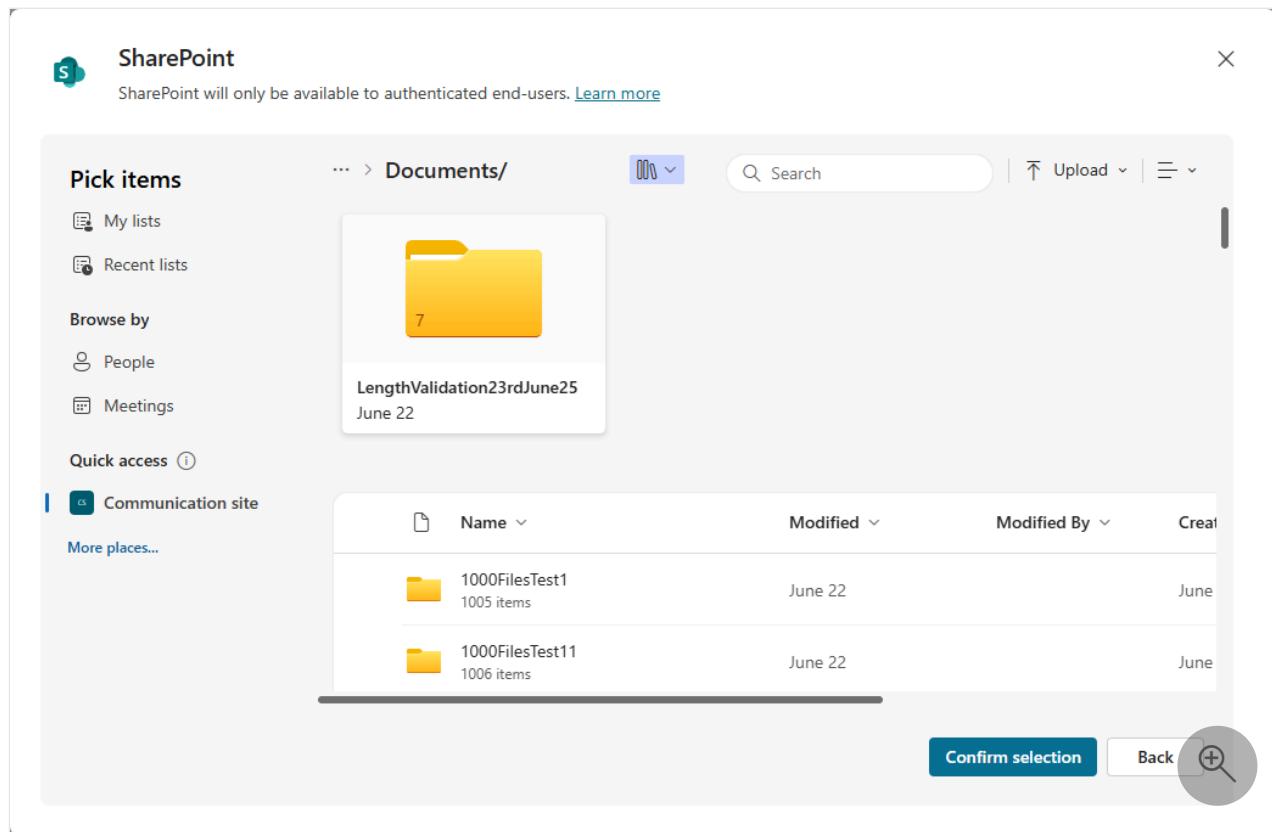
Add SharePoint lists as a knowledge source

SharePoint lists are a popular way to store tabular data and provide quick, easy access to information. Your agent can use these lists as a knowledge source. By adding a SharePoint list as a knowledge source, you create a real-time connection to the source, so the most current data is used for queries and reasoning. Users are authenticated using their SharePoint credentials, which ensures they have permission to access the data before your agent provides a response.

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. In the **Featured** section, select **SharePoint**.
4. Select **Browse items** to find your desired list, or enter the URL of a specific SharePoint list.



5. When you select **Browse items**, the dialog is automatically populated with **My Lists** and **Recent Lists**. **My Lists** only displays lists created in the SharePoint Lists app. All other lists appear in the **Recent Lists**. If you don't see a list you need, open SharePoint and access the list, and the list then appears in **Recent Lists**.



You can select up to 15 lists at a time. Each list is added to the agent's knowledge sources as individual items. To add more lists to the agent, use these steps again and select more lists.

① Note

If you want to use a SharePoint list that you have shared access to but isn't displayed in **Recent Lists**, paste the URL for the list into the URL selection.

6. After selecting your desired lists, select **Confirm selection**.
7. Add a name and a description. The description should be as detailed as possible, especially if generative AI is enabled, as the description aids [generative orchestration](#).
8. Select **Add** to finish adding the knowledge source.

Advanced authentication scenarios

By default, Copilot Studio preconfigures agents to authenticate users using Microsoft authentication to access SharePoint sources when they're connecting to an agent through Microsoft Teams, Power Apps, or Microsoft 365 Copilot. However, if you need to manually configure authentication, the following steps are required for generative answers to work with SharePoint data sources.

 **Important**

If [Restricted SharePoint Search](#) is enabled, use of SharePoint is blocked.

- To authenticate your agent, see [Authentication](#).
- For instructions on how to create the needed Microsoft Entra ID application registration, see [Configure user authentication with Microsoft Entra ID](#). When creating this registration you or your admin must specify the `Sites.Read.All` and `Files.Read.All` scopes in the Microsoft Entra ID app registration.
- Selecting **No authentication** specifies that your agent doesn't retrieve information from SharePoint.
- When you configure authentication, scopes control user access. Specify `Sites.Read.All` and `Files.Read.All` in the scopes field near the existing `profile` and `openid` values in Copilot Studio. These scopes don't give users increased permissions when using the agent, but allow their permissible content from the SharePoint site as configured for generative answers.
- This configuration only applies to generative answers, and doesn't apply to Power Platform Connectors.

Add a Dataverse knowledge source

06/28/2025

Integrating [Dataverse](#) tables as your knowledge source allows you to ground your agent in the data contained in your tables. This process involves adding synonyms and glossary definitions of the tables and columns in your data. For more information, see [Improve agent responses from Microsoft Dataverse](#).

ⓘ Note

Copilot Studio agents require Dataverse search to use this knowledge source. If you can't add a Dataverse table to an agent, ask your administrator to turn on Dataverse search in your environment. For more information about Dataverse search and how to manage it, see [What is Dataverse search](#) and [Configure Dataverse search for your environment](#).

To add Dataverse tables as a knowledge source, perform the following steps:

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages.
3. Select **Dataverse (preview)**.
4. Locate one or more of your Dataverse tables to add. Up to 15 Dataverse tables can be added per knowledge source. To narrow your selections, use the search field.

ⓘ Note

Table recommendations are based on the name of your agent.

5. Preview the tables to ensure the appropriate tables were added. The preview only displays 20 rows and a set of columns, however, all the rows and columns are included in the knowledge source.
6. Review the knowledge name and description. The description should be as detailed as possible, especially if generative AI is enabled, as the description aids [generative orchestration](#).
7. Optionally, to help improve the quality of the answers, add synonyms, and glossary terms:
 - a. Add synonyms for table columns that you selected. Select the **Back** button to accept changes.

b. Add glossary terms to define domain-specific terminology and acronyms. Select the **Back** button to accept changes.

8. Select **Add** to finish adding the knowledge source.

Synonyms and Glossary terms

Synonyms, glossary terms, and definitions for the synonyms and glossary entries aid in AI orchestration. They provide grounding data to improve generated responses. By providing extra information for the AI to understand and interpret the information in the tables, you increase the likelihood of the AI to recognize your users requests, and return responses based on the information provided to the AI.

For scenarios where your Dataverse table contains a column composed of numeric values, you need to provide a synonym for the AI to understand what's in the column. For example, your agent is providing travel assistance, and the Dataverse table contains a column named "cr_123_abc" that uses flight numbers to correspond to cities.

Column name	Synonyms ↑	Description
cr_123_abc	departure city, flight code	cr_123_abc represents the departure city of each flight represented by the flight code

Since the AI doesn't know how to qualify this information, it must be explicitly told how to interpret it. So, the maker adds a description for this column, such as the following example: "cr_123_abc represents the departure city for each flight represented by the flight code."

Sample glossary definitions

Glossary definitions are used to paraphrase the terminology in your Dataverse table, so your agent better understands user questions and can respond better.

The following table illustrates scenarios where adding definitions for glossary terms provides useful context for your agent.

 Expand table

Scenario	Glossary term	Sample description
Acronym	VP	"VP" refers to the Vice President value in the "JobTitle" column of the "Contact" table.
Custom ownership	activity owner	The "activity owner" is identified by the "PartyId" column in the "ActivityParty" table.

Scenario	Glossary term	Sample description
Custom field	opportunity revenue	"Opportunity revenue" refers to the "Custom Revenue" column in the "Opportunity" table.
Complex rules or filter	overdue task	"Overdue task" refers to the the "task" table, when the "state code" column has an open value, and the "scheduled end date" column has a value that is earlier than today.

! Note

- The descriptions in the table are examples. Test your descriptions to verify what descriptions return the best results.
- It might take up to 15 minutes for updated glossary terms and definitions to become available.

Enable Search Support for Multiline Text and File Data Types in Dataverse Tables (Preview)

With Dataverse added as a knowledge source, you can apply unstructured reasoning to get higher-quality responses from Multiline Text(`MemoType`) and File(`FileType`) columns.

Prerequisites

! Note

These steps are based on the prerequisites to perform a search on Dataverse data. Learn more in [Configure Dataverse search for your environment](#).

- Enable the Dataverse search capability.
- You must have maker or admin access to modify views in the Power Apps maker portal.

i Important

- There are extra Dataverse capacity costs incurred with the creation of indexing for search. Learn more in [How much will Dataverse search cost?](#).
- Multiline and file type attachment support is a preview feature.

Configure the Dataverse table in Power Apps

For this preview feature, you need to explicitly include the table and Multiline Text and File columns as **Searchable** in the **Quick Find View**. For more information and detailed steps for the configuration, go to [Select searchable fields and filters for each table](#).

1. Sign into [Power Apps](#) and select the environment you want.
2. Select **Dataverse** then **Tables**.
3. Select the Dataverse table you added to the Copilot Studio agent.
4. Turn on **Searchable** for each column you want to search over.
5. In the **Data experiences** pane, select **Views**.
6. From the list of views, select the **Quick Find View** type.
7. Select a searchable column from the list to add to the view.
8. Select **Edit find tables columns** to add searchable columns from the **Find by** options.
9. Select **Save and Publish** to publish the changes to the view.

Known limitations

- If you add a Dataverse knowledge source before configuring your **Multiline Text** and **File** columns in Power Apps, it might take up to two days for the system to backfill the request. To expedite, consider readding the Dataverse knowledge after configuring the field for search.
- Tables, images, and text in non-organization-based languages aren't supported in File attachments.

Upload files as a knowledge source

06/28/2025

You can upload individual documents for use as a knowledge source at the agent level, which your agent then uses to generate answers with generative AI.

(!) Note

- When you upload documents using this method, the documents are available to your agent across all levels. However, you can [upload documents in generative answers nodes](#). This other option allows you more control to [specify individual generative answers nodes that shouldn't use uploaded documents](#).
- You can upload groups of files using the file groups option. For more information, see [Create file groups](#).
- Copilot Studio agents require Dataverse search to use this knowledge source. If you can't add Dataverse files to an agent, ask your administrator to turn on Dataverse search in your environment. For more information about Dataverse search and how to manage it, see [What is Dataverse search](#) and [Configure Dataverse search for your environment](#).

When an agent user asks a question, and the agent doesn't have a defined topic to use, the agent generates an answer from your uploaded documents. The agent uses generative AI to answer the user's question and provides an answer in a conversational style.

Uploaded documents are stored securely in Dataverse. The number of documents you can upload is limited by the available file storage for your Dataverse environment.

To upload individual files as a knowledge source, perform the following steps:

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. Upload the file by either drag and drop, or browsing to the file location.
4. Add a name. By default, the file's name is used.
5. Add a description. The description should be detailed, especially if generative AI is enabled, because it helps [generative orchestration](#).

6. Select **Add** to finish adding the knowledge source.

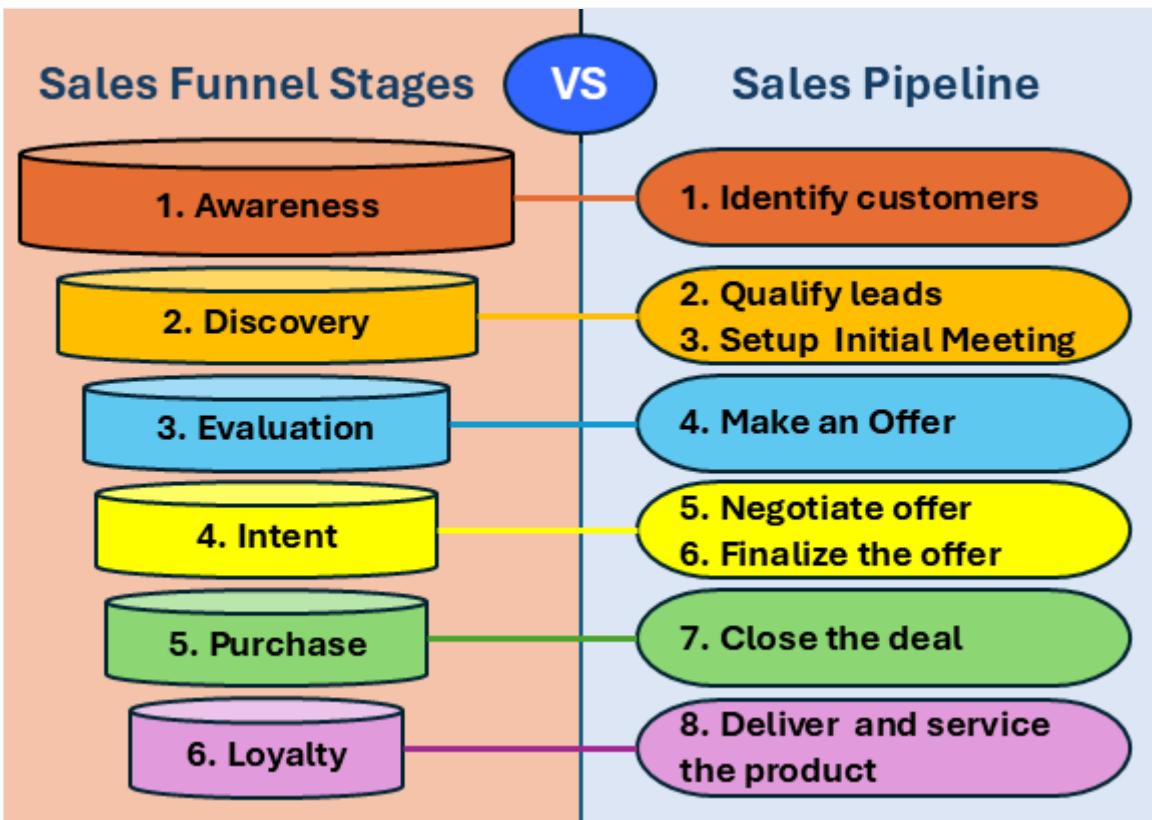
Supported document types

- Word (doc, docx)
- Excel (xls, xlsx)
- PowerPoint (ppt, pptx)
- PDF (pdf)
- Text (.txt, .md, .log)
- HTML (html, htm)
- CSV (csv)
- XML (xml)
- OpenDocument (odt, ods, odp)
- EPUB (epub)
- Rich Text Format (rtf)
- Apple iWork (pages, key, numbers)
- JSON (json)
- YAML (yml, yaml)
- LaTeX (tex)

Annotated image support

Annotated images embedded in PDF files are also supported. Annotation, also referred to as *alt-text*, is a textual description of the image. For best results, provide detailed information about the embedded image in the alt-text.

Users can ask questions and receive answers from the information contained in the annotated images. For example, the following image that was embedded in a PDF file provides information about a sales funnel and sales pipeline stages. Users of the agent containing this embedded image as a knowledge source can ask questions such as, "*What are the stages in a sales funnel?*" The agent can then return an answer based on the information in the annotated image.



(i) **Important**

- Image, video, executables, and audio files can't be used as uploaded documents.
- Images are only supported when they're embedded in PDF files.
- Embedded images are supported in all languages.
- Icons aren't supported.
- Files larger than 512 MB aren't supported.
- Files with encryption, such as sensitivity labels or password protection, aren't supported.
- An agent can include up to 500 files as knowledge.

Create file groups

08/26/2025

File groups let you group up to 500 files into a single knowledge source. Agents can narrow their search scope when asked about relevant or related data.

Querying a group of files helps agents find the right content more easily.

File group requirements

- Only locally uploaded files are supported.
- Copilot Studio agents require Dataverse search to use this knowledge source. If you can't add a Dataverse-enabled knowledge source, such as uploading a file or adding a Dataverse table to an agent, ask your administrator to turn on Dataverse search in your environment. For more information about Dataverse search and how to manage it, see [What is Dataverse search](#) and [Configure Dataverse search for your environment](#).

Note

- OneDrive and SharePoint files aren't supported.
- Copilot Studio agents require Dataverse search to use this knowledge source. If you can't add Dataverse files to an agent, ask your administrator to turn on Dataverse search in your environment. For more information about Dataverse search and how to manage it, see [What is Dataverse search](#) and [Configure Dataverse search for your environment](#).

File group components

A file group has the following three components:

1. **Name and Description:** Fields that help the agent know what the knowledge is about
2. **Instructions:** Field that tells the agent how to use this file group for a user query
3. **Knowledge:** The files with the information to ground the agent

Differences between file groups and individual files

Each individual file is its own knowledge source. A file group uses multiple files as a single knowledge source and uses instructions to help the agent find the right file and return the right information to the user.

For example, a file group helps when an organization has information that depends on specific individual factors, like geographic location or job title.

When you use individual files as a knowledge source, the agent searches all knowledge sources at the same time. The answer depends on one or more chunks the agent finds in one or more files.

When you use file groups, the agent searches for information in two stages. In the first stage, the agent uses the knowledge instructions to find the right file. In the next stage, the agent looks for the right chunks in the files that satisfy the instruction's requirements to answer the user's question. This approach returns more accurate answers from the right file.

File group example scenario

Context: Contoso International Airlines wants to build a bot for customers to ask questions about upcoming travel. One of the most common questions is about baggage policies.

Example situation: Baggage policies differ based on the type of travel (domestic or international), customer tier (economy, business, or first class), and other attributes.

Example solution: Contoso has a group of PDFs, with each policy written in detail and a document for each customer tier. In this case, each criterion, like "type of flight" and "seat type," can be set up as a variable. These variables help the agent filter the knowledge source to find the most relevant information and give the most accurate answer.

Create a file group using new files

When you select one or more files to be uploaded, the button to upload it in the **Add Knowledge** dialog is split to indicate the option to add the files to the agent as individual files (individual knowledge sources) or as a file group (single knowledge source containing one or more files).

When you select the option to **Upload as a group**, the dialog gives you the option to update the file group's name from the generic "File Group 1," its description, and add **Instructions** to help the agent filter through the group of files for more accurate searches. For more specific instructions, you can add global variables to help the agent find the right knowledge for a given user's situation.

After you add a file group as a knowledge source by selecting **Create**, a file group icon is displayed in the **Overview** page (**Knowledge** section) or in the **Knowledge** page.

The same file group can be selected at any time to review, add or delete files within, or change the name, description, or instructions fields.

Create a file group using existing files

You can start a file group from an individual file when the agent already has individual files added to it. Select the **More** icon (...) of the knowledge source's row and select **Create file group**.

When a maker uses this option, a dialog displays all of the individual files that were previously uploaded to the agent. However, it doesn't include files from SharePoint or OneDrive knowledge sources. You can select one or more files to the preselected file to create a file group.

Once you select **Create**, the individual files are no longer displayed as individual knowledge sources in the **Overview** page (**Knowledge** section) or in the **Knowledge** page. They're displayed as files inside of the file group.

View files in a file group

To see details for a file group, such as the individual files it contains, select the file group name or use the hover menu to select **Edit**. This action takes you to the **Knowledge** details page.

Add files to a file group

At any time, new files can be added to a file group. Either drag and drop files, or browse your device and select more files to add them to the file group. This latter option is only available if there are individual files added to the agent. If there are already other types of knowledge sources, even if they're files, **Select from uploaded files** is turned off.

1. On the **Overview** or **Knowledge** page, select the file group you want.
2. On the **Knowledge** page for the file group, select the **Edit** button or select on the name of the file group.
3. Drag and drop or select the option to upload one or more desired files, or use the **Select from uploaded files** option.

Note

Every change to a file group must be saved using the **Save** button.

Update or replace a file within a file group

To update a file in a file group, remove the old file and add the new file to the file group.

1. On the **Overview** or **Knowledge** page, select the file group you want.
2. On the **Knowledge** page for the file group, select the **Edit** button.
3. Select the trash icon to delete the file.
4. Drag and drop or select the option to upload one or more desired files, or use the **Select from uploaded files** option.

Note

Every change to a file group must be saved using the **Save** button.

Remove files from a file group

You can remove files from an existing file group until only one file remains. A file group needs at least one file.

To remove a file from a file group:

1. On the **Overview** or **Knowledge** page, select the desired file group.
2. On the **Knowledge** page of the file group, select the **Edit** button.
3. Select the trash icon to delete the file.
4. Repeat these steps for each file you want to remove.

Important

Deleting the files also deletes the name, description, instructions, and indexed data from the agent. This action is permanent and can't be undone. To restore the files, reupload the files. Every change to a file group must be saved using the **Save** button.

Remove a file group

A file group can be deleted from the **Overview**, **Knowledge page, or the **Knowledge details** page.

- **Overview or Knowledge page:** hover over the knowledge source to show a menu. Select the option to delete the file group.
- **Knowledge details page:** Select the **More** icon (...) next to the **Save** button, and select the option to delete the file group.

Important

Deleting the file group also deletes the files, name, description, instructions, and indexed data. This action is permanent and can't be undone. To restore the files, reupload a new file group.

File group instructions

Instructions are a rich text field where users can include global variables like a user's location or date of birth to create requirements for the agent to perform a knowledge search.

- **Geographical scenario:** If a file group has multiple files about a subject that varies by location of residence, the user can request the agent to search only within the knowledge for the user's location.

File groups without instructions

If instructions aren't provided in a file group, the agent searches all the knowledge sources in that file group instead of using specific requirements to find the most relevant information within a subset of files.

Differences between instructions and descriptions

Descriptions help the agent using generative orchestration pick the right knowledge source for an answer. Instructions add requirements to help the agent pick the right knowledge source based on a global variable, so the experience is more tailored to the end user's needs.

File group FAQs

When to create a file group

Create a file group when you add new knowledge by selecting the **Add knowledge** button or have individual files added as individual knowledge sources.

File groups are helpful in the following scenarios:

- The agent relies on numerous static files that aren't updated often.
- The agent needs to reach an unauthenticated audience.
- When the maker has to use global variables to narrow the scope of the agent's searches.

Can you create file groups and upload individual files at the same time?

You can either start a file group from new files or existing files. While it isn't possible to start a file group using both newly and previously uploaded files, it's possible to add new or existing files to an existing file group.

Can you create multiple file groups?

Create up to 25 file groups per agent. Create each group one at a time.

How many files can a file group contain?

A file group can contain up to 500 files, each up to 512 MB.

Can you select a folder to upload a file group?

You can't upload a folder directly, but you can select all files in a folder to create a file group. For file requirements, see [Upload files as a knowledge source](#).

Is it possible to ungroup a file group?

Yes, it's possible to ungroup the file group if the total number of knowledge sources is 500 or less per agent. This step converts a single knowledge source into one or more knowledge sources, depending on the number of files a file group contains.

A file group can be ungrouped from the **Overview**, **Knowledge** page, or the **Knowledge details** page.

- **Overview or Knowledge page:** hover over the knowledge source to show a menu. Select the option to ungroup the file group.
- **Knowledge details page:** Select the **More** icon (...) next to the **Save** button, and select the option to delete the file group.

Can I group or add instructions to other types of knowledge sources?

You can't create file groups or add instructions to other types of knowledge sources.

Add real-time knowledge with connectors (preview)

09/18/2025

[This article is prerelease documentation and is subject to change.]

For makers with real-time knowledge connectors, you can automatically add enterprise data that resides in your system as a knowledge source. For these systems, Microsoft only indexes metadata, such as table names and column names, and there's no data movement between systems.

Each request is processed at runtime and executed against the target system. In addition, all runtime calls are authenticated using the authentication tokens of the users, so access controls configured in the source system are retained at runtime. This configuration ensures that only users who have access to the enterprise system receive responses to their questions.

These connections are established in Power Platform, and the same connection is used with Copilot Studio. Since the same connections are used, customers govern and manage the use of the knowledge source and actions through the same [data policies](#).

Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

Supported real-time connectors

The following real-time connectors are supported:

- [Salesforce](#)
- [ServiceNow](#)
- [AzureSQL](#)
- [Azure AI Search](#)
- [SharePoint](#)

- [Dataverse](#)
- [Dynamics 365](#)
- [Snowflake](#)
- [Databricks](#)
- [Zendesk](#)
- [Confluence \(Cloud only\)](#)
- [Oracle Database](#)
- [SAP OData](#)
- [Fin & Ops Apps \(Dynamics 365\)](#)
- [Dynamics 365 Customer Insights](#)
- [Dynamics NAV](#)
- [Dynamics 365 Business Central](#)
- [Freshdesk](#)
- [Google Sheets](#)
- [Jira](#)
- [Excel](#)
- [Excel Online \(Business\)](#)
- [MySQL](#)

Add a real-time connector

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. From the **Add knowledge** dialog, select the real-time connector, and then select **Add**. If the real-time connector isn't displayed in the **Featured** list, select **Advanced** for more selections.
4. Select **Sign in**.
5. Select the target location for the connector, and then provide your credentials for the real-time connector.
6. Select **Next**.
7. Select the tables that you want to use as your knowledge source.
8. Add a name and description for the knowledge source. The default name reflects the selected tables.
9. Add [synonyms](#) and [glossary definitions](#).

 **Note**

Currently, only ServiceNow and ZenDesk connectors support adding synonyms and glossary definitions.

10. Select **Add** to complete the connection.

After you add the connector, it appears in the knowledge sources table. The status displays as **In progress** while Copilot Studio indexes the metadata in the tables. After the indexing is complete, the status is updated to **Ready**, and then you can begin [testing the knowledge source](#).

 **Note**

If you run into issues with the connector, go to [Troubleshoot enterprise knowledge sources](#).

Add Azure AI Search as a knowledge source

Article • 05/27/2025

Azure AI Search provides a powerful search engine that can search through a large collection of documents. Copilot Studio supports adding [Azure AI Search](#) as a knowledge source.

To complete the connection, you need an Azure account. If you don't have an Azure account, you can create an account at [Microsoft Azure](#).

After you create the Azure account, use the following Azure articles to set up and configure Azure AI Search. These articles provide information on the setup and configuration needed to use the Azure AI Search connection as a knowledge source:

- [Create an Azure AI Search service in the portal](#)
- [Connect to Azure AI Search using keys](#)
- [Create a vector index](#)
- [Quickstart: Vectorize text and images using the Azure portal](#)
- [Integrated data chunking and embedding in Azure AI Search](#)

Copilot Studio supports [vectorized indexes using integrated vectorization](#). Prepare your data and choose an embedded model, then use **Import and vectorize data** in Azure AI Search to create vector indexes. This method enables the system to use the same embedded model used to vectorize the data to also vectorize the incoming prompt at runtime, which reduces the need to write special functions to do the same.

Copilot Studio also supports the use of the [semantic ranker](#) feature. This feature also needs to be configured in Azure AI Search before adding the connection in Copilot Studio. For more information, see [How to get started with semantic ranker](#).

Create the connection to Azure AI Search

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. From the **Add knowledge** dialog, select **Featured**.
4. Select **Azure AI Search**.
5. Select **Create new connection**.

6. Select the **Authentication type**:

- Access Key
- Client Certificate Auth
- Service principal (Microsoft Entra ID application)
- Microsoft Entra ID Integrated

7. Type the **Azure AI Search Endpoint URL** and the **Azure AI Search Admin Key**.

8. Select **Create** again to complete the connection. A green check mark appears to confirm the connection.

9. Select **Next**.

10. Enter the Azure AI Search vector index to be used. Only one vector index can be added.

11. Select **Add** to complete the connection.

After you add the connection, it appears in the knowledge sources table. The status displays as **In progress** while Copilot Studio indexes the metadata in the tables. After the indexing is complete, the status is updated to **Ready**, and then you can begin [testing the knowledge source](#). During testing, you can verify that proper references were called by reviewing the files and citations cited by the agent.

Return citations

To return citations when using **Azure AI Search** in Copilot Studio, include a URL field with the actual link to the document in the search index. When the `metadata_storage_path` field is included in the index, Copilot Studio interprets that field as the citation. However, if that field doesn't exist, Copilot Studio considers whichever field that contains a complete URL link as the citation. For more information, see [Index file content and metadata by using Azure AI Search](#)

 **Note**

When configuring citations in Azure AI Search, ensure that the users of your agent have the necessary permissions to access the data sources the citations point to. For example, if you add a URL in the search index that points back to a website, or knowledge base, the users should have access to those sources. If the URL points to a restricted data source, the users can't access the generated citations.

Virtual Network support

Copilot Studio supports Azure AI Search indexes configured for virtual networks. This configuration allows you to use a [private endpoint](#) to securely connect your search indexes.

For instructions on how to set up the endpoint, see [Create a Private Endpoint for Azure AI Search](#)

Configure Virtual Network support in Power Platform environment, see [Set up Virtual Network support for Power Platform](#).

In Copilot Studio, follow the [steps](#) to complete the configuration of the connection.

Unstructured data as a knowledge source

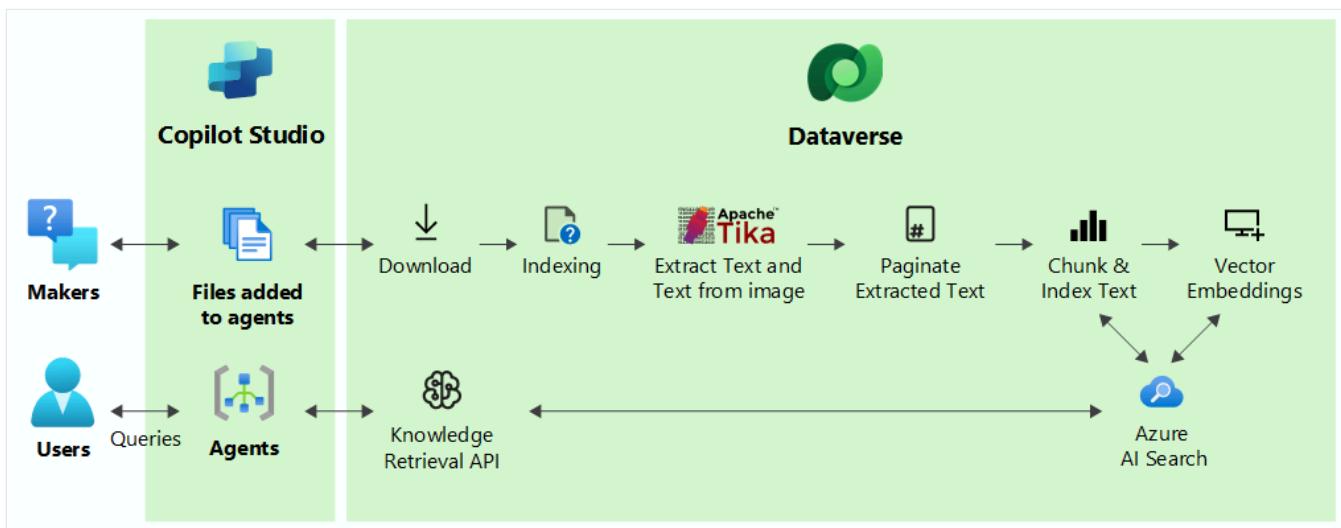
06/28/2025

Copilot Studio allows you to enhance your agents with domain-specific knowledge powered by the same trusted, familiar data sources you've been building through Power Platform connectors.

By uploading external content from your device, OneDrive, or SharePoint, you can enrich your agents with contextual knowledge tailored to your business. These files are securely stored in Microsoft Dataverse and automatically processed into semantic indexes and vector embeddings. This configuration enables your agents to generate more accurate, grounded responses based on the information you provide.

Files uploaded in Copilot Studio use [Microsoft Dataverse](#) to ingest raw files to create indexes and vector embeddings which help provide quality responses for your agents. These files can be uploaded from your computer, or by connecting to OneDrive or SharePoint.

Uploading files as knowledge sources helps makers enrich their agents with extra data, augmenting the language model's knowledge, and grounding the agent in specific information provided by the maker. Makers can upload various files which are semantically indexed as vector embeddings and then used as knowledge for agents. This knowledge used in agents can then be shared with authenticated and unauthenticated users of the agent.



To improve agent's responses, uploaded files are chunked into pieces for faster processing and vector-indexed to provide semantic matches with the user's query. The files are stored securely in Dataverse. When a user queries through an agent, Copilot Studio finds the most relevant chunks that match the intent of the user query and returns the results to the user.

Similarly, Dataverse ingests OneDrive files, SharePoint files (using the options under file upload), and unstructured content like knowledge base articles from other enterprise systems

such as Salesforce, ServiceNow, Confluence, and ZenDesk to provide better semantic results for the agent.

Power Platform connectors for unstructured data

The following [Power Platform connectors](#) are configured to work with unstructured data sources:

OneDrive

One Drive allows makers to use a file selector interface to choose the files and folders they wish to include. Once selected, the items are retrieved into Dataverse and indexed for use. Folders added include all of the supported files and subfolders within that folder up to the [total file limit](#).

SharePoint

SharePoint Documents allow makers to use a file selector interface to choose the files and folders they wish to include. Once selected, the items are retrieved into Dataverse and indexed for use. Folders added include all of the supported files and subfolders within that folder up to the [total file limit](#). Currently there's no support for Pages.

Salesforce

The Salesforce connector for unstructured data supports the ability to retrieve Knowledge Bases containing knowledge articles. Makers select a Knowledge Base and all articles within that Knowledge Base are indexed for use. Individual articles or topics can't be selected. When querying data there's no ability to specify a specific article or knowledge base. The Knowledge list shows a single object for all knowledge objects you select when you create the source.

ServiceNow

The ServiceNow connector for unstructured data supports the ability to retrieve Knowledge Bases containing knowledge articles. Knowledge Bases contain articles. Makers select a Knowledge Base and all articles within that Knowledge Base are indexed for use. Individual articles can't be selected. When querying data there's no ability to specify a knowledge base, folder, or individual article. The Knowledge list shows a single object for all knowledge objects you select when you create the source.

Confluence

The Confluence connector for unstructured data supports the ability to retrieve the spaces containing pages, subfolders are also supported. Individual pages can't be selected. When querying data there's no ability to specify a page. The Knowledge list shows a single object for all pages within the space.

Zendesk

The Zendesk connector for unstructured data supports the ability to retrieve the knowledge base containing knowledge articles. Individual articles, categories, or sections can't be selected. When querying data there's no ability to specify an article, category, or section. The Knowledge list shows a single object for all articles within the knowledge base.

Security

When a user queries an agent that is using a Power Platform Connector source, a few authorization checks are done.

Connector Access

When a maker first uses a connector-based source, they're asked to either select an existing Power Platform connector or to add one. This process ensures that data is only shared with makers who have the appropriate permissions, and provides access to the data source itself.

Content access

When a query is made, the user's connection information is used to check the data source to make sure they have permission to see the content. Even though the chunks and indexes are stored locally to Dataverse, a live check is done on the queries to make sure that the current user has access to the data before providing a summary or response.

(!) Note

- If a user doesn't have permissions for a specific set of files or knowledge base articles, a result isn't returned to them and they receive a standard message of "no results could be found." If users feel there should be results for that source, they need to work with their administrators to ensure they have permissions to the data they're trying to reach.

- Content permission information isn't stored locally. All permission checks are done live with the source to ensure they're the most up-to-date.

Synchronization and file refresh frequency

Connected files from OneDrive and SharePoint, and unstructured knowledge articles are kept fresh using a scheduled synchronization job. This job runs automatically in the background, refreshing the contents of the files and reindexing the changes to provide accurate results for queries. Refreshes manage not only changes to content, but also ensure any content deleted from the source no longer appears as part of any query responses. Currently, there isn't a way to manually trigger a refresh.

Licensing

All requests that involve knowledge are charged at the Microsoft Copilot generative answers messaging rates. For more information, go to [Billing rates and management](#).

If knowledge sources require data to be ingested, then the storage of the data and the corresponding indexes to retrieve that data would be subject to the storage entitlements the customer has. For more information on Dataverse natural language search, go to [Enhance AI-powered experiences with Dataverse search](#).

Limits and limitations

When first enabling unstructured data support, there might be a delay between 5 and 30 minutes for Dataverse configuration and indexing before the added files are processed. The length of time depends on the size of the current Dataverse environment.

Each agent can have a maximum of 500 knowledge objects. These objects could be files, folders, knowledge articles, websites, or other sources.

At this time, only five different sources can be used at a time in an agent. For example, SharePoint, Dataverse, OneDrive, or other sources.

For more information about specific limits and limitations for the supported unstructured data sources, go to [Copilot Studio unstructured data knowledge source limits](#).

Note

Copilot Studio agents require Dataverse search to use this knowledge source. If you can't add a Dataverse-enabled file to an agent, ask your administrator to turn on Dataverse search in your environment. For more information about Dataverse search and how to manage it, see [What is Dataverse search](#) and [Configure Dataverse search for your environment](#).

FAQ

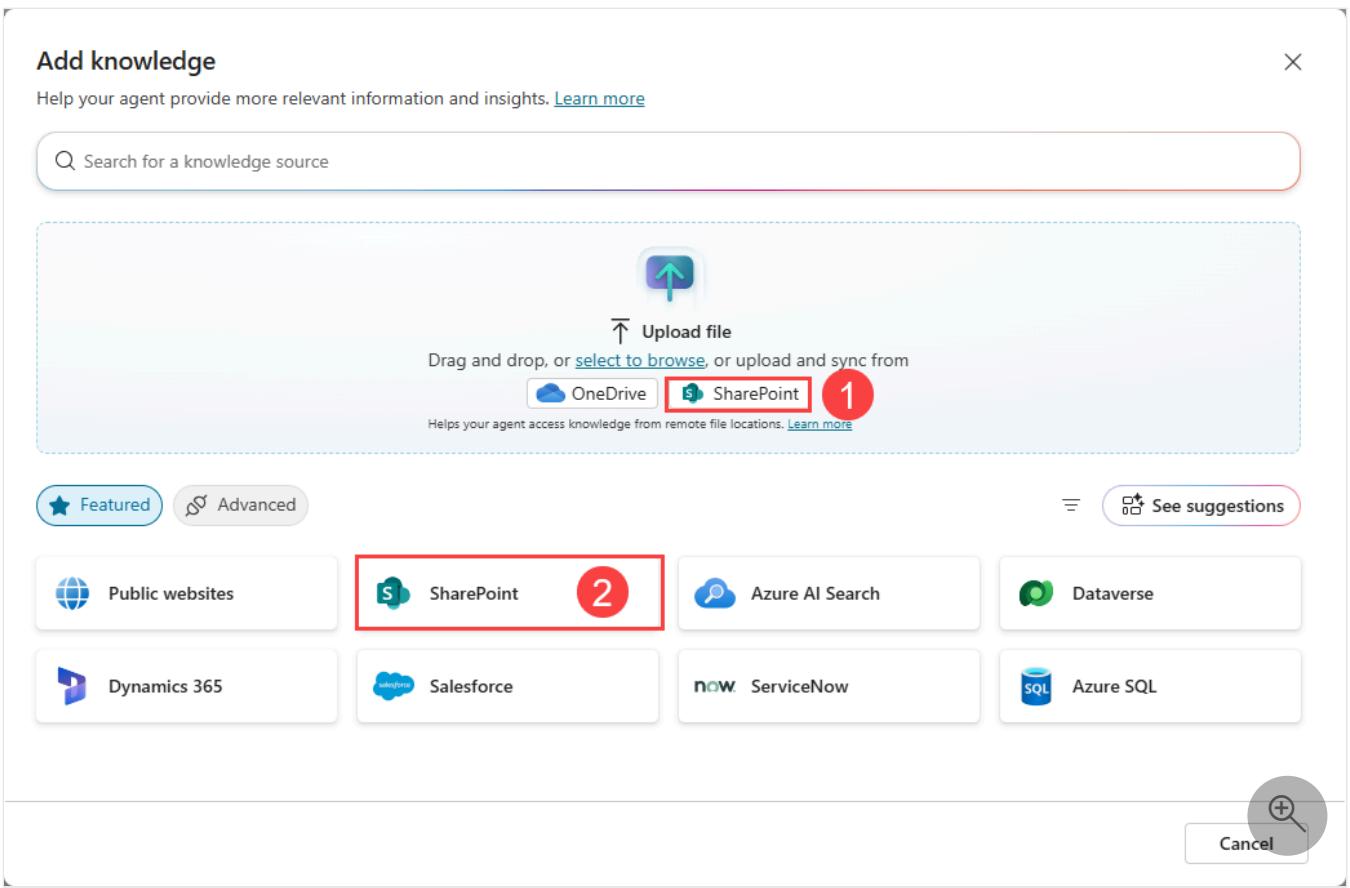
The SharePoint icon isn't displayed in the Upload files section of the Add knowledge dialog?

There's a slight delay between installing a solution and it being displayed in all existing organizations. To initiate a manual update, follow these steps:

1. Sign in to the [Power Platform Admin Center](#), using administrator credentials.
2. Select **Manage**.
3. Select **Dynamics 365 Apps**.
4. Type "PowerAIExtensions" in the search bar.
5. Select the **More icon (...)** of **Microsoft Dynamics 365 - PowerAIExtensions** and select **Install**.
6. From the drop-down menu, select your environment and then select **Install**.
7. After the installation completes, open [Power Apps](#) in a new window.
8. Select **Solutions**.
9. Select **See History**.
10. Search for "PowerAIExtensions_Anchor" and ensure it's set to 1.01.688 or higher.

In the Add knowledge dialog, what is the difference between the two SharePoint options?

In the Add knowledge dialog, there are two SharePoint options. The SharePoint option in the file upload section (1) is used to upload individual SharePoint files or folders, and enables file synchronization capabilities. The other [SharePoint](#) option (2) provides the full support of SharePoint in Copilot Studio.



What happens when I add more than 500 knowledge objects to my agent?

You're prevented from adding any further objects unless you first delete previous objects.

Does each agent have their own index of the knowledge source?

Knowledge sources are stored in Dataverse for use in the environment they were created in. If the same SharePoint folder is used in multiple agents, a single instance of the folder is used for all the agents.

What happens if I select a folder that has more than the maximum number of files, folders, and subfolders when adding a SharePoint or OneDrive source?

Copilot Studio retrieves and indexes up to the maximum number of files, folders, and subfolders, and index those. The remaining aren't processed. Currently, there isn't any messaging to indicate what was or wasn't processed.

One of the files I added (or that was part of a folder I added) is displayed as part of the knowledge source, but I can't get answers from it. Why?

This issue could be related to one of the following reasons:

- The file or folder is set to "Ready" on the **Knowledge** page.
- Ensure that the file name doesn't include an unsupported character (specifically for SharePoint files).
- Ensure that the file doesn't have a sensitivity setting of *Confidential* or *Highly Confidential*, or have password protection.
- Ensure that it's a supported file type.
- If the file or folder is from a different user's OneDrive or SharePoint site, verify that it's shared with the maker.
- If the file is knowledge base file, ensure that your account has permissions to view the content in the source system.

Add unstructured data as a knowledge source

09/19/2025

This article provides information to add unstructured data, such as OneDrive, SharePoint, and Power Platform connectors as knowledge sources for your agent. For more information about unstructured data, go to [Unstructured data as a knowledge source](#).

Add OneDrive files and folders

You can add files and folders from OneDrive for use as a knowledge source for your agent. The differences between adding files and folders using OneDrive versus [uploading files](#):

- Using the upload files method only allows you to select individual files. With OneDrive, you can add both files and folders.
- The upload files method requires that you reload the file if changes were made to the file. In the upload files method, files are static. So, if the file is updated, those updates aren't reflected in the uploaded version unless manually updated. With OneDrive, files and folders that are added are synchronized, so changes are updated automatically.
- The upload files method allows anyone with permission to the agent to access the file. With OneDrive, the user's credentials are verified to ensure they have permission to access the file before providing responses.

When you use OneDrive as a knowledge source, your files are stored in Dataverse, where they're processed and indexed for use. Your files and folders remain in Dataverse, so they do consume data storage.

OneDrive as a knowledge source currently supports the following file types:

- Word: .doc, .docx
- PowerPoint: .ppt, .pptx
- PDF: .pdf
- Excel: .xls, .xlsx

The maximum file size is 512 MB.

Documents that were protected using [sensitivity labels](#), or were password protected, can't be indexed. If added, these types of documents show as ready for use but don't provide responses.

1. Open the agent.

2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. From the **Add knowledge** dialog, select **OneDrive**.
4. Select **Browse items** to choose specific files or folders, or enter the URL of a OneDrive file.
 - a. When you select **Browse items**, you can select up to five individual files, folders, or combinations of the two. For more information about limits, go to [OneDrive limits](#).
 - b. Select **Confirm selection** once you selected all of your items.
 - c. Specify a URL for items that you have shared access to, but aren't stored in your OneDrive.
5. After you added your items from OneDrive, you can update the name and description in Copilot Studio. The description should be detailed, especially if generative AI is enabled, because it helps [generative orchestration](#). Changes to the name in Copilot Studio don't affect the files or folders in OneDrive.
6. Select **Add** to finish adding the OneDrive items to your knowledge source.
7. On the **Knowledge** page, you can view the status of the OneDrive items you added. The amount of time it takes to index your items depends on the number of files, the size of the files, and type of files. Also, if this is the first time that you're configuring this data source, extra time is required to create the Dataverse schema.

 **Note**

Individual files appear as individual items in the list. A folder is represented as one item, and includes all its subfolders and files.

8. Once the status of your items is set to **Ready**, you can ask your agent questions in the **Test panel**.
9. The first time you ask questions in the **Test panel**, you have to connect to verify your permissions to access the content.

Add SharePoint files and folders

You can add files and folders from SharePoint for use as a knowledge source for your agent. The differences between adding files and folders using SharePoint versus [uploading files](#):

- Using the upload files method only allows you to select individual files. With SharePoint, you can add both files and folders.
- The upload files method requires that you reload the file if changes were made to the file. In the upload files method, files are static. So, if the file is updated, those updates aren't reflected in the uploaded version unless manually updated. With SharePoint, files and folders that are added are synchronized, so changes are updated automatically.
- The upload files method allows anyone with permission to the agent to access the file. With SharePoint, the user's credentials are verified to ensure they have permission to access the file before providing responses.
- Currently, document libraries are not supported.

When you use SharePoint as a knowledge source, your files are stored in Dataverse, where they're processed and indexed for use. Your files and folders remain in Dataverse, so they do consume data storage.

SharePoint as a knowledge source currently supports the following file types:

- Word: .doc, .docx
- PowerPoint: .ppt, .pptx
- PDF: .pdf
- Excel: .xls, .xlsx

The maximum file size is 512 MB.

Documents that were protected using [sensitivity labels](#), or were password protected, can't be indexed. If added, these types of documents show as ready for use but don't provide responses.

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. From the **Add knowledge** dialog, select **SharePoint**.
4. Select **Browse items** to choose specific files or folders, or enter the URL of a SharePoint file.
 - a. When you select **Browse items**, you can select up to five individual files, folders, or combinations of the two. For more information about limits, go to [SharePoint limits](#).
 - b. Select **Confirm selection** once you selected all of your items.
 - c. Specify a URL for items that you have shared access to, but aren't stored in your SharePoint.

5. After you add your items from SharePoint, you can update the name and description in Copilot Studio. The description should be detailed, especially if generative AI is enabled, because it helps [generative orchestration](#). Changes to the name in Copilot Studio don't affect the files or folders in SharePoint.
6. Select **Add** to finish adding the SharePoint items to your knowledge source.
7. On the **Knowledge** page, you can view the status of the SharePoint items you added. The amount of time it takes to index your items depends on the number of files, the size of the files, and type of files. Also, if this is the first time that you're configuring this data source, extra time is required to create the Dataverse schema.

 **Note**

Individual files appear as individual items in the list. A folder is represented as one item, and includes all its subfolders and files.

8. Once the status of your items is set to **Ready**, you can ask your agent questions in the **Test panel**.
9. The first time you ask questions in the **Test panel**, you have to connect to verify your permissions to access the content.

Add knowledge base data

Knowledge bases are a key way organizations share information, both for internal users and customers. They often span sales, service, support, policy, and more. However, knowledge base content is often *unstructured*. This unstructured, nontabular storage is often less expensive than relational, table-based storage, while also allowing for more dynamic configuration than tables.

Copilot Studio currently supports the following unstructured sources:

- Confluence (cloud only)
- Salesforce
- ServiceNow
- ZenDesk

Unstructured knowledge base content is added at the "collection" level. This content might be called "Pages", "Knowledge Bases," or some other term in the individual source, but in this context, think of them as "folders." You can't select individual files.

Only published articles are used. Draft or archived content isn't used.

User credentials are used to validate that the user has permission to use the content. Before providing a summary or answer, the agent verifies permissions.

1. Open the agent.
2. Select **Add knowledge** from either the **Overview** or **Knowledge** pages, or the **Properties** of a [generative answers node](#).
3. From the **Add knowledge** dialog, select the unstructured knowledge base source, **Confluence**, **Salesforce**, **ServiceNow**, or **ZenDesk**.

 **Note**

If your admin already configured a Microsoft 365 connection to the knowledge base source, the source is listed in the **Featured** section. Otherwise, check the **Advanced** section.

4. Select the **Your connections** drop-down and select **Create new connection**. At the prompt, sign in to your knowledge base source.

 **Note**

The connection that is created during this setup allows the maker to access the data to create the knowledge source for the agent. These credentials aren't used by the agent's users after the agent is published. Those users have to sign in with their own credentials.

5. Select **Allow** to provide Copilot Studio access to your knowledge base source, then select **Next**.
6. Select your knowledge base source from the list of tables and sources available in your environment.

 **Note**

The information icon in the source provides a message to indicate that the source is unstructured.

7. Select **Preview** to review your data before it's retrieved. A sample set of the knowledge base source is displayed, and you can select individual rows to view the content you're going to use.

8. With your knowledge base source selected, select **Next**.
9. Update the name and description of the knowledge base source. The description should be detailed, especially if generative AI is enabled, because it helps [generative orchestration](#). Changes to the name in Copilot Studio don't affect the files or folders in the source files.
10. Select **Add** to complete the process.
11. On the **Knowledge** page, you can view the status of the knowledge base source you added. The amount of time it takes to index your items depends on the number of files, the size of the files, and type of files. Also, if this is the first time that you're configuring this data source, extra time is required to create the Dataverse schema.
12. Once the status of your items is set to **Ready**, you can ask your agent questions in the **Test panel**.
13. The first time you ask questions in the **Test panel**, you have to connect to verify your permissions to access the content.

Edit, view, or delete knowledge sources

Article • 05/20/2025

After knowledge sources are added, you can view details of your agent's knowledge sources, edit portions of the knowledge sources, or delete knowledge sources.

The following table shows the editable details, by knowledge source type.

 Expand table

Source	Knowledge name	Knowledge description	Other details
Public website	Yes	Yes	URL is editable
File upload	Yes	Yes	-
SharePoint	Yes	Yes	URL is editable
Dataverse	Yes	Yes	Tables, Synonyms, and Glossary are editable
Enterprise data using Microsoft Copilot connectors	-	-	-

Note

A published agent might be using the latest unpublished updates of a Dataverse knowledge source.

To edit details related to an agent's knowledge source(s):

1. Select your agent.
2. Open the **Knowledge** page.
3. Select the ellipsis of the desired knowledge source, and select **Edit** from the dropdown list.
4. Modify the knowledge source details as needed.
5. Select **Save** to confirm the changes.

Test your agent's knowledge sources

06/04/2025

Testing an agent with knowledge sources automatically grounds that agent with those knowledge sources.

After adding a knowledge source, use **Test your agent** to see if it's working in your agent. Only knowledge sources with a **Ready** status can be used in **Test your agent**.

Understanding knowledge status

The following table outlines the status of knowledge sources.

 Expand table

Status	Details
Ready	You can use this knowledge in Test your agent .
Preparing	This knowledge source isn't ready for you to use in Test your agent . This status only applies to file upload and Dataverse knowledge sources.
Not allowed	This knowledge source is blocked by a data loss prevention (DLP) policy and isn't available for use in Test your agent .
Error	This knowledge source had issues while preparing.

Test your agent's generative answers reach

1. Select **Test**.
2. In the **Test your agent** panel, ask your agent questions that take advantage of the generative answers capability.

Generative answers works well with a large variety of question types.

However, some types might produce less helpful responses, including:

- Personal questions.
- Questions that require authenticated access to content.
- Questions that have no related content at a specified URL.

 Tip

To test specific knowledge sources, turn off all the other knowledge sources in your generative answers node except the one that you want to test. Then change the [content moderation](#) setting. Once you're done testing that specific knowledge source, reactivate the other knowledge sources that you think should be relevant in that generative answers node.

Forming questions

- Agents might have difficulty answering questions that require calculations, comparisons, or form submissions. Agents might not understand comparative and superlative terms such as better or best, latest, or cheapest in a question.
- If the agent can't generate an answer to a question, it prompts you to rephrase the question. After two of these prompts, the agent initiates the [Escalate](#) system topic. System topics are topics automatically included with each agent.
- To learn more about how Bing interprets the question against the URL you specify, add `site: \<your URL here>` to the end of your question to see the top Bing results for the question.
- If your agent includes the legacy [lesson topics](#), consider [turning off or deleting these topics](#), since they might be invoked before any knowledge sources.

Tip

In your chat window, you can provide feedback on how well the AI does by selecting the "thumbs up" or "thumbs down" icon underneath the generated answer.

If you see an irrelevant or inappropriate generated response, select the thumbs down icon to let us know. You can also include more detailed feedback.

We use this feedback to improve the quality of the AI.

FAQ

These frequently asked questions (FAQ) describe how to diagnose your knowledge sources in a [generative answers node](#).

Why is my agent not answering questions using knowledge sources?

In this scenario, verify the following settings:

- Ensure that your knowledge sources are enabled in one or more of your generative answers nodes.
- Ensure not to exceed the maximum number of knowledge sources allowed for your generative answers node. For more information, see [Supported knowledge sources](#).
- Ensure that [Content moderation](#) for your generative answers node is set to the correct level.
- If the [Conversational boosting](#) system topic isn't providing enough coverage, consider adding a new topic that uses trigger phrases built for your knowledge sources.

Known issue

A published agent might be using the latest unpublished updates of a Dataverse knowledge source.

Use system topics

09/18/2025

System topics are common topics used in conversations with an agent.

System topics are configured to trigger at appropriate times in response to related events.

However, you can manually trigger system topics by [redirecting](#) a conversation to them. Some system events have trigger phrases that you can customize to fit your needs.

System topics

Web app

Unlike topics that you create, system topics are built into Copilot Studio and added to an agent automatically when you create it. System topics help your agent respond to common events like escalation and have essential behavior like ending a conversation. You can't delete system topics. You can disable system topics you don't need, however.

We recommend not customizing system topics until you're comfortable creating an end-to-end agent conversation.

System topic conditions and behaviors

This list describes the available system topics and reveals what triggers them.

Conversation Start

- Greets users and introduces the agent and its capabilities.
- Triggers when an agent first engages with a user in conversation.

Note

For agents published to the Microsoft Teams channel, the **Conversation Start** topic runs *only once*: the first time a user adds the agent to Teams, even if they remove it and add it back. The reason for this behavior is that Teams uses the same conversation ID for every session with the agent.

Conversational boosting

- Creates generative answers from external data sources.
- Triggers when the agent can't find a match for the user query.

End of Conversation

- Confirms with customers that their query is answered.
- Triggers from a redirection. Call this topic from a custom topic when you're ready to end a conversation.

Escalate

- Informs customers if they need to speak with a human.
- Triggers when "talk to agent" is matched or the **Escalate system event** is called.

For example, **Question** nodes can be configured to escalate if they don't receive a valid answer from the customer.

(!) Note

For a Copilot Studio agent configured to [hand off to OmniChannel](#), follow the instructions in [Hand off to a live agent](#) to ensure proper handoff to Dynamics 365 Customer Service.

Fallback

- Informs users their query couldn't be matched to a topic and asks them to try again.
- Triggers when the agent can't match the user's question or message to a topic.

Multiple Topics Matched

- Prompts users to choose their intended topic and sets a system variable to identify the topic triggered.
- Triggers when a user's message closely matches multiple topics.

On Error

- Informs customers that a user error occurred. (This system topic doesn't handle *system* errors.)

The message includes an error code, the conversation ID, and the error timestamp, which can be used later for debugging. If the conversation is taking place in the test panel, a detailed [error message](#) appears to help you diagnose the issue.

- Triggers when an error occurs during a conversation.

Reset Conversation

- Resets the conversation by clearing variable values and forcing the agent to use its latest published version.
- Triggers with a redirection.

Sign in

- Prompts customers to sign in when user authentication is enabled. Learn how to [add user authentication to topics](#).
- Triggers at the beginning of the conversation when users are required to sign in, or when the conversation reaches a node that uses authentication variables.

Create and edit topics

06/17/2025

In Copilot Studio, a topic defines how an agent conversation progresses.

To author topics, you can [describe what you want and have AI create the topic for you](#), or create topics from scratch.

In Copilot Studio, a *topic* represents some portion of a conversational thread between a user and an agent. You define and work with topics on an *authoring canvas*. A topic contains one or more *nodes*, which together define the conversational paths that a topic can take. Each node performs an action, such as sending a message or asking a question.

For more information about agent and topic limits, see [Quotas, limits, app registration, certificates, and configuration values](#).

Prerequisites

- To create an agent, see [Quickstart: Create and deploy an agent](#).

Selecting the right topic to respond to a user

In order to determine how to respond to users, agents use either [generative orchestration](#) or classic orchestration.

With *generative orchestration*, an agent determines the most appropriate combination of topics, actions, and knowledge to use to answer user queries or respond to event triggers. Each topic has a description that informs the agent of its purpose. Learn more about [authoring descriptions](#).

In agents configured to use *classic orchestration*, each topic has a set of *trigger phrases*—phrases, keywords, and questions that a customer is likely to use for queries related to a specific issue. These agents use natural language understanding, your customer's message, and the topic's trigger phrases to find the best topic to trigger. The customer input doesn't need to exactly match a topic trigger phrase to trigger the topic. For example, a topic about store hours might have the trigger phrase "check store hours." If a customer enters "see store opening hours," this phrase triggers your store hours topic. Learn more about [authoring trigger phrases](#).

Topics in Copilot Studio

An agent can have two types of topics, *system*, and *custom* topics. Every new agent starts with a set of predefined system and custom topics.

- System topics support essential behaviors, such as a custom request to speak to a person or end the conversation. Some system topics have trigger phrases, which you can customize to fit your agent's needs.
 - You can't create system topics.
 - You can't delete system topics, but you can disable them.
 - You can make changes to system topics. However, until you're comfortable creating complete agent experiences, we don't recommend editing the system topics.

For more information, see [Use system topics](#).

- The predefined custom topics cover common behaviors, such as greeting a customer, ending a conversation, or restarting conversation.
 - You can make changes to the predefined custom topics or remove them from your agent entirely.
 - All topics that you create are custom topics.

Node types

The following types of node are available:

[+] [Expand table](#)

Node type	Description
Message node	Send the customer a message.
Question node	Ask the customer a question.
Adaptive Card node	Create a JSON snippet to exchange with other apps.
Condition nodes	Branch your conversation based on a condition.
Variable management nodes	Set a value, parse a value, or clear variables (including the ability to clear the conversation history used by the agent).
Topic management	Redirect the conversation, transfer the user, or end the topic or conversation.
Call a tool nodes	Call a flow like Power Automate or Excel Online, or use a connector or

Node type	Description
	another type of tool.
Advanced	Send generative answers , HTTP requests , events , and more.

Tip

Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

It's not possible to rename **Trigger** nodes and **Go to step** nodes.

Node names can be up to 500 characters in length.

Create a topic

1. Go to the **Topics** page for your agent. For better visibility, close the test panel for now.
2. Select **Add a topic**, and select **From blank**.
A **Trigger** node appears on an otherwise blank topic authoring canvas.
3. Select the **More** icon (...) of the **Trigger** node, and then select **Properties**. The **On Recognized Intent properties** panel appears.
4. In **On Recognized Intent properties**, select the **Phrases** area. The **Phrases** secondary panel appears.
5. Under **Add phrases**, enter a trigger phrase for your topic.

Your agent needs 5 to 10 trigger phrases to train the AI to understand your customers' responses. To add more trigger phrases, you can either:

- Select the **Add** icon  next to the text field, and enter the desired phrase.
- Paste a set of trigger phrases, each one on a separate line, and select **Enter**.
- Type a set of trigger phrases, pressing **Shift+Enter** after each one to place it on a separate line, and select **Enter**.
- [Upload a file](#) listing trigger phrases you want to add for this topic.

You can include punctuation in a trigger phrase, but it's best to use short phrases rather than long sentences.

6. Select **Details** on the toolbar to open the **Topic details** panel.

7. Add details for your topic:

- Enter a name to identify the topic, such as "Store hours." The **Topics** page lists all the topics defined in your agent, by this name.
- If desired, enter a display name to show to the customer when the agent can't determine which topic matches the customer's message.
- If desired, use the **Description** field to describe the purpose of the topic for yourself and other agent makers on your team. Customers never see topic descriptions.

8. Select **Save** on the top menu bar to save your topic.

ⓘ Important

Avoid using periods (.) in your topic names. It isn't possible to export a solution that contains an agent with periods in the name of any of its topics.

Upload trigger phrases for a topic

From the **Phrases** panel of a topic's **Trigger** node, you can add or replace the set of trigger phrases associated with the topic.

1. Prepare a text file (maximum 3 MB) listing all trigger phrases you want to add for your topic, with each phrase on a separate line.

2. Under **Add phrases**, select **upload a file**.

3. In **Upload a file**, select the desired option:

- If you want to add new trigger phrases for your topic, select **Append**.
- To replace all existing trigger phrases with the list from your file, select **Replace**.

4. Drop your file onto the window. Alternatively, select **click to browse**, navigate to your file, and select it.

5. Review the trigger phrases that appear, and select **Add**.

6. Save your topic.

Download trigger phrases for a topic

From the **Phrases** panel of a topic's **Trigger** node, you can download the set of trigger phrases associated with the topic.

1. Under **Add phrases**, select **download**.
2. Save the resulting text file, or open it in a text editor to review its content. Each trigger phrase appears on a separate line.

Design a topic conversation path

When you create a topic, a **Trigger** node automatically appears on the authoring canvas. You can then add different types of nodes to achieve the desired conversation path for your topic.

To add a node after another node or between two nodes on your topic authoring canvas:

1. Select the **Add node** icon  below the node under which you want to add a new node.
2. Select the desired **node type** from the list that appears.

The screenshot shows the Microsoft Bot Framework designer interface. At the top, there's a toolbar with a blue 'Trigger' button and a three-dot menu icon. Below the toolbar, a 'Phrases' connector is connected to a node. A context menu is open, centered over the connector, displaying the following options:

- Paste
- Send a message
- Ask a question
- Ask with adaptive card
- Add a condition
- Variable management >
- Topic management >
- Call an action >
- Advanced >

Delete a node

Select the **More** icon (...) of the node you want to delete, and then select **Delete**.

Controls for editing nodes on the canvas

You can use controls on the toolbar to cut, copy, paste, and delete the selected node or selected adjacent nodes.

The toolbar also has a control to undo edits. Select the **Undo** icon ↺ and select:

- **Undo** to undo the most recent change
- **Undo to last save** to revert all actions back to the last save
- **Redo** to redo the previous action

Paste nodes

Once you use the **Cut** or **Copy** tools to place one or more nodes on the clipboard, there are two ways to paste them in the canvas:

- If you select a node and then select **Paste**, the nodes on the clipboard are inserted after the selected node.
- If you select the "+" to see the *Add node* menu, then select **Paste**, the node on the clipboard is inserted at that location.

Add input and output parameters for a topic

Topics can have input and output parameters. When a topic [redirects to another topic](#), you can use these parameters to pass information between the topics.

Additionally, if your agent uses [generative mode](#), it can automatically fill topic inputs from the conversation context, or after generating questions to gather values from users. This behavior is similar to how generative slot filling works for actions.

To learn more about input and output parameters for topics, see [Manage topic inputs and outputs](#).

Edit topics with the code editor

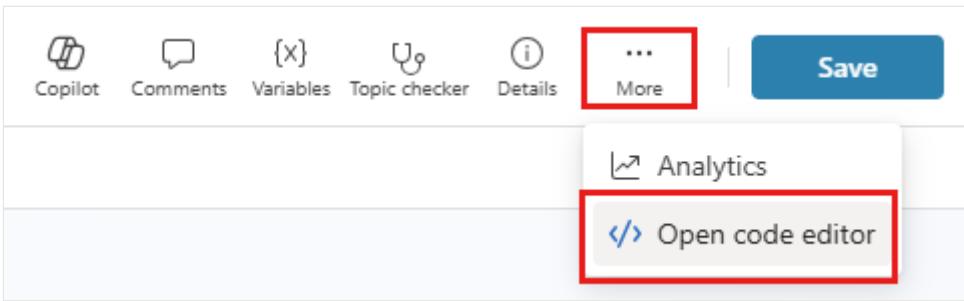
The code editor shows the topic in [YAML](#), a markup language that's easy to read and understand. Use the code editor to copy and paste topics from other bots—even ones created by other authors.

 **Important**

Designing a topic entirely in the code editor and pasting complex topics isn't fully supported.

In this example, you copy and paste YAML into the code editor to quickly add a topic that asks the customer for shipping information.

1. On the **Topics** page, select **Add a topic > From blank**.
2. In the upper-right corner of the authoring canvas, select **More**, then select **Open code editor**.



3. Select and delete the contents of the code editor. Then copy and paste the following YAML code:

YAML

```
kind: AdaptiveDialog
beginDialog:
  kind: OnRecognizedIntent
  id: main
  intent:
    displayName: Lesson 3 - A topic with a condition, variables and a
    prebuilt entity
    triggerQueries:
      - Buy items
      - Buy online
      - Buy product
      - Purchase item
      - Order product

  actions:
    - kind: SendMessage
      id: Sjghab
      message: I am happy to help you place your order.

    - kind: Question
      id: eRH3BJ
      alwaysPrompt: false
      variable: init:Topic.State
      prompt: To what state will you be shipping?
      entity: StatePrebuiltEntity

    - kind: ConditionGroup
      id: sEzule
      conditions:
        - id: pbR5LO
          condition: =Topic.State = "California" || Topic.State =
          "Washington" || Topic.State      = "Oregon"

  elseActions:
    - kind: SendMessage
      id: X7BFUC
      message: There will be an additional shipping charge of $27.50.

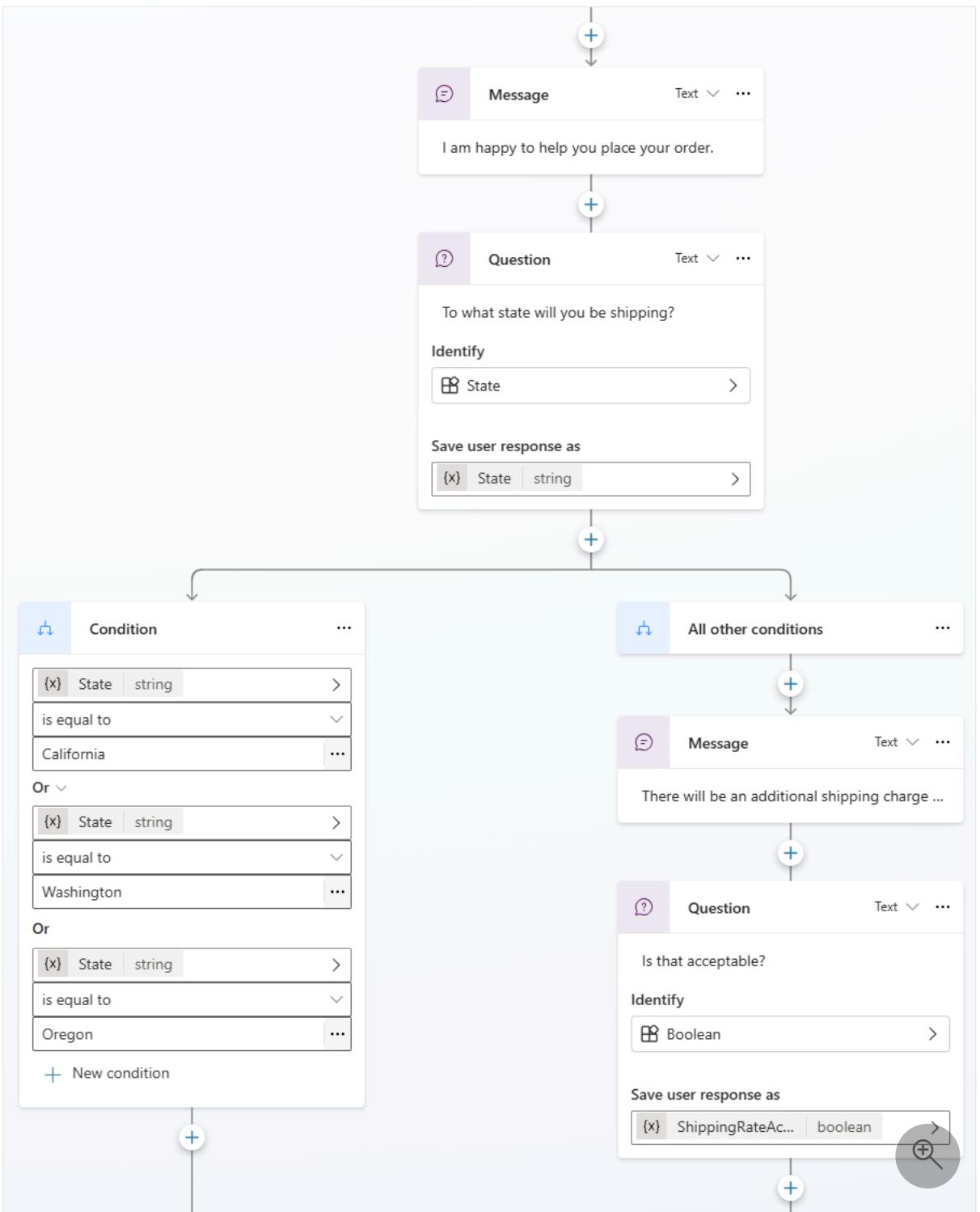
    - kind: Question
      id: 6lyBi8
```

```
    alwaysPrompt: false
    variable: init:Topic.ShippingRateAccepted
    prompt: Is that acceptable?
    entity: BooleanPrebuiltEntity

    - kind: ConditionGroup
      id: 9BR57P
      conditions:
        - id: BW47C4
          condition: =Topic.ShippingRateAccepted = true

      elseActions:
        - kind: SendMessage
          id: LMwySU
          message: Thank you and please come again.
```

4. Select **Save**, and then select **Close code editor**. Notice, the conditional logic that follows some of the **Question** nodes.



Test and publish your agent

[Test your agent](#) when you make changes to your topics, to make sure everything works as expected.

Once you're satisfied, [publish your agent to the desired channels](#).

Create and edit topics with Copilot

09/18/2025

Instead of manually [creating and editing topics](#), Copilot Studio lets you use Copilot to create and iterate on topics, by describing what you want and having AI build it for you.

AI understands what you want to do, then automatically creates a topic with different node types and conversational responses.

The [AI-based agent authoring overview](#) shows how this capability works. You can also use agent authoring with [generative answers](#) to boost conversations.

 Note

This capability might be subject to usage limits or capacity throttling.

Prerequisites

- Review the [FAQ for Copilot](#) and learn more about [Azure OpenAI](#)
- Review the list of [supported languages](#)

Use Copilot to create a topic in Copilot Studio

1. Go to the [Topics](#) page for your agent.
2. Select **Add a topic > Create from description with Copilot**.
3. In **Create from description with Copilot**, enter a name for your topic.
4. Describe the topic you want to create in simple, plain English. You can include questions you want the agent to ask, messages it should show, and details of the desired behavior.

 Tip

To get started, you can select any of the examples. Select **View more examples** to generate new suggestions.

5. Select **Create**.

The new topic appears. The topic includes trigger phrases, and one or more nodes, based on your description.

Use Copilot to edit a topic in Copilot Studio

You can make changes to any topic using the power of natural language understanding (NLU).

1. Go to the **Topics** page for your agent, and open the topic you want to modify.
2. Select **Copilot** on the toolbar.



3. In **Edit with Copilot**, describe what you'd like to change. Use simple, plain English to direct the AI, like in these examples:

- *add a question to ask the user for their date of birth*
- *add two message variations to all questions in the topic*
- *summarize the information collected from the user in an Adaptive Card*

💡 Tip

When you describe a task for your agent, there's no need to mention design elements. For example, if your description includes *ask the user their full name*, Copilot automatically generates (or updates) a **Question** node.

4. Select **Update**.

You can continue to [edit your topic](#) or describe more changes.

If you don't like the changes, select the **Undo** icon ↺ and select **Undo** to go back one action or select **Undo to last save** to go back to the last save point.

💡 Tip

To give feedback on the AI, select the "thumbs up" or "thumbs down" icon at the bottom of the **Edit with Copilot** panel.

Select the thumbs down icon and add detailed feedback. We use this feedback to improve the quality of the AI.

What's supported

Copilot can create **Message**, **Question**, and **Condition** nodes when building and iterating on topics in Copilot Studio. Other node types aren't currently supported.

Nodes generated by Copilot in Copilot Studio have the following limitations:

- Copilot understands the required properties on **Message**, **Question**, and **Condition** nodes, such as text to prompt the user or the appropriate output variables. However, Copilot doesn't understand advanced properties of these nodes, such as the reprompt and interruption settings for a **Question** node. You can still manually edit these properties.
- **Condition** nodes that branch a topic from a multiple choice question might show errors in the generated conditions. You can [troubleshoot these errors](#) manually.

Create and reply to comments in topics

Article • 05/01/2025

Comments are notes that are associated with items in your agent. Use comments to help your team collaborate on the agent design and provide feedback, or provide additional information on implementation details of your agent.

In Copilot Studio, you can attach comments to topics and to individual nodes within topics. The comments are stored in a Dataverse table in the default solution.

Replying to a comment creates a comment thread. In a thread, you can discuss a specific issue and prevent confusion from other unrelated comments. You can delete a thread when it's no longer relevant, or mark it as resolved.

(!) Note

- Comments aren't exported when you [export an agent](#).
- You can't @mention other agent makers in a comment.
- New comments in a topic become visible to [makers who are editing the same topic](#) when they refresh the page. The topic doesn't need to be saved for new comments to show.

Manage comments

The **Comments** pane lets you create new comment threads, resolve or delete comment threads, and add or delete individual comments. To open the pane for a topic, select **Comments** on the top menu bar.

Comment threads can be created at both the topic level and for individual nodes within the topic.

Both types of comment threads are shown in the **Comments** pane. Topic-level topics are displayed first, in the order they were created. Node-level topics are displayed in the order of where the node appears in the topic.

Selecting a node-level comment thread moves the authoring canvas to the associated node and highlight it.

Selecting a topic-level comment thread moves the authoring canvas to the last selected node.

Create a topic-level comment thread

To create a new topic-level comment thread, open the **Comments** pane and select **New**.

Enter your comment and select **Post comment** at the bottom of the comment entry field. You can also press **Ctrl+Enter** to post the comment.

Create a node-level comment thread

To create a new node-level comment thread, select the More icon (...) of the node, and then select **New comment**.

Enter your comment and select the **Post comment** icon. You can also press **Ctrl+Enter** to post the comment.

A comment icon appears next to a node that has comments. You can select the icon to open the **Comments** pane and go to the first comment for that node.

Edit, delete, and resolve comment threads

Select the **More thread actions** icon (...) at the top of a comment thread to edit its initial comment, delete the entire thread, or resolve the entire thread.

- Editing a comment in a thread doesn't affect other comments in the thread.
- Deleting a thread removes it completely. You can't restore deleted comments or threads.
- Resolving a thread marks it with the label **Resolved**. To reopen a resolved thread, select the **Reopen** icon ↗ at the top of the thread. Reopening a thread lets you add more comments.

Edit and delete individual comments

To edit or delete a comment, select the menu icon at the top of the comment, next to your name.

You can only edit or delete comments that you created.

Deleting a comment removes it completely from its thread. You can't restore deleted comments.

Set topic triggers

Article • 05/13/2025

For agents that use *generative orchestration*, the default topic trigger is **By agent**. It activates when the agent determines that the topic's name and description specified on the **Trigger** node match with the incoming message from the user.

For agents that use *classic orchestration*, the default topic trigger is **Phrases**. It activates when one or more of the specified trigger phrases match closely with an incoming message from a user. There are other trigger types available that determine when a topic should be executed.

Prerequisites

Become familiar with [creating and editing topics](#).

Trigger types

Triggers make it possible for your agent to intercept and handle different events or incoming activities of different types. The following trigger types determine when a topic should trigger.

Tip

When a user sends a message to an agent, the payload is a *message* of type [activity](#).

Agents can receive activities of other types depending on what happens in the chat or other events.

For example, Microsoft Teams sends hidden activities of type *invoke*, which signifies a user action, such as interacting with a Message Extension.

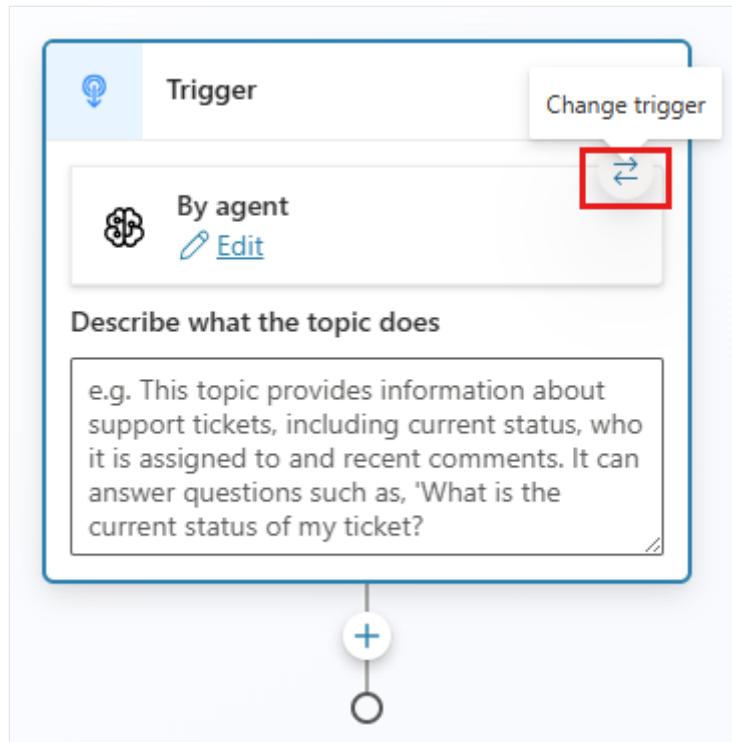
 Expand table

Type	Description
By agent	Available for agents that use generative orchestration. Fires when the agent determines that the topic's name and description specified on the Trigger node match with the incoming message from the user.
Phrases	Available for agents that use classic orchestration. Fires when one or more of the trigger phrases match with the incoming message from a user.
Message received	Fires when an activity of type <i>message</i> —the most common type of activity—is received. Received when a user types or says something to the agent.

Type	Description
Event received	Fires when an activity of type <i>event</i> is received. By default, all event activities are responded to. Use the Event name property within the trigger properties to filter for a specific event.
Activity received	Fires when an activity of any type is received. You can optionally set the Activity type property in the trigger properties to filter for a specific activity type.
Conversation update received	Fires when an activity of type <i>conversationUpdate</i> is received. For example, Teams sends an activity of this type when a user joins a conversation.
Invoke received	Fires when an activity of type <i>invoke</i> is received. Most commonly received from the Teams channel—for example, when the user interacts with a Message or Search extension in Teams.
On redirect	Fires when the topic is called explicitly by another topic. Set the desired inactivity time using the Inactivity duration property in the trigger properties.
Inactivity	Fires when a user hasn't interacted with the agent after a configured period of time.
Plan complete	Available for agents that use generative orchestration. Fires when the agent finished executing all planned steps to respond to a user query or autonomous trigger.
AI response generated	Available for agents that use generative orchestration. Fires when the agent generates a response for a user after calling one or more topics, actions, or knowledge sources. Use the Response.FormattedText system variable to see the generated response. Set the ContinueResponse variable to false if you want to prevent the orchestration response being sent (for example, if you modify the message and send your own using a message node).

Change the trigger for a topic

1. Go to the **Topics** page for your agent, and open the desired topic.
2. Hover over the box of the **Trigger** node and select the **Change trigger** icon.



3. Select the appropriate [type of trigger](#) for your topic.

- For the **By agent** trigger, describe when your agent should use this topic.
- For the **Phrases** trigger, enter the desired trigger phrases.

4. Select **Edit** on the trigger box. The properties panel for the specified trigger type appears.

5. Set the applicable properties for your trigger:

- [Activity type ↗](#)
- [Condition](#)
- [Priority](#)

Trigger condition

Specify conditions that must be met in order for the trigger to fire. For example, you might want the trigger to fire only if the channel used by an employee is Microsoft Teams.

On Recognized Intent properties

Triggered when an intent is recognized.

Condition Builder 

{x} Activity.Channel choice 

is equal to 

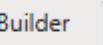
Enter or select a value ...

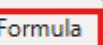
MsTeams

Directline
Facebook
Omnichannel
Telephony

If you need more complex conditions, you can switch to the [Power Fx](#) formula editor: select **Builder** then select **Formula**.

Condition Builder 

{x} Activity.Channel choice 

is equal to 

MsTeams ...

+ New condition

Trigger priority

More than one trigger can fire for a single incoming activity, such as a message. The trigger type determines the order the triggers fire.

Order of execution:

1. Activity Received
2. Message / Event / Conversation update / Invoke received
3. By agent / Phrases

If there are more than one qualifying triggers of the same type, the triggers get executed in the order of creation (oldest first).

You can set the **Priority** property to explicitly determine the order of execution.

Send a message

Article • 05/01/2025

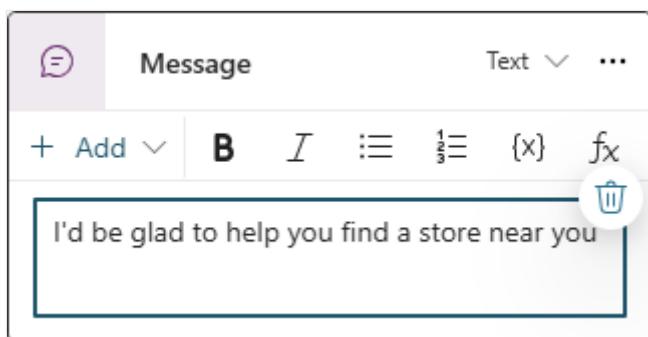
The **Message** node, within the flow of a topic, sends a message from the agent to the user. Messages can be simple text messages, but can also include richer components, such as images, videos, quick replies, and cards.

Tip

All capabilities of the **Message** node are also available to the **Question** node.

Send a text message

1. Open the desired topic.
2. Select the **Add node** icon  below the node under which you want your agent to send a message, and select **Send a message**.
3. In the text box, enter the message you want the agent to send.



4. Optionally, use the tools in the node's menu bar to add:

- basic formatting (bold, italic, bulleted, numbered)
- [variables](#)
- [message variations](#) and [speech overrides](#)
- [images](#)
- [videos](#)
- [basic cards](#)
- [Adaptive Cards](#)
- [quick replies](#)

Tip

Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

It's not possible to rename **Trigger** nodes and **Go to step** nodes.

Node names can be up to 500 characters in length.

Insert a variable in a message

1. Select the desired message.
2. In the node's menu bar, select the **Insert variable** icon ({x}).
3. In **Select a variable**, switch to the tab that matches the scope of the variable you want to use. Learn more about [variable scopes](#).
4. Select the desired variable (use the search box, if needed).

Use message variations

When you add message variations, the agent randomly picks one of them to use each time the node is triggered.

Add a message variation

1. In the node's menu bar, select **Add**, and then select **Message variation**.
2. Enter what you want the agent to say in the text box. Selecting outside of the **Message** node adds your variation to the list of messages.
3. Repeat these steps to add as many variations as you like.

Remove a message variation

1. Select a **Message** node with a variation.
2. Select the variation you want to remove, then select the **Remove variation** icon  in the corner.

Add an image

To add an image, you need one hosted through a URL.

1. In the node's menu bar, select **Add** and select **Image**.
2. In the **Image properties** panel, enter the URL of your image.
3. Optionally, enter a title for the image.

Add a video

1. In the node's menu bar, select **Add**, and then select **Video**.
2. In the **Video properties** panel, under **Media URL**, enter the URL of your video. The URL can either be a direct link to a publicly accessible MP4 file or a YouTube URL.
3. Optionally, enter a **Title**, **Subtitle**, **Image URL** (the URL of a publicly accessible image file), or **Text** to be shown alongside the video on the card. You can also add buttons, which have the same properties as [quick replies](#).

Add a basic card

A basic card is a general-purpose card used for adding text, images, and interactive elements to an agent response message.

1. In the node's menu bar, select **Add**, and then select **Basic card**.
2. In the **Basic Card properties** panel, fill in the properties for the content of your card. You can also add buttons, which have the same properties as [quick replies](#).

Add an Adaptive Card

If a message node informs users of your store's hours, you could, for example, create an [Adaptive Cards](#) showing the hours with an image of the store. Adaptive Cards are platform-agnostic cards that you can customize. In the Adaptive Card designer, you can build an Adaptive Card from elements or author the JSON payload for your card directly in the card payload editor.

1. In the node's menu bar, select **Add**, and then select **Adaptive card**.
2. In the **Adaptive Card properties** panel, select **Edit adaptive card**. The Adaptive Card designer opens.

3. Enter the JSON literal for your card in the card payload editor pane, and select **Save**. A preview of the card appears on your **Message** node.

View multiple media cards in the same node and remove a card

If you add two or more media cards to a node, the following display options are available:

- **Carousel** (default) displays one card at a time.
- **List** displays all cards as a vertical list.

To present your cards as a vertical list, select one of the cards and select the **List** icon  on the menu bar of the node. Proceed in the same fashion, selecting the **Carousel** icon  if you want to present your cards as a carousel.

To remove a card, select it and then select the **Remove** icon  in the corner of the card.

Use quick replies

Important

- Some channels don't support quick replies. In such cases, the quick-reply buttons aren't available during conversations with your agent.
- Some channels limit the number of quick replies shown at once.

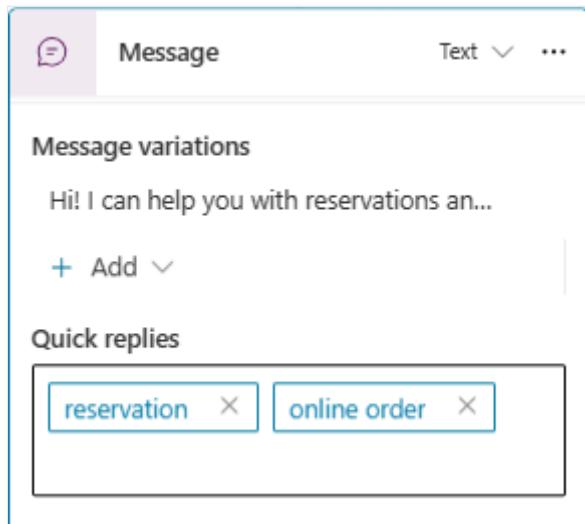
Add quick replies to a message to suggest responses or actions to the customer. Actions taken by the customer get sent back to the agent.

When the customer selects a quick reply, it appears in the chat history until the agent or the customer sends another activity.

A customer can select a quick reply or type directly in the chat. To require the user to choose an option from a list, use a multiple-choice **Question** node instead.

Add a quick reply

1. In the node's menu bar, select **Add** and select **Quick reply**.
2. In the **Quick replies** box on the **Message** node, enter the desired quick replies, one by one.



Change the action type for a quick reply

You configure quick replies to perform different types of actions.

1. Select the **Quick replies** box to open the **Quick replies properties** panel.

Quick replies properties X

Type : Send a message ⋮

Text * ⓘ >
reservation

Title ⓘ >
Enter or select a value

> Advanced options

Type : Send a message ⋮

Text * ⓘ >
online order

Title ⓘ >
Enter or select a value

> Advanced options

+ Add

2. Select the desired value for Type:

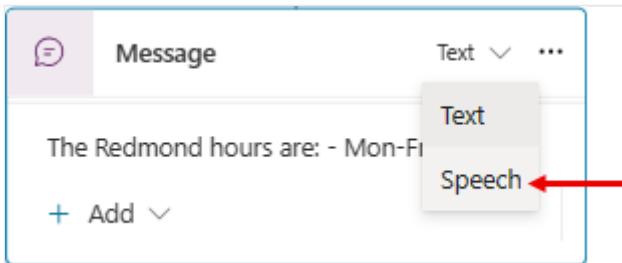
- **Send a message** (default): Send a message to the agent and show it in the chat history.
- **Open URL**: Open a URL. The URL must start with `https://`.
- **Make a call**: Call a phone number. Enter the prefix `PhoneNumber:` immediately followed by the desired phone number—for example, `PhoneNumber:1234567890`.
- **Send a hidden message to the agent**: Send a message to the agent, but hide it from the chat history.

Different properties appear, depending on the type you select. To see more properties, expand **Advanced options**.

Add a speech message override

On voice-enabled channels such as Omnichannel for Customer Service, the message text is used for both text display and voice. However, you can override the text message with a different one for speech.

1. Select the desired **Message** node.
2. In the upper-right corner, expand the selector and select **Speech**.



3. Enter what you want the agent to say.
4. Optionally, [add variations](#) for the speech message.
5. Optionally, [add SSML](#) to your speech message (and variations, if any) to control *how* the agent speaks the message.

Use SSML to customize speech responses

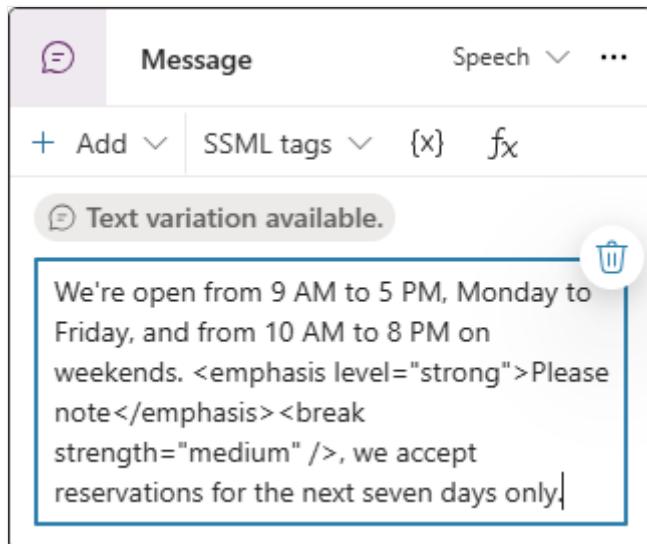
Speech Synthesis Markup Language (SSML) is an XML-based markup language that specifies how text is converted into synthesized speech. SSML is one of the many ways Copilot Studio can make your agent's responses sound like natural speech.

1. Select the speech message you want to customize.
2. In the menu bar, expand **SSML tags**, and then select the tag you want to add.

Expand table

SSML tag	Description
Audio	Add prerecorded audio.
Break	Insert pauses or breaks between words.
Emphasis	Add levels of stress to words or phrases.
Prosody	Specify changes to pitch, contour, range, rate, and volume.

You can add multiple tags to a single speech message. You can also manually enter SSML tags that don't appear on the menu. To learn more about SSML and other tags you can use, see [Improve synthesis with SSML](#) in the Microsoft Azure AI Speech service documentation.



Ask a question

05/28/2025

A **Question** node prompts a user for information and stores their response in a variable for use later on in the conversation.

The node allows you to choose the type of information to collect, such as a multiple-choice answer, a prebuilt entity, or a custom entity. [Question behavior properties](#) allow you to control the behavior of the node, such as what to do when the user enters an invalid response.

Like **Message** nodes, **Question** nodes can include images, videos, cards, quick replies, and message variations. For more information, see [Send a message](#).

💡 Tip

Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

It's not possible to rename **Trigger** nodes and **Go to step** nodes.

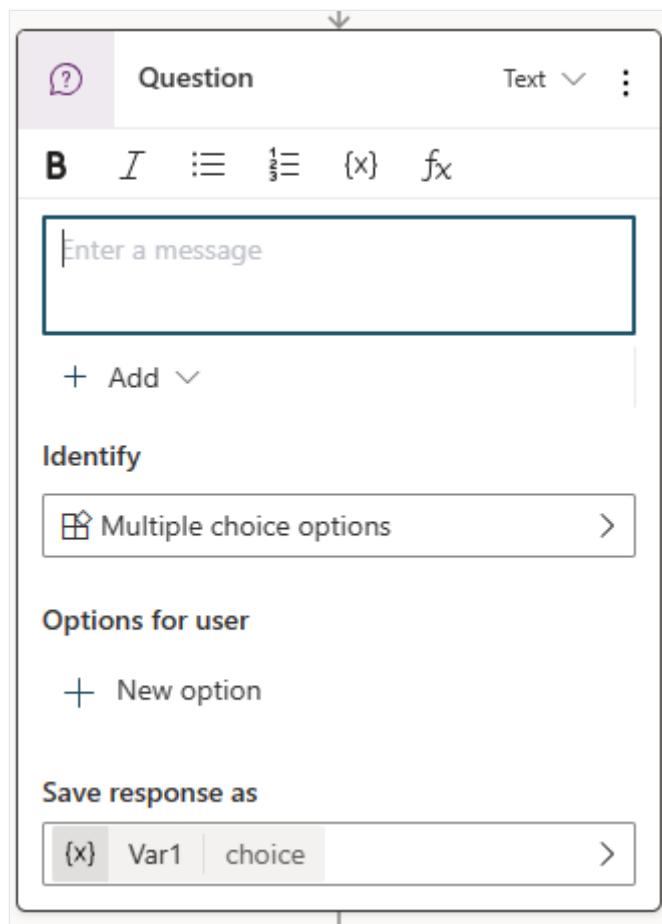
Node names can be up to 500 characters in length.

Prerequisites

- [Create and edit topics](#)
- [Send a message](#)

Add a question node

1. Select the **Add node** icon  below the node under which you want to add a new node, and then select **Ask a question**. A blank **Question** node appears.



2. In the message box, enter the question you want to ask.
3. Under **Identify**, expand the list and select the type of information the agent should identify in the user's response. You can select an existing entity or create a new one. Learn how to [use entities in a conversation](#), and how to [accept one of multiple entities at a conversation turn](#).

Depending on the **Identify** option you selected, there might be more properties to set. For example, for **Multiple choice options**, you must specify the choices the user can select in **Options for user**. During a conversation, each choice appears as a button, but users can also type their answer.
4. Select the default variable name under **Save user response as** to open the **Variable properties** panel and change the name to something meaningful, like `customerName` or `bookingDate`.

To choose a different variable or create a new one, select **>** to open the **Select a variable** panel.

5. To customize the behavior of your **Question** node, configure [question properties](#).

Configure question properties

The **Question properties** panel is where you can adjust behaviors like prompting, validation, and interruptions.

1. To open the **Question properties** panel, select the **More** icon (...) of the **Question** node, and then select **Properties**.
2. In the **Question properties** panel, select the desired category:
 - [Question behavior](#)
 - [Entity recognition](#)
 - [Interruptions](#)

Configure question behavior

Question behavior properties allow you to control whether the copilot can skip the question and how it responds to an invalid response.

Skip behavior

Skip behavior determines what the agent should do if the question node's variable already has a value from earlier in the conversation.

- **Allow question to be skipped:** Skip the question if the variable has a value.
- **Ask every time:** Ask the question even if the variable has a value.

Reprompt

Reprompt determines how your agent reacts when it doesn't get a valid answer from the user. You can tell it to try again once, twice, or move on without getting an answer. To customize what your agent does when it moves on, configure the properties under [No valid entity found](#) in the **Entity recognition** panel.

- **How many reprompts:** The number of times your agent tries to get a valid answer. **Repeat up to 2 times** is the default. You can also select **Repeat once** or **Don't repeat**.
- **Retry prompt:** To change the message, select **Customize**, and then enter the new prompt.

Configure entity recognition

Entity recognition properties allow you to expand validation beyond the default rules for the entity to collect, and to choose what happens when your copilot isn't able to elicit a valid entity value from the user.

Entity settings

Enter the name of the option set you defined in Power Fx used to reference items in the list.

Additional entity validation

By default, the **Question** node checks for a valid response based only on the entity you selected. **Additional entity validation** allows you to add criteria to the basic test. For example, the **Question** node accepts any numeric value when it identifies a number, but you might want to make sure it's less than 10. You can also change the prompt to help the user enter a valid response.

- **Condition:** Enter a Power Fx formula that returns a Boolean value (`true` or `false`); for example, `Topic.Var1 < 10`
- **Condition not met prompt:** To change the message, select **Customize**, and then enter the new prompt.

No valid entity found

No valid entity found determines what happens when your agent stops trying to get a valid response from the user. You can escalate to a human agent or provide a default value. You can also change the prompt to give the user more context.

- **Action if no entity found:**
 - **Escalate:** Redirect the user to the *Escalate* system topic. It's the default behavior.
 - **Set variable to value:** Set the output variable to a value and move on to the next node. Enter or select the value in **Default entity value**.
 - **Set variable to empty (no value):** Clear the output variable and move on to the next node. You can use a **Condition node** later to check whether the variable has a value.
- **No entity found message:** To change the message, select **Customize**, and then enter the new prompt.

Configure interruption behavior

Interruption settings determine whether the user is allowed to interrupt the flow of the current topic to switch to a different topic.

- **Allow switching to another topic:** The user can switch to a new topic when their response to the question matches a trigger for the other topic with high confidence.
- **Only selected topics:** Specify to which topics the user can switch from this question node.

 **Tip**

If you notice issues with your agent where the answer to a question triggers another topic, instead of filling the variable associated with the question node, try turning off **Allow switching to another topic**. This option prevents the agent from considering the answer to the question as a potential interruption.

Ask with Adaptive Cards

10/17/2025

Adaptive Cards let you add snippets of content to Copilot Studio agents that can also be openly exchanged with other cloud apps and services. To provide rich conversation capability for your agent, you can include text, graphics, and buttons. Because they're platform-agnostic, you can easily tailor Adaptive Cards to your needs.

With an **Adaptive Card** node, your agent can show an Adaptive Card containing one or more submit buttons, and optionally, one or more form input fields. Your agent stores user input in variables for use later in the conversation.

!**Note**

Copilot Studio supports the Adaptive Cards schema versions 1.6 and earlier. However, the appropriate schema version depends on the targeted host app:

- The Bot Framework Web Chat component (that is, the default website integration pattern) supports version 1.6 but doesn't support `Action.Execute`
- The live chat widget (used for Dynamics 365 Omnichannel for Customer Service) is limited to version 1.5
- Teams is also limited to version 1.5

In addition, Copilot Studio only renders version-1.6 cards in the test chat, not on the canvas.

For more information about the Adaptive Cards schema, see [Schema Explorer](#).

Copilot Studio includes a built-in Adaptive card designer, which offers the most useful features from the [Adaptive Cards Designer](#).

Alternatively, you can:

- Use a JSON representation for the card you want to show to the user.
- Use a [Power Fx formula](#) to include dynamic information on the card.

You can also [control the behavior of the card](#), such as what to do when the user enters an invalid response or if the node can be interrupted.

The **Adaptive Card** node is meant for *interactive* cards, where the user is expected to submit a response. **Message** and **Question** nodes can be used to present the user with a non-interactive card to display information to the user.

Tip

Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

It's not possible to rename **Trigger** nodes and **Go to step** nodes.

Node names can be up to 500 characters in length.

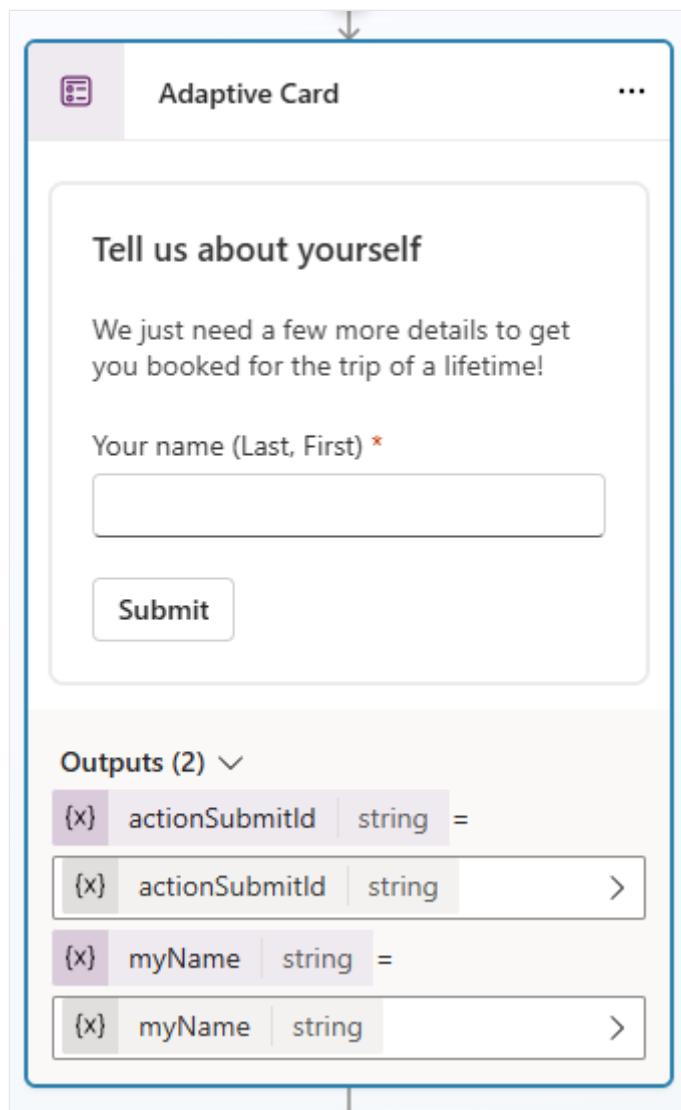
Add an Adaptive Card node

1. Select the **Add node** icon  below the node under which you want to add an **Adaptive Card** node, then select **Ask with Adaptive Card**.
2. Select the More icon (...) of the node, and then select **Properties**.
3. In the **Adaptive Card node properties** panel, select **Edit adaptive card**. The Adaptive card designer panel opens.
4. Add the desired elements for your card and configure their properties. Alternatively, in the **Card payload editor** pane, replace the default payload with the JSON literal for your card.

Tip

Your card must contain at least one submit button, as it must be an interactive card that allows a user to submit information back to the agent. If it doesn't and is only intended to show information, you should [add your Adaptive Card to a Message node](#).

5. When you're done with the initial design, select **Save** and close the designer panel. A preview of your card appears on the node. Copilot Studio automatically creates output variables based on the inputs specified in the code.



💡 Tip

If the output variables generated for your card are incorrect, you can manually update the list of variables and their types by selecting **Edit Schema** in the **Adaptive Card node properties** panel.

Your interactive Adaptive Card is now ready. When a user of your agent selects a submit button on a card, the output variables are populated with the information the user provided in their interaction with the card.

Other properties

Other properties allow you to control the behavior of the **Adaptive Card** node, such as:

- How the agent responds to an invalid response
- If it can be interrupted

If the agent is awaiting a submission from an Adaptive Card and the user sends a text message instead, this response is considered invalid, *unless* the message triggers an interruption. In this case, the following properties determine the behavior.

- **How many reprompts:** The number of times your agent tries to get a valid submission from the card. **Repeat up to 2 times** is the default. You can also select **Repeat once** or **Don't repeat**. For each retry, the card is resent to the user.
- **Retry prompt:** Use this property to define a message to be sent when a retry occurs, along with a repeat of the card. To define a retry message, select **Customize**, then enter the new prompt.
- **Allow switching to another topic:** If selected (default), an incoming message from a user when the agent is awaiting a card submission triggers an interruption and switches to another topic. If a topic switch occurs, the card is sent again to the user once the interrupting topic ends.

Submit button behavior for agents with consecutive cards

By design, Adaptive Cards allow selecting their submit buttons multiple times. If an agent has consecutive Adaptive Cards, and the user selects a button on an earlier card, the user might experience unexpected behavior.

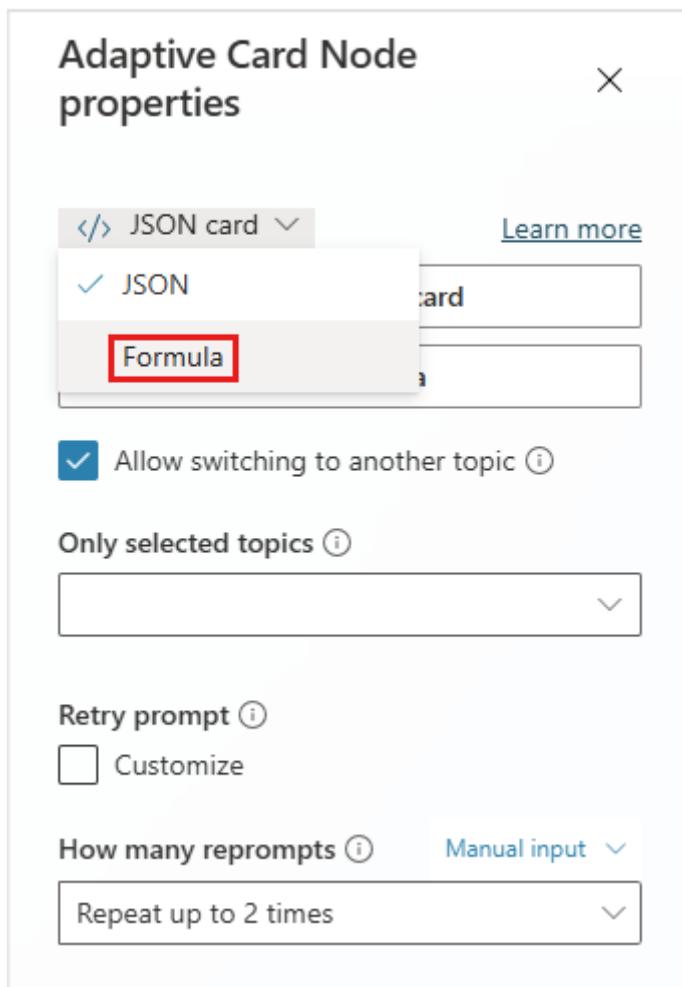
To prevent the submit action on one card from interfering with another card:

- **Isolate submit actions:** Ensure that each Adaptive Card has its own unique identifier and action handlers.
- **Use submit actions with unique data:** When defining the submit actions for your cards, include unique identifiers or data payloads that can help differentiate between the cards when the user selects a submit button.
- **Add robust event handling logic to your agent:** Define conditions based on the distinctive identifiers or payload elements associated with your submit buttons.
- **Debug and log:** Add detailed logging to your agent's event handling code to capture the sequence of actions and identify where unintended submissions occur.

Use Power Fx to make your card dynamic

You can use a Power Fx formula to include dynamic information on your card by referencing variables from your topic or agent.

1. Select the More icon (...) of the node, and then select **Properties**.
2. In the **Adaptive Card Node properties** panel, switch to **Formula**. Selecting **Formula** automatically converts the JSON representation of your card into a Power Fx formula.



For example, start with the following JSON literal for a card:

```
JSON

{
    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
    "type": "AdaptiveCard",
    "version": "1.5",
    "body": [
        {
            "type": "ColumnSet",
            "columns": [
                {
                    "type": "Column",
                    "width": 2,
                    "items": [
                        {
                            "type": "TextBlock",
                            "text": "Tell us about yourself",
                            "weight": "Bolder",
                            "size": "Medium",
                            "color": "#0078D4"
                        }
                    ]
                }
            ]
        }
    ]
}
```

```

        "wrap": true,
        "style": "heading"
    }, {
        "type": "TextBlock",
        "text": "We just need a few more details to get you booked for
the trip of a lifetime!",
        "isSubtle": true,
        "wrap": true
    }, {
        "type": "Input.Text",
        "id": "myName",
        "label": "Your name (Last, First)",
        "isRequired": true,
        "regex": "^[A-Z][a-z]+, [A-Z][a-z]+$",
        "errorMessage": "Please enter your name in the specified
format"
    }
]
}
]
],
"actions": [
    {
        "type": "Action.Submit",
        "title": "Submit"
    }
]
}

```

Here's the resulting Power Fx formula, using two variables **Topic.Title** and **Topic.Subtitle** instead of the hard-coded text from the JSON literal. (This example assumes the variables are defined in your topic.)

Power Fx

```
{
    '$schema': "http://adaptivecards.io/schemas/adaptive-card.json",
    type: "AdaptiveCard",
    version: "1.5",
    body: [
        {
            type: "ColumnSet",
            columns: [
                {
                    type: "Column",
                    width: "2",
                    items: [
                        {
                            type: "TextBlock",
                            text: Topic.Title,
                            weight: "Bolder",
                            size: "Medium",
                            wrap: true,

```

```
        style: "heading"
    },
{
    type: "TextBlock",
    text: Topic.Subtitle,
    isSubtle: true,
    wrap: true
},
{
    type: "Input.Text",
    id: "myName",
    label: "Your name (Last, First)",
   isRequired: true,
    regex: "^[A-Z][a-z]+, [A-Z][a-z]+$",
    errorMessage: "Please enter your name in the specified format"
}
]
}
]
}
],
actions: [
{
    type: "Action.Submit",
    title: "Submit"
}
]
}
```

Important

Once you begin editing in the formula panel, you *can't* go back to the original JSON code. To allow iterative design and changes, we recommend saving a copy of the original JSON in your own notes, or as a comment in the node. This precaution allows you to revert changes, if needed.

Add conditions to topics

Article • 10/11/2024

You can use **Condition** nodes to add branches in your topic, comparing the value of one variable to that of another variable or to a specific value. For example, if the value of `isClubMember` is `True`, then the topic offers a discount; otherwise, the topic presents the standard item price.

💡 Tip

Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

It's not possible to rename **Trigger** nodes and **Go to step** nodes.

Node names can be up to 500 characters in length.

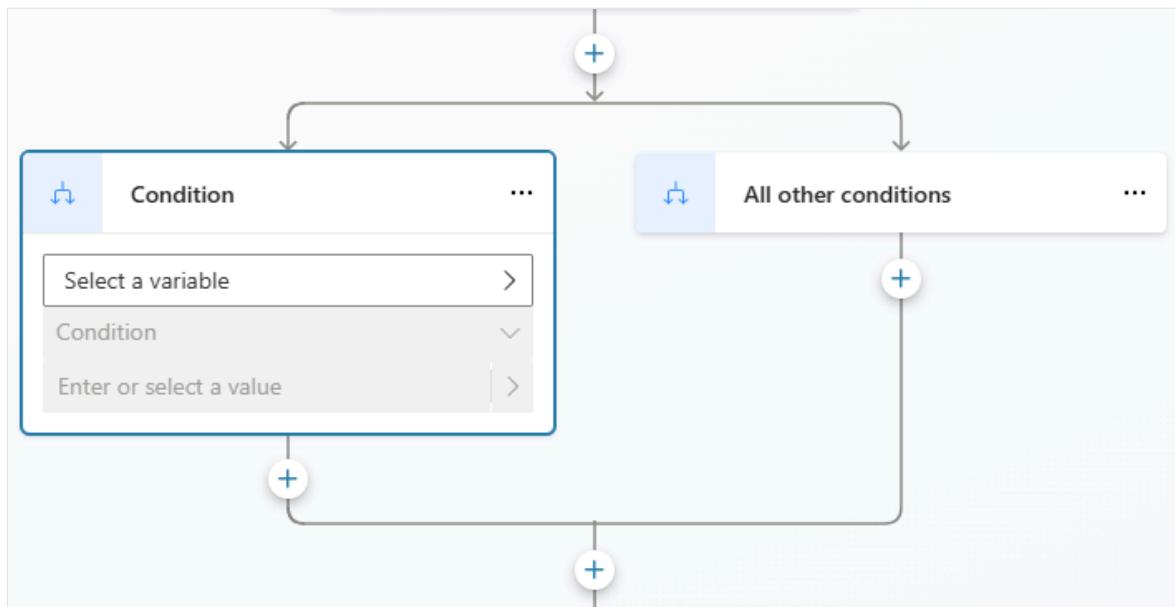
Prerequisites

- [Create and edit topics](#)

Add a condition

1. Open the desired topic.
2. Select the **Add node** icon  below the node under which you want to add a condition, and then select **Add a condition**.

This action adds two branches to your topic. One branch starts with a **Condition** node and defines the conversation path for when the condition is true. The other branch starts with an **All Other Conditions** node and defines the path for when the condition is false.



To build your condition in the **Condition** branch, you must select a variable, a logical operator, and a value.

3. Select **Select a variable** and select the variable to compare.

The logical operators available depend on the type of the variable you selected. For example, **is equal to** is selected by default, and it's available for all variable types. The operator **is greater than** is available for number types, but not for string types.

4. Select a different operator if needed.

5. Select or enter a value to compare with your variable.

The value or variable to compare must be of a compatible type, based on the variable and the operator. Some operators, such as **is Blank** or **is not Blank**, don't require you to add a comparison value.

For simple types, such as text or number, you can type a literal value. Select the **Select variable** icon (>) to select a variable or use a Power Fx formula.

6. Select **+ New condition** if you need to add more criteria to your condition. Select **And** to make the criteria cumulative, or select **Or** to make them exclusive of one another.



ⓘ Note

While configuring a condition that compares a string to a different data type, if you encounter a mismatch between the first variable and the comparison value, an error message noting the type comparison and the **Fix type mismatch** button appears. Select the button to have Copilot Studio automatically generate a [Power Fx formula](#) that resolves the mismatch.

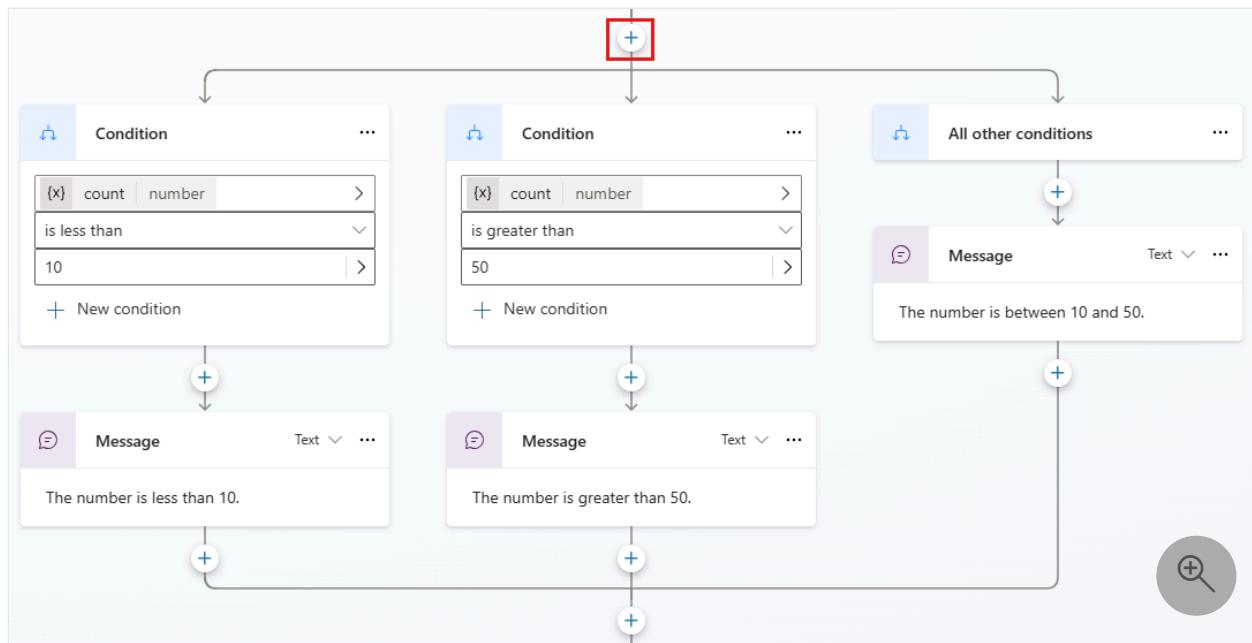
Add more Condition branches

Initially, a condition has two branches: a **Condition** branch for when the condition is true (that is, the *if* statement), and the **All Other Conditions** branch for when it's false (the *else* statement). You can add as many **Condition** branches (for any *else if* statements) as you need. Each condition is evaluated in order. The first branch with a condition that evaluates to true is followed. If all conditions evaluate to false, the **All Other Conditions** branch is followed.

For example, you might have a number variable and want three branches for the following cases:

- *If* the number is less than 10
- *Else if* the number is greater than 50
- *Else*—that is, when the number is between 10 and 50

To add another **Condition** branch, select the **Add node** icon  above your condition, and select **Add a condition**.

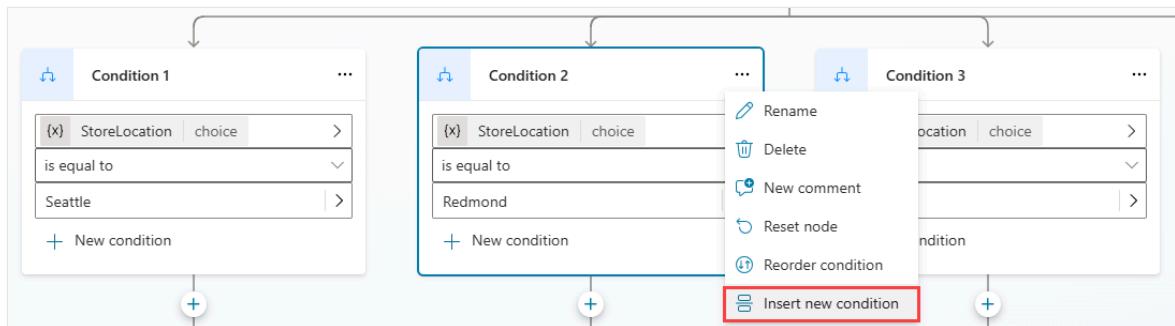


Selecting the **Add node** icon below a condition adds an entire new condition (with a **Condition** node and an **All Other Conditions** node), not a new condition branch.

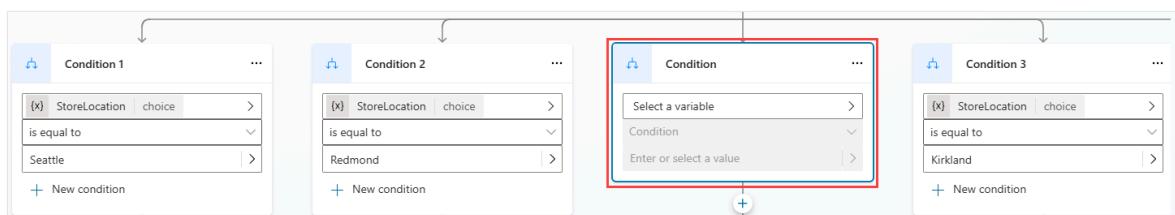
Insert a new condition to a group of conditions

You can insert a new condition into a group of conditions.

1. Select the **More** icon (...) of the **Condition** node, and then select **Insert new condition**.



2. The new condition is inserted to the right of the previously selected condition.



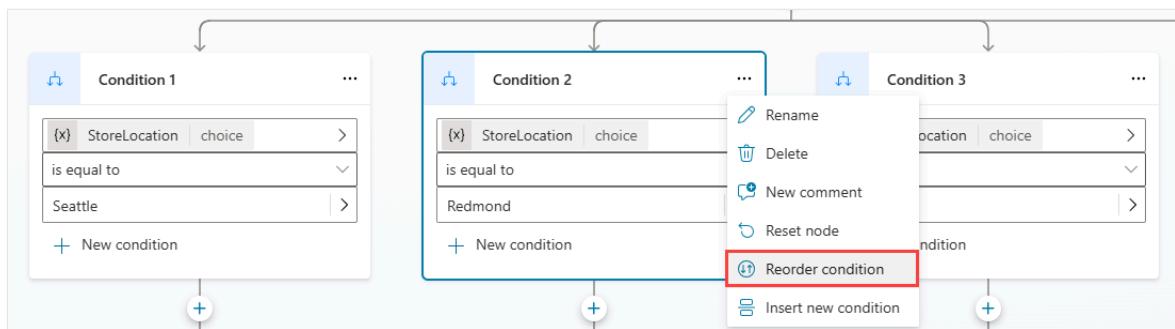
Note

You can't insert a new condition to the right of an **All Other Conditions** branch. It must remain as the last condition item on the right of the condition branch.

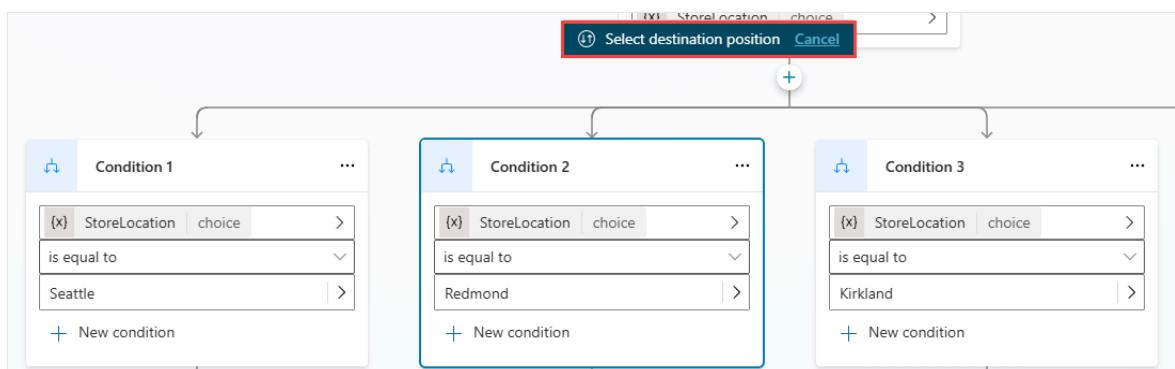
Organize conditions

For more complex condition structures, you can reorder conditions within the same condition group.

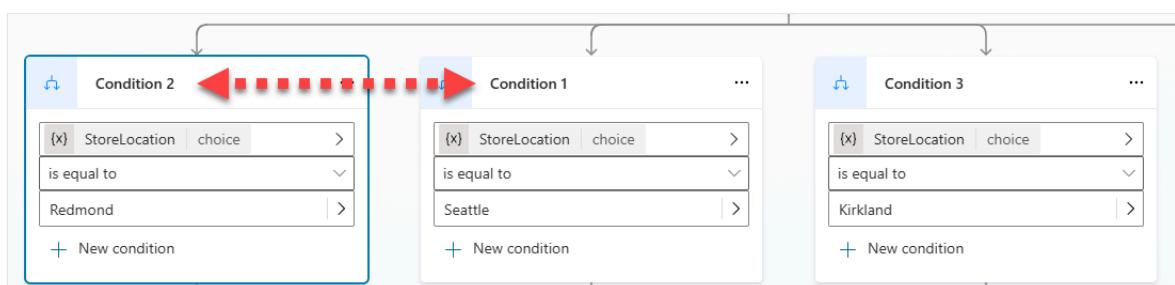
1. Select the **More** icon (...) of the **Condition** node, and then select **Reorder condition**.



2. The **Select destination position** message is displayed. Select the condition node that you want to swap positions with.



3. The condition node is now in its new position within the group.



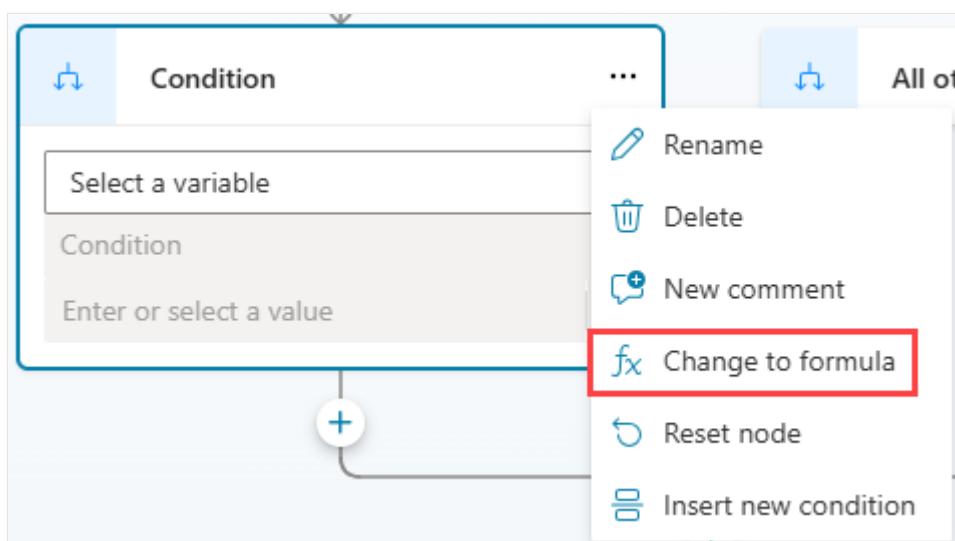
Note

- You can't move a condition to the right of an **All Other Conditions** branch. It must remain as the last condition item on the right of the condition branch.
- You can only reorder conditions in the same condition group.

Use Power Fx to define a condition

The default condition editor is good for most scenarios. If you need more flexibility, use Power Fx in a **Condition** branch.

1. Select the **More** icon (...) of the **Condition** node, and then select **Change to formula**.



The node switches to the formula editor control. If you defined a condition earlier, the formula behind that condition is shown in the formula editor.

2. To change the formula, select the **Select variable** icon (>) to open the Power Fx formula editor. The formula you enter here must return a Boolean (true or false) value.

Learn how to [create expressions using Power Fx](#).

To switch back to the simple condition editor, select the **More** icon (...) of the **Condition** node, and then select **Reset node**.

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Variables overview

Article • 03/18/2025

Variables can exist at four levels, or scopes:

- **Topic variables** can only be used in the topics in which they're created. This scope is the default for variables that you create.
- **Global variables** can be used in all topics. You can change the scope of a topic variable to make it a global variable.
- **System variables** are created automatically with your agent. They provide more contextual information about the conversation or the user. They're available in all topics.
- **Environment variables** are created in Power Platform and are read-only in Copilot Studio. They store the parameter keys and values, which then serve as input to various other application objects, including Copilot Studio. Separating the parameters from the consuming objects allows you to change the values within the same environment or when you migrate solutions to other environments. The alternative is leaving hard-coded parameter values within the components that use them.

Variable types

A variable is associated with a **base type**. The type determines what values the variable can contain and the operators that you can use when you construct a logical expression with it.

[] Expand table

Type	Description
String	A sequence of characters used to represent text
Boolean	A logical value that can only be <code>true</code> or <code>false</code>
Number	Any real number
Table	A list of values, but all values must be of the same type
Record	A collection of name-value pairs where values can be of any type
DateTime	A date, time, day of the week, or month relative to a point in time
Choice	A list of string values with associated synonyms

Type	Description
Blank	A placeholder for "no value" or "unknown value"; for more information, see Blanks in Power Fx

A variable's type is set the first time a value is assigned to it. After that, the type for that variable is fixed and it can't be assigned values of any other type. For example, a variable given the starting value of `1` is assigned the type **Number**. Attempting to assign it to a **String** value of `"apples"` results in an error.

When you're testing an agent, a variable might appear temporarily as the type **unknown**. An unknown variable hasn't been assigned a value yet.

Order of variables is determined from top to bottom of the authoring canvas. Nodes at the top of the authoring canvas are considered before nodes at the bottom.

When you create branches with **Condition** nodes, branches are ordered from left to right. Nodes in the leftmost branch are considered before nodes in the rightmost branch.

Entities

Web app

Copilot Studio uses **entities** to identify a specific type of information from a user's responses. The identified information is saved in a variable of the type that's appropriate for the information. The following table lists the variable base type associated with prebuilt entities.

 [Expand table](#)

Entity	Variable Base Type
Multiple-choice options	Choice
User's entire response	String
Age	Number
Boolean	Boolean
City	String
Color	String

Entity	Variable Base Type
Continent	String
Country or region	String
Date and time	DateTime
Email	String
Event	String
Integer	Integer
Language	String
Money	Number
Number	Number
Ordinal	Number
Organization	String
Percentage	Number
Person name	String
Phone number	String
Point of interest	String
Speed	Number
State	String
Street address	String
Temperature	Number
URL	String
Weight	Number
Zip code	String
Custom entity	Choice

Environment variables

Environment variables are a Power Platform concept. Environment variables enable the basic application lifecycle management (ALM) scenario of moving an application between Power Platform environments. In this scenario, the application stays exactly the same except for a few key external references that are different between the source environment and the destination environment.

Use *environment* variables in the same way as topic, global, and system variables. One notable difference is that environment variables are *read-only* in Copilot Studio. Agent authors can't modify environment variables in Copilot Studio. However, administrators can change the value of environment variables in Power Apps. In Copilot Studio, you can use the [Variables panel](#) to see information about an environment variable. The **Variables** panel also has a link to the native authoring experience.

The published version of an agent that uses environment variables has the values that were set for these variables when the agent was published. Whenever an administrator updates environment variables, you must republish any agents that use these variables, for the changes to be effective at runtime. However, there's one exception: you don't need to republish your agents when the value of an environment variable of type *secret* changes. Unlike other environment variables, secret variables are retrieved at runtime.

Environment variable types in Copilot Studio map to Power Apps data types as follows:

[+] [Expand table](#)

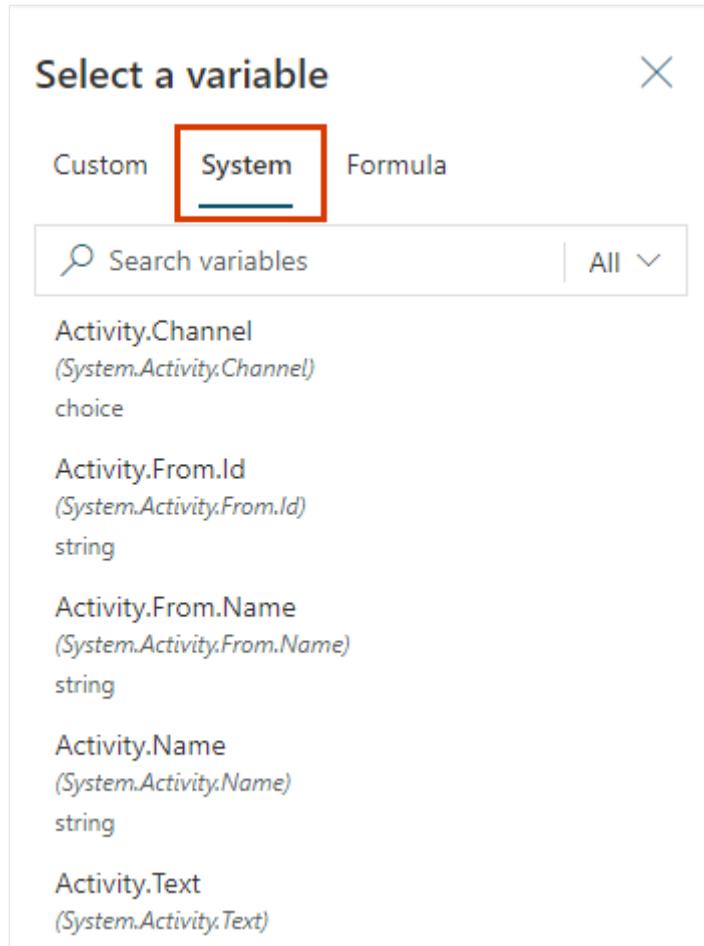
Type in Copilot Studio	Type in Power Apps
Decimal number	Number
JSON	Detect the type from the value. If not JSON => Unspecified (validation error)
Text	String
Yes/No	Boolean
Data source	String
Secret	String

 **Note**

Environment variable errors are visible in the test chat and when publishing. However, these errors aren't shown in the *Topic* list because they aren't topic variables.

System variables

Every agent comes with built-in system variables that provide additional information about a conversation.



The screenshot shows a 'Select a variable' dialog box. At the top, there are three tabs: 'Custom', 'System' (which is highlighted with a red box), and 'Formula'. Below the tabs is a search bar with the placeholder 'Search variables' and a dropdown menu set to 'All'. The main area lists several system variables:

- Activity.Channel
(*System.Activity.Channel*)
choice
- Activity.From.Id
(*System.Activity.From.Id*)
string
- Activity.From.Name
(*System.Activity.From.Name*)
string
- Activity.Name
(*System.Activity.Name*)
string
- Activity.Text
(*System.Activity.Text*)

① Note

For more information about voice-enabled agent variables, see [Use voice variables](#).

Not all system variables are shown in the list. You must access these hidden system variables with a [Power Fx formula](#).

To use system variables in a Power Fx formula, you must add `System.` before the variable name. For example, to include the system variable `User.DisplayName` in a formula, refer to it as `System.User.DisplayName`.

[+] [Expand table](#)

Name	Type	Definition
Activity.Attachments	table	The File attachments provided by the user

Name	Type	Definition
Activity.Channel	choice	The channel ID of the current conversation
Activity.ChannelData	any	An object that contains channel-specific content
Activity.ChannelId	string	The channel ID of the current conversation, as a string
Activity.From.Id	string	The channel-specific unique ID of the sender
Activity.From.Name	string	The channel-specific user-friendly name of the sender
Activity.Name	string	The name of the event
Activity.Recipient.Id	string	The incoming activity's Type property.
Activity.Recipient.Name	string	Represents the display name for the agent within the channel. In telephony channel context, the value of this variable is the phone number to which the agent is attached to.
Activity.Text	string	The most recent message sent by the user
Activity.Type	choice	Type of activity
Activity.TypeId	string	Type of activity , as a string
Activity.Value	any	Open-ended value
Bot.EnvironmentId	string	The environment ID of the agent
Bot.Id	string	The ID of the agent
Bot.Name	string	The name of your agent
Bot.SchemaName	string	The schema name of the agent
Bot.TenantId	string	The tenant ID of the agent
ClientPluginActions	choice	Collection of Dynamic Client Plugin Actions to consider for generative orchestration
Conversation.Id	string	The unique ID of the current conversation
Conversation.InTestMode	Boolean	Boolean flag that represents if the

Name	Type	Definition
		conversation is happening in test canvas
Conversation.LocalTimeZone	string	Name of the time zone to be used by the user in the IANA Time Zone database format
Conversation.LocalTimeZoneOffset	datetime	The time offset from UTC for the current local time zone
Error.Code	string	The error code for the current error. The error message for the current error.
Error.Message	string	The error message for the current error. Note: This variable is supported only if the trigger is <code>On Error</code> .
FallbackCount	number	This variable counts the times that a topic couldn't be matched to the user input. Note: This variable is supported only if the trigger is <code>On Unknown Intent</code> .
InactivityTimer.Continue	Boolean	Boolean flag that represents whether the timer needs to continue. Note: This variable is supported only if the trigger is <code>Inactivity</code> .
InactivityTimer.Count	number	Number of time <code>OnInactivity</code> timer has fired due to user inactivity after configured time. Note: This variable is supported only if the trigger is <code>Inactivity</code> .
LastMessage.Id	string	The ID of the previous message sent by the user
LastMessage.Text	string	The previous message sent by the user
Recognizer.ExtractedEntities	choice	Represents the extracted entities from triggering message. Note: This variable is supported only if the trigger is <code>On Select Intent</code> .
Recognizer.IntentOptions	choice	Represents the intent options when recognizer returns ambiguous results. Note: This variable is supported only if the trigger is <code>On Select Intent</code> .
Recognizer.SelectedIntent	choice	Represents the intent selected from recognizer.

Name	Type	Definition
		Note: This variable is supported only if the trigger is <code>On Select Intent</code> .
Recognizer.TriggeringMessage.Id	string	The ID of the user message that triggered the current topic
Recognizer.TriggeringMessage.Text	string	The user message that triggered the current topic
Recognizer.MultipleTopicsMatchedReason	string	Used to determine why multiple topics were matched. Note: This variable is supported only if the trigger is <code>On Select Intent</code> .
SignInReason	choice	Used to determine what sign-in option is needed when triggering the topic. Note: This variable is supported only if the trigger is <code>On Sign In</code> .
User.Language	choice	This variable is used to set the user language locale per conversation.

Variables for integrated authentication (default)

The following variables are available for agents configured to authenticate with Microsoft. This configuration uses Microsoft Entra ID authentication and is the default for new agents.

For more information, see [Authentication variables](#).

[Expand table](#)

Name	Type	Definition
User.DisplayName	string	The display name of the user currently talking to the agent.
User.Email	string	The email address of the user currently talking to the agent.
User.FirstName	string	The first name of the user currently talking to the agent.
User.Id	string	The unique ID of the user currently talking to the agent.
User.IsLoggedIn	Boolean	Boolean flag that represents whether the user currently talking to the agent is authenticated or not.

Name	Type	Definition
User.LastName	string	The family name of the user currently talking to the agent.
User.PrincipalName	string	The user principal name of the user currently talking to the agent.

Variables for manual authentication with Generic OAuth 2

The following variables are available for agents configured to use the Generic OAuth 2 service provider.

[\[+\] Expand table](#)

Name	Type	Definition
User.AccessToken	string	The access token for the user authenticating with the agent.
User.DisplayName	string	The display name of the user currently talking to the agent.
User.Id	string	The unique ID of the user currently talking to the agent.
User.IsLoggedIn	Boolean	Boolean flag that represents whether the user currently talking to the agent is authenticated or not.

Variables for manual authentication with Microsoft Entra ID

The following variables are available for agents configured to use the Microsoft Entra ID (formerly Azure Active Directory) service providers.

[\[+\] Expand table](#)

Name	Type	Definition
User.AccessToken	string	The access token for the user authenticating with the agent.
User.DisplayName	string	The display name of the user currently talking to the agent.
User.Email	string	The email address of the user currently talking to the agent.
User.FirstName	string	The first name of the user currently talking to the agent.
User.Id	string	The unique ID of the user currently talking to the agent.
User.IsLoggedIn	Boolean	Boolean flag that represents whether the user currently talking to

Name	Type	Definition
		the agent is authenticated or not.
User.LastName	string	The family name of the user currently talking to the agent.
User.PrincipalName	string	The principal name of the user currently talking to the agent.

Variables for voice-enabled agents

The following variables are only available for voice-enabled agents.

ⓘ Note

For more information about voice-enabled agent variables, see [Use voice variables](#).

[Expand table](#)

Name	Type	Definition
Activity.InputDTMFKeys	string	The raw DTMF key value that was received from telephony.
Activity.SpeechRecognition.Confidence	number	The confidence score for the Azure Site Recovery hypothesis entire result, 0 to 1.
Activity.SpeechRecognition.MinimallyFormattedText	string	Slightly formatted text of the Azure Site Recovery hypothesis result. For example, "Five hundred dollars". Words are spelled out, but basic capitalization and punctuation are included.
Activity.UserInputType	choice	The input type from the user's most recent input. The value can be either text, speech or DTMF.
Conversation.OnlyAllowDTMF	Boolean	Boolean flag that represents whether the IVR should be set to DTMF-only mode at runtime.
Conversation.SipUuiHeaderValue	string	The UUI header string used to pass context into IVR on call start.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Work with variables

10/04/2025

You can use variables to store customers' responses and reuse their content later in the conversation.

You can also use variables to create logical expressions that dynamically route the customer down different conversation paths. For example, save a customer's name in a variable called "customerName," and the agent can address the customer by name as the conversation continues.

Variables can also be passed to, and returned from, [other topics](#) and [Power Automate flows](#).

Create a variable



Any node that returns an output, such as a **Question** node, automatically creates an output variable of the appropriate type.

Tip

Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

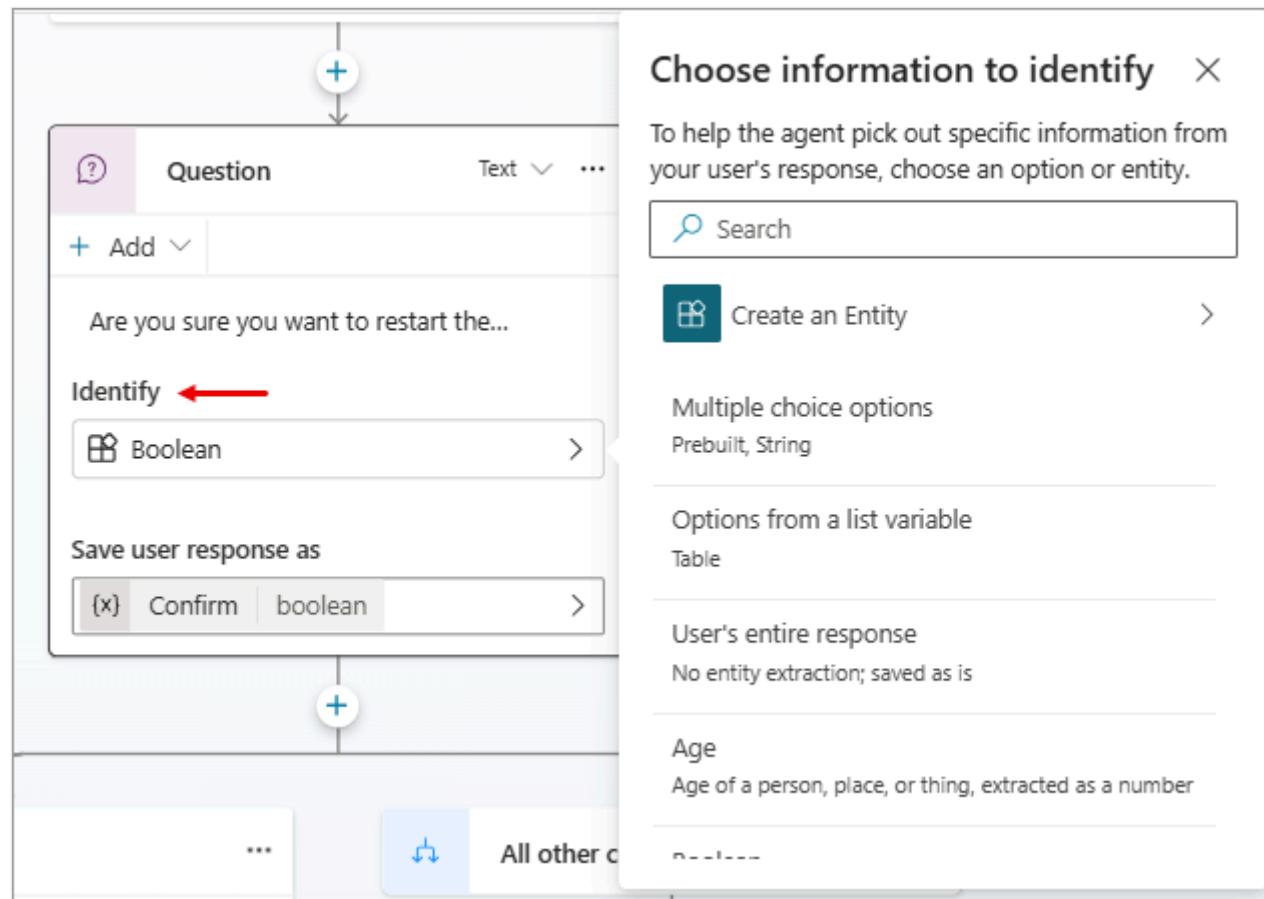
It's not possible to rename **Trigger** nodes and **Go to step** nodes.

Node names can be up to 500 characters in length.

Pick an entity to use

Web app

Question nodes are created with multiple-choice options by default. To use a different [prebuilt or custom entity](#), select the **Identify** box, and then select the desired type of information.



Rename a variable

Web app

Variables are automatically assigned a name when you create them. A best practice is to give your variables meaningful names to make their purpose clear to anyone who must maintain your agent.

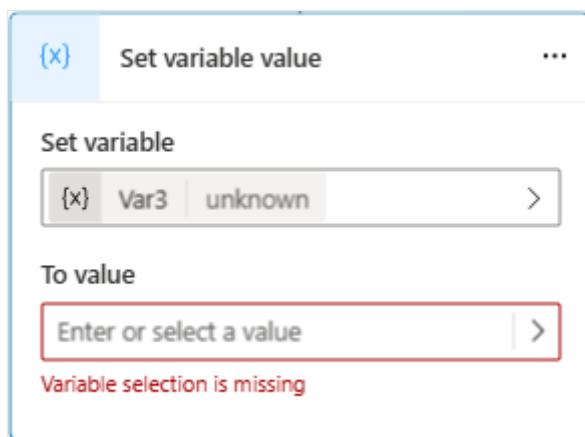
1. Select the variable. The [Variable properties](#) panel appears.

2. Under **Variable name**, enter the desired name for your variable.

Set a variable

Typically you use a **Question** node to store user input in a variable. However, there might be situations where you want to set the value of a variable yourself. In those cases, use a **Set variable value** node.

1. Select the **Add node**  icon under the node after which you want to add a **Set variable value** node.
2. Select **Variable management > Set a variable value**. A **Set variable value** node appears on the canvas.
3. Select the box under **Set variable**, and then select **Create a new variable**. A new variable is created. Its type is **unknown** until you assign a value to it.



4. For **To value**, assign a value using one of the following options:

- Type a [literal value](#).
- Select an existing variable of the same type.
- Use a [Power Fx formula](#). Power Fx formulas are useful for more complex types where literal values can't be used, such as Table and Record types.

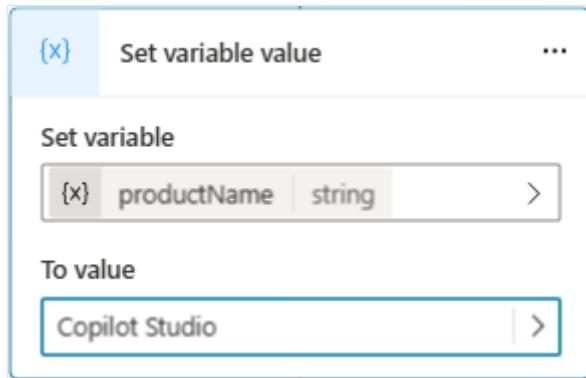
Use variables in Action nodes

When you use a variable in an **Action** node, if its base type matches a parameter type specified for a flow, or for a Bot Framework skill, you can feed it to that parameter. The output from **Action** nodes generates new variables.

For more context, see the example scenarios in [Call an agent flow from a topic](#) and [Use input and output variables to pass information](#).

Use literal values for variable inputs

In nodes where you can set a value for input parameters, you can always enter a literal value instead of selecting another variable as the value.



The node attempts to interpret literal values as a string, a number, or a Boolean. For example, 123 is interpreted as a number. If you want it to be interpreted as a string value instead, you can wrap the value in double quotes, like this: "123".

For some scenarios (for example, scenarios using complex types), you might need to use a [Power Fx formula](#) to set a specific type.

Use environment variables for Azure Key Vault secrets

An environment variable can reference a secret in the Key Vault. Secret environment variables are a special case of environment variables with unique considerations.

To create a secret environment variable in the Power Apps portal, you must [configure its Key Vault](#).

To authorize Copilot Studio to read this Key Vault, you must perform the following actions:

1. Assign the Key Vault Secrets User role to the *Microsoft Copilot Studio Service* application.
2. To authorize all agents from the environment to access the secret, create a tag `AllowedEnvironments` on the secret and add the allowed environment IDs separated by commas.
3. To authorize only specific agents from the environment to use this Key Vault, create a tag `AllowedAgents` and put the agent identifier in the format `{envId}/{schemaName}`. For

multiple values, separate the values with commas.

If you reach the maximum number of characters but still need to add more agents, add another tag with a descriptive but unique name (for example: `AllowedAgents2`).

The secret value is cached in Dialog runtime for five minutes. Unsuccessful reads are cached for 30 seconds.

Warning

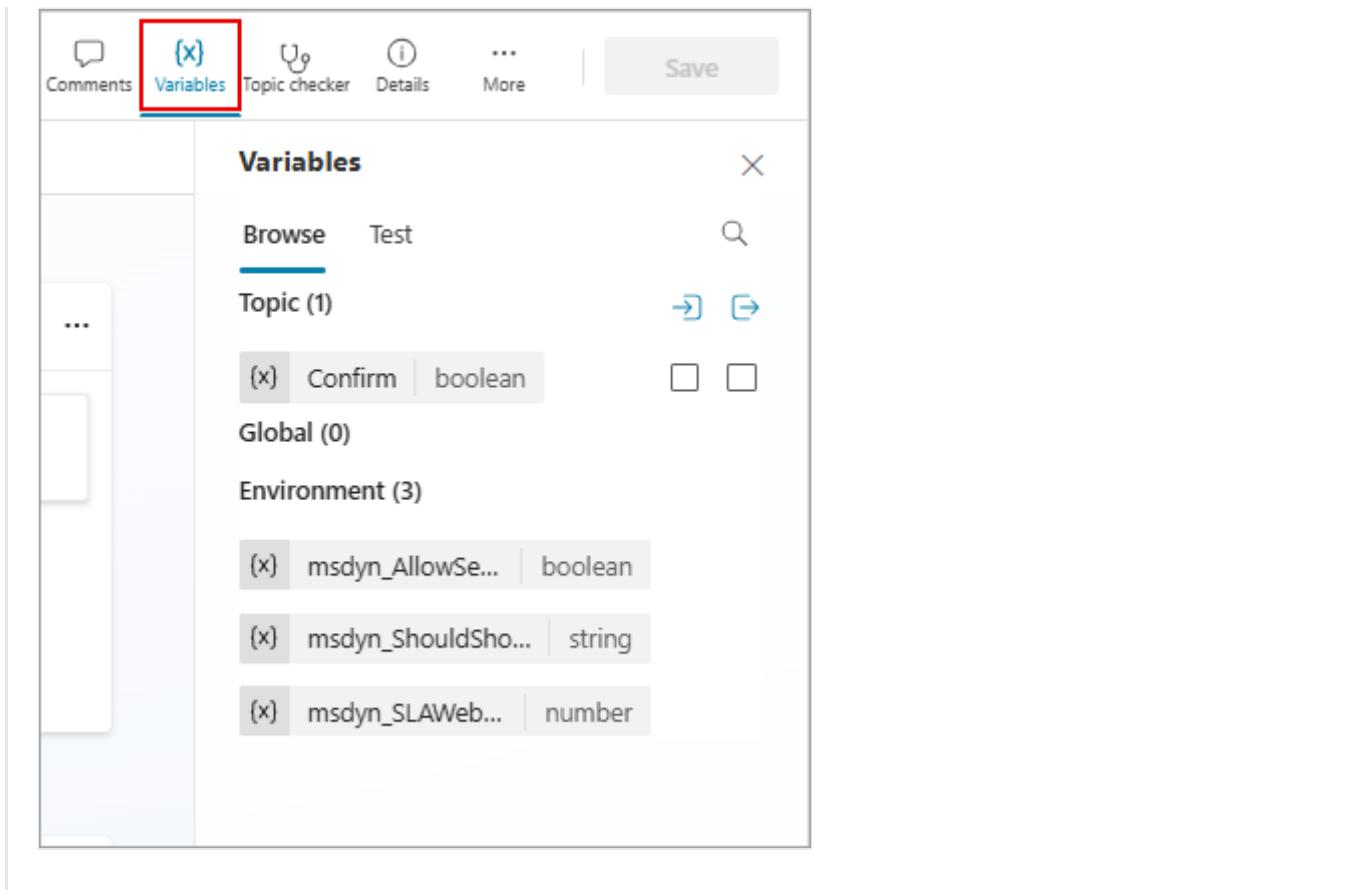
By adding the secret in the agent, you might expose the value of this secret. Anyone who can edit the agent in the environment could add a **Message** node and return the value of the secret environment variables in a message.

Variables panel

Web app

The **Variables** panel is where you can view all the variables that are available for a topic, regardless of which nodes they're defined or used in. For each variable, you can select whether it can receive its value from other topics, return its value to other topics, or both. You can also select a variable to edit its properties in the [Variable properties](#) panel.

To open the **Variables** panel, select **Variables** on the topic's menu bar.



Variable properties panel

In the **Variable properties** panel, you can rename a variable, see where your agent uses it, or convert it to a [global variable](#). You can't convert a global variable back to a topic variable, however. You can also select whether a topic variable can receive values from or pass its value to other topics.

To open the **Variable properties** panel for a variable, select the desired variable in the **Variables** panel. You can also open the **Variable properties** panel from any node that uses variables, by selecting the desired variable.

Pass variables between topics

When you redirect one topic to another, you can pass the values of variables from the originating topic to the destination topic and also return values from the destination topic to the originating topic. Passing variables between topics is especially useful when an earlier topic already collected information that a later topic needs. Your users certainly appreciate not having to answer the same questions more than once.

Receive values from other topics

When a topic defines a variable (for example, in a **Question** node), the agent asks the user the question to fill in the variable's value. If the agent already acquired the value in an earlier topic, there's no reason to ask the question again. In such cases, you can set the variable to **Receive values from other topics**. When another topic redirects to this one, it can pass either the value of a variable or a **literal value** to this variable, and skip the question. The experience for the user talking to the agent is seamless.

Web app

In this example, we use two topics, **Greeting** and **Talk to Customer**. Both topics ask for the customer's name. However, if the **Greeting** topic runs first, the **Talk to Customer** topic skips its question. Instead, it uses the value of the variable passed from the **Greeting** topic.

Here's the flow of the **Talk to Customer** topic:

```
graph TD; Trigger[Trigger] --> Question[Question]; Question --> Message[Message];
```

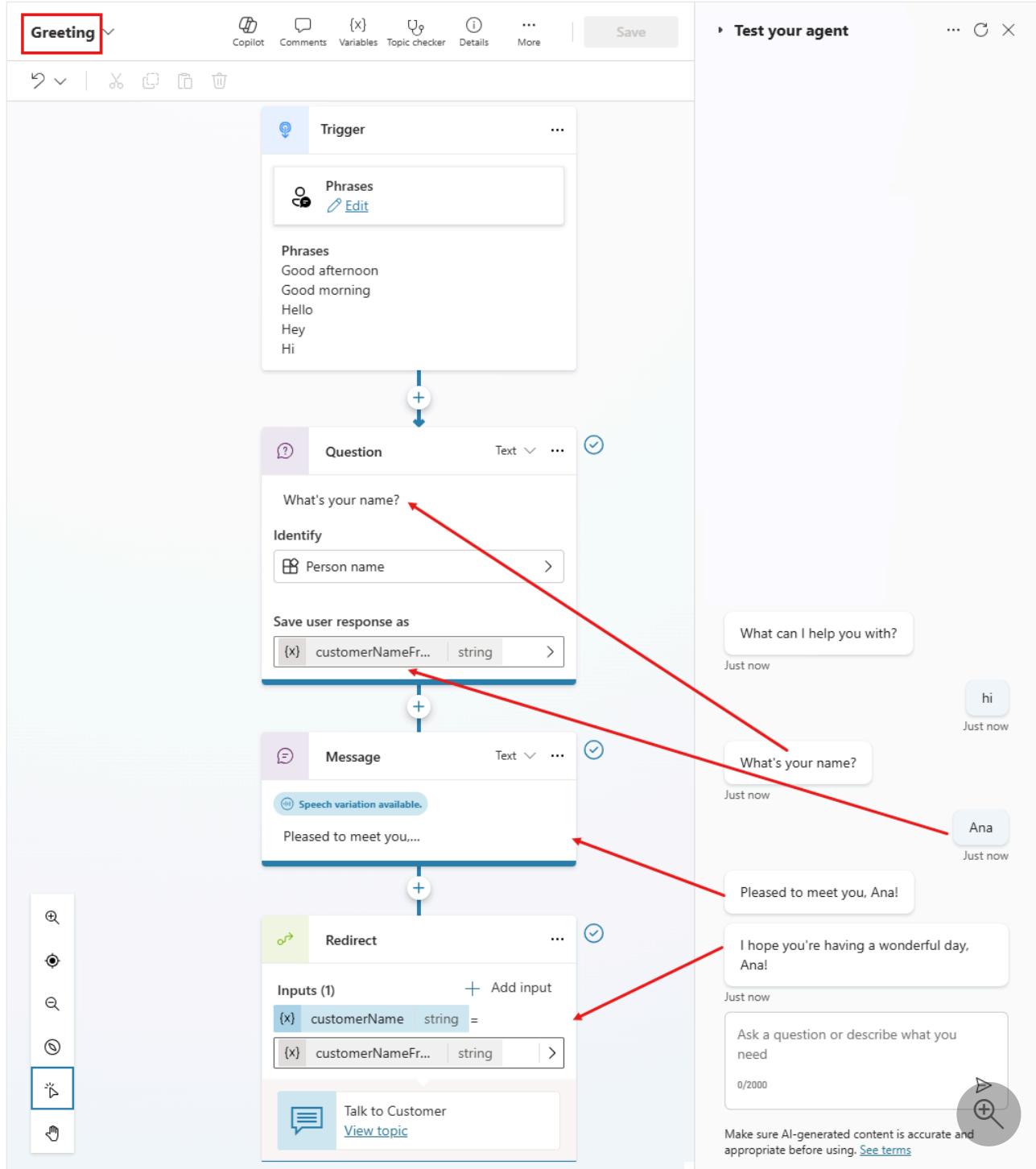
The test panel on the right shows the conversation flow:

- User message: "talk to me" (11 minutes ago)
- Bot response: "What can I help you with?" (11 minutes ago)
- User message: "What should I call you?" (11 minutes ago)
- Bot response: "Ana" (11 minutes ago)
- Bot message: "I hope you're having a wonderful day, Ana!" (11 minutes ago)
- User message: "Ask a question or describe what you need" (0/2000)

Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

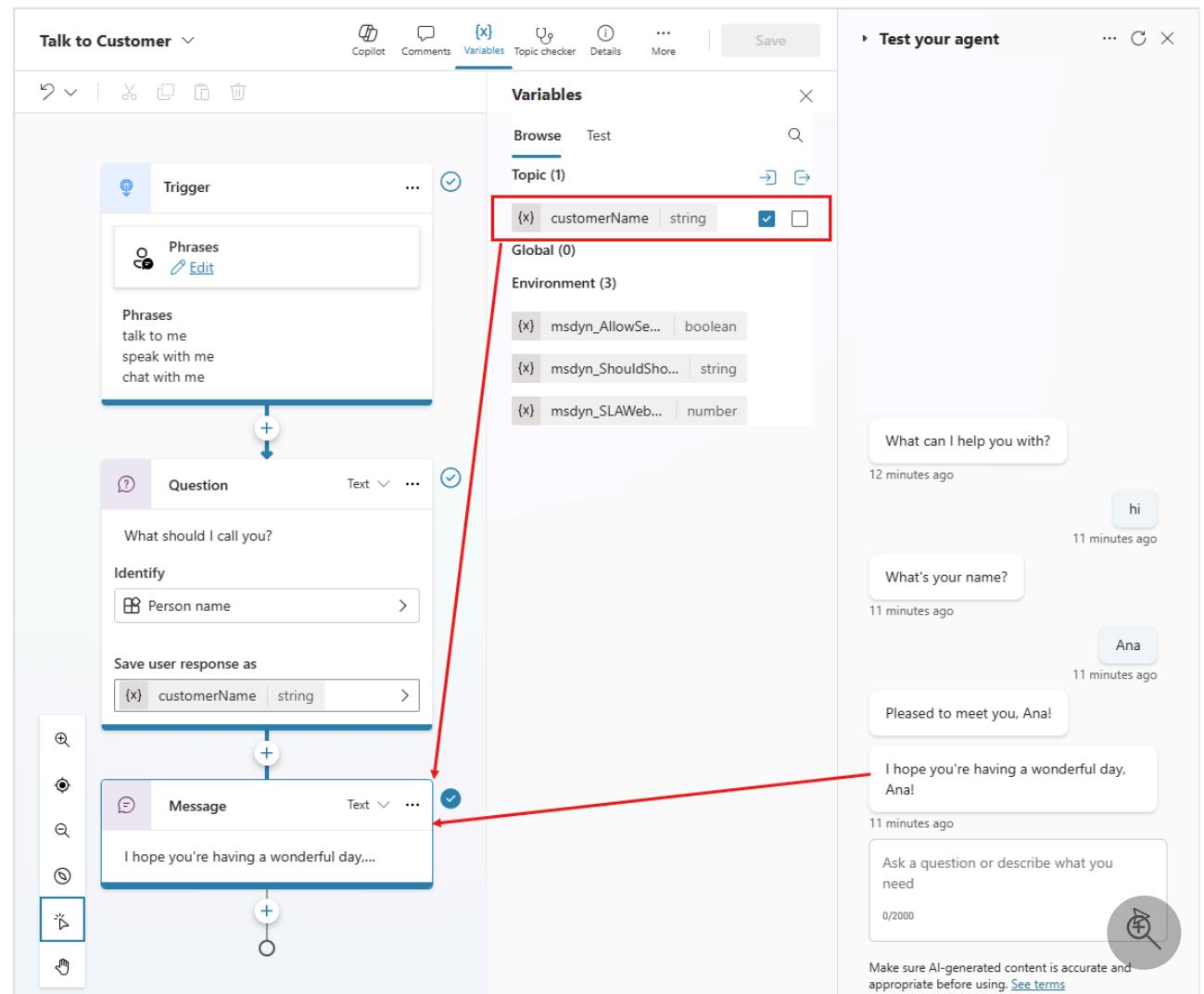
As shown in the test panel, if this topic is triggered first, it asks the user, "What should I call you?" It stores the value in a string variable called `customerName`. The `customerName` variable is also set to get its value from other topics. The topic concludes with the message, "I hope you're having a wonderful day, `customerName`!"

Here's the flow of the Greeting topic:



As shown in the test panel, if this topic is triggered first, it asks the user, "What's your name?" It stores the value in a string variable called `customerName`. The topic sends the message, "Pleased to meet you, `customerName`!" It then redirects to the **Talk to Customer** topic, which sends the message, "I hope you're having a wonderful day, `customerName`!" Note, however, that the **Talk to Customer** topic skipped asking for the user's name again. Instead, it used the value of the `customerName` variable passed from the **Greeting** topic.

Finally, here's that second conversation again, this time from the perspective of the **Talk to Customer** topic:



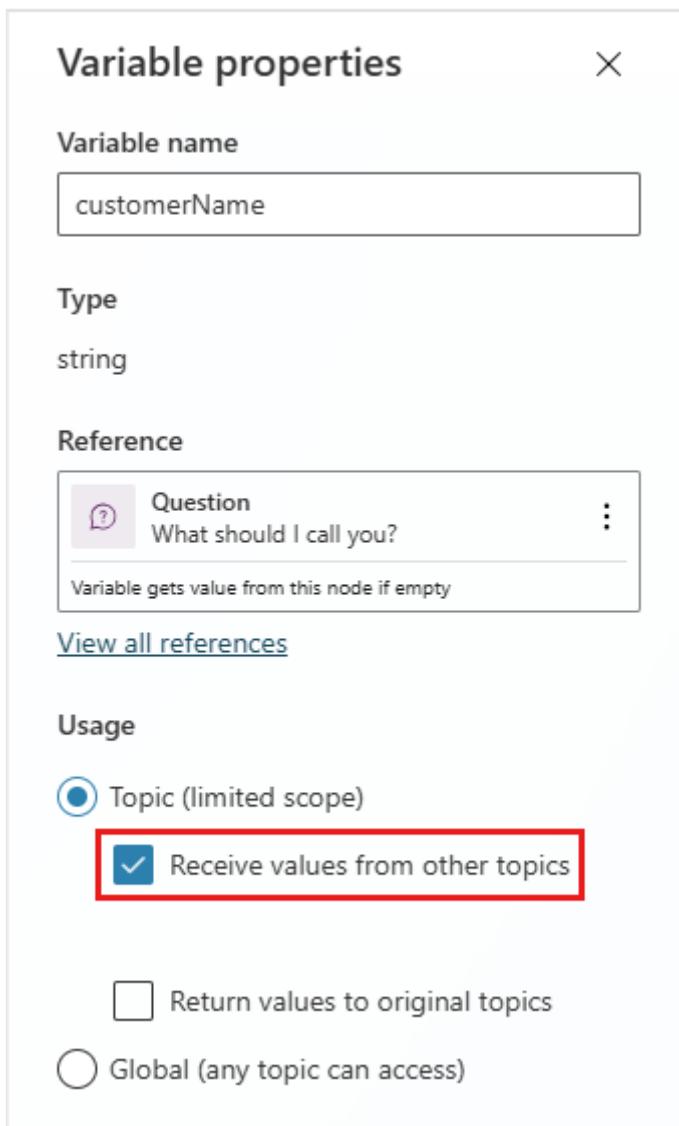
Let's walk through the steps to set up a topic to receive values from other topics. Our example scenario uses the predefined topic **Greeting** as the source topic, and a new topic **Talk to Customer** as the destination topic, but the same steps work for any topic that should use a value from an earlier topic when possible.

Set up the destination topic

The destination topic is the topic being redirected to, the one that receives values from other topics. In our example, it's **Talk to Customer**.

1. Create a new topic and call it "Talk to Customer."
2. Add trigger phrases such as "talk to me," "speak with me," "chat with me."
3. Add a **Question** node and enter "What should I call you?" for the message.
4. Under **Identify**, select the prebuilt entity **Person name**.
5. Select the name of the variable. The **Variable properties** panel opens.

6. Replace the default name with "customerName", and then select **Receive values from other topics**.



7. Add a **Message** node.

8. In the message box, enter "I hope you're having a wonderful day, ".

9. Select the **Insert variable** icon ($\{x\}$), and then select **customerName**.

10. Select the space after the variable and enter "!".

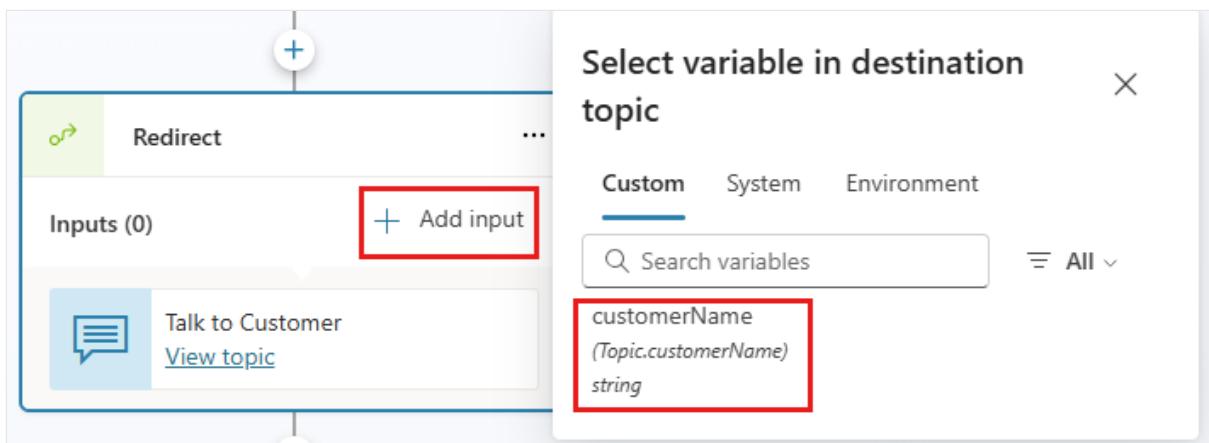
11. Save the topic.

Set up the source topic

The source topic is the topic doing the redirecting, the one that provides the value and passes it to the destination topic. In our example, it's **Greeting**.

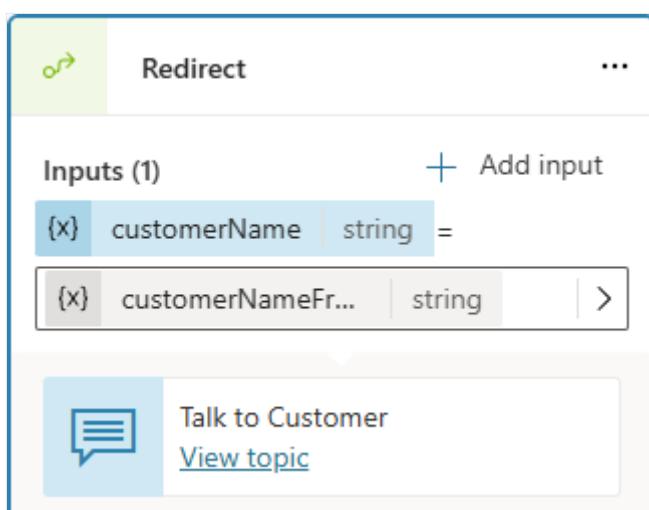
1. Go to the **Greeting** topic and delete the default nodes, except for the **Trigger** node.

2. Add a **Question** node and enter "What's your name?" for the message.
3. Under **Identify**, select the prebuilt entity **Person name**.
4. Replace the default name with "customerNameFromGreeting".
5. Add a **Message** node.
6. In the message box, enter "Pleased to meet you, ".
7. Select the **Insert variable** icon ({x}), and then select **customerNameFromGreeting**.
8. Select the space after the variable and enter "!".
9. Add a **Redirect** node, and select the destination topic **Talk to Customer**.
10. Select **Add input**, and then select the variable from the destination topic that you want to pass a value to.



11. Select the > icon, and then select the variable whose value you want to pass, **customerNameFromGreeting**, in this example.

The **Redirect** node should look like this:



12. Save the topic.

Return values to original topics

In an agent, a topic is meant to collect a specific piece of information. Multiple other topics can call it and expect it to return the information as a variable. The variable becomes part of the originating topic and can be used like any other variable. Information the agent obtains is thus available across topics, reducing the need for [global variables](#).

Web app

Let's continue with the example from the previous section. We ask a new question in the **Talk to Customer** topic, and then return the answer to the **Greeting** topic.

Set up the source topic for a returned variable

When you're returning a variable to a topic, the source topic is the topic that provides the value to pass back to the original topic. In this example, the source topic is **Talk to Customer**.

1. Go to the source topic.
2. Add a **Question** node and enter "What city do you live in?" for the message.
3. Under **Identify**, select the prebuilt entity **City**.
4. Select the variable to open the **Variable properties** panel. Name it "customerCity," and then select **Return values to original topics**.

The screenshot shows the Microsoft Bot Framework Composer interface. At the top, there's a navigation bar with 'Copilot', 'Comments', 'Variables', 'Topic checker', 'Details', 'More', and a 'Save' button. The main workspace shows a 'Talk to Customer' topic. Inside the topic, there's a 'Text' node with the message 'I hope you're having a wonderful day,...'. Below it is a 'Question' node with the question 'What city do you live in?'. Underneath the question node, there's a 'Save user response as' section where the variable '{x} customerCity' is selected as a string type. To the right, a 'Variable properties' pane is open for 'customerCity'. It shows the variable name, type (string), and a reference back to the 'Question' node. In the 'Usage' section, the 'Topic (limited scope)' option is selected, and the 'Return values to original topics' checkbox is checked.

5. Add a **Message** node.

6. Select the **Insert variable** icon (**{x}**), and then select **customerCity**.

7. After the variable in the message box, enter " must be beautiful this time of year!".

8. Save the topic.

Set up the destination topic for a returned variable

When you're returning a variable to a topic, the destination topic is the topic that receives values from the current topic. In our example, the destination topic is **Greeting**.

1. Go to the destination topic.

2. The variable you selected in the source topic should appear on the **Redirect** node as an output variable.

The screenshot shows the Microsoft Flow designer interface. A flow titled "Greeting" is being edited. The flow consists of the following steps:

- Greeting**: A message step with the text "What's your name?"
- Identify**: An identify step set to "Person name". The output variable is "{x} customerName string".
- Message**: A message step with the text "Pleased to meet you...".
- Redirect**: A redirect step. It takes the input "{x} customerName string" and outputs "{x} customerCity string". It also includes a "Talk to Customer" action with a link to "View topic".
- Message**: A message step with the text "customerCity must b...".

On the right side, there is a "Test your agent" pane showing a conversation:

- User: "What can I help you with?" (3 minutes ago)
- AI: "hi" (2 minutes ago)
- User: "What's your name?" (2 minutes ago)
- AI: "Ana" (2 minutes ago)
- User: "Pleased to meet you, Ana!" (2 minutes ago)
- AI: "I hope you're having a wonderful day, Ana!" (2 minutes ago)
- User: "What city do you live in?" (2 minutes ago)
- AI: "Philadelphia" (2 minutes ago)
- AI: "Philadelphia must be beautiful this time of year!" (2 minutes ago)
- User: "Ask a question or describe what you need" (0/2000)

A red arrow points from the "customerName" field in the Redirect step to the "What's your name?" message in the conversation. Another red arrow points from the "customerCity" field in the Redirect step to the "What city do you live in?" message in the conversation.

3. Save the topic.

Parse values

The **Parse value** node allows you to convert a value of one type, to a value of another type. A common use case for the **Parse value** node is to convert raw JSON. For example, an agent calls a flow, which in turn calls an API. The API then returns a result. Now you need a way to parse this result and return one or more primitive variables. With the **Parse value** node, you can send the entire API result as a string, such as the following:

JSON

```
{  
    "Name": "Parker",  
    "Position": "Product manager",  
    "Company": "Contoso",  
    "FormerPositions": [  
        {  
            "Position": "Customer service representative"  
        }  
    ]  
}
```

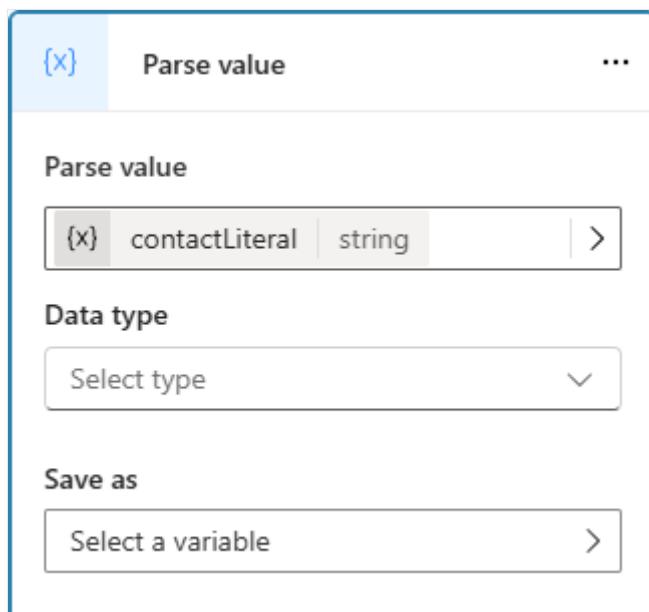
The **Parse value** node can convert the string variable that you have now into a variable of type **Record**, for which the Power Fx editor provides IntelliSense code completion and suggestions.

In addition to parsing JSON strings, a key use of the **Parse value** node is parsing untyped objects at runtime. The most common use case is when you receive an event and you need to parse the event value. Or you might want to parse the `System.Activity.ChannelData` property, which varies at runtime by channel.

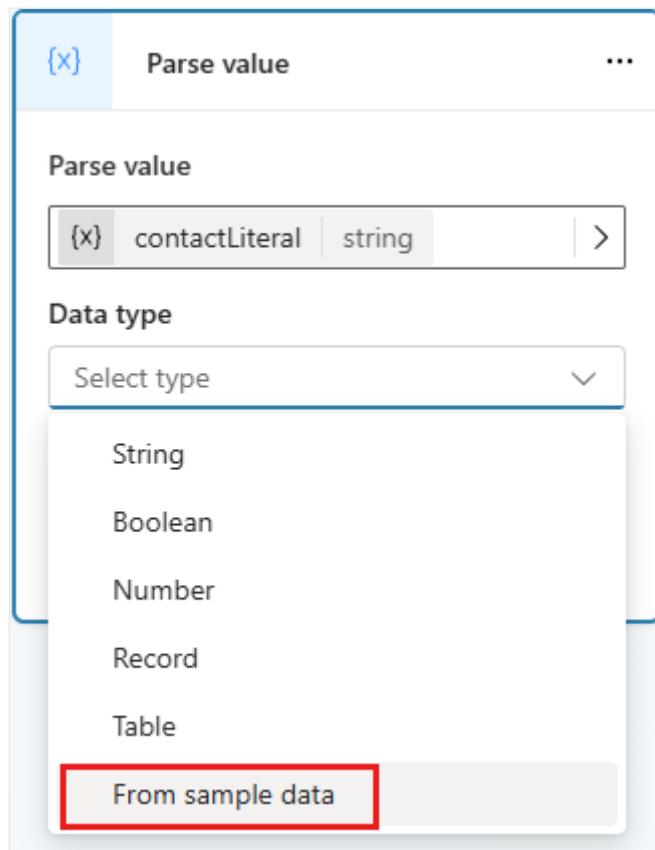
If you need to parse information from an event coming from Teams, refer to the Teams developer documentation to find an example of what the expected event might look like. You can then follow the instructions in this section, using the example event as sample data.

To add a **Parse value** node to a topic:

1. Select the **Add node**  icon under the node after which you want to parse a value, point to **Variable management**, and select **Parse value**.
2. Add a **Parse value** node, and select the variable you want to parse.



3. Select the data type. In this case choose **From Sample Data**.



4. Select **Get Schema from Sample JSON**, enter the desired JSON example in the editor that opens, and select **Confirm**.

This screenshot shows the 'Get schema from sample JSON' dialog. The title is 'Get schema from sample JSON' and the sub-instruction is 'Use a sample of the data you expect to be parsed in JSON format.' Below this is a code editor window containing the following JSON sample:

```
1 {  
2     "Name": "Parker",  
3     "Position": "Product manager",  
4     "Company": "Contoso",  
5     "FormerPositions": [  
6         {  
7             "Position": "Customer service representative"  
8         }  
9     ]  
}
```

5. Finally, select the variable to hold the parsed value. Usually you would create a new variable.

The screenshot shows two overlapping dialogs. The background dialog is titled 'Parse value' and contains the following fields:

- Parse value:** {x} contactLiteral | string
- Data type:** From sample data
- Get schema from sample JSON**
- Edit schema**
- Save as:** Select a variable

The foreground dialog is titled 'Select a variable' and has tabs for 'Custom' and 'System'. It includes a search bar and a list of variables:

- {x} Create a new variable (highlighted with a red box)
- contactLiteral
(Topic.contactLiteral)
string

The output variable now has the expected type: **record**.

The screenshot shows the 'Parse value' dialog with the following configuration:

- Parse value:** {x} contactLiteral | string
- Data type:** Record
- Edit schema**
- Save as:** {x} Var1 | record (highlighted with a red box)

Related content

- [Variables overview](#)
- [Work with global variables](#)
- [Use entities and slot filling in agents](#)
- [Customize the look and feel of an agent](#)

Work with global variables

Article • 03/21/2025

Variables store your customers' responses to questions from your agent. For example, you can save a customer's name in a variable called `UserName`. The agent can then address the customer by name as the conversation continues.

By default, a variable's value can only be used in the topic where the variable is created. However, it's possible to reuse the same value across topics. For example, a Welcome topic asks for the customer's name and email address. In the Appointment Booking topic, you want the agent to remember what the customer entered and not ask again.

One way to reuse a variable is to [pass the variable](#) from one topic to another. The other way is to make the variable global in scope, and that's what this article covers. Global variables are called that because they're available in all topics across the entire agent. They can also be set from external sources.

Global variables apply during a single user session. You specify which variables are global variables to distinguish them from topic-level variables.

Create a global variable

You create a global variable by changing the scope of a topic variable.

Note

The name of a global variable must be unique across all topics.

Web app

1. [Create a variable](#) or use the Variables panel to open an existing variable.
2. On the Variable properties panel, select **Global (any topic can access)**.

The variable name is given the prefix `Global.` to differentiate it from topic-level variables. For example, the variable `UserName` is displayed as `Global.UserName`.

3. Save the topic.

Use global variables

Web app

When you're composing a message in a **Message** node or a **Question** node, select the {x} icon to view the variables that are available to the topic. Global variables appear in the **Custom** tab along with any topic variables. Variables are listed in alphabetical order.

Find all topics using a global variable

You can find where a global variable is defined and what other topics are using it. This feature can be useful if you're working on a new agent, or if you have multiple variables and [complex topic branching](#).

Web app

1. Select the desired global variable on the authoring canvas, or in the [Variables](#) panel.
2. On the **Variable properties** panel, in the **Reference** section, select **View all references**.
3. Switch to the **Other** tab, and select any topic where the variable is used to go directly to that topic and node.

Deleting global variables

If you remove a global variable used in other topics, the references to that variable in the topics are marked as **Unknown**. You receive a warning about deleting the global variable before you can confirm the operation.

Nodes that contain references to a deleted global variable indicate that they contain an unknown variable.

Topics with nodes that contain references to deleted global variables might stop working. Ensure that you remove or correct all the topics that were using the deleted variable before publishing your agent.

Lifecycle of global variables

Web app

By default, the value of a global variable persists until the session ends. The [Clear variable values](#) node resets the values of global variables and is used in the [Reset Conversation](#) system topic. When a redirection triggers that topic (or when the user enters a phrase such as "Start over"), all global variables are reset.

Set global variables from external sources

Web app

To make sure the agent starts a conversation with some context, you can use a global variable and set its value from an external source. Let's say that your site requires users to sign in. If you store a user's name in a global variable and pass it to your agent, the agent can greet customers by name before they start typing their first question. Another example scenario is passing context from Dynamics 365 Customer Service to an agent so it can start the conversation with knowledge of what the customer wants to achieve.

To prevent undesirable latency, you can specify how long your agent can wait for a value. You can also set a default value to use when the external source fails to respond in a timely fashion.

ⓘ Note

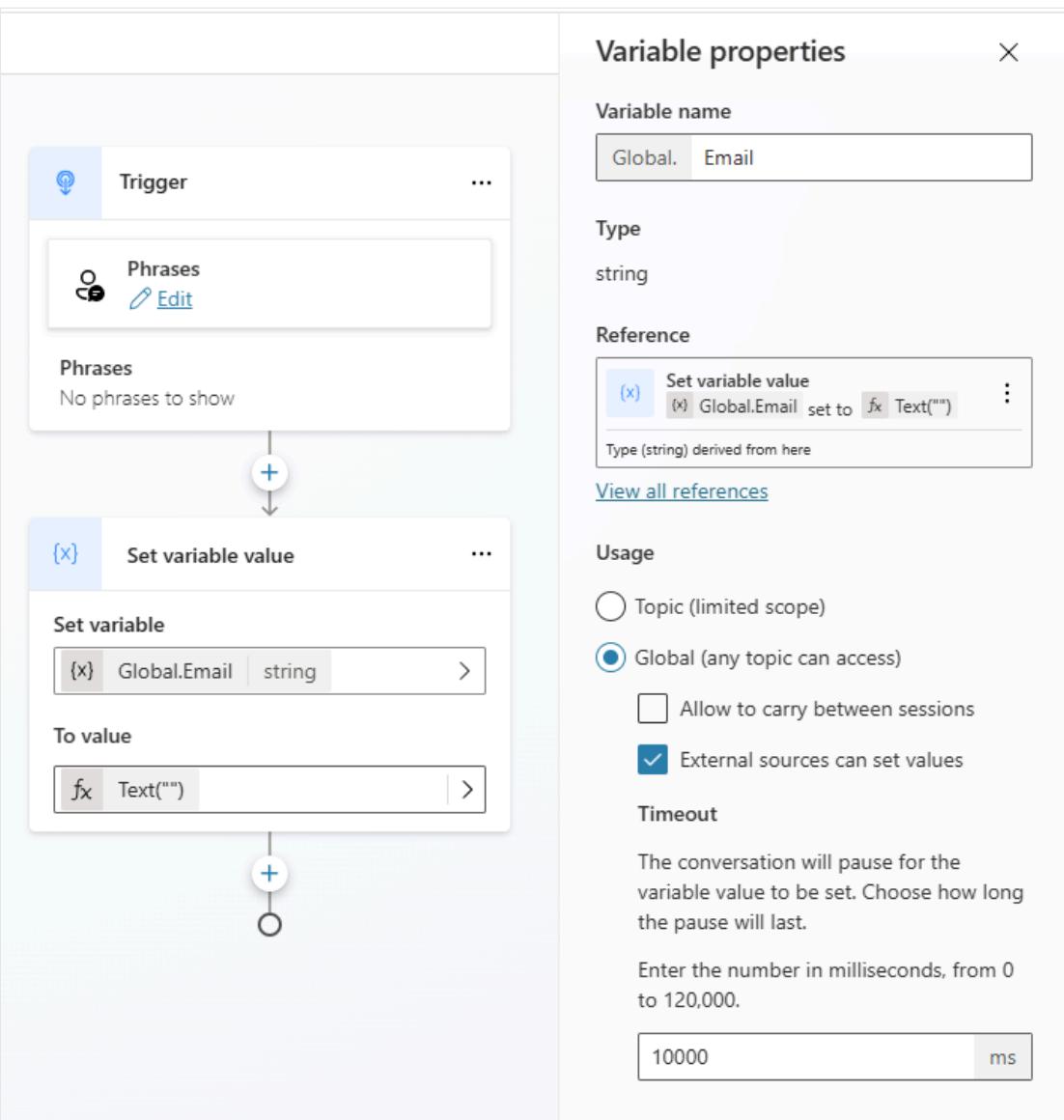
Agents that are published to the Dynamics 365 Contact Center channel for IVR use cases don't support time-out values configured for global variables set by external sources.

1. Create a dedicated topic to hold the configuration for all the variables meant to be set from external sources. You might name this topic "Set context variables" for example. This topic doesn't serve any other purposes, so it doesn't need to have trigger phrases set.
2. Add a [Set variable value](#) node to your dedicated topic.
3. Under **Set variable**, open the variable picker, and select [Create a new variable](#).

4. Select the default name of the new variable. The **Variable properties** panel appears.
5. Replace the default name to one that matches *exactly* the name of the variable being passed in from the external system.
6. Under **Usage**, select **Global (any topic can access)**, and **External sources can set values**.
7. Under **Reference**, select the three dots (:) in the upper-right corner, and select **Get value from this node if empty**.
8. (Optional) Set a time-out delay, in milliseconds. This value determines how long your agent can wait for the variable to be set by an external source before timing out and continuing with the default value you set in the **Set variable value** node. This setting is relevant in scenarios where the variable depends on a long-running or asynchronous process, but your agent must respect a maximum latency to ensure a good user experience.

For variables coming from Omnichannel for Customer Service, we suggest a value of 10 seconds (10,000 ms) as a maximum wait time.

9. In the **Set variable value** node, enter the default value to use if the time-out is reached. At runtime, your agent will expect values with the same data type. If you want this default value to be an empty string, use `Text("")` as a formula.



10. For any other values you expect to come from an external system, add more **Set variable value** nodes to your dedicated topic, and configure the required global variables in the same fashion.

Thus configured, your agent is ready to test. When the agent is invoked, instead of waiting indefinitely for all variables to be populated, your agent can immediately start sending any messages that aren't dependent on the variables being passed in. When your agent attempts to access a variable that's being set externally, it pauses until the value arrives or the time-out occurs. Learn more about [optimizing agents to minimize latency](#).

ⓘ Important

During the normal flow of a conversation, if your agent sets a variable that would otherwise come from an external value, the value set from within your

agent prevails. Any value passed in context is ignored. This rule prevents your agent from overwriting values intentionally set in topics.

Set global variables in an embedded agent

Web app

If you're [embedding your agent in a simple web page](#), you can append variables and their definitions to the agent's URL. Or, if you'd like a little more control, you can use a `<script>` code block to call and use variables programmatically.

The variable name in the query string of the URL must match the name of the global variable without the `Global.` prefix. For example, a global variable `Global.UserName` would be referred to as `UserName` in the query.

The examples that follow uses a basic declaration for the variables. In a production scenario, you might pass in as the query parameter or variable definition another variable that already stores the user's name (for example, if you have the user name from a sign-in script).

Append the variables and their definitions to the agent's URL as [query string parameters](#) in the format `botURL?`

`variableName1=variableDefinition1&variableName2=variableDefinition2.`

For example:

- You have a global variable named `Global.UserName`.
- Your agent's URL is <https://web.powerva.microsoft.com/webchat/bots/12345>.
- To pass in the user's name when starting an agent conversation on your website, attach the `UserName=` query string as:
<https://web.powerva.microsoft.com/webchat/bots/12345?UserName=Ana>.

The parameter name is case-insensitive. `username=Ana` would also work in this example.

Add global variables to a custom canvas

You can also add the variable to a [custom canvas](#).

Web app

1. In the `<script>` section on the page where you have your agent, define the variables as follows, substituting `variableName1` for the variable name without the `Global.` prefix and `variableDefinition1` for the definition. Separate multiple variables with commas (,).

HTML

```
const store = WebChat.createStore({}, ({ dispatch }) => next =>
action => {
  if (action.type === 'DIRECT_LINE/CONNECT_FULFILLED') {
    dispatch({
      type: "WEB_CHAT/SEND_EVENT",
      payload: {
        name: "pvaSetContext",
        value: {
          "variableName1": "variableDefinition1",
          "variableName2": "variableDefinition2"
        }
      },
    });
  }
  return next(action);
});
```

2. In your `<script>` section, call the `store` when you embed your agent, as in the following example where `store` is called just before where `styleOptions` is called (you must replace the `BOT_ID` with your agent's ID):

HTML

```
const BOT_ID = "12345-5678";
const theURL =
"https://powerva.microsoft.com/api/botmanagement/v1/directline/dire
ctlinetoken?botId=" + BOT_ID;

fetch(theURL)
  .then(response => response.json())
  .then(conversationInfo => {
    window.WebChat.renderWebChat(
      {
        directLine: window.WebChat.createDirectLine({
          token: conversationInfo.token,
        }),
        store,
        styleOptions
      },
    );
  });

```

```
        document.getElementById('webchat')
    );
})
.catch(err => console.error("An error occurred: " + err));
```

Authentication-related global variables

Depending on the agent's authentication setup, you have a set of global variables associated with the selected authentication provider. For details about which set of variables are available and how to use them, see [Add user authentication to topics](#).

Related content

- [Work with variables](#)
 - [Customize the look and feel of an agent](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create expressions using Power Fx

06/30/2025

Power Fx is a low-code language that uses Excel-like formulas. Use Power Fx to create complex logic that allows your agents to manipulate data. For instance, a Power Fx formula can set the value of a variable, parse a string, or use an expression in a condition. For more information, see the [Power Fx overview](#) and [formula reference](#).

(!) Note

Power Fx formulas in Copilot Studio use US-style numbering. That is, the decimal separator is a period or dot, as in `12,567.892`. This means you must use commas (,) as [parameter separators for your Power Fx formulas](#).

Prerequisites

- [Create and edit topics](#)
- [Variables overview](#)

Use variables in a formula

To use a variable in a Power Fx formula, you must add a prefix to its name to indicate the variable's scope:

- For [system variables](#), use `System.`.
- For [global variables](#), use `Global.`.
- For [topic variables](#), use `Topic.`.

For example, to use the system variable `Conversation.Id` in a formula, refer to the system variable as `System.Conversation.Id`.

The screenshot shows the Power BI formula editor interface. On the left, there's a visual representation of a flow with a plus sign icon. On the right, the 'Enter formula' screen is displayed with the 'Formula' tab selected. The formula bar at the top contains the text 'System.Conversation.Id'. Below it, under 'Variables', there's a row for 'Var1' with a 'String' type and an output of 'System.Conversation.Id'. A note says 'Press CTRL + M to disable / enable Tab character'. The 'Set Variable Value' step is highlighted with a blue border.

Use literal values in a formula

In addition to using variables in a Power Fx formula, you can enter literal values. To use a literal value in a formula, you must enter it in the format that corresponds to its [type](#). The following table lists the data types and the format of their corresponding literal values.

[Expand table](#)

Type	Format examples
String	"hi", "hello world!", "copilot"
Boolean	Only <code>true</code> or <code>false</code>
Number	1, 532, 5.258, -9201
Record and Table	<code>[1], [45, 8, 2], ["cats", "dogs"], { id: 1 }, { message: "hello" }, { name: "John", info: { age: 25, weight: 175 } }</code>
DateTime	<code>Time(5,0,23), Date(2022,5,24), DateTimeValue("May 10, 2022 5:00:00 PM")</code>
Choice	Not supported
Blank	Only <code>Blank()</code>

Common Power Fx formulas

The following table lists data types and Power Fx formulas you can use with each data type.

Type	Power Fx formulas
String	Text function Concat and Concatenate functions Len function Lower, Upper, and Proper functions IsMatch, Match, and MatchAll functions EndsWith and StartsWith functions Find function Replace and Substitute function
Boolean	Boolean function And, Or, and Not functions If and Switch functions
Number	Decimal, Float, and Value functions Int, Round, RoundDown, RoundUp, and Trunc functions
Record and Table	Concat and Concatenate functions Count, CountA, CountIf, and CountRows functions ForAll function First, FirstN, Index, Last, and LastN functions Filter, Search, and LookUp functions JSON function ParseJSON function
DateTime	Date, DateTime, and Time functions DateValue, TimeValue, and DateTimeValue functions Day, Month, Year, Hour, Minute, Second, and Weekday functions Now, Today, IsToday, UTCNow, UTCToday, IsUTCToday functions DateAdd, DateDiff, and TimeZoneOffset functions Text function
Blank	Blank, Coalesce, IsBlank, and IsEmpty functions Error, IfError, IsError, IsBlankOrError functions

Use Power Fx to set a variable

In this example, a Power Fx expression stores and outputs the customer's name in capital letters.

1. Create a topic and add a **Question** node.
2. For **Enter a message**, enter "What is your name?".
3. Under **Identify**, select the entity **Person name**.

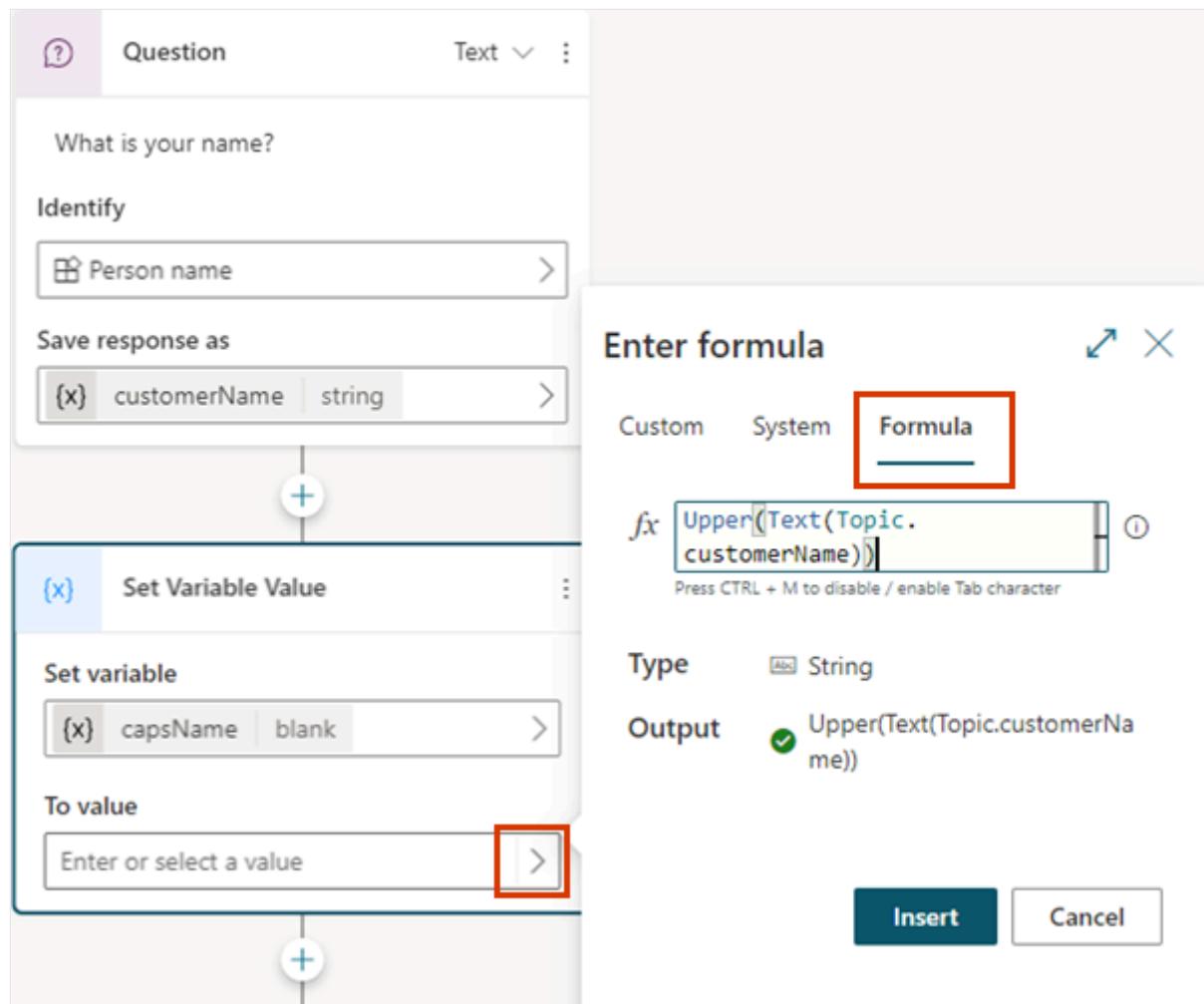
4. Select the box under **Save user response as**, and then select the variable `Var1` and name it `customerName`.

5. Under the **Question** node, select **+** and then select **Set a variable value**.

6. Select the box under **Set variable**, and then select **Create new** and name it `capsName`.

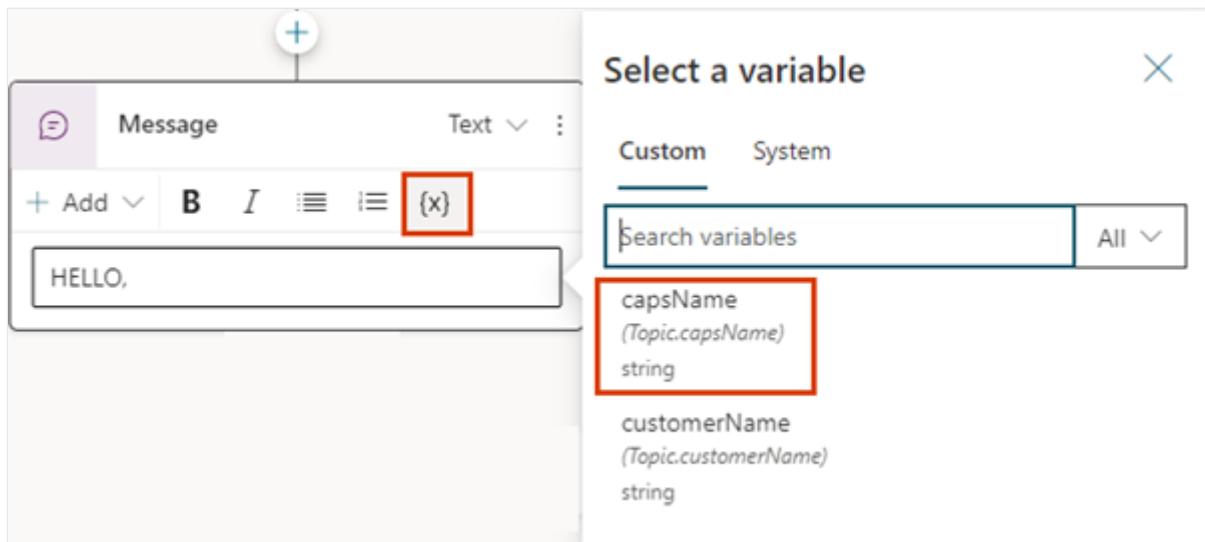
7. In the **To value** box, select the **>** arrow, and then select the **Formula** tab.

8. In the **fx** box, enter `Upper(Text(Topic.customerName))`, and then select **Insert**.



9. Under the **Question** node, select **+** and then select **Send a message**.

10. Enter "HELLO ", select `{x}`, and then select `capsName`.

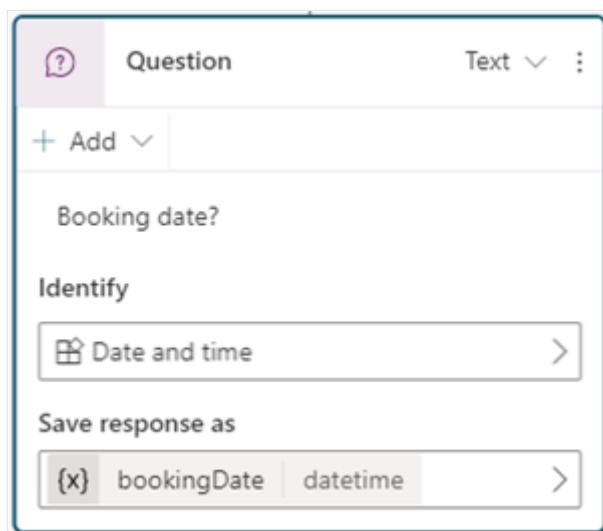


Use a Power Fx formula as a condition

To evaluate more complex expressions, set up **Condition** nodes to use Power Fx formulas.

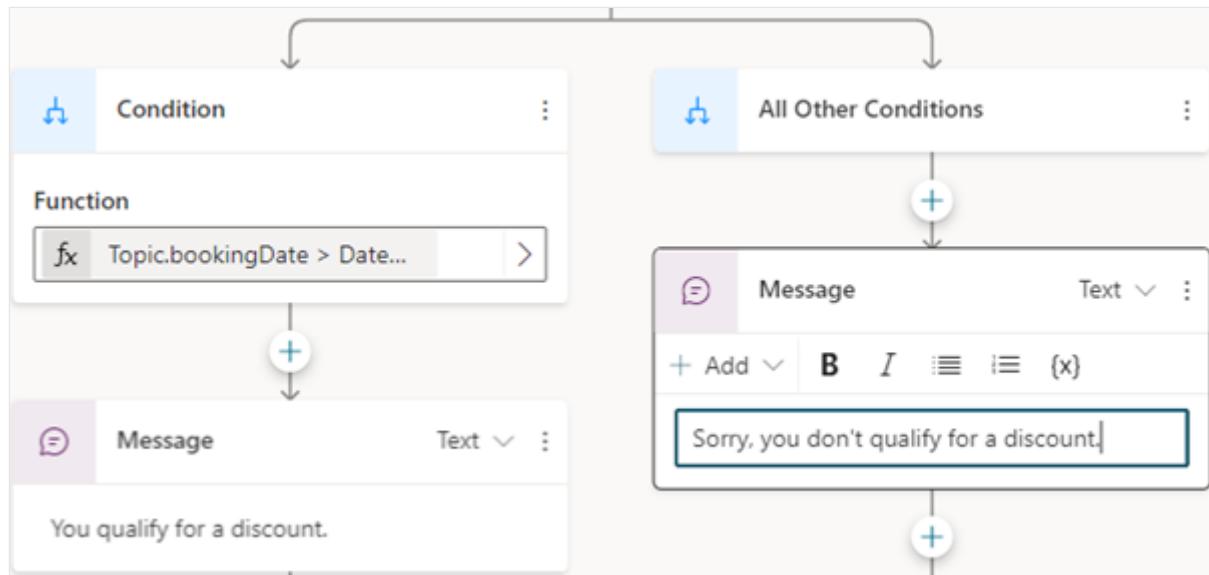
In this example, the agent determines if a booking date qualifies for a discount. To do that, it checks whether the booking date provided by the customer is 14 days or more from the current date.

1. Create a topic and add a **Question** node.
2. For **Enter a message**, enter "Booking date?".
3. Under **Identify**, select the entity **Date and time**.
4. Select the box under **Save user response as**, and then select the variable `Var1` and name it `bookingDate`.



5. Select the **Add node** icon below the **Question** node, and then select **Add a condition**.
6. Select the **More** icon (...) of the **Condition** node, and then select **Change to formula**.

7. In the **Function** box, select the > arrow, and then select the **Formula** tab.
8. Replace the contents of the **fx** box with the formula `Topic.bookingDate > (DateAdd
(Now(), 14))`, and then select **Insert**.
9. Under the **Condition** node, add a **Message** node and enter the message "You qualify for a discount."
10. Under the **All Other Conditions** node, add a **Message** node and enter the message, "Sorry, you don't qualify for a discount."



Manage topic inputs and outputs

Article • 11/19/2024

Topics can have input and output parameters. When a topic [redirects to another topic](#), you can use these parameters to pass information between topics.

If your agent uses [generative orchestration](#), it can automatically fill inputs, before running the topic, by using conversation context or generate questions to ask the user for the values. This behavior is similar to how generative slot filling for actions works.

When using generative orchestration, an agent can use topic outputs to generate a contextual response. In such cases, you don't need to configure a verbatim message from within the topic itself.

Topic inputs and outputs can share values with other topics and actions in the same generated plan.

Add inputs and outputs

To add inputs or outputs for a topic, select the **Details** in the navigation bar at the top of the authoring canvas. Alternatively you can select **Details** from the context menu for a topic on the **Topics** list page.

Once the **Details** pane is shown, you can navigate to the **Inputs** or **Outputs** tabs, and then create new input or output variables by selecting **Create a new variable**.

For every input and output parameter, you must enter a name, select a data type, and enter a description for the corresponding variable.

If your agent uses generative AI to decide how to respond to users, more properties are available for input variables. Specifically **How will the agent fill this input?** determines how the agent fills this variable with a value before running the topic. By default, each input is set to **Dynamically fill with the best option**. The agent tries to populate the value from available context, such as extracting a value from the user's message, or previous conversation. If no appropriate value is found, it generates a question to ask the user for a value. You can choose to override an input with a value instead of asking the user. To override an input parameter, select **Set as a value**, under **How will the agent fill this input?**, and enter the desired value. This value can be a literal value, an existing variable, or a Power Fx formula.

To ensure your agent captures the correct type of value for the input, you must also specify an entity type under **Identify as**.



Copilot

Comments

{x}

Topic checker

Details

More

Save

Topic details



Topic details Input Output

Input variables

Variables that the topic uses to process user input

Title

Variable name (i)

Title

How will the copilot fill this input?

Dynamically fill with best option (default)



Variable data type

String



Display name

Title

Identify as



User's entire response >

Description

The title of the ticket to be created.

> Additional settings

Create a new variable

Additional settings can be configured for inputs to control the agent's behavior when asking for a value or validating an answer from the user.

One of these properties, **Should prompt user**, is only available for topic inputs. It determines if the agent should try to fill the variable with a value before executing the topic. This property is enabled by default.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Additional settings for inputs of topics and actions

Article • 11/19/2024

When you configure [inputs for topics](#) or [inputs for actions](#), common settings allow you to determine how your agent gathers input.

Important

The additional settings are always available for action input parameters. However, for topic input parameters, these settings are only available if your agent is configured to use [generative orchestration](#).

Set how the agent should prompt users for input

By default, agents use the name and description of their input parameters to automatically generate questions to prompt the user for any missing information. However, you can override this behavior and author your own question.

1. To provide your own question, select **Customize**, under **Prompt**.
2. Enter your question. You can include references to variables or Power Fx formulas in your question.

The screenshot shows the 'Additional settings' section of a bot configuration. Under 'Agent prompts user for input', there is a 'Prompt' field with a 'Customize' button highlighted by a red box. Below it, a message template says 'How do you want to ask the user?' followed by 'The user will see this message.' A dropdown menu shows 'You create a message'. In the message editor, a question 'Which location do you want to get the weather for?' is displayed, also highlighted by a red box.

Additional settings

Should prompt user

Agent prompts user for input

Configure how the agent will ask the user to fill the input

Prompt ⓘ

Customize

How do you want to ask the user?

The user will see this message.

You create a message

Create the message to display

B I ≡ ≡ {x} fx

Which location do you want to get the weather for?

Configure the reprompt behavior for your agent

By default, when an agent doesn't get a valid answer from the user, it repeats the question two more times. You can choose to make your agent try again only once, or to move on without trying to get an answer. To customize what your agent does when it moves on, configure the properties under [No valid entity found](#).

- **How many reprompts:** The number of times your agent tries to get a valid answer. **Repeat up to 2 times** is the default. You can also select **Repeat once** or **Don't repeat**.
- **Retry prompt:** To change the question used to ask a user again for input, after the basic entity validation failed, select **Customize**, and then enter the new question. For example, if you were expecting a number, you might enter a question like "What is the ID? It should be a number."

Additional entity validation

By default, responses are validated based only on the entity you selected in the **Identify As** property. **Additional entity validation** allows you to add criteria to the basic test. For example, your input is set to **Identify As** a number, but you might want to make sure it's less than 10. You can also change the retry question to help the user enter a valid response.

- **Condition:** Enter a Power Fx formula that returns a Boolean value (`true` or `false`) —for example, `Topic.Var1 < 10`.
- **Condition not met prompt:** To change the message shown where basic entity validation is passed, but validation against a condition fails, select **Customize**, and then enter the new question. For example, if you received a number, but it was above 10, breaking the validation rule, you might enter a question like "What is the ID? It should be a number less than 10."

No valid entity found

No valid entity found determines what happens when your agent stops trying to get a valid response from the user, after the maximum retry count has been reached. You can escalate to a customer service representative or provide a default value.

- **Action if no entity found:**
 - **Escalate:** Redirect the user to the *Escalate* system topic. This is the default behavior.
 - **Set variable to value:** Set the output variable to a value and move on to the next node. Enter or select the value in **Default entity value**.
 - **Set variable to empty (no value):** Clear the output variable and move on to the next node. You can use a **Condition node** later to check whether the variable has a value.
- **No entity found message:** To specify a message to show to the user to inform them of the steps being taken, select **Customize**, and then enter the new prompt. For example, your message might say "Looks like you're struggling. I'm redirecting you to someone in our team."

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Manage topics

Article • 02/25/2025

There are many ways you can manage your topics. Copilot Studio includes a topic checker that lets you know if there are errors in your topics. You can turn topics on or off so they don't appear to people chatting with your agent. You can also duplicate topics to make it easier to create new ones.

View topic errors

Web app

When you save a topic, Copilot Studio reports any issues in the topic.

- Errors stop your agent from working. You must fix them before you can publish your agent.
- Warnings don't stop your agent from working, but might cause some topics not to work as expected. You should fix warnings when you see them.

On the **Topics** page, you can see if a topic has errors, and the number of errors, in the **Errors** column. This page only shows errors because, unlike warnings, errors prevent your agent from working.

1. Open your topic from the **Topics** page.
2. Select **Topic checker** on the toolbar.

The **Topic checker** panel appears, reporting errors and warnings for your topic.

3. Select an error to go directly to the node that contains the error.

Turn a topic on or off

Topics can have a status of **On** or **Off** to indicate whether they can be used or triggered in an agent conversation. By default, new topics are created with their status set to **On**.

To change the status of a topic, go to the **Topics** page, and use the toggle for the desired topic.

When a topic is **On**, it triggers as expected. Topics get triggered as a result of its trigger phrases or when another topic redirects to it.

When a topic is **Off**:

- The topic icon  turns gray with an indicator.
- The topic doesn't trigger at all.
- The topic doesn't trigger when trigger phrases are used in a conversation or if another topic redirects to it.
- The topic checker identifies an error if another topic redirects to the **Off** topic.
- When you publish an agent, all topics are published, whether they're **On** or **Off**.

Tip

Turn a topic **Off** to work on it in a draft state. Leave its status at **Off** while publishing changes to other topics that are ready to go live.

Copy a topic

After you create a few topics, you can use a previous topic as a baseline when creating new topics.

In the **Topics** page, select the three dots next to your topic name, then select **Make a copy**.

This duplicates the selected topic with *(Copy)* added to the name. All the topic content, such as the description, trigger phrases, and the entire conversation, is copied over to the new topic.

To prevent confusion with the original topic, which has the same trigger phrases, consider turning **Off** the new topic.

When you're done editing the new topic, turn it **On** to [test it](#). When it's ready, publish the new topic.

There's no limit to the number of times you can copy a topic. Each topic has its own internal ID.

Rename a topic

You can use the authoring canvas toolbar to quickly rename a topic. In the toolbar, select **Topics**, then select the topic name in the list. On the topic page, select the topic

name, type a new name and then press Enter.

Important

Avoid using periods (.) in your topic names. It isn't possible to export a solution that contains an agent with periods in the name of any of its topics.

Redirect to another topic

To shift the conversation from a topic to another one without starting over, add a **Redirect** node. When the conversation reaches the end of the destination topic, it resumes in the originating topic by default. You can insert more nodes after the **Redirect** node as desired.

1. Select the **Add node** icon  below the node where you want the transition to occur, point to **Topic management** and select **Go to another topic**.
2. Select the desired destination topic from the list.
3. **Pass variables between topics** if needed. If the topic you're redirecting to requires input variables or is expected to return output variables to the originating topic, enter or select a value for each one.
4. Save your topic, and then use the **Test agent** pane to confirm that your agent successfully calls the destination topic.

End the current topic or all topics

By default, a redirected conversation returns to the originating topic at the end of the called topic. To exit a topic, or to end the conversation, use the **End current topic** node or the **End all topics** node.

- An **End current topic** node ends the current topic. If the current topic is called from another topic, the conversation returns to the original topic immediately. A common use of this node is in a condition branch. One branch exits the topic early, while another branch continues the current topic.
- An **End all topics** node ends all active topics immediately. Your agent considers the next message from the user as the first message in a new conversation.

To let the user know that the conversation is over, add an **End Conversation** node *before* the **End all topics** node.

Many of the system topics use these nodes to control how conversations end or start over. Learn how to [use system topics](#).

💡 Tip

The **End all topics** node doesn't clear or reset any global variables. To clear global variables, use a **Clear all variables** node first. You can see an example of this in the system topic called [**Reset Conversation**](#) in Copilot Studio.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use generative answers in a topic

09/18/2025

The generative answers node gives you the ability to respond to your users based on [knowledge sources](#) at the topic-level. Knowledge sources allow you to give your users authoritative responses, based on information that you provide your agent.

When you first create your agent, you can [enter a URL your agent uses to generate responses](#). This URL is used in all features of the agent. However, you can boost your agent's conversations by using multiple internal and external [knowledge sources](#) within individual topics.

Generative answers as a fallback

When your agent can't find a matching intent (defined in a topic) for the user's query, it uses generative answers to try to answer the question. This behavior is known as generative answers as a fallback. If the user's intent isn't matched to topics or generative answers, the Fallback [system topic](#) is used. System topics can escalate a query for the agent.

Generative answers knowledge sources

Within an agent's topic, you can add a generative answers node. This node allows you to specify more sources that the node searches based on your inputs. Sources defined in the generative answers node override [knowledge sources](#) you specified at the agent level, which functions as a fallback. For the best results, configure your generative answers nodes with specific knowledge sources.

These sources include:

- External resources:
 - AI general knowledge
 - [Bing Web Search](#) (doesn't require external configuration)
 - [Bing Custom Search](#) ↗ (requires external configuration)
- Internal resources:
 - Azure OpenAI on your data
 - Documents uploaded to Dataverse
 - SharePoint (only [specific file formats](#) are supported)
 - Custom data (internal or external): supply your own source, such as a Power Automate Flow or from Skill.

Supported knowledge sources

Different [knowledge sources](#) have different capabilities, such as the number of inputs, the type or format of the data, and how (or if) the source needs authentication to access it.

[] Expand table

Name	Source	Description	Number of inputs supported in generative answers	Authentication
Public website	External	Searches the query input on Bing, only returns results from provided websites	Generative mode: 25 websites Classic mode: Four public URLs (for example, <i>microsoft.com</i>)	None
Documents	Internal	Searches documents uploaded to Dataverse, returns results from the document contents	Generative mode: All documents Classic mode: Limited by the Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	Generative mode: 25 URLs Classic mode: Four URLs per generative answers topic node	Agent user's Microsoft Entra ID authentication
Dataverse	Internal	Connects to the configured Dataverse environment and uses a retrieval-augmented generative technique in Dataverse to return results	Generative mode: Unlimited Classic mode: Two Dataverse knowledge sources (and up to 15 tables per knowledge source)	Agent user's Microsoft Entra ID authentication
Enterprise data using connectors	Internal	Connects to connectors where your organization data is indexed by Microsoft Search	Generative mode: Unlimited Classic mode: Two per custom agent	Agent user's Microsoft Entra ID authentication

! Note

- Agent user authentication for knowledge sources means that when a specific user asks a question of the agent, the agent only surfaces content that the specific user can access.

- Knowledge sources in generative answers nodes currently don't support Bing Custom Search, Azure OpenAI, or Custom Data. Instead, from the generative answers node properties, use the **Classic data** option for [Bing Custom Search](#), [Azure OpenAI](#), or [Custom Data](#) sources.
- For websites, you need to confirm which website(s) your organization owns that Bing will search through Copilot Studio.
- You can perform language-agnostic querying across all supported file types and languages.
- If you're using unstructured data, such as individual SharePoint files and folders, OneDrive files and folders, or connectors, there are different limits and limitations. For more information, go to [Limits and limitations](#).
- Currently, citations returned from a knowledge source can't be used as inputs to other tools or actions.

Add a generative answers node

1. Go to the [Topics](#) page and open the desired topic.
2. Select the **Add node** icon  below the node under which you want to use generative answers, point to **Advanced**, and select **Generative answers**. A new node called **Create generative answers** appears.
3. Select the **More** icon (...) of the node, and then select **Properties**. The **Create generative answers properties** pane appears.

You can specify and configure your new data sources:

- [Search public data or use a Bing Custom Search for generative answers](#)
- [Connect your data to Azure OpenAI for generative answers \(preview\)](#)
- [Use uploaded documents for generative answers](#)
- [Use content on SharePoint content for generative answers](#)
- [Use a custom data source for generative answers](#)

Learn how to customize the agent's response at [Obtain feedback for every response](#).

Authentication

Some sources require authentication, since the agent makes calls on behalf of the user in the chat window. Calls use the account credentials from copilotstudio.microsoft.com.

The authentication settings configured in the agent need manual authentication with the **Service Provider** type of Microsoft Entra ID.

To configure authentication and create a Microsoft Entra ID, see:

- [Configure user authentication in Copilot Studio](#).
- [Configure user authentication with a Microsoft Entra ID](#).

Include delegated permissions for **Microsoft Graph**:

1. `Files.Read.All`
2. `Sites.Read.All`

Supported content

Generative answers obtain content stored in these formats:

- SharePoint pages (modern pages)
- Word documents (docx)
- PowerPoint documents (pptx)
- PDF documents (pdf)

 **Note**

For a list of limits and supported SharePoint functionality, see [Copilot Studio web app SharePoint limits](#).

Content moderation

The [content moderation settings](#) for a generative answers node can be set from the **Properties** pane. The content moderation settings allow your agent to provide more answers. However, the increase in answers might affect the allowance of [harmful content](#) from the agent.

1. Select the More icon (...) of the node, and select **Properties**.
2. Select the desired moderation level.

The moderation levels range from **Lowest** to **Highest**. The lowest level generates the most answers, but they might contain harmful content. The highest level of content

moderation generates fewer answers, and applies a stricter filter to restrict harmful content. The default moderation level is **High**.

3. Select **Save** at the top of the page.

 **Note**

If your generative answers node is set to moderate content, answers might not be returned. For more information on how to diagnose answers not being returned, see [Content blocked by content moderation](#).

Search public data or use a Bing Custom Search for generative answers nodes

08/27/2025

At the topic level, you can search publicly available data by using the Bing search engine APIs within a [generative answers node](#).

ⓘ Important

Knowledge sources defined in generative answers nodes take priority over [knowledge sources](#) at the agent level. Copilot level sources function as a fallback.

When you specify the URL, the generative answers node uses the Bing Search configuration, which is based on [Grounding with Bing Search](#). Bing Search helps you boost your agent by including many sources without configuring each source individually.

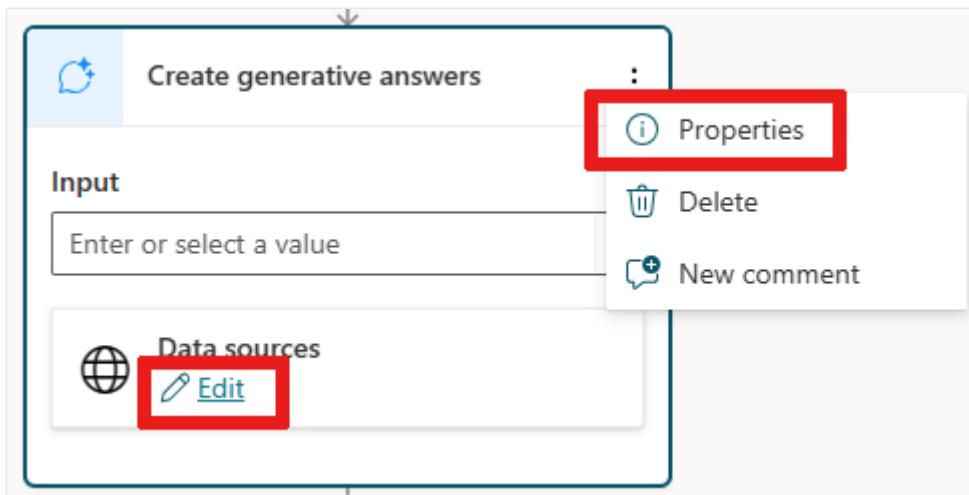
The other option is to build a tailored search for specific content using [Bing Custom Search](#). This option is based on [Grounding with Bing Custom Search](#). To use Bing Custom Search as a source, you need a custom search instance connected to the [Create generative answers node](#).

ⓘ Note

- To create a Bing Custom Search, see [Quickstart: Create your first Bing Custom Search instance](#).
- For more information on using public websites in a generative answers node, see [Generative answers based on public websites](#).

To use Bing Search or Bing Custom Search:

1. Open the **Data source** configuration pane from one of two places:
 - a. On the [Create generative answers node](#), select **Edit** under **Data sources**.



b. Alternatively, in the **Create generative answers** node select the ..., then select **Properties** and select **Data source**.

2. Select the **Class data** options.

✓ Classic data

Data sources

Choose up to 4 public websites and 4 Microsoft internal sites for your bot to use to create dynamic, generative answers.

Search public data ⓘ

Search public websites ⓘ

Public websites ⓘ

Manual input ⓘ

Enter text



To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

SharePoint ⓘ

Manual input ⓘ

Enter text



To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

Azure OpenAI Services on your data

Add a connection from Azure OpenAI as a data source.

Add connection

Connection properties

Custom data ⓘ

Enter or select a value



3. Under **Search public data**, select how you want to search:

Data source

Data sources

Choose up to 4 public websites and 4 Microsoft internal sites for your bot to use to create dynamic, generative answers.

Search public data (?)

Search public websites ▼

Search public websites

ual input ▼

Search with Bing Custom Search



- To use Bing Search, select **Search public websites**.
- To use a Bing Custom Search, select **Search with Bing Custom Search** and enter your Custom Configuration ID in the **Configuration ID** input box.

💡 Tip

To get the **Configuration ID**:

- From the [Bing Custom Search web portal](#), select the search instance you want to use.
- Select **Production** from the top menu, and under the **Endpoints** section, select the **Web API** tab.
- Copy the **Custom Configuration ID**.

4. Save changes to your topic.

Connect your data to Azure OpenAI for generative answers (preview)

06/04/2025

[This article is prerelease documentation and is subject to change.]

When you connect data through Azure OpenAI Service in Copilot Studio, your agents can use resources from Azure through the [generative answers node](#). [Azure OpenAI on your data](#) works with the powerful language models. The Azure OpenAI models and Azure AI Search index your data and provide responses while complying with your organizational policies.

Important

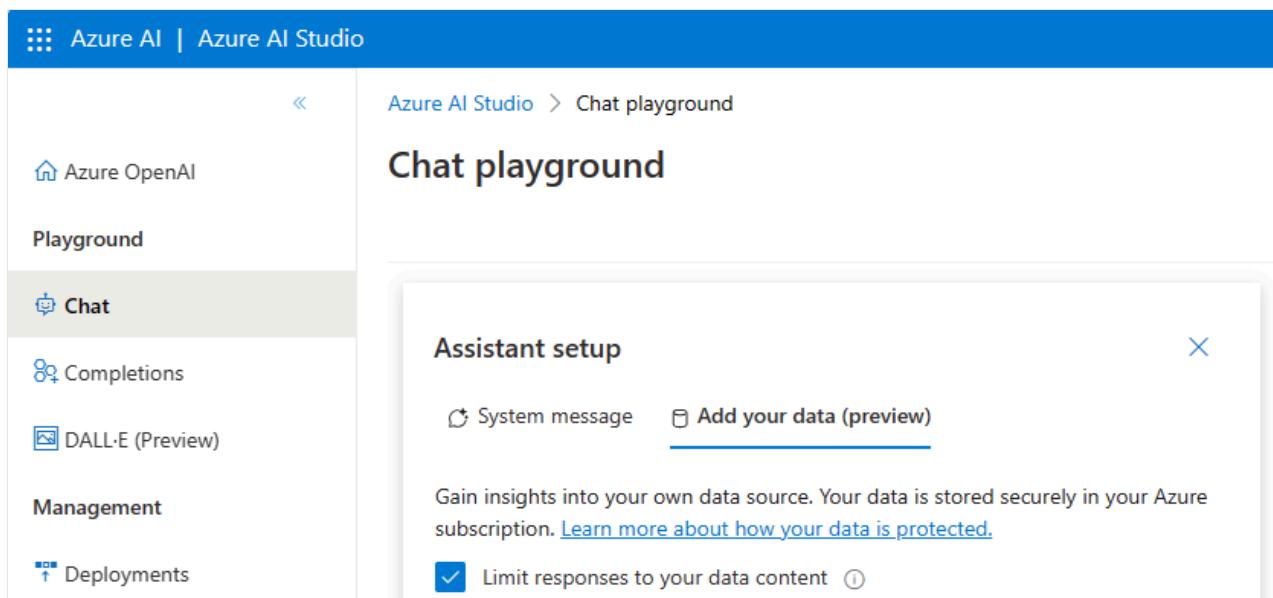
This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

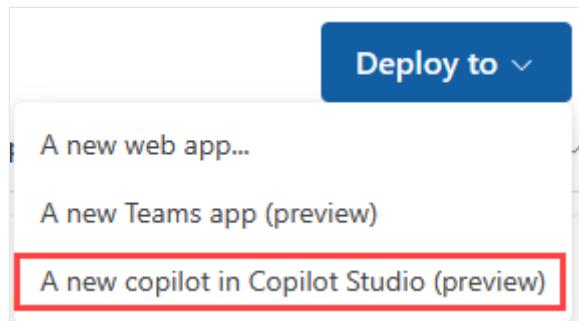
If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

Prerequisites

- Connect your data source to AI models using [Azure OpenAI Service in Azure AI Foundry](#).



- Connect a Copilot Studio agent to your data by selecting **Deploy to** and then **A new Microsoft Copilot Studio bot**. An agent is created and automatically connects to your Azure resource in your default environment.



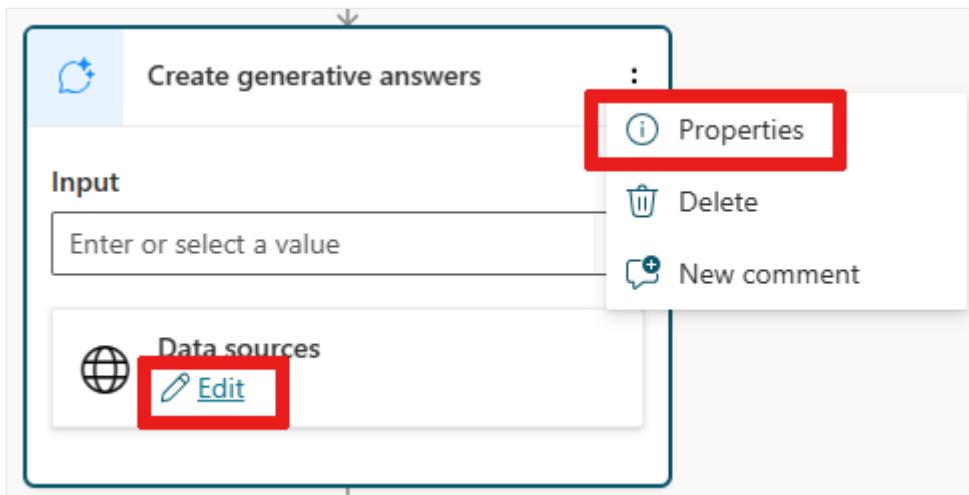
! Note

When you create an agent from Azure OpenAI Service, a Conversation Boosting system topic is automatically generated.

Type	Name ↓	Trigger ↑
Conversation Start		On Conversation Start
Conversational boosting		On Unknown Intent
End of Conversation		On Redirect

Use Azure OpenAI in Copilot Studio

1. In the relevant topic's **Create generative answers** node, open the **Data source** configuration pane from one of two places:
 - a. Select **Edit** under **Data sources**.



b. Alternatively, select ..., and then select **Properties** to open a pane, and finally select **Data source**.

2. Open the **Classic data** options.

✓ Classic data

Data sources

Choose up to 4 public websites and 4 Microsoft internal sites for your bot to use to create dynamic, generative answers.

Search public data (i)

[Search public websites](#) ▼

Public websites (i)

Manual input ▼

Enter text



To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

SharePoint (i)

Manual input ▼

Enter text



To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

Azure OpenAI Services on your data

Add a connection from Azure OpenAI as a data source.

Add connection

Connection properties

Custom data (i)

Enter or select a value



3. Edit the connection to the Azure OpenAI Service.

4. Select **Save** to save any changes to your topic.

The topic now generates answers from the information sources defined by your Azure OpenAI connection.

Important

Knowledge sources defined in generative answers nodes take priority over [knowledge sources](#) at the agent level. Agent level sources function as a fallback.

Use SharePoint content for generative answers

07/03/2025

Generative answers nodes using SharePoint as a knowledge source at the topic level work by pairing your agent with a SharePoint URL, such as *contoso.sharepoint.com/sites/policies*. When a user asks a question and the agent doesn't have a topic to use for an answer, the agent searches the URL and all subpaths. For example, a URL such as *contoso.sharepoint.com/sites* also includes subpaths like *contoso.sharepoint.com/sites/policies*. Generative answers summarize this content into a targeted response.

Important

Knowledge sources defined in generative answers nodes take priority over [knowledge sources](#) at the agent level. Agent level sources function as a fallback. For instructions on how to add SharePoint as a knowledge source at the agent level, see [Add SharePoint as a knowledge source](#). Agent responses that use SharePoint as a knowledge source at either the topic or agent level aren't included in [conversation transcripts](#).

When you publish your agent, the calls using generative answers are made on behalf of the user chatting with the agent, using the [authentication settings](#) configured in the agent. By default, agents created in Copilot Studio and in Microsoft Teams are configured with the **Authenticate with Microsoft** authentication option, which works in environments such as Microsoft Teams, Power Apps, and Microsoft 365 Copilot.

Note

It's possible to use generative answers with SharePoint data in Microsoft Teams chats, and not require manual authentication. To use this method for a previously published agent, reconfigure the agent to use **Authenticate with Microsoft** and then publish it again to Microsoft Teams. It might take a few hours before this change takes effect. If a user is in the middle of a conversation and the change doesn't appear to have taken effect, they can type "start over" in the chat to force the conversation to restart with the latest version of the agent. These changes are now available for Teams 1:1 chats between the user and the agent. They're not yet available for group chats or channel messages.

If you want to use manual authentication for your agent, see [Advanced authentication scenarios](#).

(!) Note

- For a list of limits and supported SharePoint functionality, see [Copilot Studio web app SharePoint limits](#).
- Copilot Studio agents require Dataverse search to use this knowledge source. If you can't add a Dataverse-enabled file to an agent, ask your administrator to turn on Dataverse search in your environment. For more information about Dataverse search and how to manage it, see [What is Dataverse search](#) and [Configure Dataverse search for your environment](#).

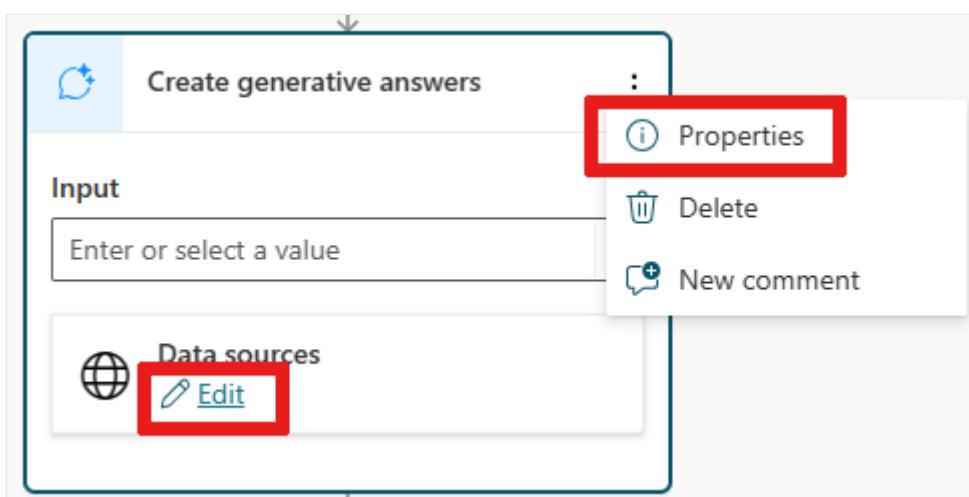
Use SharePoint in a generative answers node

(i) Important

When using a [generative answers node](#) in a topic, you might encounter an error on the authoring canvas and in the Topic Checker when Integrated Security has been chosen. This error is benign, and doesn't prevent the feature from working. This error will be resolved in an upcoming deployment.

1. Open the **Data source** configuration pane from one of two places:

a. On the **Create generative answers** node, select **Edit** under **Data sources**.



b. Alternatively, in the **Create generative answers** node select the **...**, then select **Properties** and select **Data source**.

2. Select the **Class data** options.

✓ Classic data

Data sources

Choose up to 4 public websites and 4 Microsoft internal sites for your bot to use to create dynamic, generative answers.

Search public data (i)

Search public websites ▼

Public websites (i)

Manual input ▼

Enter text



To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

SharePoint (i)

Manual input ▼

Enter text



To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

Azure OpenAI Services on your data

Add a connection from Azure OpenAI as a data source.

Add connection

Connection properties

Custom data (i)

Enter or select a value



- Under **SharePoint** in the **Data source** pane, add the SharePoint URLs you want to use. Separate multiple URLs with manual line breaks (use **Shift + Enter**).

The screenshot shows the Copilot Studio interface with two main sections for input. The top section is labeled 'Public websites' with a help icon and a 'Manual input' dropdown. Below it is a text input field with a placeholder 'Enter text' and a '+' button. A note below says: 'To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation'. The bottom section is labeled 'SharePoint' with a help icon and a 'Manual input' dropdown. It also has a text input field with a placeholder 'Enter text' and a '+' button. A note below says: 'Shift+Enter to create line separation'. The 'SharePoint' section is highlighted with a red rectangular border.

Note

For a list of limits and supported SharePoint functionality, see [Copilot Studio web app SharePoint limits](#).

4. Select **Save** at the top to save your topic changes.

5. Test your agent with phrases that you expect to return content.

Note

If the user account you used to sign in to *copilotstudio.microsoft.com* doesn't have access to the SharePoint site, you don't get content or you might see a system error.

Advanced authentication scenarios

By default, Copilot Studio preconfigures agents to authenticate users using Microsoft authentication to access SharePoint sources when they're connecting to an agent through Microsoft Teams, Power Apps, or Microsoft 365 Copilot. However, if you need to manually configure authentication, the following steps are required for generative answers to work with SharePoint data sources.

Important

If [Restricted SharePoint Search](#) is enabled, use of SharePoint is blocked.

- To authenticate your agent, see [Authentication](#).

- For instructions on how to create the needed Microsoft Entra ID application registration, see [Configure user authentication with Microsoft Entra ID](#). When creating this registration you or your admin must specify the `Sites.Read.All` and `Files.Read.All` scopes in the Microsoft Entra ID app registration.
- Selecting **No authentication** specifies that your agent doesn't retrieve information from SharePoint.
- When you configure authentication, scopes control user access. Specify `Sites.Read.All` and `Files.Read.All` in the scopes field near the existing `profile` and `openid` values in Copilot Studio. These scopes don't give users increased permissions when using the agent, but allow their permissible content from the SharePoint site as configured for generative answers.
- This configuration only applies to generative answers, and doesn't apply to Power Platform Connectors.

Use a custom data source for generative answers nodes

Article • 04/01/2025

There might be scenarios where your data might not reside in one of the supported knowledge sources. In these scenarios, you can provide your own data through [Power Automate flows](#) or through [HTTP requests](#). These options typically return a JSON object literal, which you can then parse into a [Table](#) format to generate answers.

ⓘ Important

Knowledge sources defined in generative answers nodes take priority over [knowledge sources](#) at the agent level. Agent level sources function as a fallback.

The custom data field takes a [Table](#) as an input, with the below properties:

[] Expand table

Name	Required	Description
Content	Required	Source content used to summarize and answer the input question
ContentLocation	Optional	URL for the citation of the source content
Title	Optional	Title for the citation of the source content

Here's a sample table in Power Fx format, using a parsed JSON object literal:

```
Power Apps

[
{
  Content: "This is a sample piece of text that was provided for testing purposes, to be replaced with content of your choice",
  ContentLocation: "https://contoso.com/p1.htm",
  Title: "Contoso Sample"
},
{
  Content: "This is a second bit of sample text that can be replaced with content of your choice",
  ContentLocation: "https://fabrikam.com/p2.htm"
},
{
  Content: "This is a third bit of sample text that can be replaced with content of your choice",
}
```

```
Title: "Adventure Works Cycles Sample"  
}  
]
```

Agent answers are generated from `Content` and include the link to the data source in `ContentLocation`. If a `Title`, is it used for the citation.

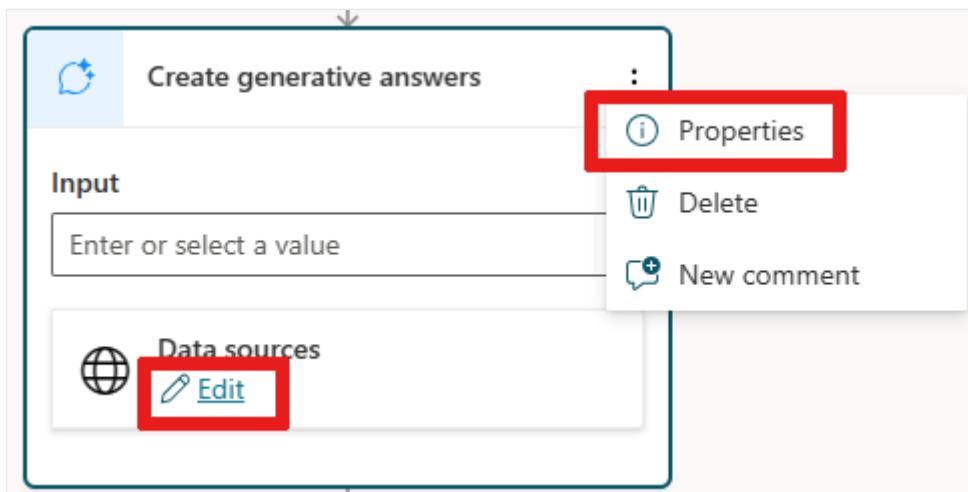
ⓘ Important

Only the first three records of the table are used to generate an answer.

Use custom data

1. Open the **Data source** configuration pane from one of two places:

a. On the **Create generative answers** node, select **Edit** under **Data sources**.



b. Alternatively, in the **Create generative answers** node select the **...**, then select **Properties** and select **Data source**.

2. Select the **Class data** options.

The screenshot shows the 'Data sources' section of the Microsoft Bot Framework configuration interface. It includes fields for 'Search public data', 'Public websites' (with a 'Manual input' dropdown), and 'SharePoint' (also with a 'Manual input' dropdown). There are also sections for 'Azure OpenAI Services on your data' and 'Custom data'.

Classic data

Data sources

Choose up to 4 public websites and 4 Microsoft internal sites for your bot to use to create dynamic, generative answers.

Search public data ⓘ

Search public websites ⓘ

Public websites ⓘ Manual input ⓘ

Enter text +

To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

SharePoint ⓘ Manual input ⓘ

Enter text +

To add items in bulk, paste in line-separated items or use Shift+Enter to create line separation

Azure OpenAI Services on your data

Add a connection from Azure OpenAI as a data source.

+ Add connection

Connection properties

Custom data ⓘ

Enter or select a value >

3. In the **Custom data** field, enter the JSON array you want to use, or the variable for the array.
4. Select **Save** to save any changes to your topic.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Use uploaded files with generative answers nodes

You can upload your own files at the topic level to use with [generative answers nodes](#). The files are used across your agent, but you can [specify individual nodes that shouldn't use uploaded files](#).

Important

Knowledge sources defined in generative answers nodes take priority over [knowledge sources](#) at the agent level. Agent level sources function as a fallback. For more information on how to upload files as knowledge sources at the agent level, see [Upload files as a knowledge source](#).

When an agent user asks a question, and the agent doesn't have a defined topic to use, the agent generates an answer from your uploaded files. The agent uses generative AI to answer the user's question and provides an answer in a conversational style.

Uploaded files are stored securely in Dataverse. The number of files you can upload is limited by the available file storage for your Dataverse environment.

Note

Generative answers can process uploaded files that are up to 512 MB in size.

Image, audio, video, and executable files aren't supported. See [Supported file types](#) at the end of this article for a full list.

Prerequisites

- One or more [supported file types](#), each of which can be up to 512 MB.

Uploading a file

Warning

Contents of the files you upload are available to all users.

Uploaded file content is available to anyone chatting with the agent, regardless of file permissions or access controls.

To upload a file as a knowledge source in a generative answers node, perform the following steps:

1. Open the agent.
2. Select **Topics**.
3. Select the topic with the generative answers node you want to add files to or **Add a topic**.
4. In the **Create generative answers** node for the topic, select the **More icon (...)** of the node, and then select **Properties**. The **Create generative answers properties** pane appears.

 **Note**

For information on how to add a generative answers node itself, see [Add a generative answers node](#).

5. Under **Knowledge sources**, select **Add knowledge**. Ensure that **Search only selected sources** is turned on.
6. Upload the file by either drag and drop, or browsing to the file location.
7. Select **Add to agent** to finish adding the knowledge source.

 **Note**

To upload files, ensure that **File uploads** is turned on in the Settings page for your agent. For more information, see [Turn on file uploads](#).

After uploading your files

- Once uploaded, your content in the file might take a few minutes to be ready for use by the agent. When the file is ready, the status of the file changes from **In progress** to **Ready**.
- Uploaded files create new copies and don't overwrite existing files, even if they have the same name.

- The uploaded file becomes part of the agent solution, therefore exporting and importing an agent solution includes the files.

Deleting a file

To delete a file as a knowledge source in a generative answers node, perform the following steps:

1. Open the **Knowledge** page for an agent.
2. Locate the file you want to delete and select **More options (...)** then select **Delete**.
3. Confirm you want to delete the file by selecting **Delete**.

Warning

Deleting a knowledge source is permanent and cannot be undone.

Download a file

To download a file as a knowledge source in a generative answers node, perform the following steps:

1. Open the **Knowledge** page for an agent.
2. Locate the file you want to download and select **More options (...)** then select **Download**.

Disable files as a data source for a generative answers node

Files used as a data source for the agent are used as a data source by default for all generative answers nodes.

To exclude a specific generative answers node from using files as a data source:

1. Open the topic with the generative answers node you want to exclude.
2. On the top menu bar, select **More options > Open code editor**.
3. Insert the following lines of code immediately after the line `variable: Topic.Answer:`

`YAML`

```
fileSearchDataSource:  
  searchFilesMode:  
    kind: DoNotSearchFiles
```

4. Select **Save** at the top of the page.

Supported file types

- Word (doc, docx)
- Excel (xls, xlsx)
- PowerPoint (ppt, pptx)
- PDF (pdf)
- Text (.txt, .md, .log)
- HTML (html, htm)
- CSV (csv)
- XML (xml)
- OpenDocument (odt, ods, odp)
- EPUB (epub)
- Rich Text Format (rtf)
- Apple iWork (pages, key, numbers)
- JSON (json)
- YAML (yml, yaml)
- LaTeX (tex)

Important

Image, audio, video, and executable files aren't supported. In addition, files with encryption, such as sensitivity labels or password protection, aren't supported.

Last updated on 10/28/2025

Make HTTP requests

10/03/2025

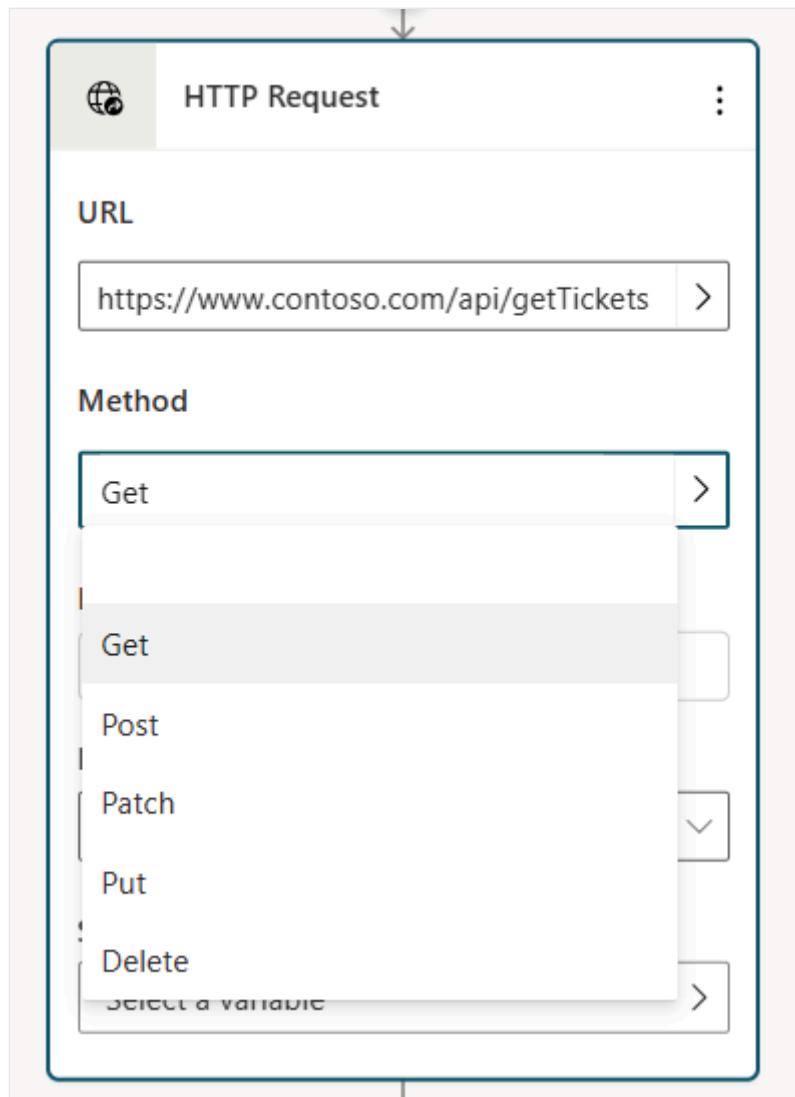
In Copilot Studio, you can extend an agent's capabilities by calling external REST APIs using the **HTTP Request** node. This call can be helpful when you need to retrieve data from an external system or manipulate data on that system.

Prerequisites

- [Create and edit topics](#)

Use the HTTP Request node

1. Select the Add node  icon under the node after which you want to call an external API, point to **Advanced**, and select **Send HTTP request**.
2. In the **URL** box, enter the URL for the API endpoint you want to call.
3. Select the appropriate request **Method** from the dropdown. GET, POST, PATCH, PUT, and DELETE methods are supported.



4. Under **Headers and body** on the node, select the **Edit** button, which opens the **HTTP Request properties** panel.
5. You can optionally add one or more headers to the HTTP request. Select the **Add** button to add each header's key and value.

The screenshot shows the 'Headers' section of the properties panel. It has a placeholder text 'Give additional information with values.' Below it is a 'Key' input field containing 'Content-Type' and a 'Value' input field containing 'application/json'. At the bottom left is a '+ Add' button.

You can use the headers section to pass authentication tokens, content type, or any other required headers for the API you're calling. For example, you might add a header like

`Authorization: Bearer <your_token>` to authenticate the request.

6. By default, no content is sent in the body of the HTTP request, but you can [specify content in the body](#).

7. On the node, choose the **Response data type** from the available options.

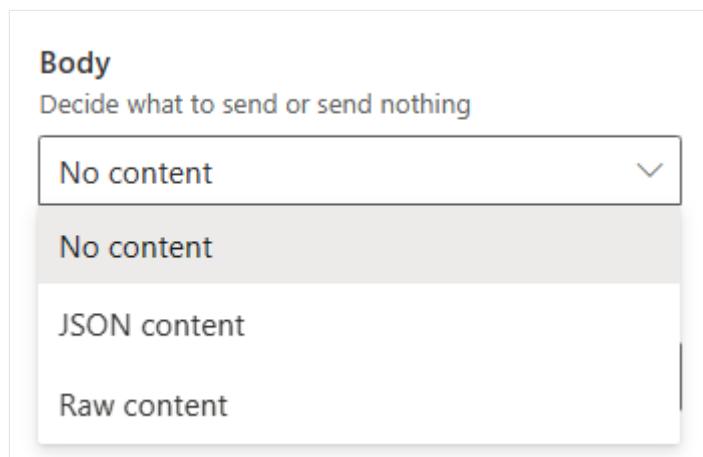
You can provide an example JSON response, which you can usually find in the documentation for the API you're calling. It generates a Power Fx variable, allowing you to use it elsewhere in your authoring canvas, with intellisense support in the Power Fx editor. Select **From Sample Data** and then select **Get schema from sample JSON**. Paste your sample response into the editor and select **Confirm**.

8. Under **Save user response as** choose where you want the HTTP request response to be stored, either creating a new variable or selecting an existing one.

Send content in the HTTP request body

1. Under **Headers and body** on the node, select the **Edit** button, which opens the **HTTP Request properties** panel.

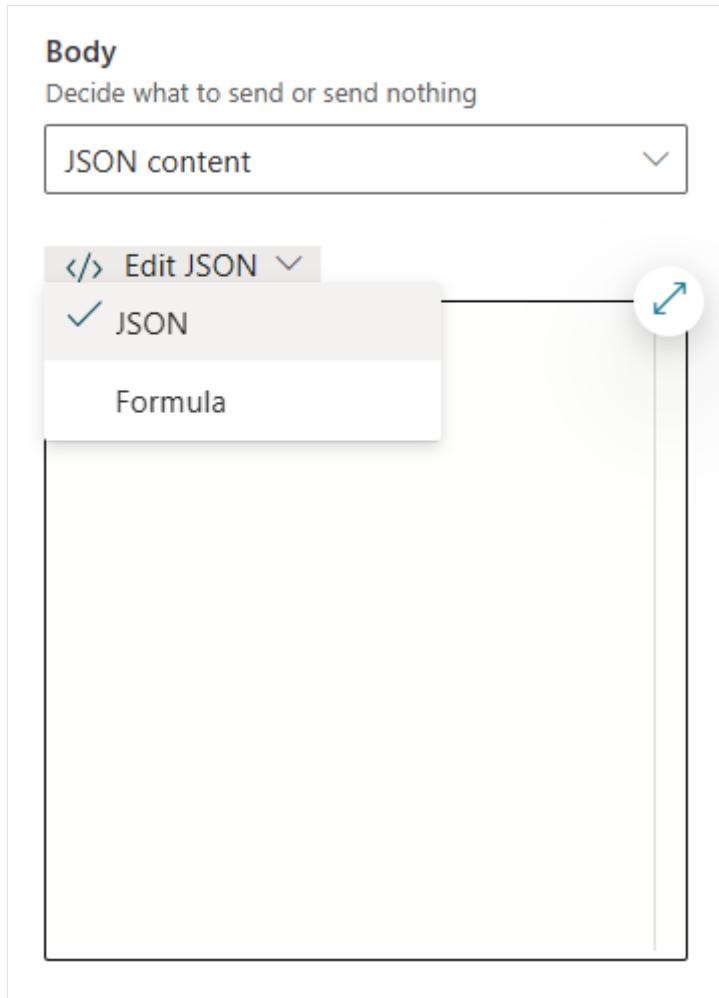
2. Under **Body**, select the appropriate content type. By default, **Body** is set to **No Content**, where no content should be present within the request body, which is most commonly used with the GET request. To add content to the body, select from one of the available options.



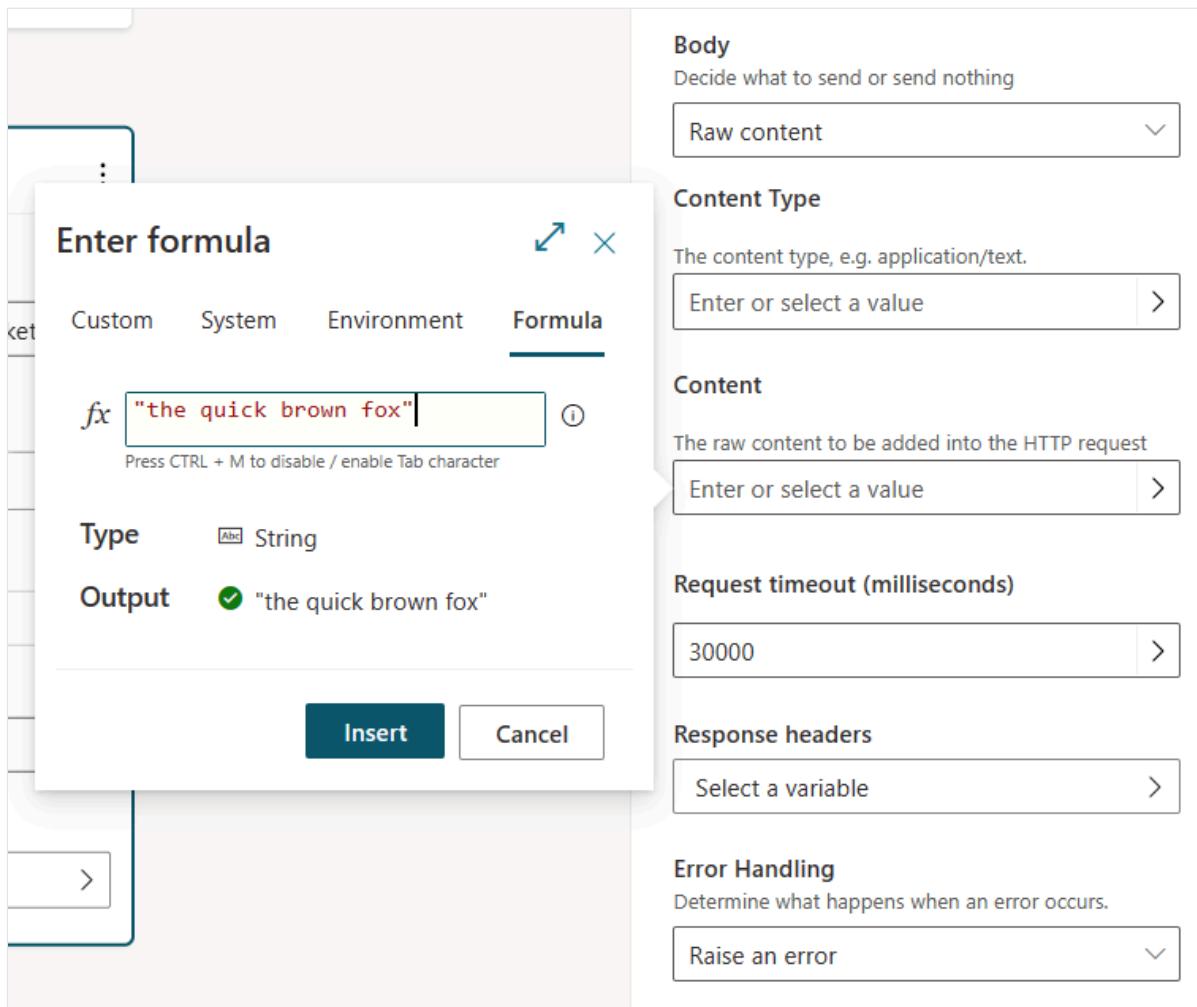
- **JSON Content:** JSON content allows you to provide a JSON object in the request body. This object is often used when making POST or PUT requests. When this option is selected, an editor allows you to enter your JSON content.

You can also specify your JSON content using Power Fx, allowing you to include dynamic values and reference variables. Select the **Edit JSON** button, choose **Formula**, and then enter a Power Fx object (any existing JSON that was entered is

converted into a Power Fx object for you), which is converted into JSON when the HTTP request is made.



- **Raw content:** Raw content lets you insert a string of data into the request body, and this string can be of any content type you specify. Raw content is entered using a Power Fx formula, such as the string content shown in the following example.



Error handling and time out

The **HTTP Request** node has multiple options for handling situations where the HTTP request might fail or return an error.

By default, the feature raises an error. This means that when an HTTP request returns an error or fails to execute, the agent stops its operation and triggers the **On Error** system topic, displaying an error message.

However, you can configure the error handling to not trigger the **On Error** system topic. Instead, you can make it store the HTTP status code and any error response body into variables that you specify. You can interrogate these variables as needed, and the topic can continue to run. This configuration is useful in scenarios where you want the agent to continue operation, even if a particular HTTP request fails.

To configure the error handling behavior of an **HTTP Request** node:

1. Under **Headers and body**, select **Edit**. The **HTTP Request properties** panel appears.
2. Under **Error handling**, select the desired behavior: **Raise an error**, or **Continue on error**.

3. If you select **Continue on error**, configure variables for the status code and error response body.

The screenshot shows the 'Error Handling' configuration panel. It includes sections for 'Status code' and 'Error response body', each with a variable reference and a 'More options' button.

Error Handling
Determine what happens when an error occurs.

Status code
{x} StatusCode number >

Error response body
{x} ErrorResponse any >

In this example, if the HTTP request fails, the agent stores the HTTP status code in the `Topic.StatusCode` variable, and the response body in the `Topic.ErrorResponse` variable. The agent continues to its next steps in the topic.

The **Error response** variable type is **Any**. You can use a [Parse value](#) node to convert this variable into a Power Fx record.

In the **HTTP Request properties** panel, you can also set a **Request timeout** property, in milliseconds. The default value is 30 seconds.

Send an event or activity

06/18/2025

Copilot Studio provides a set of nodes that you can use for activities that aren't messages: the **Event activity** node, the **Invoke activity** node, and other activity-related nodes that the *activity protocol* supports.

What is the activity protocol?

The *activity protocol* is a core transport concept in Copilot Studio. The activity protocol allows seamless communication between users and agents. It defines a standardized structure for all interactions, categorizing them into various activity types.

Event activities and *message* activities are two distinct types within the activity protocol, each serving unique purposes in communication workflows.

Message activities represent the foundational communication units between users and agents, primarily carrying textual, media, or adaptive card content. They're designed for direct interactions, where a user makes a query or statement, and the agent responds accordingly.

Event activities, on the other hand, are typically used to signify nonverbal actions or system-generated updates. They allow for asynchronous communication, triggering specific functionalities or workflows without requiring direct user interaction. For instance, an event activity can notify an agent about a user joining a conversation or changes in surrounding context.

Sending events

The **Event activity** node is designed for sending event activities. Event activities are sent from the agent and can be intercepted and used by the channel, which decides if and how to use the activities. When you send an event, you give it a name and then you can set a value for the event. This value can be in any format:

- A primitive literal value
- A variable reference
- A Power Fx formula.

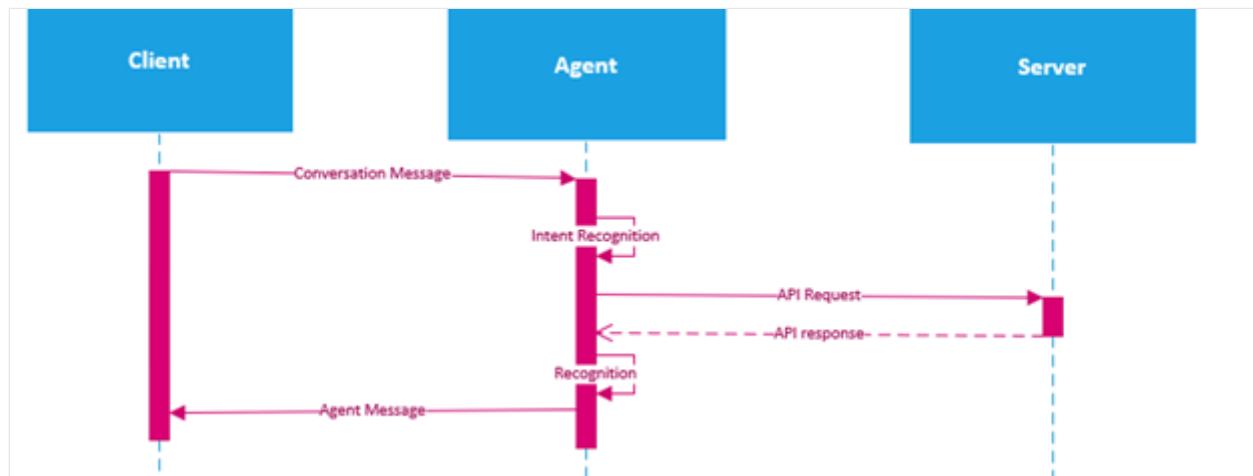
The value gets serialized as a JSON literal and added to the outgoing activity.

Uses of this node include:

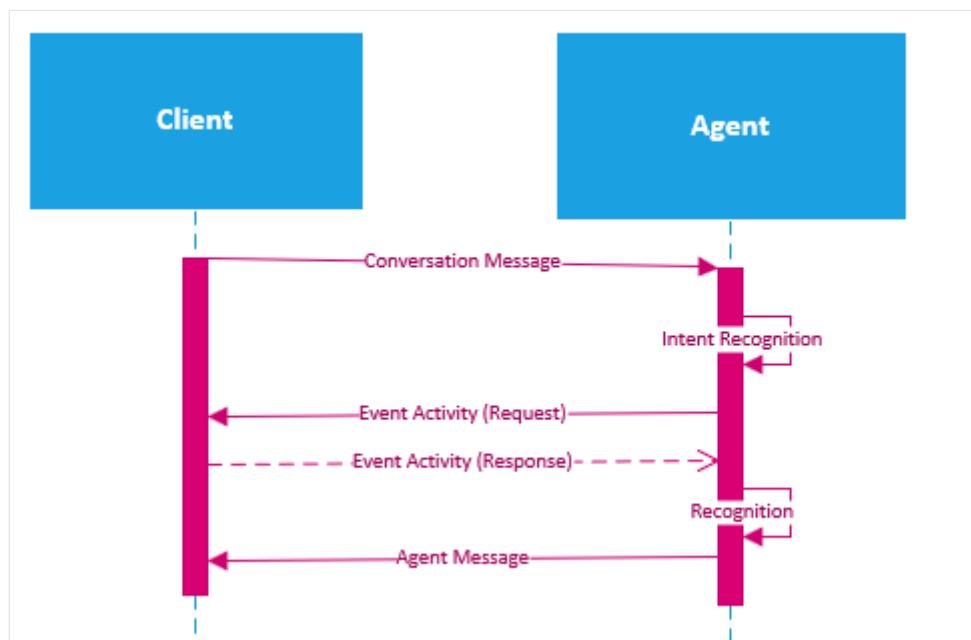
- Configure a custom Web Chat control to handle events sent from the agent. For example, you could look for an event coming back from the agent and take an action on the page. The [04.api/c.incoming-activity-event](#) sample in the [Microsoft Bot Framework Web Chat samples repository on GitHub](#) demonstrates how event handling could work.
- Use event activities to control AudioCodes recording services—for example, to start or stop call recording. For more information, see [Recording calls](#).

Using client tools

During orchestration, we provide the language model with a set of tools. Most people think about tools as calling into external systems.



However, we can also use event activities to execute tools from the client.



When the orchestrator decides to execute a client tool, the agent sends out an event activity to the client, including the inputs defined in the tool.

The agent sends out the activity, then waits for the client to perform the action and return a result. When the client is done, it sends an event activity containing the response to the agent. The agent takes this response as the tool response and continues orchestration.

Sample payloads for client tools

Suppose we have a client tool that retrieves the text on a PowerPoint slide.

The inputs are the page number, and the output is the text on the slide.

The agent could send a payload like:

```
JSON

{
  "type": "event",
  "timestamp": 1738709828,
  "from": {
    "id": "d9c0dcf9-4045-8062-535b-73fb4dfee954",
    "role": 0
  },
  "name": "getSlideContent",
  "replyToId": "f617c120-7b36-496a-a096-ac692efdad04",
  "value": {
    "page": 5
  }
}
```

After the operation finishes, the client sends the agent something a payload like:

```
JSON

{
  "type": "event",
  "timestamp": 1738709828,
  "name": "getSlideContent",
  "replyToId": "f617c120-7b36-496a-a096-ac692efdad04",
  "value": {
    "content": "QA slide. Next steps"
  }
}
```

Setting up client tools

There are two ways to register client tools.

As part of your agent topic content, you can register a dedicated client task action on your agent, using the code editor.

For example:

YAML

```
description: this tool retrieves the content of a powerpoint slide
schemaName: GetSlideContent
dialog:
  kind: TaskDialog
  action:
    kind: InvokeClientTaskAction
    clientActionInputSchema:
      kind: Record
      properties:
        page:
          displayName: Page Number
          description: The number of the slide
         .isRequired: true
          type: Number
    clientActionResponseSchema:
      kind: Record
      properties:
        content:
          displayName: Slide Content
          description: The content of the slide
          type: String
```

Sometimes, the available tools are dynamic, based on the context of the hosting client. To allow more flexibility, makers can set a [system variable](#) to dynamically set other client tools that can be used for this session.

YAML

```
- kind: SetVariable
id: setVariable_76NZWK
variable: System.ClientPluginActions
value: |-
=[
{
  Description: "this tool retrieves the content of a powerpoint slide",
  Identifier: "GetSlideContent",
  Name: "GetSlideContent",
  Response: {mode: "Generated"},
  Inputs: [
    {
      Description: "The name of the menu for the form to launch",
      IsAutomatic: true,
     IsRequired: true,
      Name: "Page Number",
```

```
        PropertyName: "page",
        Type: {
            '$kind': "Number"
        }
    ],
    Outputs: [
        {
            Description: "The content of the slide",
            Name: "Slide Content",
            PropertyName: "content",
            Type: {
                '$kind': "String"
            }
        }
    ]
}
```

Sending other activity types

In addition to event activities, you can send activities of other types using the [Invoke activity](#) node. The types of activities you can send are a subset of the ones offered in the [Bot Framework Schema - ActivityTypes Class](#). When using this node, you choose the type of the activity and then optionally set a name or value.

Common types are:

- **Typing** sends a typing activity, which the channel can choose to pick up and show a typing indicator on the client.
- **Invoke** and **Invoke response** are used for Microsoft Teams. You create a topic with an invoke trigger to intercept an incoming Invoke from Teams, and use an **Invoke response activity** node to send an appropriate response back to Teams.
- **Handoff** sends a handoff activity with explicit control over the value. A handoff is used for external channels, such as AudioCodes.

Related content

- [Work with global variables](#)
- [Customize the look and feel of an agent](#)

Add other agents (preview)

10/27/2025

[This article is prerelease documentation and is subject to change.]

Copilot Studio lets you enhance your agents by connecting them to other agents, allowing them to hand off user interactions or respond to autonomous triggers. Scale out your solutions efficiently and effectively by using modular agents tailored to particular tasks or data sets.

All agents you add appear on the [Agents](#) page.

Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

Creating child agents or connecting to existing agents

The following considerations are important when deciding if you should create child agents, within your main agent, or connect your agent to a separate agent.

A *child* agent is a lightweight agent that lives within your main agent. It's similar to tools and topics within an agent.

A *connected* agent is a full-fledged agent, independent from your main agent. Copilot Studio currently supports connecting other Copilot Studio agents that exist in the same environment, and Microsoft Fabric data agents.

Create child agents within your agent when:

- A single developer or a small, cohesive team manages the entire agent solution.
- You want to logically group tools, instructions, and knowledge into clearly defined sub-agents within a larger agent.

- You don't need separate configuration settings, authentication, or deployment capabilities for the sub-agents.
- You don't intend to publish these agents separately or make them available independently.
- You don't need to reuse the agent across multiple agents.

Connect existing agents when:

- Multiple teams or developers manage different agents independently.
- You need to publish and maintain agents separately, and make them available directly on independent channels.
- Agents must have their own dedicated settings, including authentication.
- You need independent application lifecycle management (ALM) processes for each agent.
- You want to make your agent reusable.

You can mix connected agents and child agents in your solution. For example, you could break out parts of your solution into separate agents that users could also access directly. Each of those agents could have their own child agents for key purposes.

Create a child agent

1. Go to the **Agents** page for your main agent and select **Add an agent**.
2. Select **Create an agent**.
3. Enter a name for your new agent.
4. Determine when your agent should be used. By default, agents respond to users or triggers, based on their description.
 - If you want to use the default behavior (**The agent chooses**), enter a brief description of your agent's purpose.
 - Otherwise, expand the list under **When will this be used?** and select the desired behavior. For more information on the supported behaviors, see [Determine when your agent should be used](#).
5. Enter clear instructions that you want your agent to follow when invoked. To reference tools, variables, or add Power Fx formulas in your instructions, enter a slash (/) and select the desired option from the menu that appears.

 **Important**

When referencing existing tools in your instructions, decide if the tool should be accessible directly by your main agent or only when explicitly called within instructions. Limiting a tool's usage to explicit instruction references helps prevent confusion when similar tools or agents exist. For example, if you have a "Check account balance" agent and a similar "Get account balance" tool, restrict the tool to be called only by the agent to avoid overlap. To restrict a tool to only be available when referenced from another agent, clear the **Allow agent to decide dynamically when to use this tool** property in the **Additional details** section on the tool's detail page.

6. Optionally, add knowledge and tools that only this child agent might use:

- In the **Knowledge** section, select **Add** and proceed in the same fashion as when you [add knowledge](#) to your main agent.
- In the **Tools** section, select **Add** and proceed in the same fashion as when you [add tools](#) to your main agent.

7. If you don't want this agent to be active yet, turn off **Enabled**. You can [make your child agent active](#) later.

8. Select **Save**.

Determine when your agent should be used

By default, agents respond to a user query, based on their description. You can also configure agents to intercept other events and respond to them.

 Expand table

Event	Description
A message is received	Called when an activity of type <code>message</code> —the most common type of activity—is received. Received when a user types or says something to the agent. By default, the agent responds to any messages. To limit the agent response to a specific type of message, use the Activity type list under Additional details .
A custom client event occurs	Called when an activity of type <code>event</code> is received. By default, the agent responds to any events. To limit the agent response to a specific event, use the Event name property under Additional details .
An activity occurs	Called when an activity of any type is received. To limit the agent response to a specific type of activity, use the Activity type list under Additional details .
The conversation changes	Called when an activity of type <code>conversationUpdate</code> is received. For example, Teams sends an activity of this type when a user joins a conversation.

Event	Description
It's invoked	Called when an activity of type <code>invoke</code> is received. Most commonly received from the Teams channel—for example, when the user interacts with a message or search extension in Teams.
It's redirected to	Called when the agent is called explicitly from within a topic. <i>Not yet supported.</i>
The user is inactive for a while	Called when a user hasn't interacted with the main agent after a configured period of time. Select the desired inactivity threshold from the Inactivity duration list under Additional details .
A plan completes	Called when the main agent finishes executing all planned steps to respond to a user query or autonomous trigger.
An AI-generated response is about to be sent	Called when the main agent generates a response for a user after calling one or more topics, actions, or knowledge sources. Use the <code>Response.FormattedText</code> system variable to see the generated response. Set the <code>ContinueResponse</code> variable to false if you want to prevent the orchestration response being sent (that is, if you modify the message and send your own using a message node).

Additional details

Depending on your selection for **When will this be used?**, more properties might be available. The following properties are always available.

Condition

Specify conditions that must be met in order for the agent to be called. For example, you might want the agent to be called only if the channel used by an employee is Microsoft Teams.

Additional details

Condition

Activity.Channel choice

is equal to

Enter or select a value

- Directline
- Facebook
- MsTeams
- Omnichannel
- Telephony

If you need more complex conditions, you can switch to the [Power Fx](#) formula editor: select **Builder** then select **Formula**.

The screenshot shows the Power Fx formula editor interface. A dropdown menu is open, with the 'Builder' option selected (indicated by a red box). Another red box highlights the 'Formula' option in the list. The formula being edited is '{x} Activity.Channel choice', with the condition 'is equal to' and a placeholder 'Enter or select a value'. A 'New condition' button is also visible.

Priority

More than one agent can fire for a single incoming activity, such as a message. By default, the option you select for **When will this be used?** determines the order the agents fire.

Child agents and topics share the same set of triggers or events that they can respond to. If an agent and a topic are configured to respond to the same event, such as user inactivity, you can use the priority property to determine which should go first.

Order of execution:

1. **Activity Received**
2. **Message / Event / Conversation update / Invoke received**
3. **By agent**

If multiple agents or topics must be called based on the same type of event happening, they're called in the order of creation (oldest first).

You can set the **Priority** property explicitly. A lower number indicates a higher priority.

Connect an existing agent

To connect an existing agent to your agent:

1. On the **Agents** page, select **Add an agent**.
2. Select the type of agent that you would like to connect to:
 - [Copilot Studio](#), for a Copilot Studio agent.
 - [Microsoft Fabric](#), for a Microsoft Fabric data agent.
3. Select **Add agent**.

Once added, your agent appears in the **Agents** list and you can immediately test it by asking something in the test chat that should cause the agent to be called.

! Note

If you make changes to a connected agent, be sure to publish those changes—your main agent can only use the latest version after it's published. Also, keep in mind: once an agent is connected, you control its *description* locally. Any updates to the original agent's description doesn't automatically sync with your main agent. You must update the description manually if you want to reflect those changes or make any other changes.

Connect a Copilot Studio agent

To connect your main agent to another agent, the other agent must be:

- In the same environment as your main agent
- Published
- Configured to [allow connections from other agents](#)

To connect an existing agent to your agent:

1. Select the desired agent from the list of available agents. The agent's name, instructions, and description appear.
2. Adjust the description (if needed) to make it more contextual for the main agent (that is, making the description more specific if you have other tools or agents where the descriptions might overlap). We recommend you update the description to ensure Copilot Studio can understand when to invoke this agent. Learn about [writing effective metadata](#).
3. If you want to prevent conversation history from being passed to this agent when it's called, clear the **Pass conversation history to this agent** box. This method limits the information being passed to the agent to just the explicit task that the main agent wants the other agent to complete.

Connect a Microsoft Fabric Data agent

1. Select the desired connection from the list of available connections or create a new connection between Microsoft Fabric and your Copilot Studio agent.
2. Select **Next**.

3. Select the desired agent from the list of Fabric data agents you have access to.
4. Adjust the description (if needed) to make it more contextual for the main agent (that is, making the description more specific if you have other tools or agents where the descriptions might overlap). We recommend you update the description to ensure Copilot Studio can understand when to invoke this agent. Learn about [writing effective metadata](#).

Make a Copilot Studio agent available to other agents

By default, any agent can connect another agent in the same environment.

1. Go to the **Settings** page for your agent.
2. Turn on **Let other agents connect to and use this one** if it isn't.

If you want to prevent other agents from connecting to this agent, turn off this option.

Reference an agent in your main agent's instructions

It can be useful, especially for autonomous agents, to be able to reference child agents you created or other agents you connected. With this method, you can break down a long instruction set for an agent into smaller focused pieces.

1. Go to the **Overview** page for your main agent.
2. In the **Instructions** field, enter a slash (/) and select the desired agent. Copilot Studio automatically saves the instructions.
3. Test your agent. When you use the test panel in Copilot Studio, you should see your agent referenced in the activity map.

Manage child and connected agents

You can make any child or connected agent momentarily unavailable or remove it entirely from your solution.

Turn an agent on or off

On the Agents page for your main agent, use the **Enabled** toggle next to the agent you want to turn on or off.

Turning off a child agent or a connected agent makes it inactive—that means it doesn't respond to users or triggers.

Delete a child agent

On the Agents page for your main agent, select the **More** icon (...) next to the child agent you want to remove and select **Delete**.

Remove a connected agent

On the Agents page for your main agent, select the **More** icon (...) next to the connected agent you want to remove and then select **Disconnect agent**.

Known limitations

The following limitations apply to child and connected agents.

- The **Redirect** node doesn't support redirecting to child and connected agents.
- Citations might be missing for information a child agent passes to the main agent. As a workaround, consider utilizing the instructions for the child agent. For example, instruct the child agent to always send a message immediately once it finds an answer to a user's question: "Once you have searched knowledge and found an answer, always send a message with the answer immediately." You can also instruct the child agent to maintain citations within any summary it generates: "When producing a summary, always maintain all citations and include them in the summary."
- While child and connected agents respect the **Use general knowledge** setting of the main agent, by not using general knowledge as part of answers generated by the built-in knowledge tool, they might sometimes use underlying LLM knowledge when generating questions or messages.
- You can't connect to an existing agent that is already connected to other agents.

Event trigger overview

08/26/2025

You can configure custom agents to perform actions or call topics in response to something happening using *event triggers*. Unlike topic triggers, which require input from a user, event triggers allow your agent to act autonomously in response to the defined event occurring.

Important

Enabling event triggers can impact how billing is calculated. For more information, see the [Copilot Credits and events scenarios](#) table.

Note

This feature is only available for agents with [generative orchestration](#) enabled.

How event triggers work

Event triggers require a chosen event which generates a *trigger payload* and sends it to the agent through a connector. The payload contains information about the event, including variables for specific kinds of data. When the agent receives the payload, it executes the directions provided by the agent author in the agent's instructions and instructions sent through the trigger payload.

Agents only act based on their author's design and instructions. For example, you can add a trigger for *when a new team member is added*, and designate the response to be the action *send the new employee a welcome message with onboarding resources*.

Event triggers activate based on events external to the agent. They're different from [topic triggers](#), which are used for activating topics or actions based on trigger phrases entered by users.

This article explains how event triggers work, their limitations, and troubleshooting strategies. For more information about adding an event trigger, see [Add an event trigger](#).

Other examples of event triggers include:

- When an item is created in SharePoint
- When a file is created in OneDrive
- When a task is completed in Planner

- A set amount of time passed (a **Recurrence** trigger)

ⓘ Important

What triggers are available depends on your organization's [data policies](#), configured in Power Automate by an administrator.

Billing and usage for event triggers

Event trigger activity counts towards units of consumption for usage and billing. A message is defined as any request or message sent to the agent that triggers an action or a response from the agent. This includes payloads sent to agents from event triggers. For example, a recurrence trigger set to activate every 10 minutes sends a trigger payload as a message to an agent every 10 minutes.

For more information, go to [Billing rates and management](#).

The trigger workflow

A trigger is one part of a workflow containing multiple parts:

1. An event trigger registers that a specific *event* occurred.
2. The trigger sends a *payload* containing information about the event and instructions.
3. The agent has *instructions* to choose one or more *actions* or *topics* based on the payload.

Find the trigger that fits your event

Copilot Studio has a library of triggers for a range of events that can occur in Microsoft and partner services. The trigger configuration determines the parameters of the event that initiates the trigger. It also determines the contents of the trigger payload.

Add a trigger

X

In addition to user input, your agent can respond to events. Search for an event trigger to enable the agent to take action automatically.

All

Featured

Library

Search

9 triggers



Recurrence



When an item is created
SharePoint



When a file is created
OneDrive for Business



When a row is added, modified or deleted
Microsoft Dataverse



When a task is completed
Planner



When a new response is submitted
Microsoft Forms



When an item is created or modified
SharePoint



When a new channel message is added
Microsoft Teams



When a new email arrives (V3)
Office 365 Outlook

Cancel

Most triggers allow you to specify parameters about the event that activates the trigger. For example, in the *When a row is added, modified or deleted* trigger for Dataverse, you select which table's changes activate the trigger.

Define the trigger payload

The trigger payload is a JSON or plain text message that contains information about an event and is sent to your agent as a [message](#). When [adding a trigger](#), you can keep the default payload contents for that trigger, or add your own instructions. Later, you can [modify the payload contents](#), including adding variables and string operators, in Power Automate.

For example, the default message in the *When a row is added, modified or deleted* trigger is *Use content from Body*. When the agent receives the payload, it has the content from the row, and instructions to use that content.

Add trigger

X

Manage how your agent responds to user input and external events. This is a paid preview feature. [Learn more.](#)

When a row is added, modified or deleted

Trigger your agent with certain message upon event: When a row is added, modified or deleted.

* Table name
Account Research Agent Triggers

* Scope
Organization

Select columns
Enter a comma-separated list of column unique names. The flow triggers if any of:

Filter rows
Odata expression to limit rows that can trigger the flow, eg. statecode eq 0

Delay until (Optional)
Enter a time to delay the trigger evaluation, eg. 2020-01-01T10:10:00Z

Run as (Optional)
Select an item

Additional instructions to the agent when it's invoked by this trigger
Use content from Body

[Back](#)

[Create trigger](#)

[Cancel](#)

You can add instructions to send to your agent inside the payload that direct the agent on how to act when activated by the trigger. If you have multiple triggers, each trigger payload can have specific and detailed instructions. As a result, you can avoid writing long and complicated guidance in the agent's general instructions or confusing the agent about which instructions apply to which trigger.

For example, in a *When a row is added, modified or deleted* trigger payload, you can add *Send a summary of the changes in the chat*. When the agent receives the payload, it summarizes the changes for the user in the agent chat.

However, avoid writing payload instructions that conflict with the agent's general instructions. Conflicts in instructions can cause an error or unexpected results. Make sure to [test all changes involving event triggers](#).

After creating a trigger, you can [add or change variables or string operators](#) in a trigger payload and modify existing payload instructions [using Power Automate](#).

Agent instructions versus payload instructions

Payload instructions are specific to how to react to one event. You can also use agent instructions to direct your agent in how to handle information from a trigger and how to act when it receives a trigger payload, as well as for determining your agent's general behavior. For example, for an agent that checks for duplicate account names in new Dataverse table rows, the instructions could be: *When a new row is added, check if it's a duplicate account. If there's a duplicate, create a To Do task to investigate, and include details about the changes and duplicates.*

Details  Edit

 **Name**
Dataverse agent

Description
Testing the Dataverse event trigger

Instructions When a new row is added, check if it's a duplicate account. If there's a duplicate, create a To Do task to investigate, and include details about the changes and duplicates.

Orchestration
Use generative AI to determine how best to respond to users and events (preview). [Learn more](#)  Enabled

However, agent instructions might not work best for all situations. If your agent has multiple triggers or multiple complex goals, you should use instructions in the [trigger payload](#) instead.

Continuing the last example, you could add an instruction in the trigger payload to *Look for duplicate account names in the same Dataverse table*. When the agent receives the payload, it's instructed to look for duplicate account names. The agent's general instructions then says, *If there's a duplicate, create a To Do task to investigate, and include details about the changes and duplicates.*

Add trigger

Manage how your agent responds to user input and external events. This is a paid preview feature. [Learn more.](#)

When a row is added, modified or deleted

Trigger your agent with certain message upon event: When a row is added, modified or deleted.

* Table name
Select an item

* Scope
Select an item

Select columns
Enter a comma-separated list of column unique names. The flow triggers if any of:

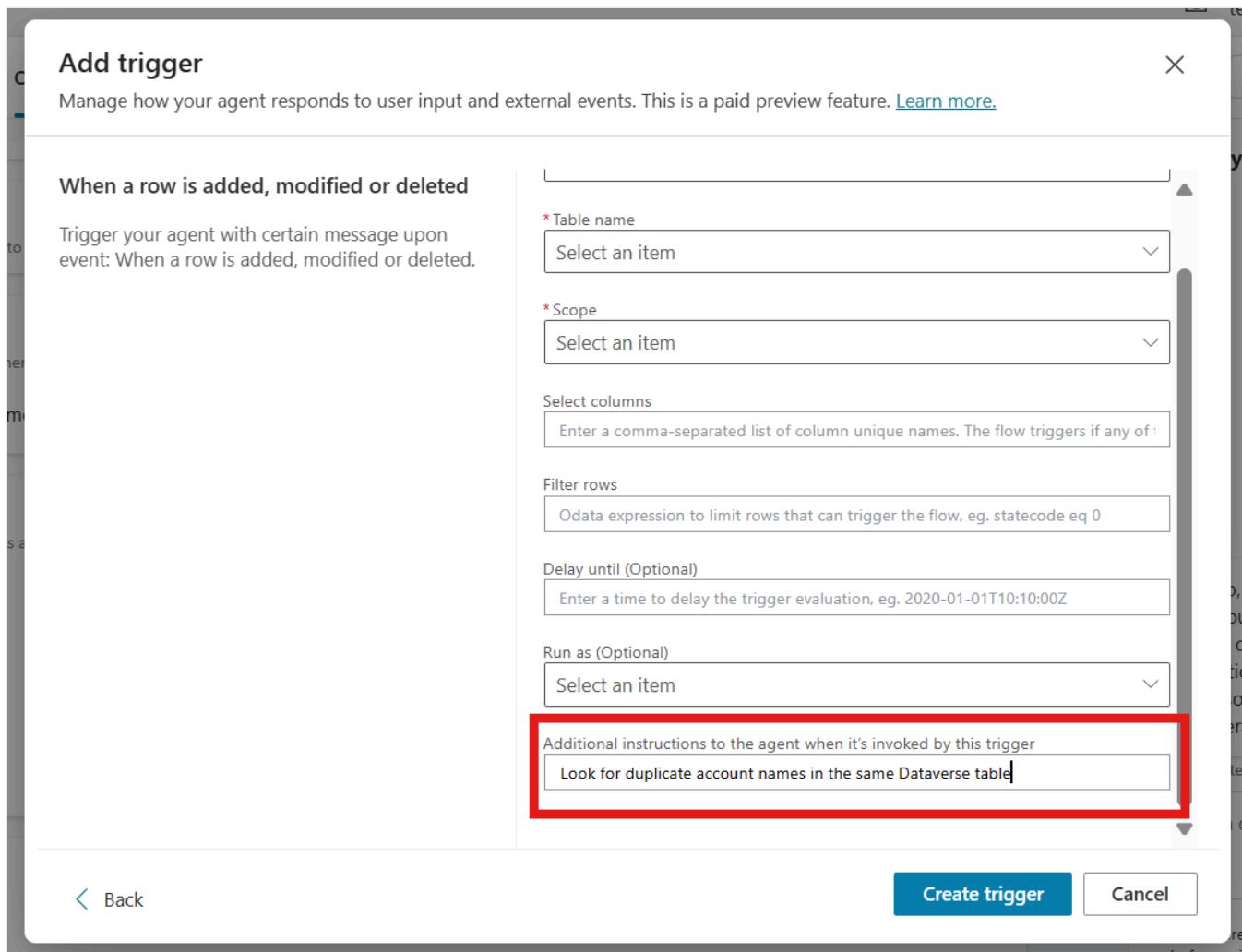
Filter rows
Odata expression to limit rows that can trigger the flow, eg. statecode eq 0

Delay until (Optional)
Enter a time to delay the trigger evaluation, eg. 2020-01-01T10:10:00Z

Run as (Optional)
Select an item

Additional instructions to the agent when it's invoked by this trigger
Look for duplicate account names in the same Dataverse table

[Back](#) [Create trigger](#) [Cancel](#)



Creating instructions for agents takes practice. Go to [writing guidance](#) for more tips, and always test all changes you make to your agent.

Call an action or topic

When an agent receives a trigger payload, the instructions you provide it determine the action or topic it calls in response, based on the information it has available.

Your agent doesn't create a new action or topic on the spot. As the agent author, you need to define the actions or topics it can select from. To learn how your agent determines which action or topic to call, go to [How does generative orchestration work?](#)

If your agent isn't reacting as expected, you can use the [activity map](#) to see if it's missing any key input information.



Add a to-do (V3)

Connector



Description

When you receive action items from new email trigger, create a To Do task with: - a short summary of the action item for the title - the due date set - any other details added to the note field

Inputs (2)

{x} Title | String =

Draft proposal for paperclip

{x} To-do List | String =

⚠️ Missing required information

Outputs (13)

{x} Content | String =

{x} Content Type | String =

{x} Modified Date... | DateTime =

› See more (10)

✖ Session incomplete

Some improvements you can make to instructions include:

- Adding more detailed instructions in the trigger payload or agent instructions. Your agent might need more direction, like what information to use in a specific input field for action.
- Including instructions in the [trigger payload](#).
- Instructing your agent to call a specific action or topic.
- Checking for conflicting instructions between trigger payload and agent instructions.
- Adding to the descriptions of the actions, so the agent has more information to determine when to call it.
- Adding descriptions to the action input fields to help your agent fill in the parameters correctly. If an action's inputs are the same every time, you can set the value yourself.
- Calling fewer than 15 actions or topics consecutively. Complex agents that run many actions or topics as a single sequence can struggle to run them reliably.

If your agent still struggles to call the expected action, consider adding a [Power Automate flow](#) that fulfills your goal as an action for your agent.

Publishing agents with event triggers

Before you publish your agent with a new event trigger, the agent doesn't automatically react to that trigger. Make sure you test the agent thoroughly before publication, because after you [publish an agent](#) with a new trigger, your agent reacts automatically each time its triggers are activated. You can see a step-by-step record of your agent's triggers and reactions in the [Activity](#) page.

For information on activating triggers during testing, go to [Test a trigger](#).

Event triggers can only use the agent maker's credentials for authentication (that is, the credentials you used to [authorize the connections](#)) for your trigger. This might allow users of an agent to use the agent to access data and systems using that same authorization. For more information, go to [Troubleshooting and limitations](#).

Authenticating actions after publishing

If your agent is missing authentication to perform an action or is configured to request user authentication, it sends a message to the user asking for credentials. If an agent's flow is interrupted because it can't receive information or an action failed, it can't continue the session. If you want your agent to run autonomously, each action must be configured with working maker authentication that doesn't require user input. You can also instruct your agent to not request credentials from users.

Because triggers use maker authentication, be aware of what data potential users can access through a published agent that has triggers. See the [Data protection with triggers](#) section for more information.

Troubleshooting and limitations

Quota limitations

If triggers activate too frequently, then your agent might end up using more resources than expected. Your agent might then exceed service load [quota limits](#), and your service could be throttled.

Administrators can [monitor resource usage through Power Platform](#). They can also [block event triggers](#) from being used in an environment.

To avoid exceeding quota limits:

- Take care when adding very frequent triggers, or triggers that recur indefinitely. For example, a recurrence trigger activates whenever a set amount of time passes. The smaller the amount of time between activations, the more resources the trigger uses.
- Keep track of how many triggers are active in an environment.

Data protection for agents with triggers

Currently, event triggers can use only the agent author's credentials for authentication. As well, in order for the agent to run autonomously, all triggers and actions that require authentication must use the maker's credentials.

If you publish an agent with authenticated event triggers, users might be able to access information or prompt the agent to perform actions using the author's credentials. If makers configure an agent with an authenticated event trigger, they are reminded of this limitation before they publish the agent.

Publish this agent

Review the status of this agent or choose Publish to make the content available across all channels this agent is connected to.

 There are risks that should be reviewed. 2 risks 

Your agent runs actions that use the author's credentials View 2 actions

Anyone you share with can use the original author's credentials to access information or complete a task. [Learn more](#)

Your agent includes triggers that use the author's credentials

If the instructions in these triggers share data with other users, those users can use the original author's credentials to access information or complete a task. [Learn more](#)

Publish Cancel

Trigger payloads can potentially contain sensitive information that an agent can output in other locations through actions. For example, if you built an agent that uses information from incoming emails to populate new rows in Dataverse tables.

Administrators can block Copilot Studio users from using event triggers with their agents. For more information, see [Block event triggers](#).

Data protection best practices

To help prevent users from accessing sensitive data:

- Evaluate whether and how data and systems requiring authentication are used by agents with event triggers.
- Understand what information triggers may include in their payloads, and what information actions output using those payload inputs through [testing](#).
- Narrow the scope of what activates a trigger using the [trigger parameters](#), and provide instructions to your agent for how to [handle payload data](#).

Solution-aware cloud flow sharing must be turned on

Makers can only add event triggers to agents in environments where [solution-aware cloud flow sharing](#) is turned on.

If you can't add triggers, and your organization's [data policies allow for triggers](#), your environment may have this option turned off. Contact your environment's admin to turn on solution-aware cloud sharing.

Troubleshoot connector issues

Agents might not be able to run every connector successfully. If an agent repeatedly fails to call a connector, consider creating a [Power Automate flow action](#) that uses the problematic connector to complete the action.

Add an event trigger

Article • 03/26/2025

Event triggers allow your agent to react automatically when an event occurs in other systems. You can then define a specific flow of actions for the agent to take in response, or give your agent instructions for how to determine the best actions to take.

ⓘ Important

Enabling event triggers can impact how billing is calculated. Learn more about [billing for generative mode](#).

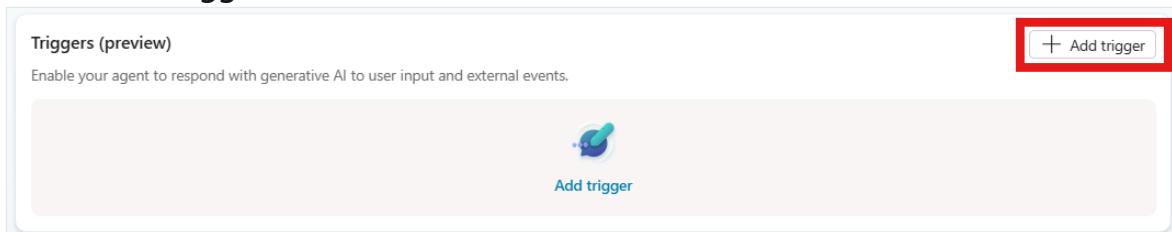
For in depth guidance on how event triggers work and relate to other components of an agent, go to [Event trigger overview](#).

ⓘ Note

This feature is only available for agents with [generative orchestration](#) enabled.

To add an event trigger to your agent:

1. On the [Overview](#) page for your agent, go to the **Triggers** section.
2. Select **Add trigger**.



3. Select the desired trigger.
4. Provide your authentication details, if necessary.
5. Select **Next**.

ⓘ Note

You can only authenticate trigger connectors using the agent maker's account (agent author authentication). The agent maker can only set a trigger for your

agent based on events in systems you have permission to access as agent author. For more information, see the [event trigger limitations](#).

6. Configure your trigger as needed and define the event parameters that activate your trigger and the trigger payload. Triggers provide default payload content, but you can add your own content and instructions. For more information, go to [Define the trigger payload](#).

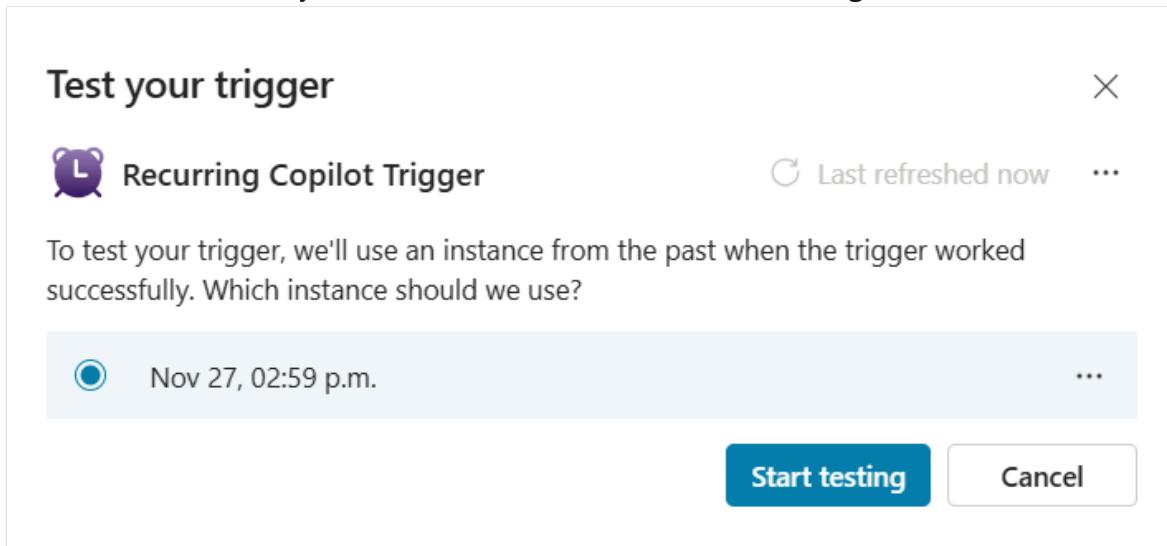
In order for an agent to react to an event trigger, the agent needs actions or topics to call in its response, and instructions for which ones to call. For more information, see [Call an action or topic](#).

Test a trigger

Always test a trigger and its reaction before publishing. You can test how your agent reacts to your defined trigger using the [test agent chat](#).

Until you publish your agent with a new trigger, the agent doesn't react automatically to the trigger. Instead, you must manually activate your agent by selecting a trigger payload to send to your agent.

1. Run the triggering event once to generate a payload. For example, if using the Microsoft Planner trigger *When a new task is assigned to me* for Microsoft Planner, assign a test task to yourself in Microsoft Planner to activate your trigger.
2. Select **Test trigger icon**  beside the trigger in the agent's **Overview** page. The list includes all recent instances of the event.
3. Choose the instance you want to test and select **Start testing**.

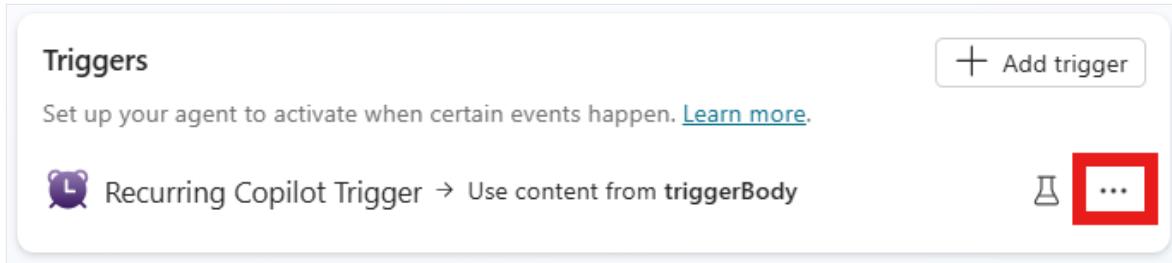


You can use the [activity map](#) to see how your agent reacts to the trigger payload during your tests. To open the activity map, select the map icon  at the top of the test pane.

Modify a trigger

After you create a trigger, you can change the event parameters and payload through Power Automate. To open the trigger from Copilot Studio:

1. In the **Overview** page, locate the trigger.
2. Select the more icon (...), then **Open in Power Automate**.



3. Select the trigger, then go to the **Parameters** and **Code** tabs to make your changes. For information on what variables and string operators you can add, go to the **About** tab.
4. Make and save your changes. For more information on making changes in Power Automate, go to [Get started with triggers in Power Automate](#).

Publish an agent with a trigger

Before you publish an agent with event triggers, you receive a warning about the agent using the author's credentials. Event triggers can only use the agent author's credentials for authentication (that is, the credentials you used to [authorize the connections](#)). This might allow users of an agent without their own authorization the ability to access data and systems they shouldn't. For more information, go to [data protection best practices for triggers](#).

Publish this agent

Review the status of this agent or choose Publish to make the content available across all channels this agent is connected to.



There are risks that should be reviewed.

2 risks 

Your agent runs actions that use the author's credentials

[View 2 actions](#)

Anyone you share with can use the original author's credentials to access information or complete a task. [Learn more](#)

Your agent includes triggers that use the author's credentials

If the instructions in these triggers share data with other users, those users can use the original author's credentials to access information or complete a task. [Learn more](#)

[Publish](#)

[Cancel](#)

After you [publish an agent](#) with event triggers, it reacts automatically each time the agent's triggers are activated. You can see a step-by-step record of your agent's triggers and reactions in the [Activity page](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) 

Add tools to custom agents

08/28/2025

Tools are the building blocks that enable your agent to interact with external systems. Tools expand the functionality of your agent, allowing it to perform various actions in response to user requests or autonomous triggers. Each tool represents a specific capability that your agent can perform. For example, you can equip your agent with tools that do things like:

- Send emails using the Office 365 Outlook connector
- Check the current weather conditions and forecasts
- Read and write data from Dataverse
- Read and post messages to Teams

Mechanisms for adding tools

You can extend the capabilities of your custom agent by adding one or more **tools**. Your agent can use tools to respond to users automatically, using [generative orchestration](#). You can also call tools explicitly from within a topic.

With [generative orchestration](#) (active by default), your agent can automatically select the most appropriate tool or topic, or search across knowledge, to respond to a user. This orchestration mode creates a more dynamic and intelligent conversation experience.

In classic mode (generative orchestration turned off), an agent can only use topics to respond to the user. However, you can still design your agent to call tools explicitly from within topics.

There are several mechanisms available to you to add tools to your agent:

- Prebuilt connector: Choose from a selection of preset connections to thousands of popular APIs from both Microsoft and non-Microsoft services.
- Custom connector: Define a connection to a custom service or system to enable custom tool options using Power Platform Connectors. The connector needs view and share permissions for the organization for the agent to use the connector.
- Agent flow: Define an agent flow, including one or more actions to carry out.
- Prompt: Single turn model-based prompt that can reference knowledge you provide.
- REST API connection: Define a connection to a REST API, and select one or more API endpoints and methods to add as tools.
- Model Context Protocol (MCP): Connect to an MCP server to access tools and resources.
- Computer use (preview) agent: Lets your agent interact with any system that has a graphical user interface, for websites and desktop apps, selecting buttons, choosing menus, and entering text into fields on the screen.

When we talk about tools in this article and this section of the documentation, we mean one of the mechanisms in this group. These tool types can be added to your agent and configured in similar ways.

There are two other mechanisms you can use to add tool-like behavior to your agent:

- Skills: Container for a set of related tools.
- Client tool: Send an event activity to the client so that the client carries out an action and returns a response.

For more information on skills and client tools, see the links in the [Related content](#) section.

Create and add a new tool at agent level

Creating new tools directly within Copilot Studio streamlines the development process and ensures proper integration with your agent. Tools added to an agent are available for automatic orchestration throughout your agent's conversations.

1. Open your agent by choosing **Agents** in the left hand navigation pane and selecting your agent from the list.
2. Go to the **Tools** page for the agent.
3. Select **Add a tool**.
4. In the **Add tool** pane, select **New tool**.
5. Select the type of tool you want to add from the list that appears:
 - **Prompt**: Create AI-powered text generation and analysis tools
 - **Agent flow**: Build multi-step processes with conditional logic
 - **Computer use**: Enable GUI automation for legacy systems
 - **Custom connector**: Connect to proprietary APIs and services
 - **Model Context Protocol**: Connect to an MCP server to access tools and resources
 - **REST API**: Direct integration with web services
6. Perform the configuration steps specific to the type of tool you selected. For example, if you select **Prompt**, you must perform the following steps:
 - Define the prompt template and instructions
 - Specify input parameters
 - Configure knowledge sources
 - Set response format and constraints
7. Test your tool configuration using the preview panel to ensure it works as expected.

8. Select **Save** to create the new tool.

9. Select **Add to agent** to add the tool to your agent. You can see the new tool on the **Tools** page for the agent.

View and make changes to your tool configuration

You can view and edit the configuration of your tool at any time: go to the **Tools** page for your agent, and select the tool from the list of tools.

The **Tool details** page opens, showing the configuration of your tool. The details are displayed in three sections:

- Details
- Inputs
- Completion (Tools for MCP connectors)

Details

The **Details** section lets you configure basic details about your tool.

Here, you can view and update:

- **Name:** The name of the tool. This name appears in the list of tools for your agent. Choose a name that clearly indicates the tool's function.
- **Description:** A description of the tool. Generative orchestration relies on this description to determine when your agent should use the tool. Write clear, specific descriptions including what the tool does and when it should be used.
- **Additional details:**
 - **Allow agent to decide dynamically when to use the tool:** When this option is selected, the agent can use generative orchestration to determine when to use the tool. If this option isn't selected, the agent only uses the tool when it's explicitly called from a topic. (By default, this option is selected when generative orchestration is enabled.)
 - **Ask the end user before running:** In the end user chat experience, ask the agent for confirmation before running the tool. This option is set to **No** by default.
 - **Authentication:** Select whether to use the user's or maker's credentials for the tool. By default, user authentication applies.
 - **Description:** Optionally, you can add a description of the tool that is shown to the user when the tool is run. This description lets the user know what they're being asked to authenticate.

Inputs

Here, you can view and configure the inputs for your tool. The inputs are used to gather information from the user to fill the required inputs for the tool. The information is displayed as a table, one line for each input.

By default, the **Fill using** column value for each input is set to **Dynamically fill with AI**. The agent tries to extract the value from available context—for example, from the user's message. If no appropriate value is found, it generates a question to ask the user for a value. Select **Customize** to access more fine-grained customizations for input collection and filling:

- **Display name and Description:** How the input appears to the user.
- **Identify as:** How the user's response is interpreted, for example as string of text or mapped to a predefined entity.
- **Retry logic:** If the agent doesn't identify an entity in the user's statement, it can ask the question again.
- **Input validation:** Configure extra validation behavior on the user's input beyond the default for the entity type.

You can also choose to override an input with an explicit value instead of letting the agent extract it. To set an override, set **Fill using** to **Custom value**, and enter a value, select a variable, or use a Power Fx formula. If an input is overridden, the agent doesn't ask the user for a value.

Completion

Here, you can select what you want to happen when the tool is done running.

You can have the agent automatically generate a contextual response for a user, based on their query and the result of the tool.

Alternatively, you can choose to author a specific, formatted response for your tool to return. You can insert references to output variables from the tool using the variable picker. You can also use Power Fx formulas to format the response.

Under **After running**, select one of a few different options for how the tool should respond to the user after the tool runs:

- **Don't respond (default):** The agent incorporates the tool output into its response
- **Write the response with generative AI:** Let AI craft a contextual response using the tool outputs
- **Send specific response:** Author a templated response with variable insertion
- **Send an adaptive card:** Create rich, interactive responses with buttons and actions

You can also configure which output variables to make available to the agent and other tools.

Tool selection and input collection

When you define a tool in Copilot Studio, you also provide information that describes its purpose. This information allows the agent to identify when to use the tool and also use generative AI to generate questions. These questions are used to gather information to fill the inputs needed to perform the tool. As a result, you don't need to manually author question nodes to gather all inputs needed, such as inputs in a flow. Input collection is handled for you at runtime.

The agent uses several factors to determine tool selection:

- The tool's name and description
- The current conversation context
- User intent derived from their message
- Available inputs and outputs
- Previous tool usage in the conversation

By default, when using generative mode, tools return their information back to the agent, so that the agent can generate a contextual response to the user's query. Alternatively, you can instruct your tool to always respond immediately, either generating a message or authoring an explicit message.

Tip

When using generated questions from a tool, inform your users that some of the conversation is generated by AI.

For example, add an extra message in the [Conversation Start](#) system topic. This topic determines the message shown to your users when a new conversation starts.

Call an existing tool from within a topic

You can call a tool explicitly from within a topic. Depending on your use case, you might use your tool as part of a wider topic, which uses more nodes. Or, like in the weather example, adding a node to a topic might be all you need.

To call a tool from within a topic:

1. In Copilot Studio, go to the [Topics](#) page for the agent you want to edit.
2. Create a new topic, and give it a name, for example, **Get weather**.
3. Add the following **Trigger phrases**:

- *will it rain*
- *today's forecast*
- *get weather*
- *what's the weather*

4. Select **Add node** (+) and then select **Add a tool**. Select the tool from the available tools.

There are three tabs showing different types of tools:

- Basic tools
- Connector
- Tool

Your **Action** node is now added to your topic.

5. Select **Save**.

MCP connector information

For MCP connector, you can view the names and descriptions of the MCP tools and resources that are made available by the MCP connector. Information for the tools and resources is displayed in a table, one line for each tool.

Authentication considerations for tools

Some tools require authentication to work correctly, such as Dynamic Prompt or others that call a Dataverse API. Proper authentication configuration ensures security while maintaining a smooth user experience.

Tools are always run in the agent's runtime in the user context and can't be run unless authentication is enabled. Two types of authentication methods are supported:

- **User authentication:** The agent uses the user's credentials to authenticate with the service. This method ensures users only access data they're authorized to see.
- **Copilot author authentication:** The agent uses the credentials of its author to authenticate with the service. Use this authentication mode for shared resources or when users shouldn't need individual access.

Test your tool

With [generative orchestration](#), the orchestrator selects your tool when it deems it relevant to a user query.

Alternatively, you can [call an existing tool from within a topic](#).

Related content

- [Use connectors](#)
- [Use prompts](#)
- [Use agent flows with your agent](#)
- [Model context protocol \(MCP\)](#)
- [Add tools from a REST API](#)
- [Use skills](#)
- [Client tools](#)
- [Computer use \(preview\)](#)

Use prompts to make your agent perform specific tasks

06/25/2025

You can author custom prompts to configure instructions to an Azure OpenAI Service or Azure AI Foundry model for your custom agent.

You can add custom prompts to your agent at either of two levels:

- Agent level
- Topic level

Add a prompt as a tool to an agent

1. Select **Agents** in the left pane and select the agent to which you want to add a new prompt as a tool.
2. Select the **Tools** tab.
3. Select **New tool**.
4. Select **Prompt**.

The custom prompt editor opens. You can use this editor to [create a custom prompt for your agent](#).

Add a prompt node in a topic

1. Open a topic and select the **Add node** icon  below the node under which you want to add the new node.
2. Select **Add a tool > New prompt**.

The custom prompt editor opens. You can use this editor to [create a custom prompt for your agent](#).

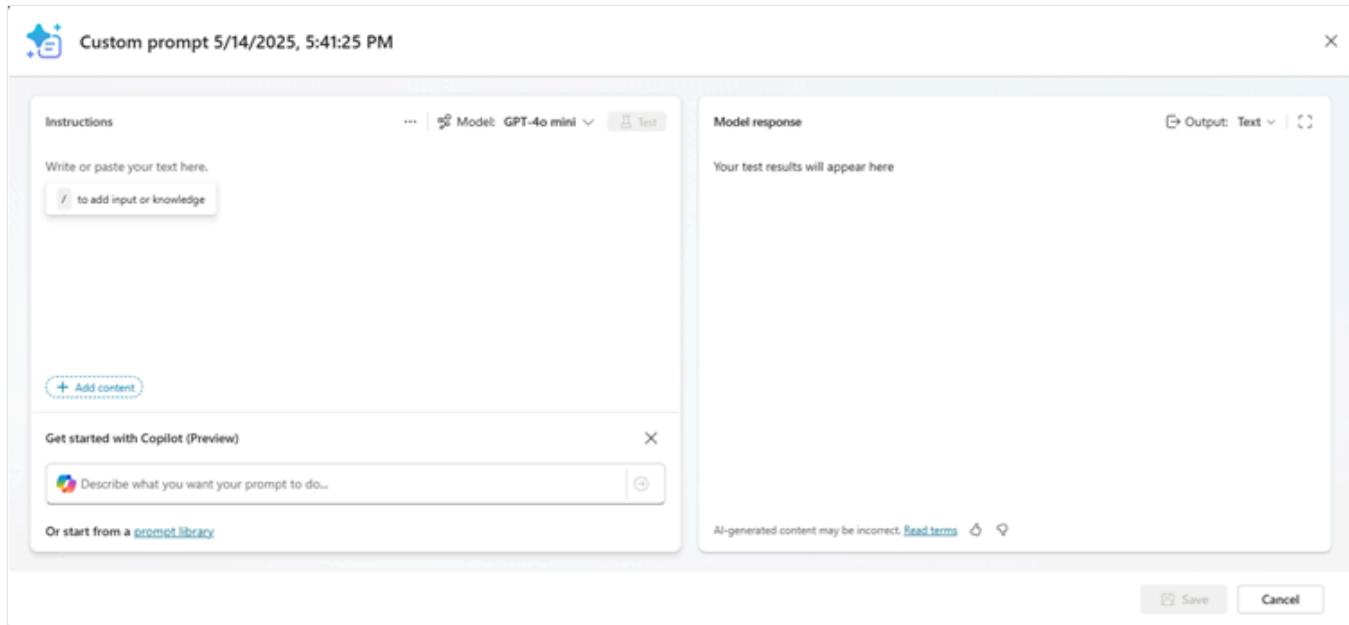
Note

Agents configured to allow anonymous users (no authentication) can't use Dataverse tables as knowledge sources. However, you can still specify input parameters for the

prompt, choose the desired model to run the prompt, and set the desired temperature to control the randomness and creativity of the generated text.

Configure and test a prompt with the embedded AI Builder prompt editor

The custom prompt editor is a powerful tool that allows you to create, customize, and test prompt templates for your agent. The editor uses the same interface as the AI Builder prompt editor, which is designed to help you create effective prompts for your agent.



The prompt editor lets you write your prompt instructions in a few different ways. You can:

- Write out your own instructions manually
- Use Copilot to generate instructions for you based on your description of what you want the prompt to do
- Use a preset template from the prompt library to create a prompt

You can configure several aspects of the prompt, including:

- Chat model to use for the prompt, and settings for the model such as temperature, settings for knowledge retrieval, whether to include links in the response, and whether to enable code interpreter and reasoning.
- User inputs to the prompt such as text and images, and sample data to test with.
- Knowledge to process and include from Dataverse tables.
- Formatting to apply to output.

When you're done creating your prompt, you can test it by specifying sample data for any inputs and selecting **Test**. The editor generates a response based on the prompt and the

sample data you provided.

For full details on how to use the editor to create and test prompts, see [Create a prompt](#), in the AI Builder documentation.

Best practices for prompt instructions

When using the prompt node, it's important to follow best practices for prompt engineering. Here are some tips to help you get the most out of this feature:

- **Be specific:** Custom instructions should be clear and specific, so the agent knows exactly what to do. Avoid vague or ambiguous language that could lead to confusion or incorrect responses.
- **Use examples:** Provide examples to illustrate your instructions and help the agent understand your expectations. Examples help the agent generate accurate and relevant responses.
- **Keep it simple:** Avoid overloading your custom instructions with too many details or complex logic. Keep your instructions simple and straightforward so the agent can process them effectively.
- **Keep it brief:** Custom instructions should be concise and to the point. Instructions that are too long can lead to latency, timeouts, or issues handling the prompt.
- **Give the agent a way out:** Give the agent an alternative path for when it's unable to complete the assigned task. For example, when the user asks a question, you might include "respond with 'not found' if the answer isn't present." This alternative path helps the agent avoid generating false responses.
- **Test and refine:** It's important to test your custom instructions thoroughly to ensure they're working as intended. Make adjustments as needed to improve the accuracy and effectiveness of your agent's responses.

Important

- AI Builder prompts run on models powered by Azure OpenAI Service and Azure AI Foundry.
- This capability is limited to specific regions.
- This capability might be subject to usage limits or capacity throttling.

Use code interpreter to generate and execute Python code

Code interpreter in Microsoft Copilot Studio and AI Builder prompts lets makers build AI agents that:

- Execute Python code for data analysis, processing Word, Excel, PowerPoint, and PDF files, and visualizations
- Use language models for generative responses

This feature is tailored for developers, business analysts, and low-code makers. Code generation and execution integrate seamlessly with the Copilot Studio ecosystem to create powerful, AI-driven solutions.

Developers should read [Code interpreter for developers](#) and review the [Code interpreter Power Apps Component Framework \(PCF\) component sample](#) to understand how they can use the Dataverse `Predict` message to invoke prompts and process the responses returned.

Requirements

- **Licensing:** [Microsoft Copilot Studio billing and licensing](#). Code generation and execution count as [text and generative AI tools \(premium\)](#) features.
- **Supported regions:** Available for all public clouds. Sovereign clouds aren't currently supported.

Code interpreter capabilities in a glance

Code interpreter lets agents write and execute Python code for advanced tasks. Use cases include:

- Excel file processing: Create, copy, and update Excel worksheets in a workbook, read StyleNames, apply StyleNames, copy formatting across cells, copy and update formulas across cells, and so on.
- Word and PowerPoint file processing.
- PDF file processing: Create and copy PDF files, read tables, and text paragraphs.
- Dataverse table data processing.
- Complex mathematical and statistical computations.
- Data analysis, manipulation, and visualization (for example, charts and graphs).

Code interpreter adds the following capabilities to prompts:

- Support for files as an output type in prompts
- Support for Excel files as input and output types for prompts

You can create prompts in two ways:

- Using the **Build your own prompt** option in AI Hub in Power Apps
- Within a prompt tool added to an agent in Copilot Studio at the agent level

For more information on prompts, see [Prompts overview](#).

You can also use code interpreter in agent chat in Copilot Studio.

Turn on code interpreter for the environment in Power Platform admin center

For all uses of code interpreter, you first need to turn on code interpreter for the environment in Power Platform admin center.

Important

A tenant admin must turn on code interpreter using Power Platform admin center. The setting is a *per environment* setting and is configured as **Off** by default. Once you turn on the setting in Power Platform admin center, you can turn on code interpreter in your agents and prompts. Make sure you carry out this step before proceeding with the rest of the setup.

To turn on code interpreter for the environment:

1. In Power Platform admin center, go to **Copilot**, and then select **Settings**.
2. Scroll to the **Copilot Studio** section and select **Code generation and execution in Copilot Studio**. A pane with list of environments opens.
3. Select the environment and select **Add**. A pane opens.
4. Select **On** to turn on code generation and execution in Copilot Studio.
5. Select **Save**.

Create a new empty prompt and turn on code interpreter for the prompt

First, you start by creating a new empty prompt in either AI Hub or Copilot Studio. Then, you turn on code interpreter in the prompt settings.

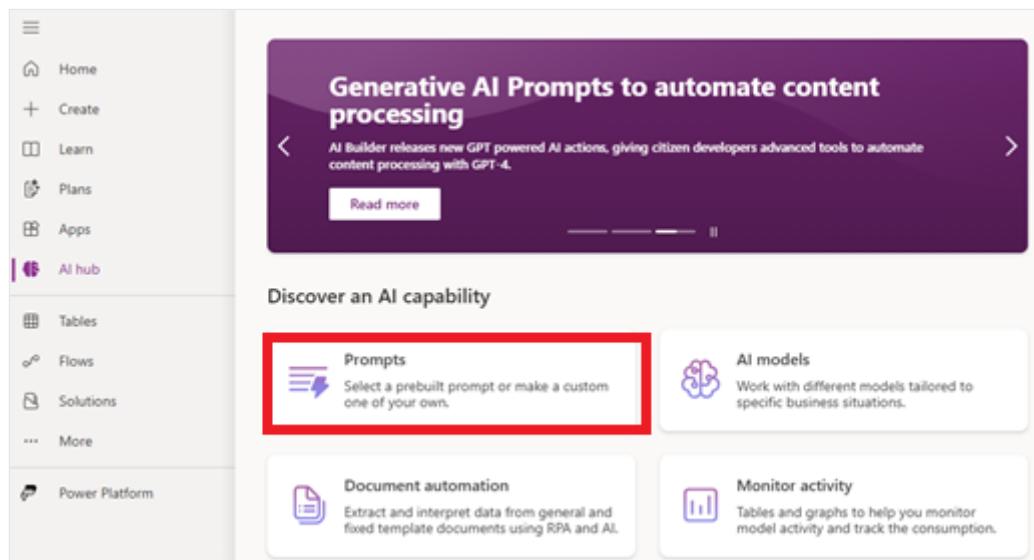
Create prompt in Power Apps AI Hub and turn on code interpreter

You can create a new empty prompt in AI Hub in Power Apps and turn on code interpreter in the prompt settings. You can see examples of how to write and configure prompts that use code interpreter, see [Use code interpreter in prompts examples](#).

i Important

Before you begin, make sure that code interpreter is already [turned on for the environment in Power Platform admin center](#).

1. Go to [Power Apps](#).
2. Select **AI hub** in the left pane.
3. Select **Prompts**.



4. Select **Build your own prompt**.

The screenshot shows the Microsoft Copilot Studio interface. On the left, there's a sidebar with various options like Home, Create, Learn, Plans, Apps, AI hub, Tables, Flows, Solutions, Prompts (which is selected and highlighted in purple), More, and Power Platform. The main area is titled 'Prompts' and contains a section for 'Popular templates'. One template, 'Build your own prompt', is highlighted with a red box. Below it, there are four other templates: 'Summarize text', 'Extract information from text', 'Classify text', and 'Run sentiment analysis on text'. Each template has a small icon, a name, a count (e.g., 69), and a brief description.

5. Provide the name of your prompt.
6. In the **Instructions** section, select ... > **Settings**.

This screenshot shows the 'Instructions' section of Copilot Studio. It includes a placeholder text 'Use copilot to create your i...' and a note about adding inputs with '/'. Above the text is a 'Model: GPT-4.1 mini' dropdown with a three-dot menu icon to its left. A dropdown menu is open, listing 'Clear prompt', 'Prompt library', and 'Settings'. The 'Settings' option is highlighted with a gray background.

7. On the **Settings** page, turn on code interpreter.
8. Close the settings page to return to the prompt editor.

An empty prompt is created with code interpreter turned on. Next, you can write the prompt instructions and configure the prompt as needed.

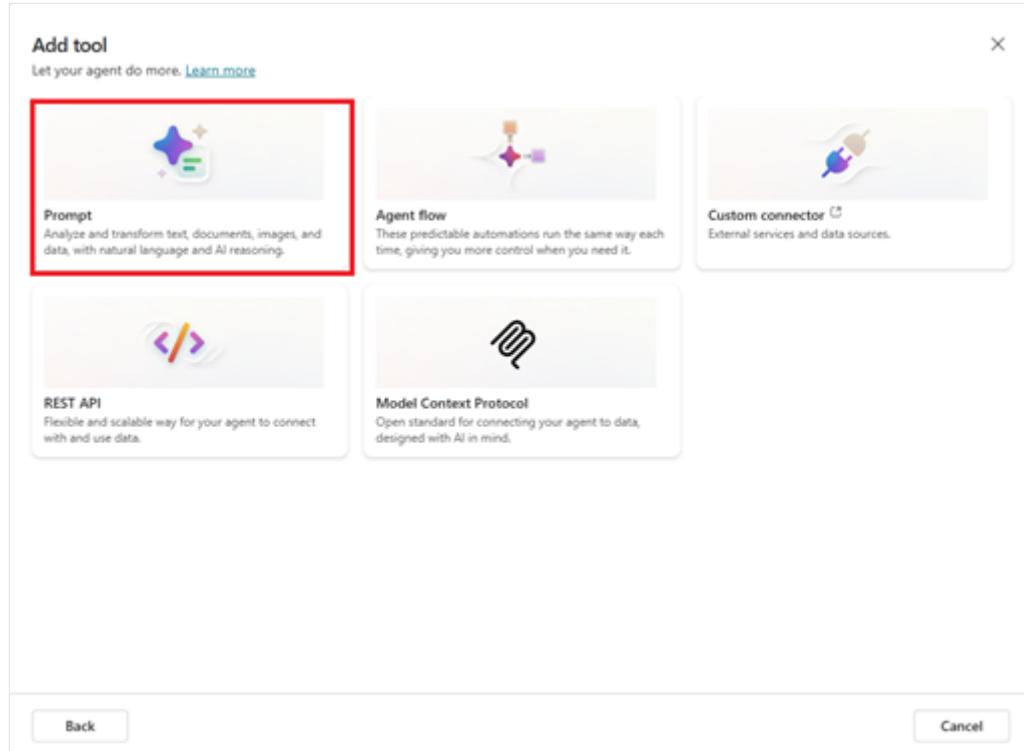
Create a prompt as a tool within an agent and turn on code interpreter

You can create a new empty prompt as a tool within an agent in Copilot Studio and turn on code interpreter for that prompt. Later in this article we show you how to write and configure a prompt that uses code interpreter.

Important

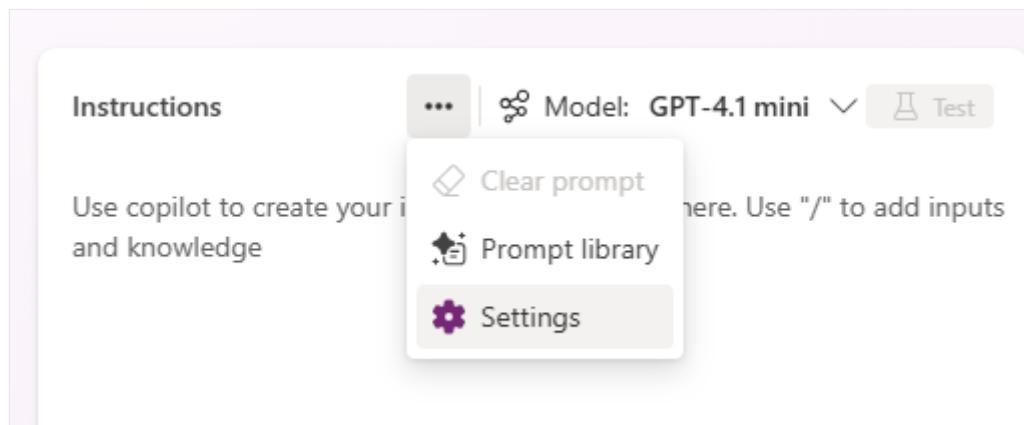
Before you begin, make sure that code interpreter is already [turned on for the environment in Power Platform admin center](#).

1. In Copilot Studio, go to the agent where you want to turn on code interpreter.
2. Select the **Tools** tab, and then select **Add a tool > New tool > Prompt**.



The embedded AI Builder prompt authoring interface opens in Copilot Studio.

3. Select ... > **Settings** on the information bar.



4. On the **Settings** page, select **Enable code interpreter**.
5. Close the settings page to return to the prompt editor.

An empty prompt is created with code interpreter turned on.

Next steps with code interpreter in prompts

Next, you can write the prompt instructions and configure the prompt as needed.

For detailed examples of how to use code interpreter in prompts, see [Code interpreter prompt examples](#).

Best practices for code interpreter in prompts

- Use other AI tools like Copilot to get the optimized prompt to start.
- Use precise prompts with examples (few-shot prompting).
- Specify outputs (for example, "Return as JSON" or "Excel" or "PDF").

Use code interpreter in agent chat (preview)

This functionality is in public preview and is subject to change.

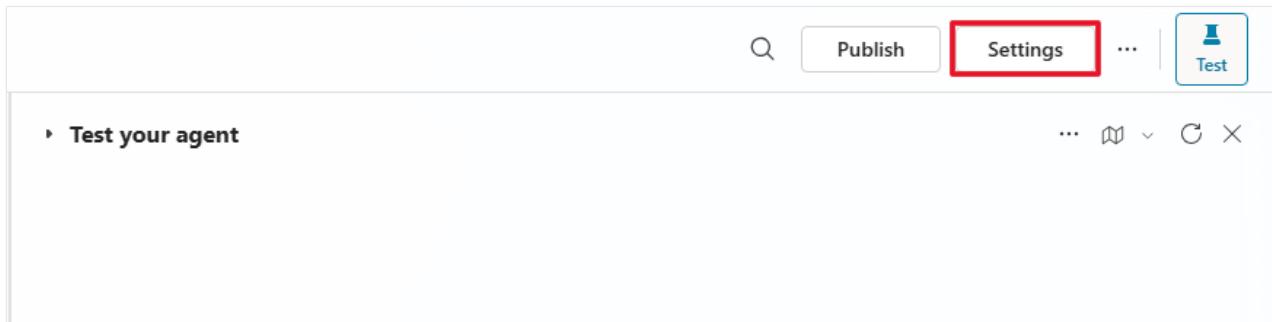
You can also turn on code interpreter at the agent level in Copilot Studio and then trigger it in agent chat using prompt instructions that the agent needs code to perform.

Turn on code interpreter in agent settings

i Important

Before you begin, make sure that code interpreter is already [turned on for the environment in Power Platform admin center](#).

1. Go to your agent and select **Settings**.



2. On the **Generative AI** page, turn on **Code interpreter**.

Settings

Generative AI (highlighted with a red box)

- Agent details
- Security
- Connection Settings
- Authoring Canvas
- Entities
- Skills
- Voice
- Languages
- Language understanding
- Component collections
- Advanced

User Feedback (Preview)

Collect user reactions to agent messages On

Users can give thumbs-up or thumbs-down and an optional comment. Feedback goes to your organization, not to Microsoft. [Learn more](#)

Disclaimer

This message will be shown in supported channels to users when they're submitting feedback. [See supported channels](#).

What should your end users know before they submit feedback? For example, "Your feedback will only be used to improve our services. Don't include any personal information or passwords. See our privacy policy here."

0/500

Knowledge

Use general knowledge On

The foundational knowledge that the generative AI was trained on. To ground your agent only with your specific knowledge sources, turn this off. [Learn more](#)

Use information from the Web Off

Let your agent browse the Web using Bing Web search. [Learn more](#)

File processing capabilities

File uploads On

Users can upload .PDF, .TXT, .CSV and images (.png, .webp, .jpeg and non-animated .gif) that agents can use in conversations. Uploads are limited to 15 MB. [Learn more](#)

Code interpreter (highlighted with a red box)

Generate and run code on demand to process files, create reports, and more. Note that file type and size limits apply. [Learn more](#)

0/500

3. Save the settings and close the settings page to return to your agent.

File processing capabilities

File uploads On

Users can upload .PDF, .TXT, .CSV and images (.png, .webp, .jpeg and non-animated .gif) that agents can use in conversations. Uploads are limited to 15 MB. [Learn more](#)

Code interpreter On

Generate and run code on demand to process files, create reports, and more. Note that file type and size limits apply. [Learn more](#)

Search

Tenant graph grounding with semantic search (Premium) On

Can provide improved search performance for Microsoft 365 Copilot tenants. Availability varies by data source. [Learn more](#)

Save

Trigger code interpreter with a prompt in agent chat

1. Enter the following instructions in the **Test your agent** chat:

Copilot prompt

1. You are given:

- An Excel file with thousands of procurement transactions, including vendor name, invoice amount, PO number, and transaction date.

2. Your tasks:

- Process the Excel file to identify:
 - Transactions over \$10,000 with missing or blank PO numbers.
 - Flag transactions that meet either of the following criteria:
 - High-value transaction with missing PO.
- For each flagged transaction:
 - Highlight the row in red.
 - Add a comment explaining the issue (for example, "PO Missing").

3. Generate a summary report listing:

- Total flagged transactions.
- Vendor names involved.
- Reasons for flagging.

4. Return:

- The updated Excel file with highlights and comments.
- A summary of updates made in text.

2. Upload an Excel file with procurement transactions as input to the chat. You can use the agent chat to generate a sample Excel file with procurement transactions as needed.

3. In a few seconds, you should see the summary output in text as the agent response along with the updated Excel file. You can download the file to verify.

The screenshot shows an AI interface with the following components:

- Activity map:** Shows a "Code (Preview)" step completed in 34.48s.
- Code (Preview) Step:** Status "Complete".
 - Summary: "Flagged Transactions Report Total flagged transactions: **34** Vendor names involved: Acme Supplies, Alpha Manufacturing, Bright Future LLC, Delta Office, Everest Solutions, Global Traders, Highway Imports, NorthStar Systems, Omega Partners, Prime Logistics, QuickBuild, Redline Corp, SafeHands, SecureTech, SkyHigh Reason for flagging: PO Missing ..."
 - See more link.
- Python Editor:**

```
import logging
from workerinterfaces import ExecutorInterface,
ConnectorClient, ExecutionResult, read_file_from_input,
write_file_to_output
from typing import Any, Dict, Optional
import io

import openpyxl
from openpyxl.styles import PatternFill
from openpyxl.comments import Comment

class PromptExecutor(ExecutorInterface):
    async def execute(self, logger: logging.Logger,
                    connector_client: ConnectorClient, input: Dict[str,
                    Any]) -> ExecutionResult:
        output = ("text": "", "error": {}, "files": [])
        error_message = ""
        flagged_rows_count = 0
```
- Rationale:** "Show rationale" button.
- Test your agent:**

Vendor	Invoice Amount	PO Status
Highway Imports	22158.09	103 PO Missing
Bright Future LLC	19038.16	104 PO Missing
SkyHigh	21936.47	109 PO Missing
Bright Future LLC	35520.54	112 PO Missing
Highway Imports	23902.04	119 PO Missing
SkyHigh	32626.57	127 PO Missing
QuickBuild	38272.89	128 PO Missing
Prime Logistics	18769.5	131 PO Missing
Everest Solutions	18308.99	138 PO Missing
Prime Logistics	19400.73	148 PO Missing
Global Traders	30484.28	169 PO Missing
SafeHands	11124.46	176 PO Missing
Everest Solutions	14101.12	189 PO Missing
Bright Future LLC	10910.02	196 PO Missing
Highway Imports	44885.04	197 PO Missing
Alpha Manufacturing	43506.48	198 PO Missing
SkyHigh	37648.6	199 PO Missing
- Files:** "flagged_transactions.xlsx".
- Message from AI:** "If you need any further assistance, feel free to ask!"
- Input Area:** "Ask a question or describe what you need".
- Footer:** "Make sure AI-generated content is accurate and appropriate before using. [See terms](#)".

Limitations

- Analyzing multiple files uploaded in a single prompt isn't supported.
- Returning multiple file outputs in a single prompt isn't supported.
- Asking questions about an uploaded file over multiple turns isn't supported.

Related information

- [Code interpreter for developers](#)
- [Code interpreter PCF component sample](#)

Last updated on 10/30/2025

Use code interpreter in prompts examples

The following examples demonstrate the functionality of code interpreter for handling complex prompts.

The examples assume you already:

- [Enabled code interpreter in Power Platform admin center for your environment](#)
- [Created a new AI Builder prompt and turn on code interpreter in the prompt settings.](#)

Example scenario 1: code interpreter prompt to audit transactions

The purchasing manager at Contoso regularly audits purchase transactions. They receive a periodic PDF report identifying high-risk vendors, and follow a policy requiring purchase orders for transactions of over \$10,000. The purchasing manager uses the PDF to audit the transactions in Excel. The purchasing manager uses prompts with code interpreter to automate this audit process.

Edit and test the transactions prompt

With the prompt editor open and code interpreter turned on, follow these steps to create the prompt:

1. Enter the following instructions for the prompt:

Copilot prompt

1. You are given:
 - An Excel file <parameter A> with thousands of procurement transactions, including vendor name, invoice amount, PO number, and transaction date.
 - A PDF file <parameter B> listing vendors flagged as "High Risk" in the latest compliance audit.
2. Your tasks:
 - Process the Excel file to identify:
 - Transactions over \$10,000 with missing or blank PO numbers.
 - Extract vendor names from the PDF file that are marked "High Risk".
 - Cross-reference the vendor names in Excel with those from the PDF file.
 - Flag transactions that meet either of the following criteria:
 - High-value transaction with missing PO
 - Vendor is classified as High Risk
 - For each flagged transaction:
 - Highlight the row in red
 - Add a comment explaining the issue (for example, "High Risk Vendor" or "PO Missing")

3. Generate a summary report listing:
 - Total flagged transactions
 - Vendor names involved
 - Reasons for flagging

4. Return:
 - The updated Excel file with highlights and comments
 - A summary audit report formatted as PDF
 - A summary of updates made in text

2. In the instructions, replace `<parameter A>` with a file input parameter for the Excel file that contains procurement transactions. To make the change, replace the text with `/` and select **Image or document** from the context menu. Provide a name for this parameter such as "procurement transactions," and upload a sample Excel file with transactions using the expected format.

Make sure to have transaction ID, vendor name, invoice amount, PO number, and transaction date as columns in the transaction file. You can use another prompt with code interpreter turned on to create this sample Excel file as needed.

3. Follow the same steps to replace the text `<parameter B>` with an uploaded input PDF document for the vendor risk report. You can have vendor names as a list in the PDF file. Again, you can use another prompt with code interpreter turned on to create the sample PDF file as needed.

4. Select Test.

In a few seconds, you should see the summary output in text as the model response.

Note

You can leave the default model as GPT-4o or select any other as needed.

5. Under **Model response > Output**, select **Document** for the model response format. The model response should also show the updated Excel file with flagged transactions and the summary report in PDF format. You can download these files to verify the results.

Important

To retest with a different file, you have to change the prompt text and save. Simply uploading or swapping the sample file without modifying the prompt doesn't trigger reprocessing.

This example illustrates the power of code interpreter enabling prompts to perform complex scenarios using files.

Example scenario 2: code interpreter prompt to create a visual summary of Dataverse tables

This example shows how to create a prompt with code interpreter turned on to read records from Dataverse tables, analyze the data, and generate a visual summary (chart) and a text summary.

Edit and test the Dataverse prompt

With the prompt editor open and code interpreter turned on, follow these steps to create the prompt:

1. Enter the following instructions for the prompt:

Copilot prompt

1. You are given:
 - Ticket data that gives the created date/time, first response date/time, and the first response SLA date/time:
2. Your tasks:
 - Create a histogram of the tickets by day of the week created and show the first response SLA status
 - Create insights into trends in the data
3. Return:
 - A graph. Make sure to return the graph as a base64 encoded image
 - A summary of insights

2. Ground the prompt by linking to relevant Dataverse tables (for example, tickets, staff schedules).

- a. Use / to open the input and knowledge menu and select **Dataverse** under **Knowledge**.

Instructions

... Model: GPT-4.1 mini Test

1. You are given:
-Ticket data that gives the created date/time, first response date/time and the first response SLA date/time:

/

Input

- Text
- Image or document
- Power Fx

Knowledge

Dataaverse

The screenshot shows the 'Knowledge' section of a Microsoft Power Automate canvas. A mouse cursor is hovering over the 'Dataaverse' button, which is highlighted with a red rectangle. To the right of the button, there is explanatory text: 'he week created and show the first response SLA status' and 'd image'.

- b. Select a table and select necessary fields (for example, ticket status, assigned agent, SLA status) to the prompt.
- c. Select **Add** to add the fields from the table to the prompt.

Home

Create

Agents

Flows

Tools

Ticket SLA Graph

Instructions

... Model: GPT-4.1 mini Test

1. You are given:
-Ticket data that gives the created date/time, first response date/time and the first response SLA date/time:

/

< Ticket

Created DateTime

Created On

Description

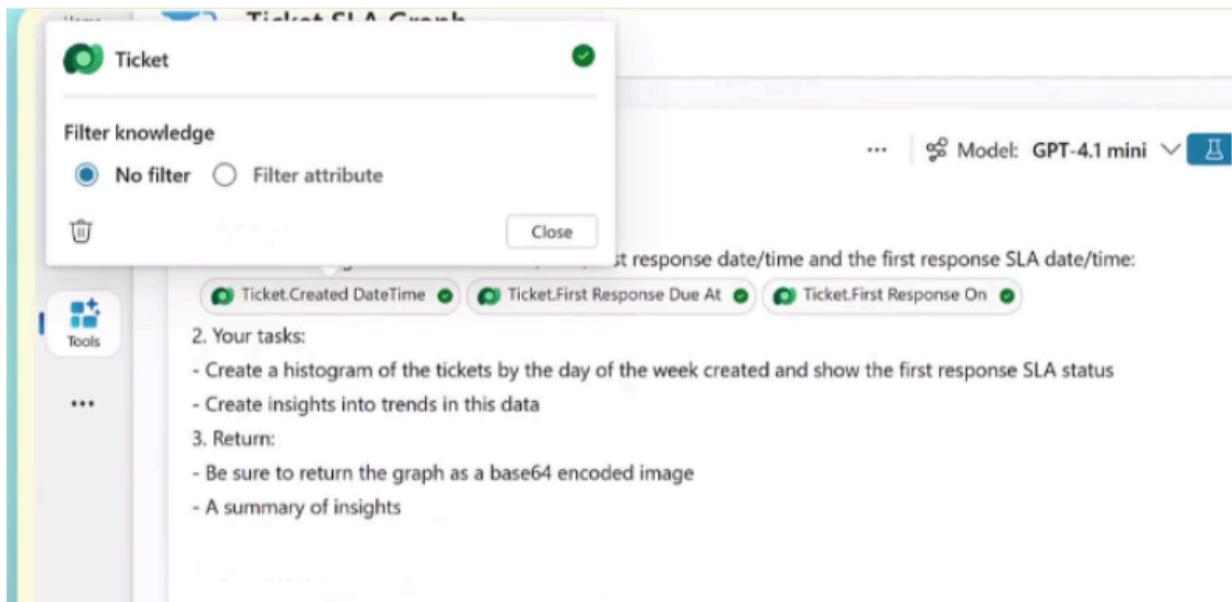
First Response Due At

First Response On

Handoff Count

The screenshot shows the 'Ticket SLA Graph' canvas. On the left, there is a sidebar with icons for Home, Create, Agents, Flows, Tools (which is selected and highlighted with a red rectangle), and more. The main area displays the 'Instructions' and a list of ticket fields. The 'Handoff Count' field is highlighted with a red rectangle. To the right of the list, there is explanatory text: 'he week created and show the first response SLA status' and 'd image'.

- d. Select **Filter attribute**, as needed, to filter the table data on specific attributes and values.



3. Select **Test** to run the prompt.
4. Verify that the prompt reads ticket data, applies assignment rules, and updates records in Dataverse as expected.



5. Save the prompt.

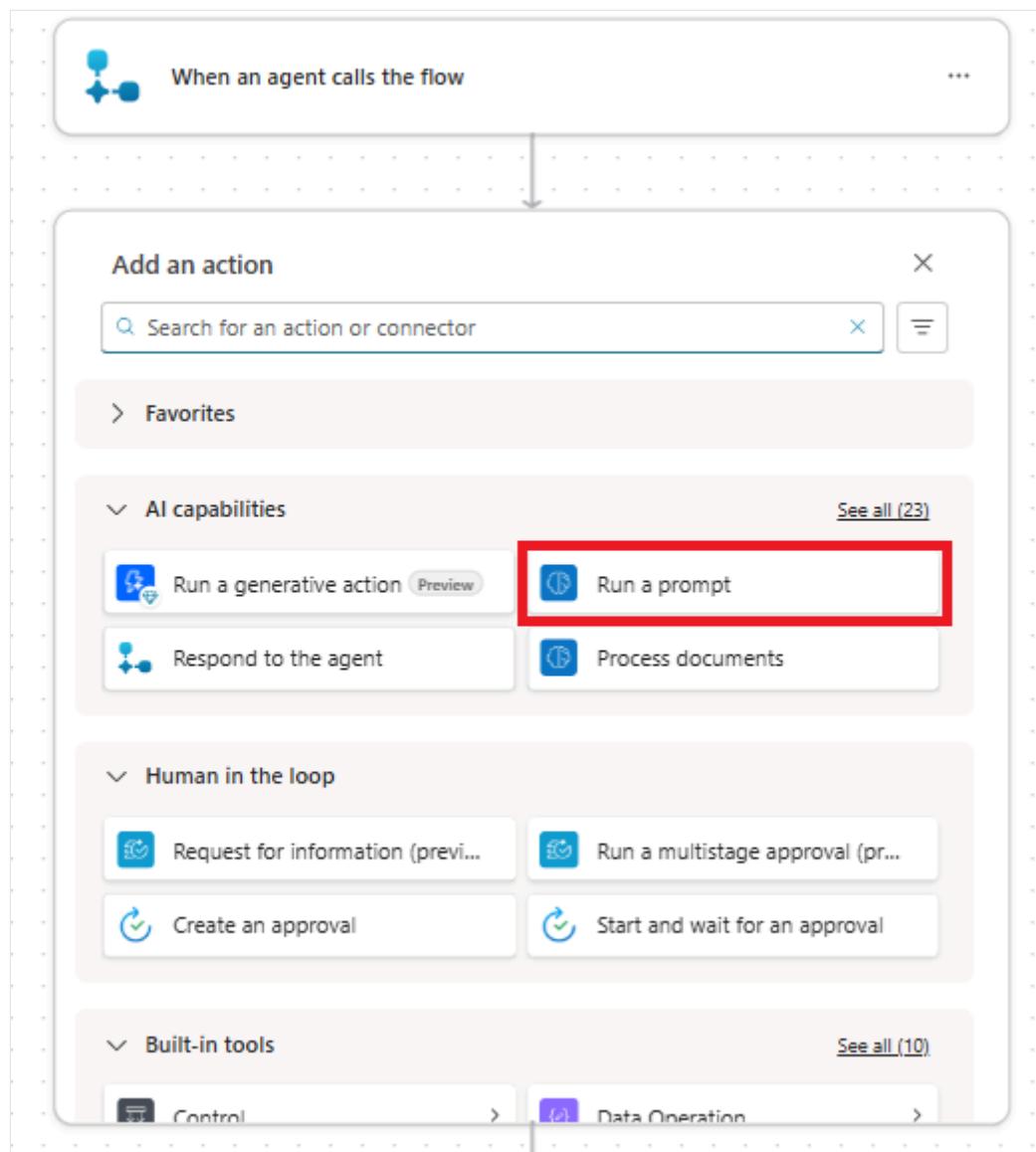
Integrate the prompt into an agent flow in Copilot Studio

Next, you can create an agent flow to wrap the prompt and connect it to Dataverse.

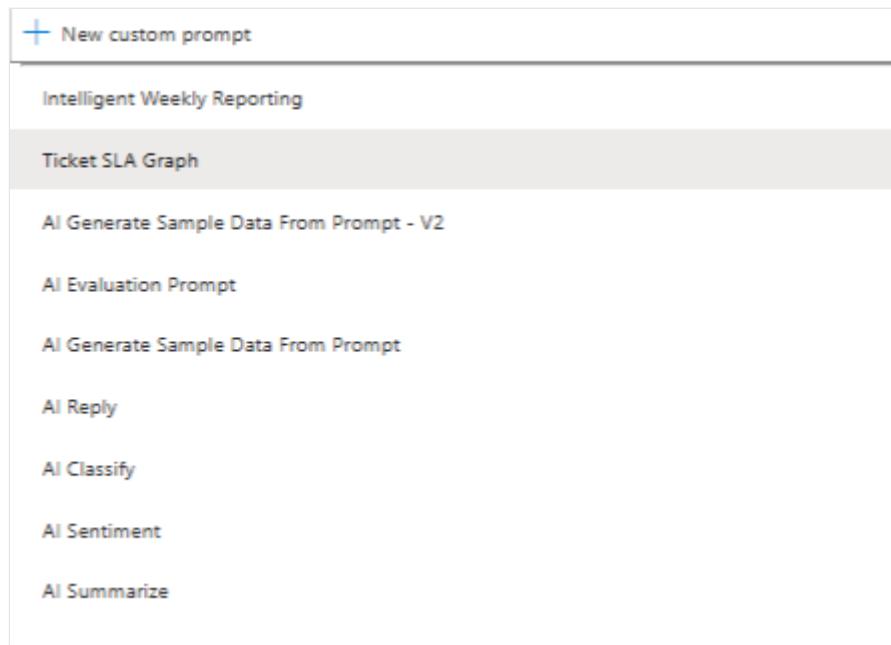
1. Return to the **Tools** page for the agent and select **Add a tool > New tool > Agent flow**.

The agent flow designer appears with an initial **When an agent calls the flow** trigger.

2. In the designer, select **Insert > Run a prompt** to add a new **Run a prompt** action.



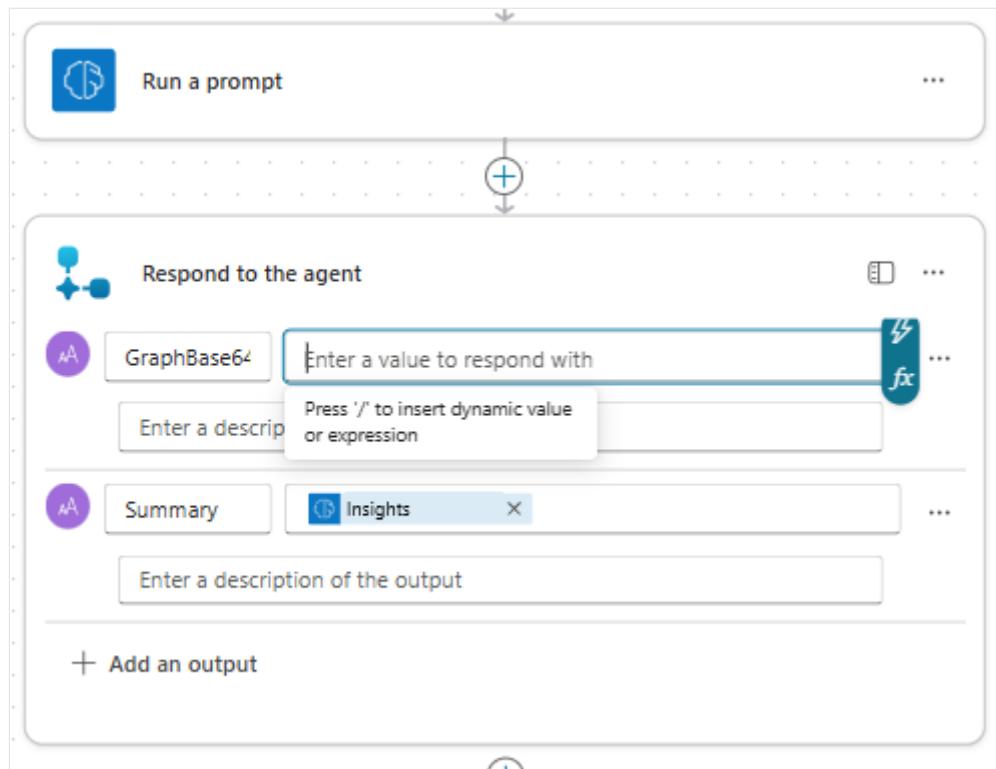
3. In the **Run a prompt** node, select your code interpreter prompt.



4. In the designer, select **Insert > Respond to the agent** to add a new **Respond to the agent** action after the **Run a prompt** action.
5. Next, open the **Respond to the agent** node editor to select parameters to pass back to the agent as outputs.
6. Use **Add an output** to add the summary text and the graph image data from the prompt as outputs for the flow.

Name one of the outputs **Summary**, and use the dynamic value option to select the **Insights** output variable from the **Run a prompt** action.

Name the second output **GraphBase64**, and define the output value using the expression option. Select the Base64 image output from the **Run a prompt** action.



7. Select **Save draft**.

8. On the **Overview** tab, under **Details**, give the flow a name.

Publish and use the agent

1. Save and publish the agent.

The agent can now process Dataverse records and generate a summary and chart.

2. Go to the agent **Tools** page and select the specific agent flow that uses the prompt.

3. Under **Completion**, set the **After running** action to **Send an adaptive card (specify below)**.

4. Under **Adaptive card to display using a PowerFx formula**, select **Formula** to define the card using a formula.

The adaptive card formula editor appears.

5. Define the contents and source for the adaptive card using the following formula:

```
powerfx
{
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "type": "AdaptiveCard",
  "version": "1.0",
  "body": [
```

```
{  
    "type": "Image",  
    "url": "data:image/png;base64," & Topic.Output.GraphBase64,  
    "altText": "Ticket analysis generated graph"  
},  
{  
    "type": "TextBlock",  
    "text": Topic.Output.Summary,  
    "wrap": true  
}  
]  
}
```

Here, `Topic.Output.GraphBase64` and `Topic.Output.Summary` are the output variables defined earlier for the agent flow.

6. Save your agent and test it.
7. Publish your agent so that other users can generate dynamic content based on the Dataverse tables.

Example scenario 3: code interpreter prompt to create a chart from a user-supplied Excel file

This example shows how to create a prompt with code interpreter turned on to read an uploaded Excel file, convert the data into a chart image, and return the image to the user. The prompt is triggered from a topic when the user uploads an Excel file and asks for a chart.

Create and configure the chart generation prompt

With the prompt editor open and code interpreter enabled, follow these steps to create a prompt:

1. Give your prompt a meaningful name (for example, *Image gen prompt*), and enter the following instructions for the prompt:

Copilot prompt

Read the attached Excel (.xlsx) and convert it into a chart named mychart.png.

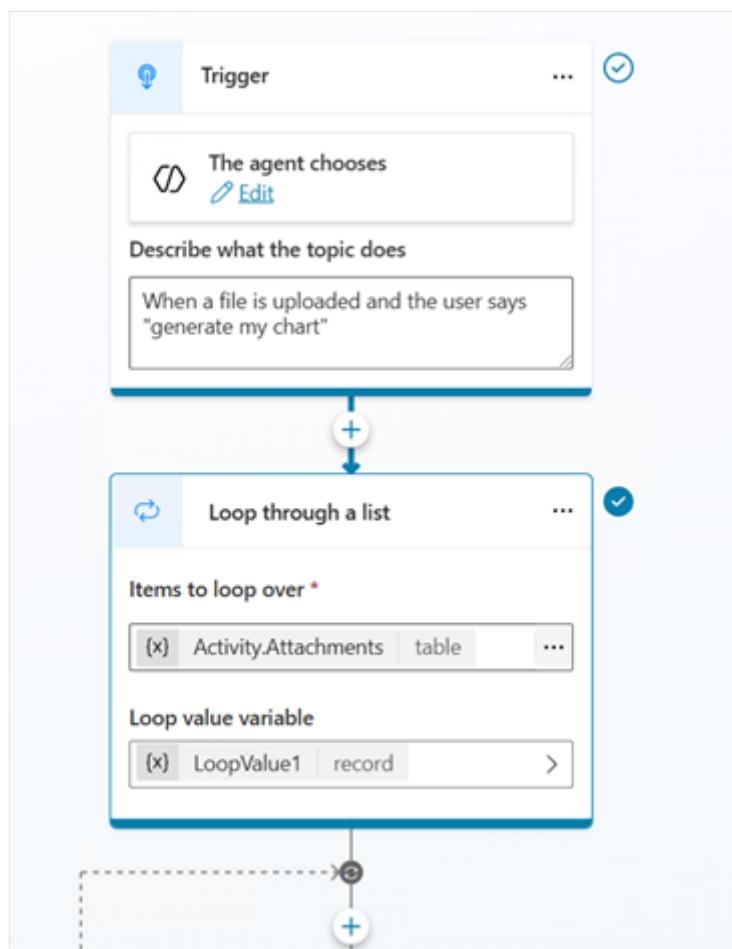
2. Use / to open the input and knowledge menu and select **Image or document** under **Input**.
3. Select an Excel file with sample data to upload as the document input for testing.

4. Select Save.

5. Select **Add to agent** to add the prompt as a tool.

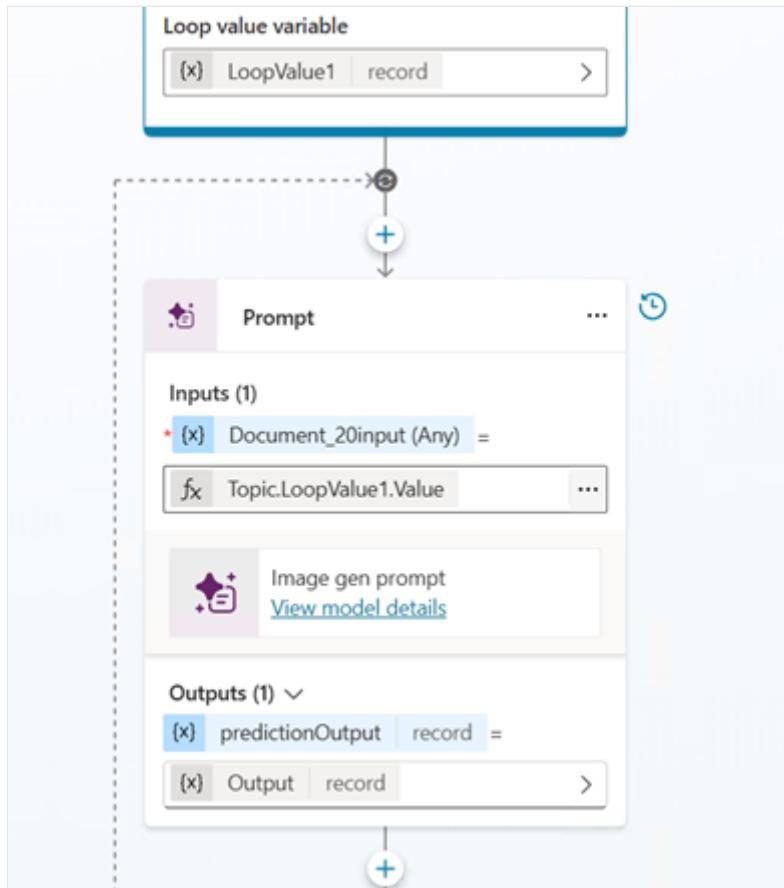
Create a topic that calls the chart prompt

1. Go to the **Topics** tab in Copilot Studio and create a new topic by selecting **Add a topic > From blank**.
2. Under the **Trigger note**, set **Describe what the topic does** to "When a file is uploaded and the user says 'generate my chart.'"
3. Select **Add node**, and then select **Variable management > List management > Loop through a list** to add a loop node.
4. Select the **Loop through a list** node to open the node editor. Under **Items to loop over**, select **Select variable > System > Activity.Attachments**.



5. Within the loop, select **Add node**, and then select **Add a tool** and select the prompt tool you created earlier to add the prompt tool to the topic.
6. Configure input and output variables accordingly. For the input, set using a formula to **Topic.LoopValue1.Value** and select **Insert**. For the output, initialize a variable named

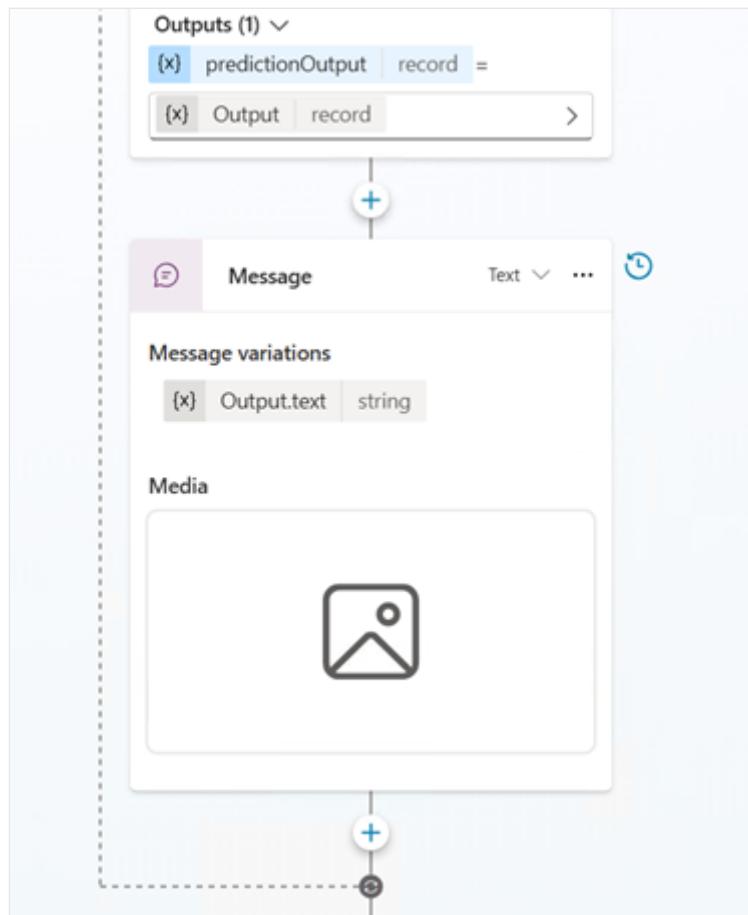
`Output` to capture the prompt output.



7. Within the loop, select **Add node > Create a message** to add a message node. The message node is used to relay the prompt output back to the user. In this example, the prompt initializes the variable `Output` with the response. The text is stored in `Output.text`.

8. In the text portion of the message, select **Insert variable** and select `Output.text`.

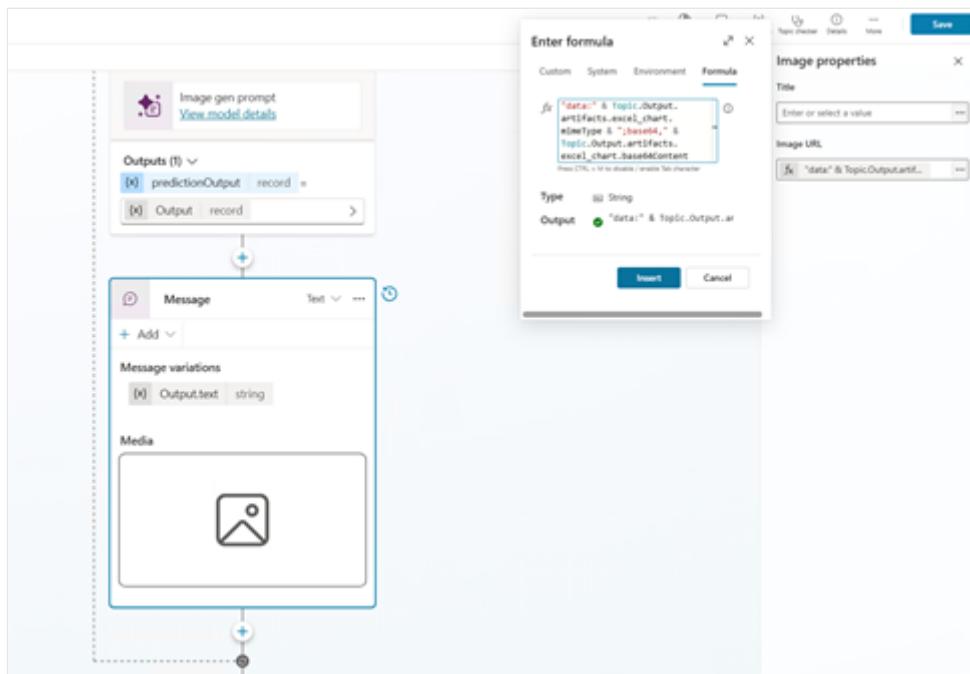
9. Select **Add > Image** to add an image to the message.



10. Next, format to display the image correctly. Select **Media** and under **Image URL**, select **Select variable > Formula**, and enter the following formula:
- ```
"data:" &
Topic.Output.artifacts.mychart.mimeType & ";"base64," &
Topic.Output.artifacts.mychart.base64Content
```

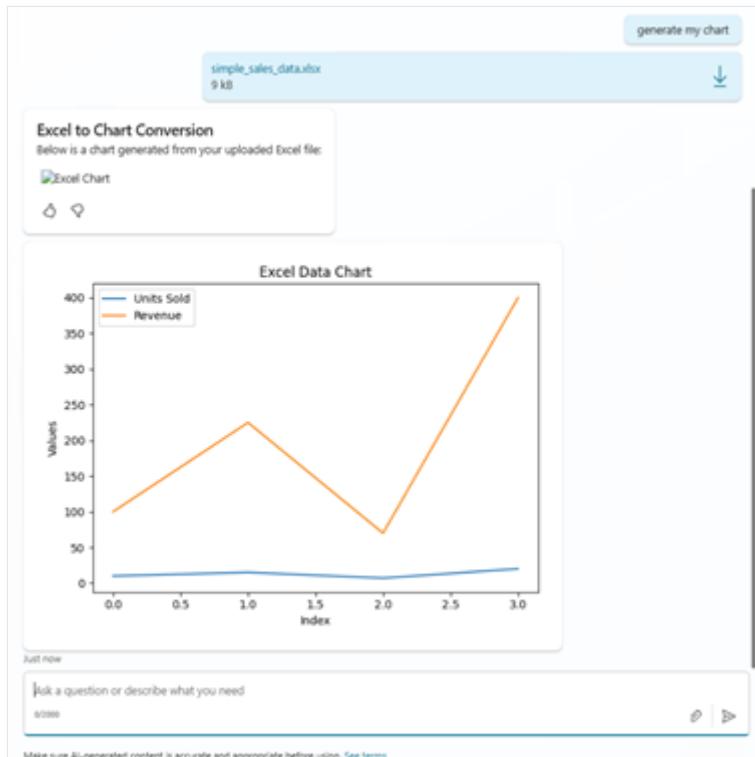
#### ! Note

This format is required in order for the chat to render the image correctly. In this example, `mychart` represents the name we selected for the file output in our prompt and `Output` is the variable name. The details can change in other examples depending on how the prompt output and topic variables are named.



11. Select **Save** to save your topic.

12. Select **Test**. You should see an image displayed when the prompt is triggered.



## More examples of prompts using code interpreter

The following table provides more examples of prompts that can be used with code interpreter turned on. These examples show how to prompt code interpreter to perform complex tasks.

[] Expand table

| Prompt                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Expected result                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Create a PDF file to capture order details                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Generate an order capture form in PDF form                                                                                                                                                                                    |
| Create a copy of the PDF file <parameter A>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Create a copy of the provided PDF file input.                                                                                                                                                                                 |
| Create an Excel file with a synthetic data set of sales transactions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Create an Excel file with sample sales transaction. You can improve on the performance by giving more instructions in the prompt to get more advanced synthetic data generated. The next example shows a more complex prompt. |
| <p>Create an Excel file with a realistic synthetic dataset simulating weekly sales transactions across four regions ("North," "South," "East," "West") for a full calendar year. Each record should include:</p> <ul style="list-style-type: none"> <li>- Date of transaction (weekly)</li> <li>- Region</li> <li>- Salesperson Name (at least three unique names per region)</li> <li>- Product Category (for example, "Electronics," "Furniture," "Appliances", "Clothing")</li> <li>- Product Name (realistic names based on category)</li> <li>- Units Sold</li> <li>- Unit Price (varies by category)</li> <li>- Total Sale Amount (calculated)</li> <li>- Target Met (Yes/No, based on a regional threshold)</li> <li>- Quarter</li> <li>- Channel (Online, Retail, Distributor)</li> <li>- Customer Segment (for example, "Enterprise," "SMB," "Consumer")</li> </ul> <p>Inject some random variation and seasonality:</p> <ul style="list-style-type: none"> <li>- Boost sales for "Electronics" in Q4</li> <li>- Lower sales in "North" during Jan–Feb (simulate winter slowdown)</li> <li>- Higher clothing sales in Q2 (spring)</li> </ul> <p>The final dataset should be at least 2,000 rows.<br/>Add light randomness for realism.</p> | This prompt generates an Excel file with richer synthetic data than before. This prompt illustrates the effectiveness of code interpreter and how we should give clear and detailed instructions in the prompt.               |
| <p>Autocreate Summary Sheet – using &lt;parameter A&gt; Generate:</p> <ul style="list-style-type: none"> <li>- Monthly rollups</li> <li>- Top-performing regions</li> <li>- Conditional formatting for underperformers</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Perform analysis of an Excel file by passing the file as the input into this prompt. This prompt creates the summary in Excel but, can be saved into a PDF file as well by prompting accordingly.                             |

| Prompt                                                           | Expected result                                                                                                                                                                                                                                   |
|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Highlight cells in green color if sales > 1000 else, make it red | <p>Ask the code interpreter to perform Excel formatting based on conditions and formulas. The formatting is expressed in the prompt as illustrated.</p> <p>You can apply the same idea to other use cases in Excel formatting/style handling.</p> |

Last updated on 10/30/2025

# Use connectors in Copilot Studio

06/10/2025

Connectors from Microsoft Power Platform are proxies or "wrappers" around APIs that allow Microsoft Copilot Studio, Microsoft Power Automate, Microsoft Power Apps, and Azure Logic Apps to talk to other apps and services. Connectors let you connect your accounts and use prebuilt tools and triggers to build your apps and workflows.

With connectors, you can access various services (both within the Microsoft ecosystem and outside it) to perform a wide array of tasks automatically.

There are [many connectors](#) available, ranging from connections between and to Microsoft services like Office 365, SharePoint, and Dynamics 365, all the way to connections to non-Microsoft services like Twitter, Google services, Salesforce, and more. These connectors are categorized as:

- **Standard Connectors**, such as SharePoint, which are included with all Copilot Studio plans.
- **Premium Connectors** that are available in select [Copilot Studio plans](#).
- **Custom Connectors**, which let you connect to any publicly available API for services not covered by existing connectors.

## Integration with Copilot Studio

Connectors are essential tools that greatly extend the functionality of Copilot Studio agents by connecting various external services and applications to perform a wide range of tasks. With these connectors you can create more dynamic, responsive, and useful agents, tailored to specific business needs and processes.

You can call connectors as tools in your agent, from an **Action** node in a [topic](#), or at the agent level.

## Add a tool from a prebuilt connector to an agent

1. Select **Agents** and select the agent you want to add a connector to.
2. Go to the **Tools** page and select **Add a tool**.
3. Select **Connector**. The different services with connectors available are displayed.

4. Select the service you want to connect to, or search for the service in the search box. You can see a list of tools available for the service connector.
5. Select the tool you want to add. The **Add tool** pane opens.
6. If the connection doesn't already exist, select **Create new connection**. The details of setting up the connection depend on the connector you selected.
7. Select **Submit** or **Create** as applicable when you're done.
8. Select **Add to agent**. You're taken back to the **Tools** page, where you can see the new tool listed.

By default the connection is configured to use user credentials. For more information about the supported authentication modes, see [Configure user authentication for tools](#). To change this behavior, see the following section.

## Add a tool from a prebuilt connector in a topic

1. Select **Agents** and select the agent you want to add a connector to.
2. Go to the **Topics** page and select the topic you want to add a connector to.
3. Select **Add node (+)** on the authoring canvas.
4. In the node selection window, select **Add a tool > Connector**, and search for the connector tool you want to add.
5. Set up connection details as needed for the connector.
6. Select **Submit**.

By default the connection is configured to use user credentials. For more information about the supported authentication modes, see [Configure user authentication for tools](#). To change this behavior, see the following section.

## Create a custom connector to add to an agent

1. Select **Agents** and select the agent you want to add a connector to.
2. Go to the **Tools** page and select **Add a tool**.
3. Select **New tool > Custom connector**. You're taken to the Power Apps portal under the **Custom connectors** section.

4. Select **New custom connector** and select the method you want to use to create the connector.

## Use connectors with agent author's credentials

Connectors require a valid set of credentials. By default, connectors are configured to ask users (users of your agent) to provide their credentials for the associated service, when the tool is invoked. To have your agent use the author's credentials or credentials for a proxy account, perform the following steps:

1. Configure your agent to use an [authenticated channel](#).
2. Add a connector tool to your agent as a plugin tool, and configure it.
3. Go to the connector tool properties.
4. Under **End user authentication**, select **Agent author authentication**.
5. Publish and test the experience in the **Test your agent** pane, or in the desired channel.

## Share connection

To share your connection with others:

1. Go to [make.powerapps.com](https://make.powerapps.com) .
2. Select **Connections** in the left navigation bar.
3. Select the connection and select **Share**.
4. In the **Share** dialog, search for the desired user and select the user.
5. Under **Permission**, next to the user, select **Can use + share**.

# Use agent flows with your agent

Article • 05/19/2025

Extend the capabilities of your agent with [agent flows](#) that you build in Copilot Studio using low-code, drag-and-drop tools. You can create a flow from the Copilot Studio **Flows** page or convert flows that already exist in your [Power Platform environment](#).

## In this section

The examples in this section help you learn to use flows to do more with your agents.

 [Expand table](#)

| Article                                                            | Description                                                            |
|--------------------------------------------------------------------|------------------------------------------------------------------------|
| <a href="#">Create a flow</a>                                      | Create an agent flow that provides a weather forecast.                 |
| <a href="#">Call a flow as a tool</a>                              | Call an agent flow from an agent topic as a tool.                      |
| <a href="#">Use input and output variables to pass information</a> | Pass variables between an agent flow and a Copilot Studio agent.       |
| <a href="#">Return a list of results</a>                           | Return a list of results from an agent flow to a Copilot Studio agent. |

# Create an agent flow as a tool

06/03/2025

Learn how to use a cloud flow in Copilot Studio by giving an agent the ability to produce a weather forecast on request. In this first example, you create a flow that fetches a weather forecast. In the next example, you set up a topic to [call the flow as an action](#) so the agent can respond with the forecast when a customer asks about the weather.

To work as a tool with agents, an agent flow requires the following:

- Have the **When an agent calls the flow** trigger and a response action of **Respond to the agent**.
- Be configured to respond in real time, not asynchronously. The **Asynchronous response** toggle must be set to **Off** under **Networking** in the **Respond to the agent** action settings.
- Respond to the agent within the 100 second action limit. Optimize the flow logic, queries, and the amount of data returned so that a typical run is below this 100 second limit. Actions in the flow that need to run longer can be placed after the **Respond to Copilot** action to continue to run up to the flow run duration limit of 30 days.

When you create an agent flow from Copilot Studio, the required flow trigger and response action are added for you. You can [modify an existing flow](#) for use with agents by updating it to use this trigger and this response action.

## Prerequisites

- [Access to Copilot Studio](#) with a license that includes the use of flows in Copilot Studio.

## Create a flow you can use with an agent

1. Go to the [Topics](#) page for your agent.
2. Open the topic from which you want to call a flow.
3. Select the **Add node** icon  below any node, and select **Add a tool**.
4. On the **Basic tools** tab, select **New Agent flow**.

The **Flows** page opens, with a starter flow template open in the agent flows designer for editing.

Now add actions to your flow. In this example, you add an action that fetches the current forecast from MSN Weather.

5. Select the name of the flow and replace it with "Get weather forecast."

6. Select the **When an agent calls the flow** flow trigger and add the following input parameters:

- City (Text)
- ZIP code (Number)

7. Select the **Add** icon below the **When an agent calls the flow** flow trigger.

8. In **Add an action**, search for "MSN weather", and select the **Get forecast for today** action.

9. In the **Get forecast for today** panel, in the **Location** box, enter a slash (/), select **Insert dynamic content**, and then select your **City** input parameter.

10. Proceed in the same fashion to add your **ZIP code** input parameter.

11. In the **Respond to the agent** action, add the following output parameters and variables:

  Expand table

| Output parameter | Type   | Variable        |
|------------------|--------|-----------------|
| day_summary      | Text   | Day Summary     |
| location         | Text   | Location        |
| chance_of_rain   | Number | Day Rain Chance |

Parameters   Settings   Code view   About

A day\_summary ● Day Summary x ...  
Enter a description of the output

A location ● Location x ...  
Enter a description of the output

B chance\_of\_ra ● Day Rain Chance x ...  
Enter a description of the output

+ Add an output

12. Save and publish the flow. A notification appears when the flow is created successfully.

13. Select **Go back to agent** to return to your topic in Copilot Studio.

To continue, follow the instructions in [Call an agent flow from an agent](#).

## Manage which connections are used by the flow

In a supported authenticated agent, cloud flows can be [configured to use user credentials](#) when they're run as part of a generative orchestration plan or called from a topic. Using the flow in a supported channel doesn't require any special configuration in your agent. Customers are able to [manage their connections](#) while using the agent.

Cloud flows in environments using [customer-managed keys \(CMK\)](#) can't yet be run with customer credentials as part of generative orchestration plans or from topics. Ensure the run-only settings for cloud flows in CMK environments are set to specific connections instead of **Provided by run-only user**.

## Related content

- [Call a flow as an action](#)
- [Use input and output variables to pass information](#)
- [Add user authentication to topics](#)

# Modify an existing flow to use with an agent

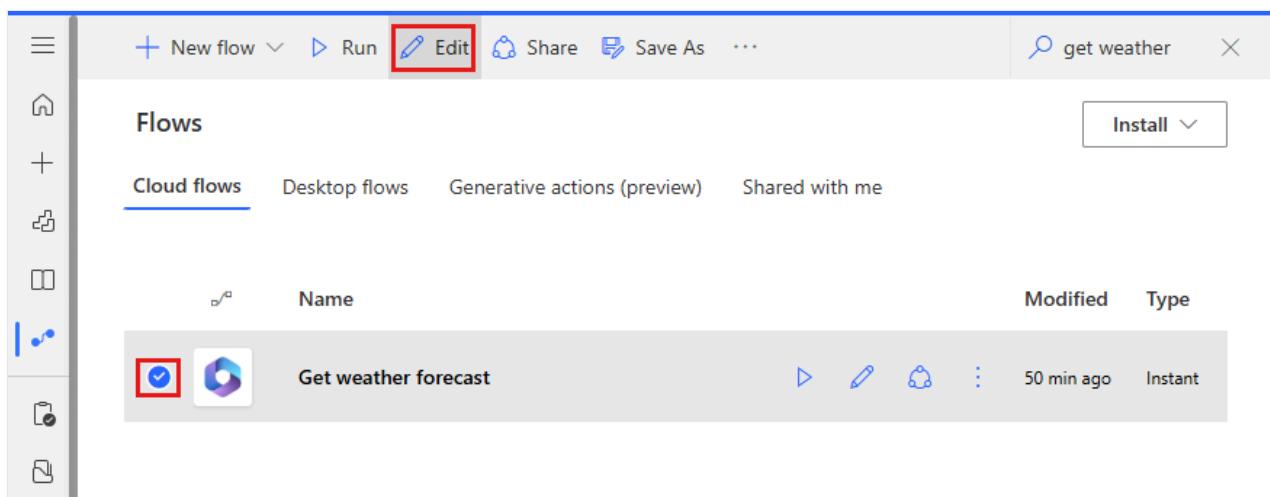
Article • 05/19/2025

To work as an action with agents, an agent flow requires the following:

- Have the **When an agent calls the flow** trigger and a response action of **Respond to the agent**.
- Be configured to respond in real time, not asynchronously. The **Asynchronous response** toggle must be set to **Off** under **Networking** in the **Respond to the agent** action settings.
- Respond to the agent within the 100 second action limit. Optimize the flow logic, queries, and the amount of data returned so that a typical run is below this 100 second limit. Actions in the flow that need to run longer can be placed after the **Respond to the agent** action to continue to run up to the flow run duration limit of 30 days.

If you have a flow with a different trigger that you wish to use with your agent, you can change the flow to use the **When an agent calls the flow** trigger and **Respond to the agent** response action, and make sure it's in a solution in the same environment as the custom agent.

1. To choose the environment where your flow is stored, go to the [Power Automate](#) portal and use the environment selector at the top right.
2. Select **My flows** in the left navigation pane and search for the flow you want to modify.
3. Select your flow, and select **Edit**.



## Add the agent trigger and response action

If your flow doesn't already include the required flow trigger and response action, you must add them before you can use the flow with your agent.

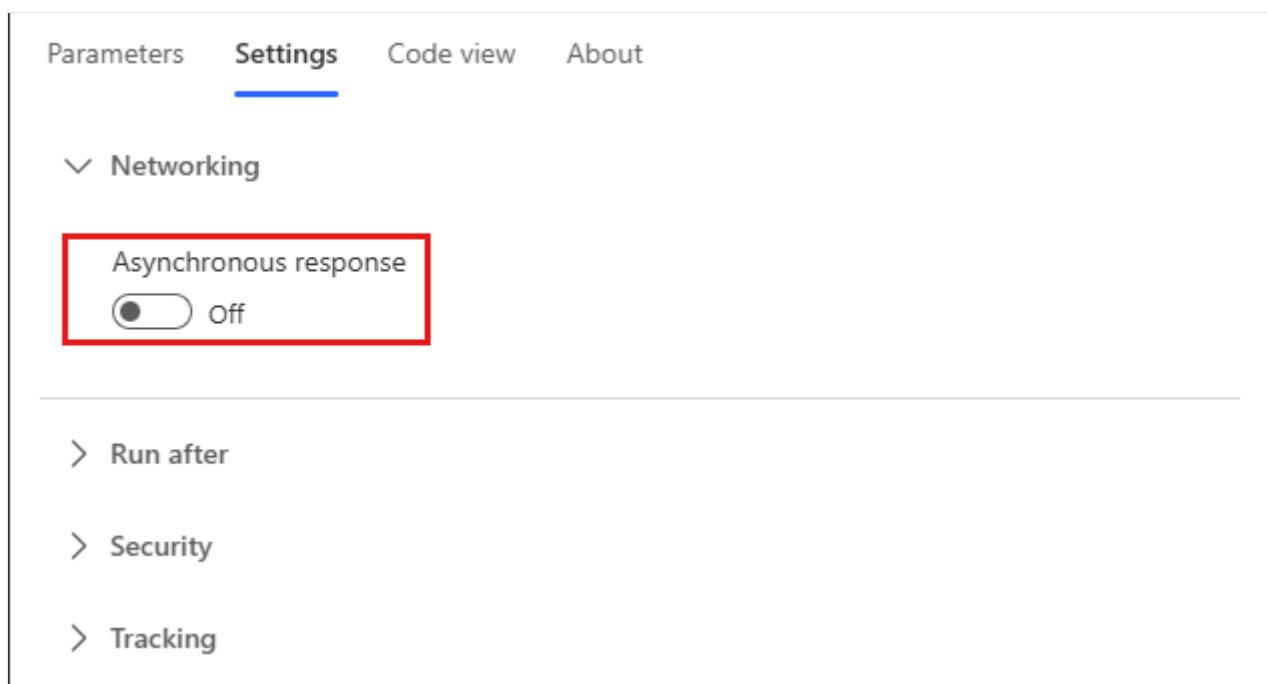
1. Select **New step**.
2. Search for **Copilot** and select **When an agent calls the flow**.
3. Select **New step**.
4. Search for the **Copilot** connector and select **Respond to the agent**. The response action can be used at multiple branches in the flow, but must have the same outputs at each usage.
5. Save and publish the flow.

## Ensure response action is set to return in real time

Flows that you want to use in an agent must return values in real time, or [synchronously](#). Flows that run in the background, or asynchronously, might cause an error when your agent tries to run them. Instead of running the flow, the agent says, "Something unexpected happened. We're looking into it. Error code: 3000."

When you create a flow from Copilot Studio, [asynchronous responses](#) are turned off by default. If you modified an existing flow that has asynchronous responses turned on, follow these steps to check the setting and ensure it's turned off:

1. Open your flow and find the **Respond to the agent** actions.
2. Select the **Settings** tab.
3. Set **Asynchronous Response** to **Off**, and then select **Save**.



# Add the flow to a solution

To be available to agents, flows must be stored in a [solution](#) in the same Power Platform environment. If a flow is in a solution, a Solutions tile is visible on the flow's detail page that lists which solutions the flow is in. Flows can be added to the **Default Solution** for use with agents, or moved into a different solution using the below steps:

1. Go to [Power Automate](#).
2. Go to **Solutions** and either select an existing solution or create a solution for your flow.
3. If you want to create a new solution:
  - a. Select **New solution**.
  - b. Give your new solution a name, select **CDS Default Publisher** in the **Publisher** field, and enter a **Version** number.

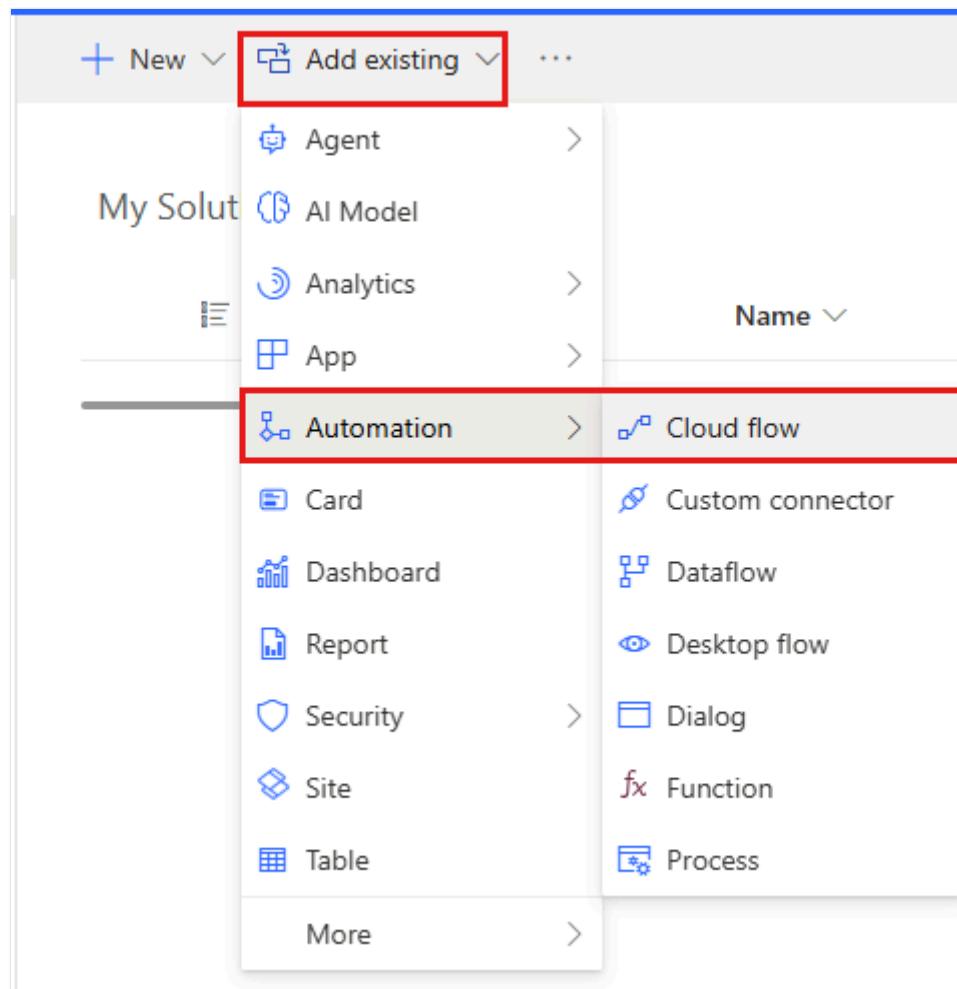
The screenshot shows the 'New solution' dialog box. It has fields for 'Display name' (containing 'My Solution'), 'Name' (containing 'MySolution'), 'Publisher' (set to 'CDS Default Publisher (Cr071a8)'), and 'Version' (set to '1.0.0.0'). There is a 'More options' dropdown and 'Create' and 'Cancel' buttons at the bottom.

|                                                                                      |                                                                                                                   |
|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Display name *                                                                       | My Solution                                                                                                       |
| Name *                                                                               | MySolution                                                                                                        |
| Publisher *                                                                          | CDS Default Publisher (Cr071a8) <span style="font-size: small;">▼</span> <span style="font-size: small;">✎</span> |
| + New publisher                                                                      |                                                                                                                   |
| Version *                                                                            | 1.0.0.0                                                                                                           |
| More options ▾                                                                       |                                                                                                                   |
| <span style="background-color: blue; color: white; padding: 5px 10px;">Create</span> | <span style="border: 1px solid #ccc; padding: 5px 10px;">Cancel</span>                                            |

- c. Select **Create**.

4. If you want to add your flow to an existing solution, select the desired solution, and select **Edit**.

5. Select **Add existing > Automation > Cloud flow**.



The **Add existing flows** panel appears.

6. Select your flow, and then select **Add**.

## Add existing cloud flows

X

Select cloud flows from other solutions or cloud flows that aren't in solutions yet. Adding cloud flows that aren't already in solutions will also add them to Dataverse.

1 cloud flow selected

Search cloud flows

From Dataverse       Outside Dataverse

| <input checked="" type="checkbox"/> | Display name         | Modified       |
|-------------------------------------|----------------------|----------------|
| <input checked="" type="checkbox"/> | Get weather forecast | ... 34 min ago |

Add

Cancel

# Call an agent flow

09/18/2025

Learn how to use an agent flow in Copilot Studio by giving an agent the ability to produce a weather forecast on request.

## Prerequisites

This example depends on a previous example where you [created an agent flow](#) that fetches a weather forecast.

If you haven't already created the weather forecast flow, do that now.

## Call an agent flow from an agent

Here we show how to add the weather forecast flow as a tool to an agent. This method allows the agent orchestrator to call the flow when it needs to provide a weather forecast.

1. From the left navigation pane in Copilot Studio, select **Agents** and select the agent you want to add the flow to.
2. Go to the **Tools** page and select **Add a tool**.
3. Select **Flow** to show the list of available published flows.
4. Select the **Get weather forecast** flow you created earlier.
5. Select **Add and configure**.

The flow is added to your agent, and appears in the list of tools for the agent. The configuration pane for the tool used by the agent opens. You can use this pane to configure the tool for the agent.

6. Under **Name and Description**, enter a display name for the tool and a clear description that helps the agent orchestrator understand when to use the flow. For example, "Get today's weather forecast at a provided city name or zip code."
7. Under **Inputs**, select how you want the agent to fill the variable value.
8. Under **Completion**, select what you want the agent to do after it finishes using the tool.
9. When you're done with the configuration, select **Save**.

# Call an agent flow from a topic

Here we show how to use an **Action** node in a topic to call the flow when a customer asks about the weather.

1. In Copilot Studio, go to the [Topics](#) page for your agent.

2. Create a new topic, and name it **Get weather**.

3. Add the following trigger phrases:

- will it rain
- today's forecast
- get weather
- what's the weather

The screenshot shows the 'Topics' page in Copilot Studio. A new topic named 'Get weather' has been created and is selected. The 'Trigger' node is open, showing the four trigger phrases added: 'will it rain', 'today's forecast', 'get weather', and 'what's the weather'. To the right, a modal window titled 'On Recognized Intent' is displayed, showing the 'Phrases' section with the same four trigger phrases listed. A red box highlights the 'Add phrases' section, which contains a text input field with 'Enter text' placeholder and a '+' button, along with instructions for adding items in bulk or uploading files. Below this, the four trigger phrases are listed again, each preceded by a small blue circle icon.

4. Select the **Add node** icon below the **Trigger** node, and select **Send a message**.

5. On the **Message** node, enter **I can help you with that.** in the text box.

6. Select the **Add node** icon  under the **Message** node and select **Ask a question** to add a **Question** node for the first piece of information for your flow, the city.
7. On the **Question** node:
  - a. Enter the question "What is your city?" in the text box.
  - b. Under **Identify**, select **User's entire response**.
  - c. Note the name of the variable associated with this response, `Var1`. You can leave the name as is, or replace it with a more meaningful name such as "City" (see [Rename a variable](#), if needed).
8. Proceed in the same fashion to add another **Question** node for the other piece of information for your flow, the ZIP code:
  - a. Enter the question "What is your ZIP code?" in the text box.
  - b. Under **Identify**, select **Number**.
  - c. Note the name of the variable associated with this response, `Var2`. You can leave the name as is, or replace it with a more meaningful name (for example, "ZIPcode").
9. Select the **Add node** icon  under the **Question** node for the ZIP code, select **Add a tool**, and then select the flow you created earlier, **Get weather forecast**.
10. Set the flow inputs to the output variables from the question nodes. **City (String)** gets its value from `Var1` (or `City` if you renamed it) and **ZIP code (Number)** gets its value from `Var2` (or `ZIPcode` if you renamed it).

The screenshot shows a Power Automate flow interface. At the top, there is a header with a lightning bolt icon labeled "Action" and a three-dot menu icon. Below the header, the "Power Automate inputs (2)" section contains two items: "{x} City (String) = {x} City | string" and "{x} ZIP code (Number) = {x} ZIPcode | number". Under the "Get weather forecast" action node, the "Outputs (3)" section lists three items: "{x} day\_summary | string", "{x} location | string", and "{x} chance\_of\_rain | number".

11. Under the Action node, add a **Message** node, and enter a message that uses output information from the flow. For example:

"Today's forecast for `location: day_summary`"

"Chance of rain is `chance_of_rain %`"

Where `location`, `day_summary`, and `chance_of_rain` are output parameters from your flow.

The screenshot shows a "Message" node configuration. The message text is set to "Text" and contains the following content:  
"Today's forecast for  
{x} location | string ;  
{x} day\_summary | string  
Chance of rain is  
{x} chance\_of\_rain | number %"

12. Select Save.

## Manage an agent flow used in a topic

Once you have an **Action** node that calls a flow, you can use the following options from the node's menu to manage the flow:

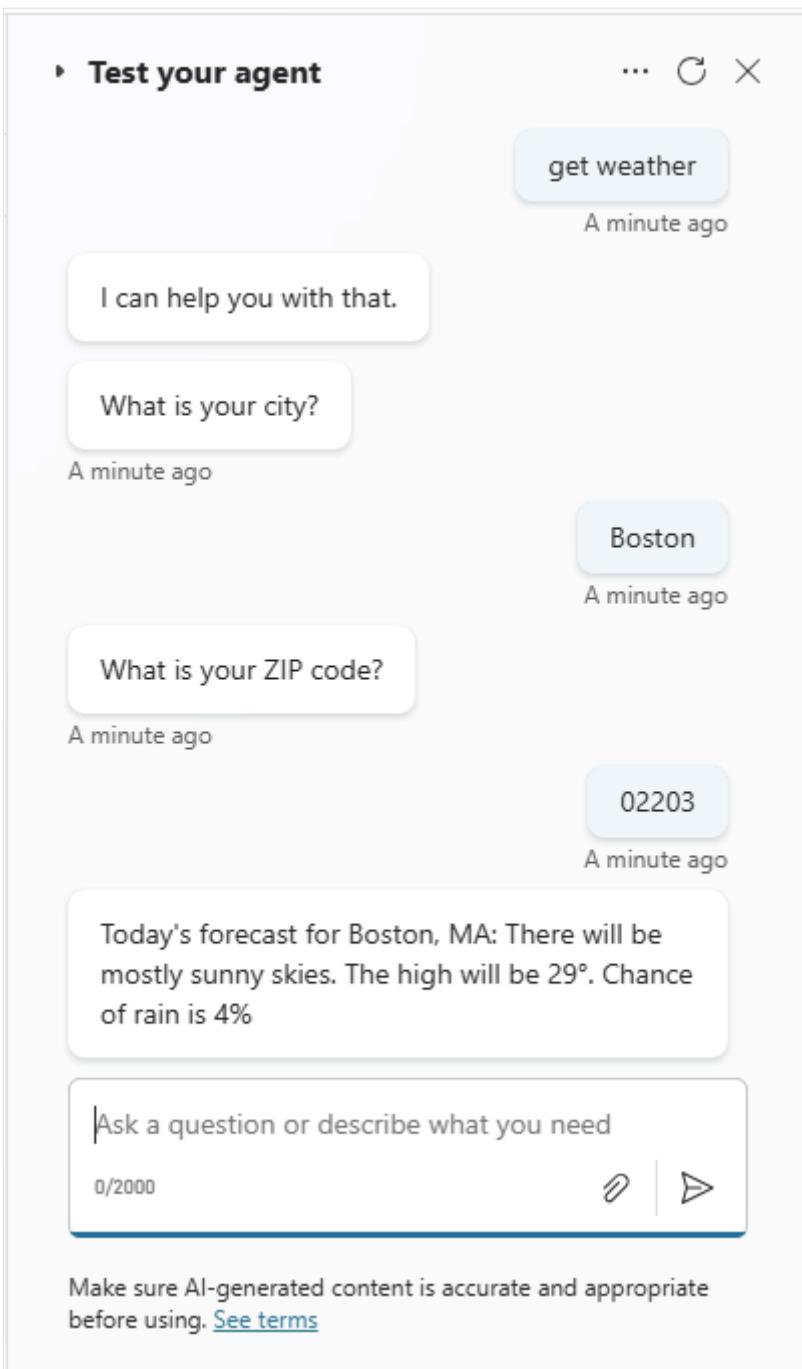
- **Delete** permanently deletes the node and removes the flow from the topic.
- **Refresh** reloads the flow to obtain any changes you made in Power Automate, and validates it in the topic. You must fix any problems with the flow before you can save the topic.

If you would like to select or create a different flow, use the **Change flow** icon  that appears when you hover over the flow in your **Action** node.

## Test your flow and topic

It's a good idea to test the flow and the topic that calls it.

In the **Test your agent** panel, type one of the topic's trigger phrases to start a conversation with the agent. Enter your city and ZIP code when the agent asks for them. Verify that the agent returns today's weather forecast for your city and that the message looks the way you expect.



## Troubleshoot your agent

Always [test your agent](#) when you make changes to topics and flows, to make sure everything still works. If the agent encounters a problem during a conversation, it responds with an [error message](#).

Find most flow-related issues in the flow checker on the [flow designer](#) tab. Use the [topic checker](#) to identify issues on the authoring canvas.

# Use input and output variables to pass information

06/03/2025

Use variables as input and output parameters to pass information between agent flows and Copilot Studio agents.

## Important

An agent can only *receive* up to 1 MB of data from a flow in a single action. However, there's no limit on how many variables or the amount of data an agent can send.

In this example, you create a flow with an input parameter and return it to an agent as an output parameter.

## Prerequisites

- Understand how to [create a flow](#) from the Copilot Studio authoring canvas

## Supported parameter types

Copilot Studio agents support the following types of input and output parameters for agent flows:

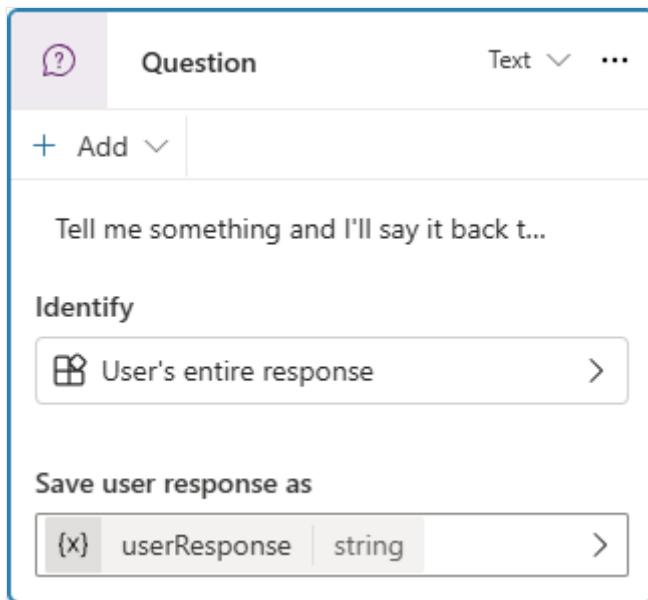
- Number
- String
- Boolean

The following data types aren't supported:

- Object
- Date
- Timestamp
- List [String]
- List [Number]
- List [Boolean]
- List [Object]
- List [Date]
- List [Timestamp]

# Create a topic and a flow

1. Go to the [Topics](#) page for your agent.
2. Create a new topic called **Echo message**.
3. Add the trigger phrase **echo**.
4. Add a **Question** node and enter the message "Tell me something and I'll say it back to you."
5. For **Identify**, choose **User's entire response**.
6. For **Save user response as**, rename the variable to **userResponse**.

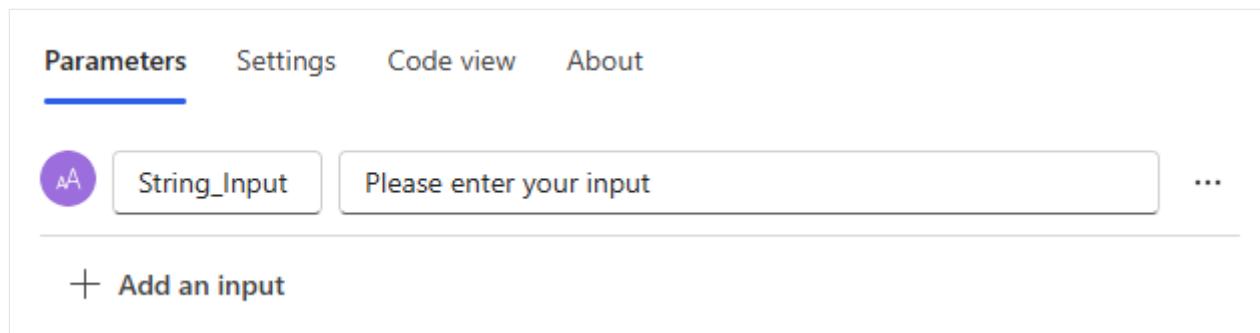


7. Select the **Add node** icon under the **Question** node, and select **Add a tool**.
8. On the **Basic actions** tab, select **New Agent flow**. The agent flow designer tab opens with a starting template for an agent flow called by an agent. The flow includes two actions:  
**When an agent calls the flow** and **Respond to the agent**.
9. Select **Save draft**.
10. On the **Overview** tab of the flow, edit the details for the flow, replacing the name with **Echo parameter**. Select **Save**.
11. Return to the **Designer** tab.

## Add an input parameter to the flow

1. Select the **When an agent calls the flow** flow trigger, and select **Add an input**.

2. Select **Text** and enter the name **String\_Input**.



## Add an output parameter to the flow

1. Select the **Respond to the agent response** action, and select **Add an output**.
2. Select **Text** and enter the name **String\_Output**.



3. For the value, enter a slash (/) in the box, select **Insert dynamic content**, and then select the **String\_Input** variable.
4. Select **Save draft** and **Publish**. Copilot Studio informs you when the agent flow is published.
5. Select **Go back to agent** to return to the agent topic editor. The new action flow is added to the topic.

## Call the flow from the topic

1. Select the new action node.
2. Set the flow input parameter to the output variable from the **Question** node: **String\_Input (string)** gets its value from the **userResponse** variable.

The screenshot shows a Power Automate flow configuration window. At the top, there is a green header bar with a lightning bolt icon and the word "Action". Below the header, the title "Power Automate inputs (1)" is displayed. A single input node is shown, labeled "String\_Input (String) = {x} userResponse string". Below this, under the heading "Outputs (1)", a single output node is shown, labeled "String\_Output string = {x} string\_output". A blue "Echo parameter" button is visible next to the output node.

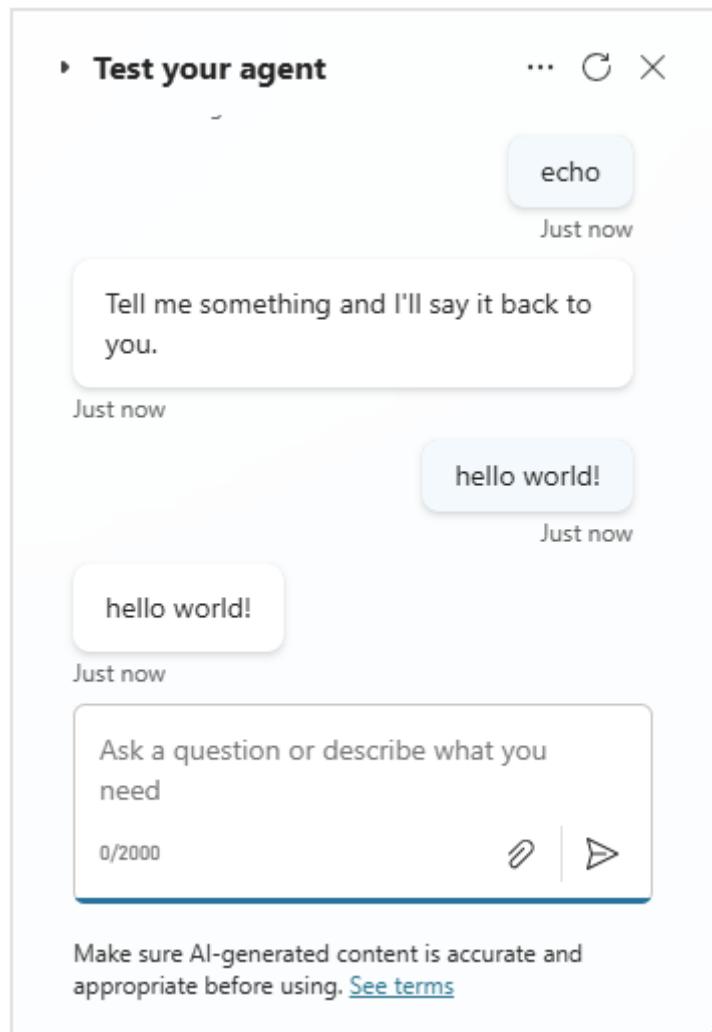
3. Add a **Message** node.

4. Select **Insert variable**, and then select **String\_Output**.

The screenshot shows a "Message" node configuration window. The node has a purple icon with a speech bubble and the word "Message". The "Text" field contains the placeholder "Text" with a dropdown arrow and three dots. Below the text field, there is a variable insertion placeholder: "{x} String\_Output string".

5. Select **Save**.

6. Test your agent.



## Related content

- [Variables overview](#)

# Return a list of results

Article • 05/19/2025

It's common to ask an agent to look for data in an external system and return a list of results. To do that, an agent can call a Power Automate flow to:

1. Authenticate and connect to an external solution.
2. Run a query based on the user input.
3. Format the results.
4. Return the results to the agent.

This example uses the Dataverse connector in Power Automate to search for accounts. The connector returns a list of results that includes the account name, city, and account number to your agent.

## Prerequisites

- Understand how to [create a flow](#) from the Copilot Studio authoring canvas
- [Use input and output variables to pass information](#)
- [Configure Dataverse search for your environment](#)

## Set up a Dataverse Account table

The Account table is a [standard table](#) available by default in Power Platform environments. However, it doesn't come with account data. Before your agent can search for accounts, you must populate the Account table.

If you already have an Account table with data, skip this step and move on to [Create a topic](#). However, remember to use search terms relevant to your data when you test your agent in later steps.

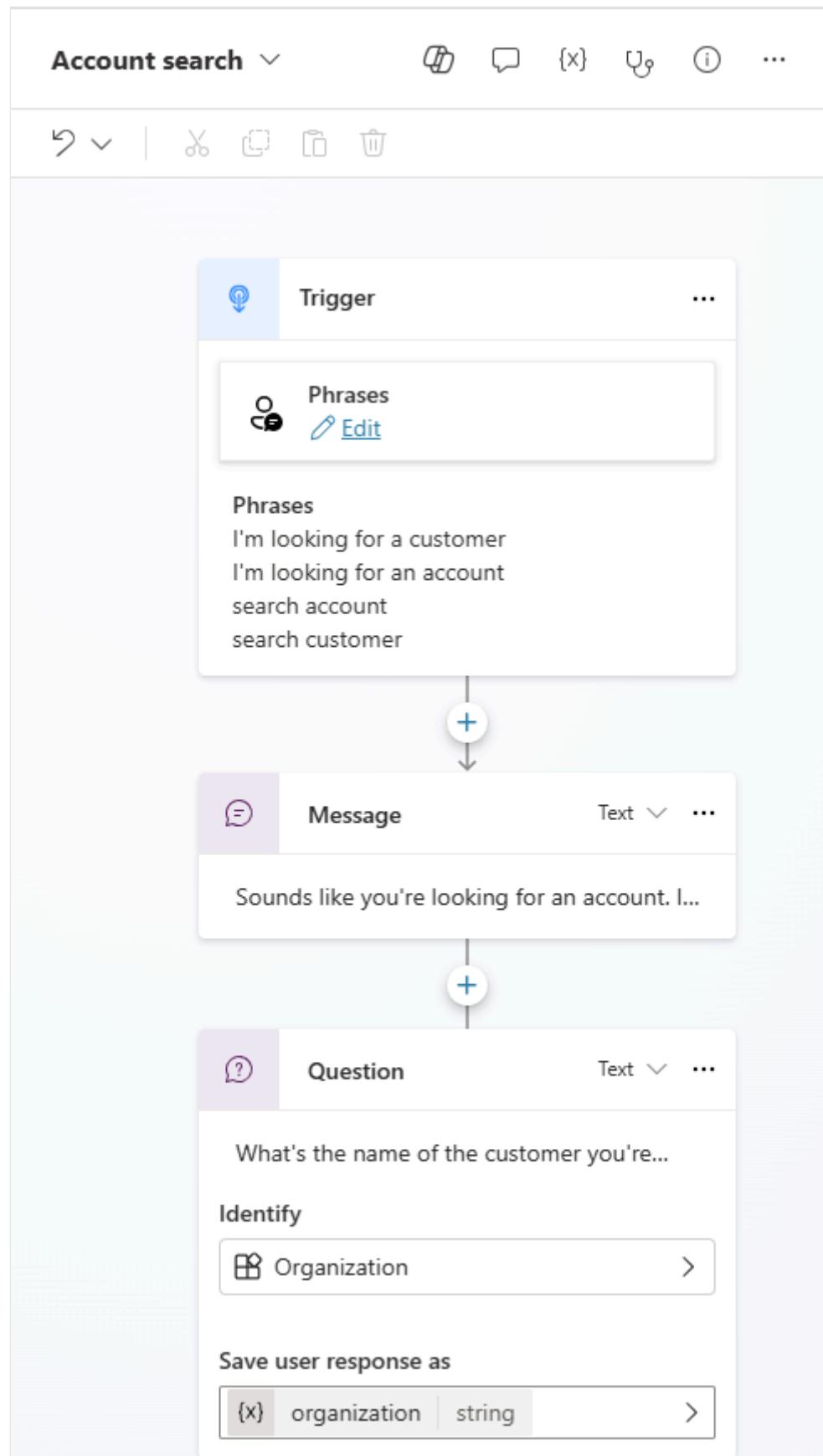
1. Go to the [Power Apps portal](#).
2. In the navigation pane, select **Tables**.
3. Select the **Account** table, and then on the table properties page, select **Edit**.
4. Add the following rows to the table:

 [Expand table](#)

| <b>Address 1: City</b> | <b>Account Name</b>   | <b>Account Number</b> |
|------------------------|-----------------------|-----------------------|
| Seattle                | Contoso Inc           | AC0125                |
| San Francisco          | Contoso Residences    | AC0255                |
| Olympia                | Contoso Meal Delivery | AC0035                |

## Create a topic

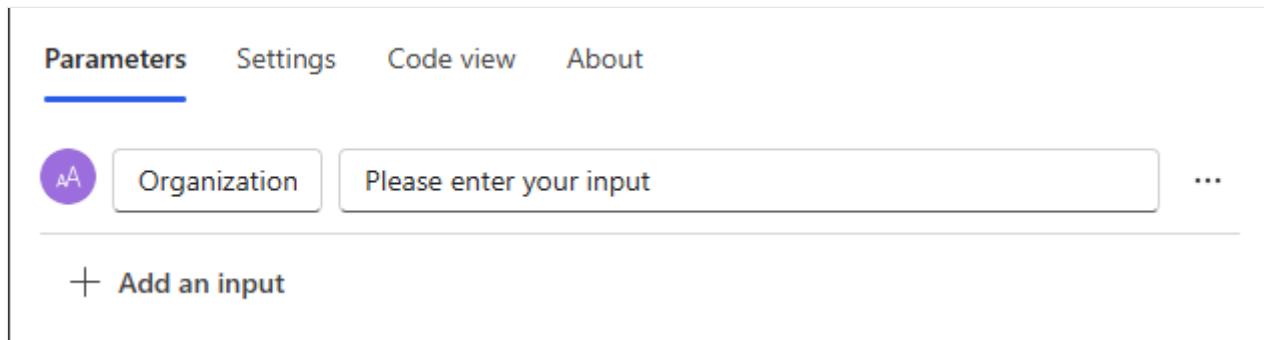
1. In Copilot Studio, go to the [Topics](#) page for your agent.
2. Create a topic called **Account search**.
3. Add the following trigger phrases:
  - I'm looking for a customer
  - I'm looking for an account
  - search account
  - search customer
4. Add a **Message** node and enter the message "Sounds like you're looking for an account. I can look that up for you."
5. Add a **Question** node and enter the message "What's the name of the customer you're looking for?"
6. For **Identify**, select **Organization**.  
  
 User's entire response would also work. However, by selecting the **Organization** entity, you take advantage of the agent's language understanding capabilities to extract the organization name from the response.
7. For **Save user response as**, rename the variable to "organization".



## Create a flow

1. Select the Add node icon below any node, and select Add an action.

2. On the **Basic actions** tab, select **New Agent flow**. The agent flow designer tab opens with a starting template for an agent flow called by an agent. The flow includes two actions: **When an agent calls the flow** and **Respond to the agent**.
3. Select **Save draft**.
4. On the **Overview** tab of the flow, edit the details for the flow, replacing the name with **Search account**. Select **Save**.
5. Return to the **Designer** tab. Select the **When an agent calls the flow** flow trigger and add a **Text input parameter** named **Organization**.



6. Select the **Add** icon below the **When an agent calls the flow** flow trigger.
7. In **Add an action**, search for "Microsoft Dataverse", and select the **Search rows** action.  
This action uses fuzzy matching to find relevant accounts in your [Dataverse Account table](#).
8. In the **Search term** box, enter a slash (/), select **Insert dynamic content**, and then select your **Organization** input parameter.
9. For **Advanced parameters**, select **Show all** and set the following values:
  - **Table filter Item:** `account`
  - **Sort by Item - 1:** `search.score desc`
  - **Sort by Item - 2:** `name asc`

The screenshot shows the 'Search rows (preview)' configuration screen in Power Automate. The interface includes a header with a logo, a search bar, and a help/ellipsis icon. Below the header, there are several input fields and sections:

- \* Search term:** Organization
- Search type:** Enter whether simple or full search syntax should be used (default is simple)
- Search mode:** Enter whether any or all of the search terms must be matched (default is any)
- Row count:** Enter the number of search results to be listed (default = 50)
- Row filter:** Enter an Odata style filter expression to narrow the search

Below these are three expandable sections:

- Table filter Item - 1:** Contains the value "account" and a "+ Add new item" button.
- Sort by Item - 1:** Contains the value "search.score desc" and a "... Edit" button.
- Sort by Item - 2:** Contains the value "name asc" and a "... Edit" button.

Another expandable section follows:

- Facet query Item - 1:** Contains the placeholder "Enter a facet query to narrow the search" and a "+ Add new item" button.

At the bottom are two more input fields:

- Skip rows:** Enter the number of search results to be skipped
- Return row count:** Choose an option (dropdown menu)

A "Hide advanced options" link is located at the bottom left.

## Format results

The **Search rows** action returns the **List of rows** variable, which contains JSON data. Before you can use the data, you must analyze it with the **Parse JSON** action.

1. Select the **Add** icon below the **Search rows** action.
2. In **Add an action**, search for "Data Operation", and select the **Parse JSON** action.
3. In the **Content** box, enter a slash (/), select **Insert dynamic content**, and then select **List of rows**.

4. Copy the following JSON schema and paste it in the **Schema** box:

JSON

```
{
 "type": "array",
 "items": {
 "type": "object",
 "properties": {
 "@@search.score": {
 "type": "number"
 },
 "name": {
 "type": "string"
 },
 "address1_city": {
 "type": "string"
 },
 "accountnumber": {
 "type": "string"
 }
 },
 "required": [
 "name"
]
 }
}
```

{v} Parse JSON

⋮ <>

Parameters Settings Code view About

Content \*

List of rows x

Schema \*

```
{
 "type": "array",
 "items": {
 "type": "object",
 "properties": {
 "@@search.score": {
 "type": "number"
 },
 "name": {
 "type": "string"
 },
 "address1_city": {
 "type": "string"
 },
 "accountnumber": {
 "type": "string"
 }
 },
 "required": [
 "name"
]
}
```

Use sample payload to generate schema

The screenshot shows the 'Parse JSON' action configuration in Microsoft Power Automate. The 'Content' field is set to 'List of rows'. The 'Schema' field displays the following JSON schema:

```
{
 "type": "array",
 "items": {
 "type": "object",
 "properties": {
 "@@search.score": {
 "type": "number"
 },
 "name": {
 "type": "string"
 },
 "address1_city": {
 "type": "string"
 },
 "accountnumber": {
 "type": "string"
 }
 },
 "required": [
 "name"
]
}
```

A blue link at the bottom of the schema editor says 'Use sample payload to generate schema'.

5. Select the **Add** icon below the **Parse JSON** action.
6. In **Add an action**, search for "Variable", and select the **Initialize variable** action.
7. In the **Name** box, enter "ListOfAccounts".
8. For **Type**, select **String**.

The screenshot shows the 'Initialize variable' screen in PowerApps. At the top, there's a purple button with the code '{x}' and the text 'Initialize variable'. To the right are three small icons: a vertical ellipsis, a double left arrow, and a double right arrow. Below this is a navigation bar with tabs: 'Parameters' (which is underlined in blue), 'Settings', 'Code view', and 'About'. The main area is titled 'Name \*' with a required asterisk. A text input field contains the value 'ListOfAccounts'. Below it is a 'Type \*' section with a dropdown menu set to 'String'. There is also a 'Value' section with an empty text input field.

9. Select the **Add** icon below the **Initialize variable** action.
10. In **Add an action**, search for "Control", and select the **Apply to each** action.
11. In the **Select an output from previous steps** box, enter a slash (/), select **Insert dynamic content**, and then select the **Body** variable under **Parse JSON**.
12. Select the **Add** icon *inside* the **Apply to each** action.
13. In **Add an action**, search for "Variable", and select the **Append to string variable** action.
14. For **Name**, select **ListOfAccounts**.
15. Copy the following text and paste it in the **Value** box:

```
PowerApps Formula
- @{items('Apply_to_each')['accountnumber']}: @{items('Apply_to_each')['name']} - @{items('Apply_to_each')['address1_city']}
```

16. Add a line break after the snippet to make each result appear on its own line.

The screenshot shows the 'Append to string variable' action configuration. At the top, there's a header with the action name and navigation icons. Below it, tabs for 'Parameters', 'Settings', 'Code view', and 'About' are visible, with 'Parameters' being the active tab. Under 'Name \*', the value 'ListOfAccounts' is entered. In the 'Value \*' section, three dynamic content items are listed: 'Body accountnu...', 'Body name', and 'Body address1\_city'. The 'Value' field has a dropdown arrow icon.

17. In the **Respond to the agent** action, add a **Text** output.
18. For **Name**, enter "FoundAccounts".
19. For the value, enter a slash (/) in the box, select **Insert dynamic content**, and then select the **ListOfAccounts** variable.

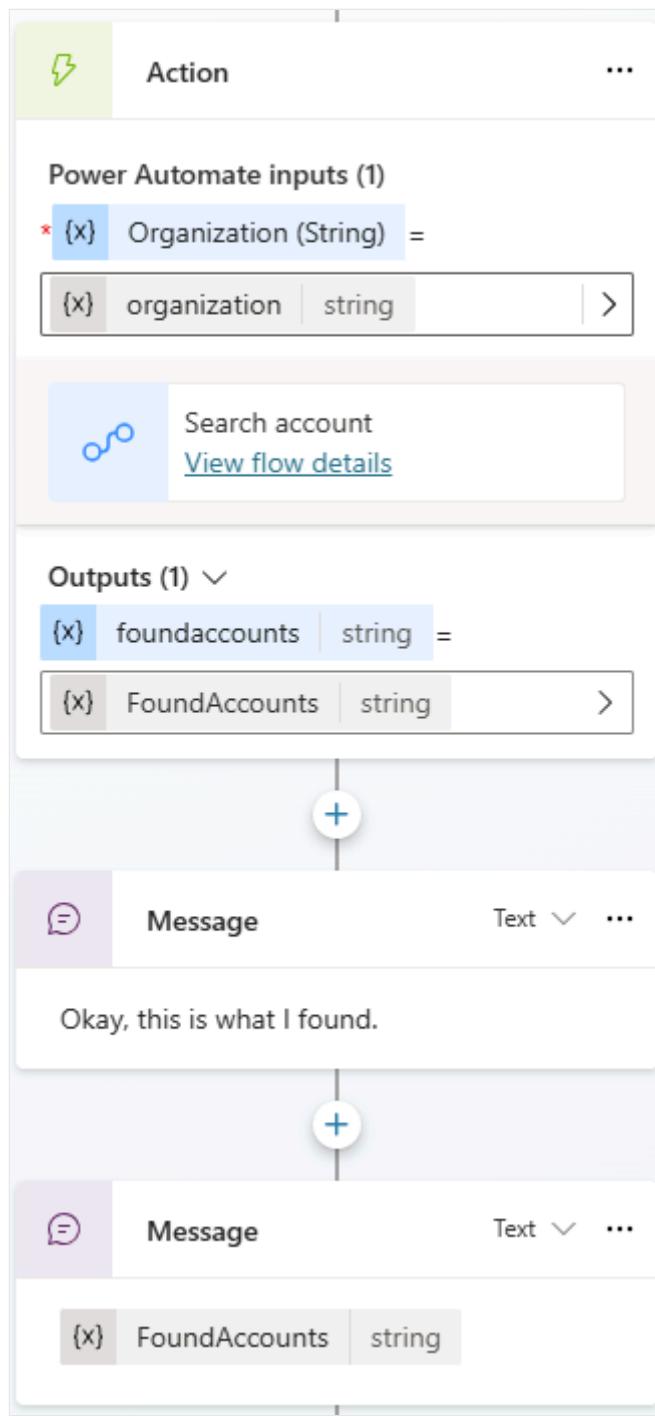
The screenshot shows the configuration for a 'Text' output. The 'Parameters' tab is selected. A list of outputs is shown, with 'FoundAccou...' and '{x} ListOfAccounts x' currently selected. An input field 'Enter a description of the output' is present. A button '+ Add an output' is at the bottom.

20. Save and publish your flow. Copilot Studio informs you when the agent flow is published.
21. Select **Go back to agent** to return to the agent topic editor. The new action flow is added to the topic.

## Call the flow from Copilot Studio

1. Select the new action node in the topic.
2. Set the flow input parameter to the output variable from the **Question** node: **Organization (String)** gets its value from the **organization** variable.
3. Add a **Message** node and enter the message "Okay, this is what I found."

4. Add a second Message node. Select the **Insert variable** icon, and then select **FoundAccounts**.



5. Select Save.

6. Test your agent.

▶ Test your agent

... C X

search accounts

A minute ago

Sounds like you're looking for an account. I can look that up for you.

What's the name of the customer you're looking for?

A minute ago

contoso

Just now

Okay, this is what I found.

- AC0125: Contoso Inc - Seattle
- AC0035: Contoso Meal Delivery - Olympia
- AC0255: Contoso Residences - San Francisco

Just now

Ask a question or describe what you need

0/2000



Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

# Extend your agent with tools from a REST API (preview)

06/04/2025

[This article is prerelease documentation and is subject to change.]

You can use REST APIs (including OpenAI API) to connect an agent you create with external systems and access available data for use within your agent. You can connect your agent to a REST API by providing Copilot Studio with three things:

- An OpenAPI specification defining the API's functions and available actions
- Details on the type of authentication needed and the authentication details for users to connect to the API to access the external system
- Descriptions to help the language model determine when to invoke the API to utilize the data

REST APIs can be added to Copilot agents and custom agents through Copilot Studio.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

Copilot agents allow a maker to combine multiple data sources like connectors, APIs, prompts, and knowledge sources into a single agent. You can use this agent to extend Microsoft-branded agent experiences like Microsoft 365 Copilot.

Custom agents are standalone agents that contain connectors, APIs, prompts, and knowledge sources. You can use custom agents directly by integrating them into websites or other channels.

## Note

REST API tools must be created from an OpenAPI v2 specification. This requirement is due to the behavior of the Power Platform in processing API specifications. If a v3 specification

is submitted, it's automatically translated to a v2 specification during the creation process.

# Prerequisites

A few prerequisites are needed to extend Microsoft-branded agent experiences with connector actions.

For all agent experience types, you must have maker-level credentials, and a Copilot Studio license.

You also need a copy of the OpenAPI specification, knowledge of the type of authentication needed, and the authentication details.

## Add a REST API action to an agent

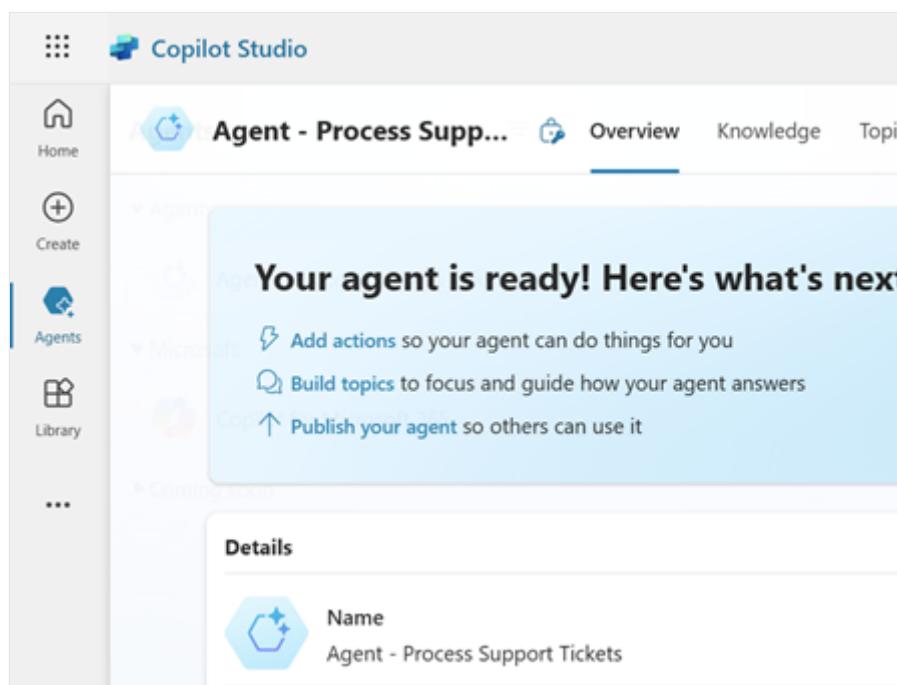
Adding a REST API to your agent involves a few steps. The following sections walk you through the process.

The process to add a REST API is identical for both custom and Copilot agents.

There are a few steps in the process.

### Add an action to the agent

1. Start at the **Overview** page of your agent.



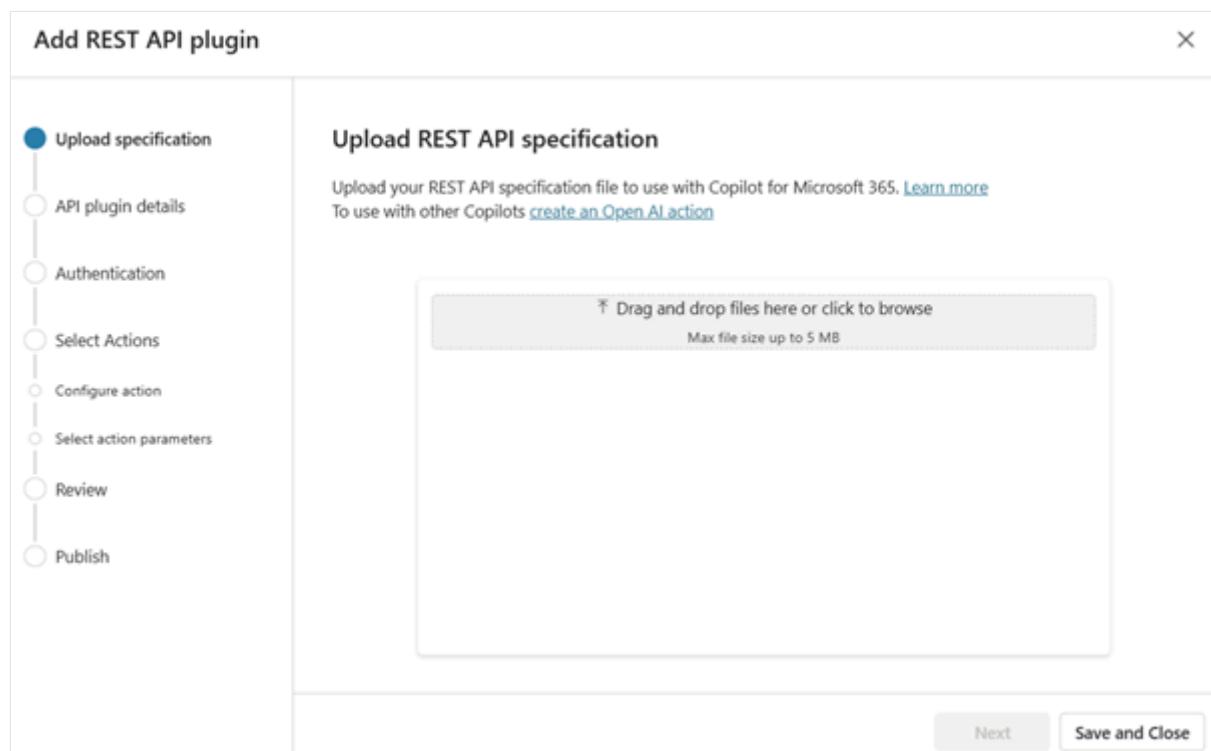
2. In the **Tools** section, select **Add tool**. You can also go to the **Tools** tab and select **Add a tool**.

The **Add tool** page is displayed.

3. Select **New tool > REST API**.

## Provide API specification, description, and solution

1. Upload an OpenAPI specification file for the REST API you want to connect to. You can either drag and drop the specification file into the **Upload a REST API** screen or browse your system to locate the file you wish to use.



### ⓘ Note

The OpenAPI specification must be a JSON file in v2 format. If a v3 specification is submitted, it's automatically translated to a v2 specification during the creation process.

After you upload the specification, the screen updates to indicate the specification file name and the details.

## Upload REST API specification

Upload your REST API specification file to use with Copilot for Microsoft 365. [Learn more](#)  
To use with other Copilots [create an Open AI action](#)

The screenshot shows the 'Upload REST API specification' step of a Copilot setup wizard. It includes fields for 'File Name' (set to 'swagger 3.json') and a 'Browse API specification' button. Below this is a preview area titled 'API specification' showing the first few lines of a JSON schema:

```
0. {
 1. "openapi": "3.0.0",
 2. "info": {
 3. "title": "SunnyADO Ticket Manager Service",
```

At the bottom right are 'Next' and 'Save and Close' buttons.

In the steps that follow, we ground the procedure in a specific example of SunnyADO, an ADO ticket management system. In the example, the intention is to allow the users to retrieve and update their tickets via the agent.

2. Verify the details, then select **Next**.

You're presented with an **API plugin details** page where you can provide additional information about the API.

The screenshot shows the 'API plugin details' step of the wizard. It has fields for 'Action name \*' (set to 'SunnyADO Ticket Manager Servic') and 'Description \*' (set to 'A simple service to manage tickets'). Below these is a 'Solution' section with a note: 'If no solution is selected, one will automatically be created for you.' A dropdown menu is shown with the placeholder 'Select a solution'. At the bottom right are 'Back', 'Next', and 'Save and Close' buttons.

The description field is initially populated based on the description in the API specification you uploaded. Provide a detailed description, because your agent

orchestration uses the description to determine when to use the particular tool. Provide details, including synonyms, to help your agent with the selection process.

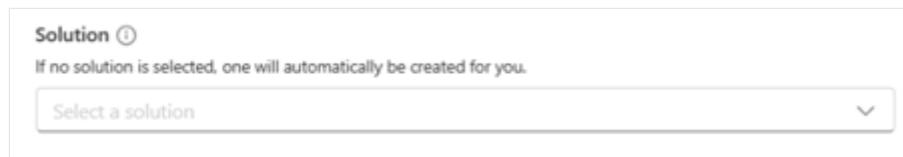
For example, the initial description provided is: "A simple service to manage tickets."

A better description is: "A system used to get, retrieve, find, and display existing tickets from SunnyADO. It allows users to update, change, and manage tickets to provide more data to improve the records."

3. Enter an improved description under the **Description** field.

4. Under **Solution**, a dropdown lists all solutions available within the current environment.

Select the solution you want to use. For more information on what solutions are, see [Solution concepts](#).



If you have a preferred solution, or your selected connector is already in the solution, that solution is automatically selected.

You can either select a solution or leave it blank. If you leave the solution blank, a solution is created for you with the action name and default publisher. Storing your action in a solution lets you move it easily across environments.

**① Note**

If you don't see the default solution or the CDS default solution as an option in this case as we recommend having a custom solution for easy management. For more information, see: [Default solution vs. custom solution](#).

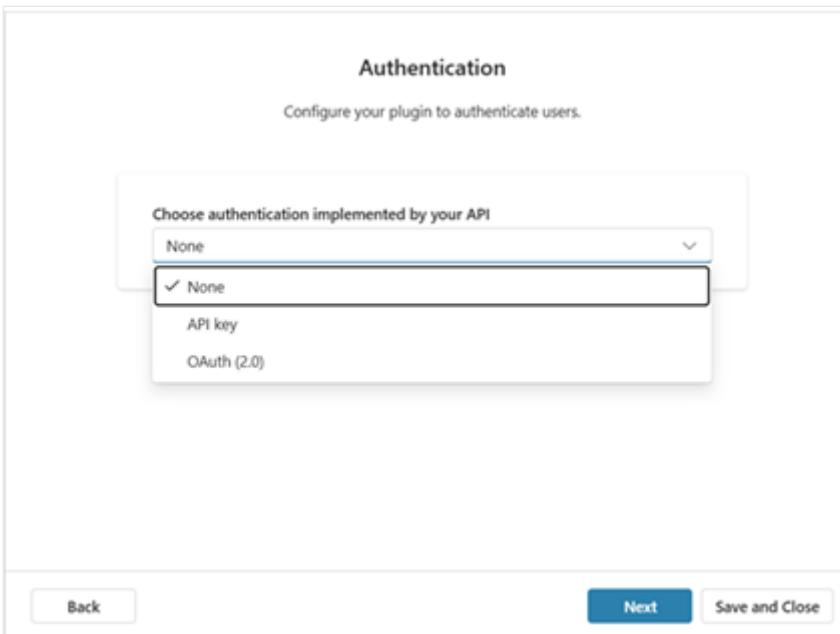
5. With a solution selected, select **Next** to proceed.

## Provide authentication details

The **Authentication** page is displayed, to select which type of authentication to use for the API.

**① Note**

Currently the available options are None, Auth 2.0, and API.



1. Select an authentication method from the list.
2. Fill in the required fields for the authentication method. The fields vary based on the authentication method.
  - **None:** No other fields are required.
  - **API key:**
    - Parameter Label: A text label for the API parameter.
    - Parameter Name: A text name for the API parameter.
    - Parameter Location: The position where the parameter can be found.
  - **Auth 2.0:**
    - Client ID: Client GUID for the target service.
    - Client Secret – Secret value for the client. The secret isn't displayed when the user opens up the edit panel afterwards. However, store the secret, because you'll need it if you choose to make further edits.
    - Authorization URL: URL used to authorize the source system.
    - Token URL: URL where the token can be retrieved.
    - Refresh URL: What URL you're redirected to in a refresh scenario.
    - Scope: The Scope URL assigned to the API for Microsoft Entra apps.
    - Which Microsoft 365 organization accesses the endpoints: This limits access to the source to either the organization of the maker, or all organizations.
    - Which app (Client) can use the endpoints: GUID that defines the client system that can be used to access this data. Apps could include Microsoft 365, Power Automate, and other options.

3. Once all fields are completed, select **Next**.

You're presented with a **Select and configure your plugin tool** page where you can select tools to enable for the API.

## Select and configure your plugin action

Please select the required action for Copilot for Microsoft 365.

Available actions (4)

 Search

 List all tickets

 Create a new ticket

 Delete an existing ticket

 Update an existing ticket

[Back](#)

[Next](#)

[Save and Close](#)

## Select tools for the API

Choose the API-supported tools to enable. Generally, a REST API offers a range of tools through the various combinations of endpoint and HTTP method (get, put, post, delete, and so on) defined in the API specification. In some cases, you might not want the agent's users to have the ability to execute *every* action the API generally offers. For example, your API specification might include actions to update and delete but you only want users of your agent to be able to create records.

1. Select a tool from the list to configure.

The **Configure your plugin action** page is displayed.

## Configure your plugin action

Select the connector action you wish to leverage and enter details to configure your plugin action.

The screenshot shows a configuration interface for a plugin action. At the top, there's a header "Configure your plugin action" and a sub-instruction "Select the connector action you wish to leverage and enter details to configure your plugin action." Below this, there's a list of available actions: "List all tickets". A specific action is selected: "Action name \* List all tickets". An "Action description \* ⓘ" field contains the text "Returns a list of tickets with their details. Find, locate, and get tickets or records.". At the bottom right are three buttons: "Back", "Next", and "Save and Close".

2. Configure the selected action. As with the overall API, you're asked to provide an **Tool name** and **Tool description**. Descriptions are initially prepopulated from the descriptions in the API specification. The name doesn't need to be unique but it should represent the tool itself. The description, as with the overall API description, should be specific enough to provide the language model with details to better identify that your query aligns with this specific tool.

3. Once the fields are completed, select **Next**.

The **Review your tool's parameters** page is displayed.

## Review your action's parameters

Review your input and output values and update descriptions as needed

The screenshot shows a configuration interface for a plugin tool. At the top, there is a title bar with the text "Review your action's parameters" and a subtitle "Review your input and output values and update descriptions as needed". Below this, there is a section titled "List all tickets" with a green icon. The main area contains two tables: one for inputs and one for outputs.

| Input name | Input description *                            |
|------------|------------------------------------------------|
| text       | Filter tickets by text in title or description |
| assignedTo | Filter repairs by who they're assigned to      |
| status     | Filter repairs by their status                 |

| Output name | Output description *                |
|-------------|-------------------------------------|
| id          | The unique identifier of the ticket |
| title       | The short summary of the ticket     |

At the bottom of the form are three buttons: "Back" (gray), "Next" (blue), and "Save and Close" (gray).

This page shows the values provided as part of the possible input and output values.

These values can't be changed, however, the *descriptions* of the inputs and outputs can be updated. All content in this page is pulled directly from the uploaded API specification.

4. Fill in values as needed for the descriptions. The descriptions provide a definition of what the values are used for. If any of the descriptions are blank, they must be completed before you can move forward. You can paste in the name if you don't have a better description.
5. After completing the descriptions, select **Next**.

The first tool is now configured and appears in the list of **Selected tools** on the **Select and configure your plugin tool** page.

## Select and configure your plugin action

Please select the required action for Copilot for Microsoft 365.

### Selected actions (1)



List all tickets

### Available actions (3)



Search



Create a new ticket



Delete an existing ticket



Update an existing ticket

Back

Next

Save and Close

6. Add in any other actions you wish to include at this time. Once you're done adding tools you want your agent to support, select **Next**.

The **Review your tool** page is displayed. The page provides the details of the configured REST API tool.

## Review your action

Almost done! Make sure you review your action's details before creating it.

### Action details

#### Action name

SunnyADO TMS – Get, Update

#### Action description

A system used to get, retrieve, find, and display existing tickets from SunnyADO. It will allow users to update, change, and manage tickets to provide additional data to improve the records.

#### Selected actions



List all tickets



Update an existing ticket

Back

Next

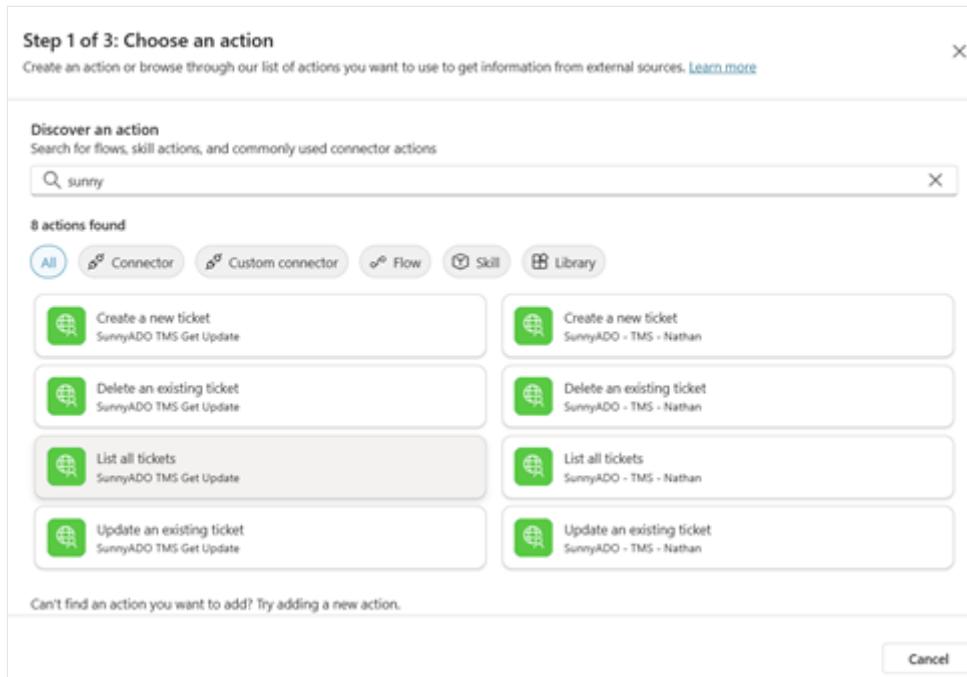
Save and Close

## Review and publish

1. If you need to make any updates, you can select **Back** and make your changes. Otherwise, select **Next**.

A screen is presented indicating that your tool is being published while the process is being completed. You're informed once the publish is complete.

2. After publishing completes, you're returned to the **Choose a tool** screen. Here you can add the newly configured REST API into your Copilot agent or custom agent and complete the configuration of the component.



The REST API tool is now available for use in your agent.

### 💡 Tip

To more easily find your tool, use the Search bar to locate it.

# Extend your agent with Model Context Protocol

10/23/2025

Another way to extend your agent with tools is via Model Context Protocol (MCP).

## What is Model Context Protocol?

[Model Context Protocol \(MCP\)](#) allows users to connect with existing knowledge servers and data sources directly within Copilot Studio. Connecting to an MCP server gives you access to:

- Resources: File-like data that an agent can read for more context (like API responses or file contents)
- Tools: Functions a language model can call to perform an action
- Prompts: Predefined prompt templates to accomplish specific tasks

 Note

Copilot Studio currently supports MCP tools and resources.

## How does MCP work?

Each tool or resource published by a connected MCP server is automatically made available for use in Copilot Studio. Name, description, inputs, and outputs are inherited from the server. As tools and resources are updated or removed on the MCP server, Copilot Studio dynamically reflects these changes, ensuring users always have the latest versions and that obsolete tools and resources are removed. A single MCP server can integrate and manage multiple tools and resources, each accessible for use by Copilot Studio agents.

When you connect to a Non-Microsoft Product, including an external MCP server, you're responsible for the tools and resources you access from within Copilot Studio.

 Note

Generative Orchestration must be enabled to use MCP.

# What is the process to extend an agent using MCP in Copilot Studio?

To integrate MCP in Copilot Studio:

1. [Connect your agent to an existing MCP server](#) based on a YAML schema template.
2. [Create an MCP server](#) if you don't already have an MCP server.
3. [Add MCP server tools and resources to your agent](#) so that your Copilot Studio agent can use them.
4. (Optional) [Publish your MCP connector](#) to allow the connector to be used across tenants.

For more information on troubleshooting MCP integration, see [Troubleshooting Model Context Protocol \(MCP\) integration](#).

## Related content

- [Model context protocol ↗](#)
- [Introduction to the Model Control Protocol ↗](#)
- [Core MCP concepts ↗](#)
- [Microsoft Copilot Studio - MCP Lab ↗](#)

# Connect your agent to an existing Model Context Protocol (MCP) server

If you have a Model Context Protocol (MCP) server already set up, you can connect the MCP server to your agent.

There are two ways to connect your agent to an MCP server in Copilot Studio:

- Add the MCP server in Copilot Studio using the *MCP onboarding wizard* (recommended)
- Create a custom connector to your server via Power Apps

If you don't yet have an MCP server set up, see [Create a new MCP server](#) for information on how to get started.

## Supported transports

In MCP, [transports](#) are the foundation for client-server communication. Transports handle the mechanics of sending and receiving messages. Currently, Copilot Studio supports the Streamable transport type.

### Note

Given that SSE transport is [deprecated](#), Copilot Studio no longer supports SSE for MCP after August 2025.

## Option 1: Use the MCP onboarding wizard (recommended)

The simplest way to connect to an existing MCP server is directly within Copilot Studio using the MCP onboarding wizard.

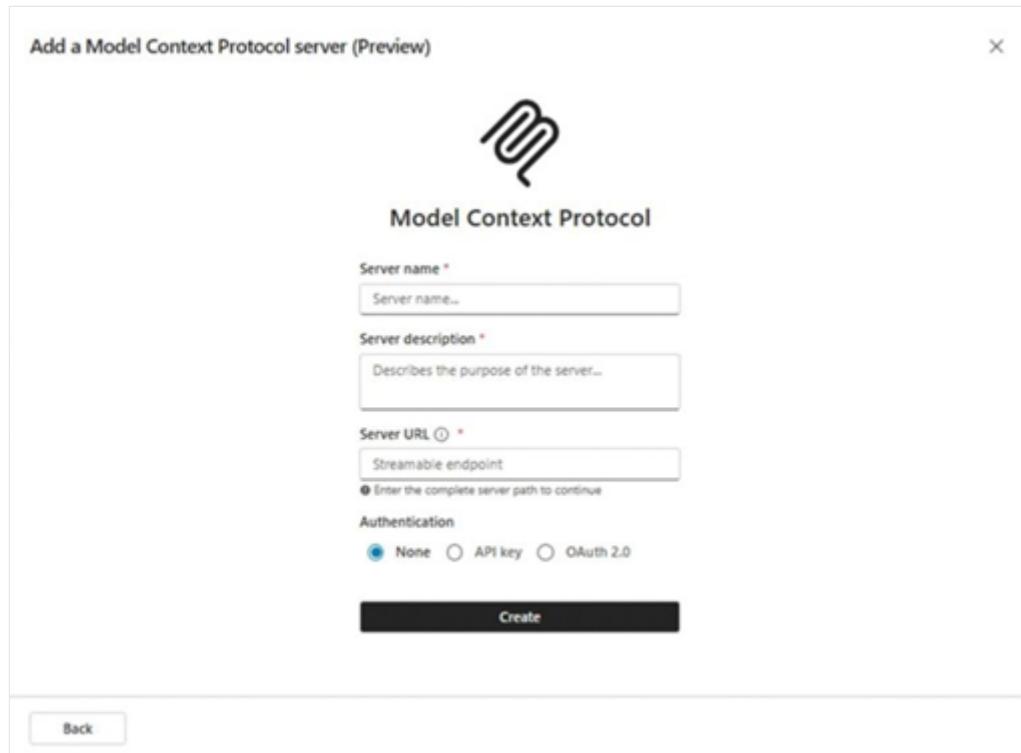
There are two parts to this method:

- Configure the basic MCP server details
- Configure authentication with your MCP server

### Configure basic server details

1. Go to the **Tools** page for your agent.

2. Select **Add a tool**.
3. Select **New tool**.
4. Select **Model Context Protocol**. The MCP onboarding wizard appears.



5. Fill in the required fields **Server name**, **Server description**, and **Server URL**. Make sure to write a brief, but clear description of what your MCP server does. The agent orchestrator uses this information to identify whether to call your server at runtime.
6. Select the authentication type for your MCP server, if applicable. You have three options:
  - **None**
  - **API key:** [Configure API key authentication](#)
  - **OAuth 2.0:** [Configure OAuth 2.0 authentication](#)
7. If you selected **None**, select **Create**. The **Add tool** dialog appears. Skip to [Create a new connection and add the MCP server to your agent](#).

## Configure authentication with your MCP server

If your server requires authentication, you have two options:

- **API key:** Select this option if your MCP server requires an API key for authentication. An API key is a simple option when only your application (agent) needs to authenticate with the server.

- **OAuth 2.0:** Select this option if your MCP server uses OAuth 2.0 for authentication. OAuth 2.0 lets individual users authenticate with the server and grant permissions to your application (agent) without sharing their credentials.

If you choose to use authentication, you need to configure the authentication details. The steps depend on the authentication type you select.

## Configure API key authentication

1. Select **API key** as the authentication type. More fields appear for you to configure the API key.
2. Select the **Type** of API key to use. You have two options:
  - **Header:** Select this option if your MCP server requires the API key to be sent in the request header.
  - **Query:** Select this option if your MCP server requires the API key to be sent as a query parameter in the URL.
3. Enter the appropriate header or query string for the API key.
4. Select **Create**. The **Add tool** dialog appears. Skip to [Create a new connection and add the MCP server to your agent](#).

## Configure OAuth 2.0 authentication

1. Select **OAuth 2.0** as the authentication type.
2. Select the **Type** of OAuth 2.0 authentication to use. You have three options:
  - **Dynamic discovery:** Select this option if your MCP server supports the OAuth 2.0 dynamic client registration (DCR) with discovery mechanism. The client can use a discovery endpoint to automatically discover the necessary endpoints and register itself with the identity provider.
  - **Dynamic:** Select this option if your MCP server supports dynamic OAuth 2.0 DCR, but doesn't support the dynamic discovery mechanism. The client can still register itself with the identity provider, but you need to provide the necessary endpoints manually.
  - **Manual:** Select this option if your MCP server requires manual configuration of OAuth 2.0 settings.

## Dynamic discovery

If your MCP server supports the OAuth 2.0 Dynamic Client Registration (DCR) mechanism, you can use the dynamic discovery option. If supported, DCR with discovery is the simplest method to configure OAuth 2.0 authentication with your MCP server in Copilot Studio.

Add a Model Context Protocol server (Preview)

Server name...

Server description \*

Describes the purpose of the server...

Server URL ⓘ \*

Streamable endpoint

● Enter the complete server path to continue

Authentication

None  API key  OAuth 2.0

Type

Dynamic discovery  Dynamic  Manual

Your MCP server must support OAuth 2.0 Dynamic Client Registration (DCR) to enable Dynamic Discovery Authentication.

If DCR is not supported, authentication may fail. Please verify that your server exposes a valid registration\_endpoint in its OpenID configuration. [Learn more](#)

Create

1. Select **Dynamic discovery** as the OAuth 2.0 authentication type.
2. Select **Create** to add the server. The client uses the discovery endpoint to automatically find the necessary endpoints and register itself with the identity provider.
3. Select **Next** to continue. The **Add tool** dialog appears.
4. Continue to [Create a new connection and add the MCP server to your agent](#).

## Dynamic

1. Select **Dynamic** as the OAuth 2.0 authentication type.
2. Fill in the required fields:
  - **Authorization URL:** The URL of the identity provider server where the client registration and authorization endpoints can be accessed.
  - **Token URL template:** The endpoint where your agent exchanges an authorization code (or refresh token) for an *access token* and *refresh token*. The access token lets your agent use the MCP server on behalf of the user. Refresh tokens let your agent get new access and refresh tokens from the refresh endpoint when the previous access token expires.

3. Select **Create** to add the server. Depending on the configuration, a *callback URL* might appear. The callback URL is where the identity provider responds with the authorization code once the user signs in and grants permissions to your agent.
4. If you see the callback URL, copy the callback URL to add to your application's registration with your identity provider.
5. Select **Next** to continue. The **Add tool** dialog appears.
6. Continue to [Create a new connection and add the MCP server to your agent](#).

## Manual

1. Select **Manual** as the OAuth 2.0 type. More fields appear for you to configure the OAuth 2.0 settings.
2. Fill in the required fields:
  - **Client ID:** The client identifier the identity provider issues when you register your app. The client ID lets the identity provider know what app is making the request.
  - **Client secret:** The client secret the identity provider issues when you register your app. Your agent sends the client secret along with the client ID to prove your agent is authorized to request access tokens for the MCP server.
  - **Authorization URL:** The identity provider endpoint where your agent redirects the user to sign in and grant permissions to your agent (consent card presented in the agent chat). The user authenticates here and then the identity provider responds back to the agent at the *callback URL* with an *authorization code*.
  - **Token URL template:** The endpoint where your agent exchanges the authorization code (or refresh token) for an *access token* and *refresh token*. The access token lets your agent use the MCP server on behalf of the user. Refresh tokens let your agent get new access and refresh tokens from the refresh endpoint when the previous access token expires.
  - **Refresh URL:** The endpoint to request a new access token using a refresh token (so that the user doesn't have to sign in again when the token expires).
  - **Scopes (Optional):** The permissions your app is asking for, as a space-separated list.
3. Select **Create** to add the server. A *callback URL* appears. The callback URL is where the identity provider responds with the authorization code once the user signs in and grants permissions to your agent.
4. Copy the callback URL to add to your application's registration with your identity provider.
5. Select **Next** to continue. The **Add tool** dialog appears.

6. Continue to [Create a new connection and add the MCP server to your agent](#).

## Create a new connection and add the MCP server to your agent

1. On the **Add tool** dialog, select **Create a new connection** for your MCP server or use an existing one.
2. Select **Add to agent** to finish adding the MCP server to your agent.

## Option 2: Create a custom MCP connector in Power Apps

You can create a custom connector in Power Apps manually to configure a connection to the server.

To carry out this procedure, you need a schema file for your MCP server. The schema file is an OpenAPI specification YAML file that describes the API of your MCP server.

For guidance on what the specification file should look like, check out some of the provided [MCP server schema example](#).

1. Go to the **Tools** page for your agent.
2. Select **Add a tool**.
3. Select **New tool**.
4. Select **Custom connector**. You're taken to Power Apps to create a new custom connector.
5. Select **New custom connector**.
6. Select **Import OpenAPI file**.
7. Navigate to your schema file and select **Import** to import the file.
8. Select **Continue** to complete the setup in Power Apps. You can read more about the setup process in the Power Apps documentation at [Import the OpenAPI definition](#).

## MCP server schema example

Here's a sample OpenAPI schema file for an MCP server using fictional data, in YAML format. You need to fill in the details for your own MCP server. This sample uses the Streamable

transport type.

### YAML

```
swagger: '2.0'
info:
 title: Contoso
 description: MCP Test Specification, YAML for streamable MCP support in Copilot Studio
 version: 1.0.0
host: contoso.com
basePath: /
schemes:
 - https
paths:
 /mcp:
 post:
 summary: Contoso Lead Management Server
 x-ms-agic-protocol: mcp-streamable-1.0
 operationId: InvokeMCP
 responses:
 '200':
 description: Success
```

## Edit an MCP connection

To edit your MCP connector or add custom parameters:

1. Go to the Power Apps or Power Automate portal and select **Custom connectors**.
2. Locate your connector file in the list of connectors and make the necessary updates using one of the available methods.

---

Last updated on 10/25/2025

# Create a new Model Context Protocol (MCP) server

10/18/2025

You can use [MCP software development kits \(SDKs\)](#) to set up an MCP server in one of the supported languages.

If you already have an MCP server set up, see [Add an existing Model Context Protocol \(MCP\) server to your agent](#) for information on how to add the server to your agent.

## Authentication support

When you create an MCP server, you can choose to implement authentication or not. If you choose to implement authentication, you can use one of the following methods:

- **API key:** A simple way to secure your server by making your application include a key with requests.
- **OAuth 2.0:** A more robust authentication method that lets individual users grant access to their data without sharing their credentials with the agent.

Choose the method that best fits your needs and follow the implementation guidelines for that method.

You must register your application with an identity provider to obtain the necessary client credentials. The credentials are either an application key for API key authentication or a client ID and client secret for OAuth 2.0 authentication.

You must provide the authentication credentials from your identity provider when you add the MCP server to your agent in Copilot Studio. For details, see [Configure authentication with your MCP server](#).

If you use OAuth 2.0 authentication, you receive a callback URL from Copilot Studio after you add the MCP server. You need to update your app registration with your identity provider to add this URL. This URL is where the identity provider responds once the user signs in and grants permissions to your agent.

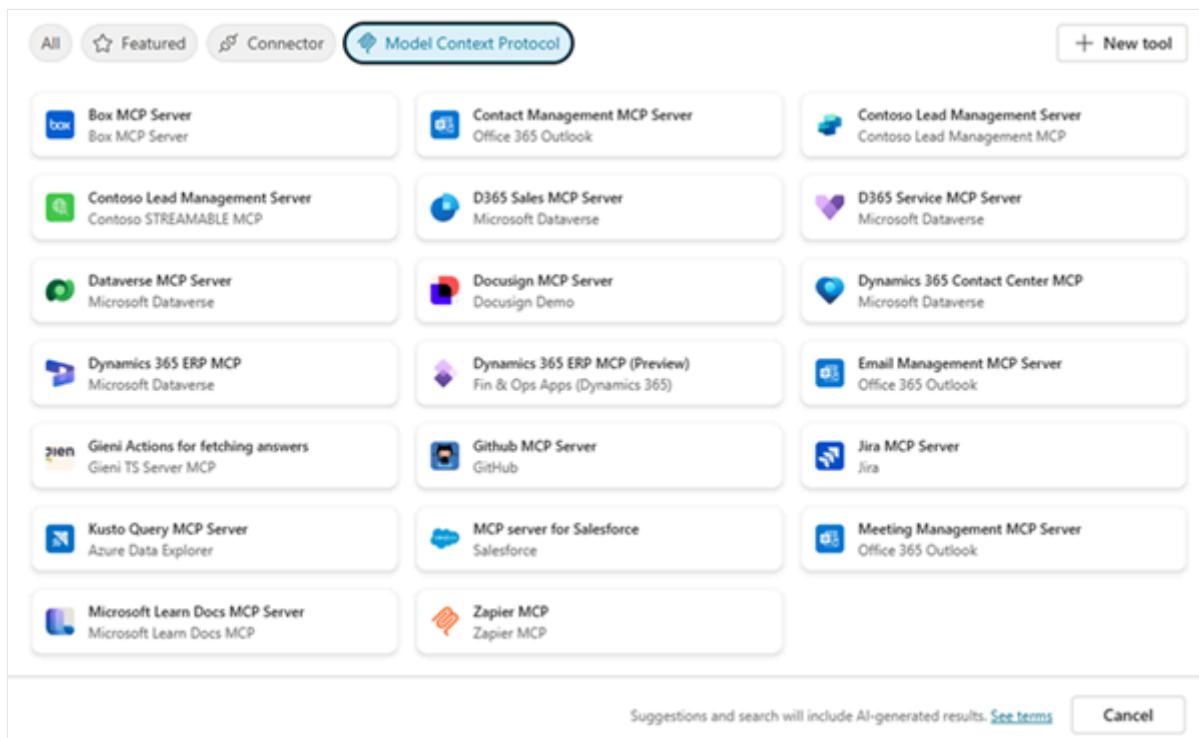
# Add tools and resources from a Model Context Protocol (MCP) server to your agent

10/25/2025

Once an existing MCP server is connected to Copilot Studio, you can add the server to an agent as a tool, and access server tools and resources. The process is the same, whether for prebuilt Microsoft MCP connectors or for an MCP server to which you configured a connection.

To add an MCP server to an agent, follow these steps:

1. Go to the **Tools** page for your agent.
2. Select **Add a tool**.
3. Select **Model Context Protocol**. A list of available MCP connectors is displayed.



4. Select the desired MCP connector from the list.
5. Authorize the connection, entering any information that is needed.
6. When you're done, select **Add to agent** or **Add and configure** to proceed.

The first option simply adds the MCP server and its tools and resources to the agent. The second adds the server and its tools and resources while also giving you an opportunity to configure details at the same time.

The MCP server is now connected to your agent.

## View tools and resources in an existing MCP server

Each MCP server in Copilot Studio is a tool that you can add to your agent. Once you add the MCP server to your agent as a tool, the server appears as an tool on the agent's **Tools** tab.

The MCP server provides access to all the *MCP tools and resources* offered by the server. To view information about the MCP tools and resources that come with an MCP server:

1. Go to the **Tools** tab for your agent.
2. Select the MCP server from the list of tools.

A settings page for the MCP server appears.

From the settings page, you can view and configure details related to your agent's use of the MCP server. As with any agent tool, you can view and update **Details** and **Inputs**.

For MCP connectors, you can view two other MCP-specific sections with details on functionality offered by the server:

- **Tools:** A list of names and descriptions for the MCP tools offered by the server.
- **Resources:** A list of names and descriptions for a sampling of the MCP resources offered by the server.

### **Important**

For a resource to be used by Copilot Studio agents, the MCP server owner needs to configure the resource as an output of one of the MCP tools on the server.

## Customize tool selection from an MCP server in your agent

If you don't want to use all the tools offered by an MCP server, you can selectively turn off MCP server tools directly in your agent.

All tools are turned on by default when you add an MCP server. An **Allow all** toggle is turned on.

To turn individual tools on and off:

1. Turn off the **Allow all** toggle. Toggles become available for each of the individual tools.

2. Use the individual toggles to turn off tools that aren't needed to ensure your agent only uses the most relevant features.

[!NOTE> When you turn off **Allow all**, any new tools added to the MCP server are turned off by default.

# Copilot Studio MCP Catalog

08/12/2025

For cross tenant consumption and to make your MCP server available in our Copilot Studio catalog, see [Certifying your connector](#).

The following are MCP connectors currently available in Copilot Studio:

- Dataverse
- Dynamics 365
  - Sales
  - Finance
  - Supply Chain
  - Service
  - Enterprise Resource Planning (ERP)
  - Contact Center
- Fabric
- Office 365 Outlook
  - Contact Management
  - Email Management
  - Meeting Management
- Kusto Query
- GitHub
- Learn MCP
- Box.com
- Gieni Actions for fetching answers

# Troubleshooting Model Context Protocol (MCP) integration

10/23/2025

This article provides troubleshooting guidance for integrating your agent with Model Context Protocol (MCP) servers.

## Known issues and workarounds

The following are identified issues the product team is aware of and working on:

  Expand table

| Issues                                                                                                                                                                               | Resolution/Workaround                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Currently, the endpoint returned in the <a href="#">Open SSE connection</a> call must be a full URI.                                                                                 | Not applicable.                                |
| A <code>System.FormatException</code> is thrown when the <code>exclusiveMinimum</code> property is set to an integer (instead of a Boolean) in the MCP tool definition input schema. | Not applicable.                                |
| The MCP tool definition input schema definition is truncated when a type in a tool definition is an array of multiple types instead of a single type.                                | Make the values for type fields a single type. |
| Tools with reference type inputs in the schema are filtered from the list of available tools for MCP server. Reference type inputs and outputs aren't supported.                     | Not applicable.                                |
| Tools with enum type inputs in the schema are interpreted as string instead of enum.                                                                                                 | Not applicable.                                |

# Automate web and desktop apps with computer use (preview)

[This article is prerelease documentation and is subject to change.]

Computer use is a tool in Copilot Studio that lets your agent interact with and automate tasks on a Windows computer. It works with websites and desktop apps by selecting buttons, choosing menus, and entering text into fields on the screen. Describe in natural language what you want computer use to do, and it performs the task on a computer you set up by using a virtual mouse and keyboard. With computer use, agents can complete tasks even when there's no API to connect directly to the system. If a person can use an app or website, computer use can too. You can use computer use for tasks like automated data entry, invoice processing, and data extraction.

## i Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

Computer use is powered by Computer-Using Agents (CUA), an AI model that combines vision capabilities with advanced reasoning to interact with graphical user interfaces (GUIs). Because it's AI-powered, it adapts to interface changes. For example, when buttons or screens change, the tool continues working without breaking your flow. It's simple to use. Just describe what you want in natural language. You don't need to write code.

Watch this video to learn how the computer use tool lets an agent interact with a web application.

<https://learn-video.azurefd.net/vod/player?id=373502d9-940f-48db-aa3d-b5f2463a5618&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Fcomputer-use>

## Requirements

- The feature is available for environments where the region is set to United States.
- Make sure your agent has the [generative orchestrator enabled](#) to use computer use.

## Add computer use to your agent

On a new agent or an existing agent in Copilot Studio, add computer use as a tool by following these steps:

1. Go to the **Tools** section in your agent and select **Add tool**.

2. In the **Add tool** dialog, select **New tool**.

3. Select **Computer use**.

4. Provide the instructions that describe the task the tool should perform on the computer. You see some instruction templates to get started with. To learn more about how to best write instructions for computer use, check [best practices for instructions for computer use](#).

5. Choose the machine where computer use runs. You can either use the hosted browser, a ready-to-use machine for automating tasks on public-facing websites, or select a Windows machine that you can configure to use for computer use. Learn more in [Configure where computer use runs](#).

6. Select **Add and configure**.

7. On the configuration page, configure these three fields:

- **Name:** Enter the display name for the computer use tool. This name helps you differentiate it from other tools you add to your agent.
- **Description:** Enter a short description of what this tool does and when to use it. This text lets your agent know when to use this tool.
- **Instructions:** List the steps the tool should perform, including URLs and application names.  
For tips, see [Best practices for instructions for computer use](#).

8. Review these other fields and settings on the configuration page that might be relevant:

- **Inputs:** Use Inputs to define dynamic values that change each time computer use runs. For example, if you want to fill out a form with a different value on every run, create an input for that field. At execution time, computer use combines your instructions with the input values to complete the task.
- **Stored credentials:** Define the credentials that computer use uses to sign in to websites and applications. During execution, if a sign-in prompt appears, computer use securely uses any credentials you defined in this section for that site or application. Password values of these credentials are stored in an Azure Key Vault that you provide. Learn how to create an Azure Key Vault in [Create a key vault using the Azure portal](#).

To configure credentials in the tool, first enter the subscription ID, resource group name, and Key Vault name. All this information is available from the [Overview](#) page of your Key Vault.

To use Azure Key Vault secrets with Power Platform:

- The Azure subscription that has the vault must have the "PowerPlatform resource provider" registered.
- The user who creates the environment variable must have appropriate permissions to the Azure Key Vault resource.

If you haven't already, follow the steps in [Configure Azure Key Vault](#). Then provide the following details:

- **Name:** the display name for the credential.
- **Username:** the username used to sign in to the target website or application.
- **Login domain:** the domain where the credentials should be entered (for example, `login.microsoft.com`). Make sure to verify this domain, as it might differ from the main site URL.
- **Azure secret name:** the name of the secret in the Key Vault that stores the password for the website or application.

 **Note**

Password fields are supported on all websites and most Windows applications (WinForms, WPF, UWP, WinUI, Win32), which covers most customer scenarios. Some app types, such as Electron, Java, Unity, games, command-line interfaces, Citrix, or other virtualized environments, might not be supported.

- **Access control:** By default, computer use can operate on any website or application. If you want to restrict this access, enable access control to define the specific URLs and desktop applications that computer use should be limited to. You can configure both websites and applications:
  - **Websites:** Enter the main website address (for example, `example.com`). All pages on that website are included automatically. You can also use wildcards (\*) for subdomains.
    - Examples: `www.contoso.com`, `*.contoso.com`, `contoso.com`
  - **Desktop applications:** Enter the application product name or process name. To find it, press Ctrl + Shift + Esc to open Task Manager. Then check the **Processes** tab.
    - Examples: `Microsoft Edge`, `msedge`, `Notepad`.

 **Note**

Access control only prevents the model from taking actions on websites or applications that aren't in the allowlist. It doesn't stop the model from opening them. For example, if only `microsoft.com` and `Microsoft Edge` are in the allowlist, the model can still use the Edge search bar to open Bing. However, once Bing is opened, any attempt to interact with it fails because it isn't in the allowlist.

- **Machine:** This value is the name of the machine where computer use runs. Select **Manage machine** to view more machine settings.
- **Connection:** This value is the connection for this tool. Update the connection to change the credentials and machine for computer use.
- **Authentication:** Specify how computer use authenticates during execution:

- **Copilot author authentication** (default): This option uses the maker's credentials and is suitable for autonomous agents.

 **Warning**

If you [share](#) an agent with this setting, anyone using it can act with the original author's access on the configured machine.

- **User authentication:** This option uses the credentials of the person interacting with the agent. Each user must have access credentials to the machine.

9. Select **Save**.

## Configure where computer use runs

To interact with websites and desktop apps for task automation, computer use runs on a Windows machine. When you add computer use to an agent, you have the following options:

### Hosted browser (powered by Windows 365)

The hosted browser lets you get started quickly without any machine setup. It provides both web automation using Microsoft Edge and access to built-in Windows applications.

The hosted browser runs in a Microsoft-managed environment. It isn't Microsoft Entra joined to your tenant or managed by your Intune policies. It's designed for quick web automation and early experimentation, but it doesn't support enterprise resource access, custom desktop apps, or organization-specific device management.

 **Note**

The hosted browser lets you get started quickly with computer use. However, it's not recommended for production use. Usage might be throttled based on demand. For more information, see [Hosted browser limitations](#).

### Machine

This option lets you use Windows machines that you own and manage. You register and manage machines in Power Automate.

Here are the key requirements to use a machine for computer use:

- Install [Power Automate for desktop](#) version 2.59.169.25241 or later. Include the Power Automate web extension installation for web browser interactions.
- After registering the machine, [turn on computer](#) use in the machine settings.

Learn more in [prerequisites and limitations](#).

## Register a new machine

Because computer use runs on the machine, dedicate a machine to computer use to avoid interruptions.

### ⓘ Note

If your environment has [Power Automate Hosted Process](#) capacity, create a hosted machine that already has Power Automate for desktop installed and automatically registered to the environment.

To register a new machine:

1. Install the [latest version of Power Automate](#) on your machine. During installation, make sure you check the **Install the machine-runtime app to connect to the Power Automate cloud portal** option.
2. Open the **Power Automate machine runtime** app and sign in.
3. Register the machine to the environment you want to use with computer use.

Learn more about registering machines in [register a new machine](#).

## Enable computer use

After you register your machine, enable it for computer use.

1. Sign in to [Power Automate](#).
2. Go to **Machines**.
3. Select the machine you registered.
4. On the machine detail page, select **Settings**.
5. Turn on **Enable for computer use**.
6. Select **Save**.

## Run scheduling and queuing

When a **Machine** is in use, run requests targeting the same machine are **queued** and executed sequentially:

1. The first run executes on the target machine.
2. Subsequent runs are added to the queue and marked as **Queued**.
3. When a run completes, the next run starts and is marked as **Next to run**.

To monitor the run queue:

1. Sign in to [Power Automate](#).
2. Go to **Machines**.

3. Select the machine.
4. On the machine detail page, select **Run queue**.

## Test computer use

Testing computer use is a key step in the authoring journey. After you enter a name, description, and instructions, and save the tool, select **Test** to start the test experience.

After a short loading period, the test experience appears:

- The left panel shows your instructions and a step-by-step log of the tool's reasoning and actions.
- The right panel shows a preview of the actions on the machine you set up for computer use.

When the task finishes, you see a **Test completed** message. While the test is in progress, you can select **Stop testing** to immediately stop all actions on the machine.

If the result isn't what you expect, go to the configuration page and refine your instructions. Add more detail to improve accuracy. For guidance, see [best practices for writing effective instructions](#).

## Publish an agent with computer use

Set up computer use, then publish your agent. How your agent runs depends on the scenario. It can be autonomous or conversational:

- Autonomous agents run automatically and perform tasks in the background.
- Conversational agents let users interact through channels like Microsoft Teams.

Computer use works best for autonomous agents, performing tasks in the background without user interaction.

You can also apply computer use in conversational experiences, but keep these considerations in mind:

1. If you select **User authentication** as the authentication setting, each user interacting with the agent in a conversation needs valid credentials for the machine used by computer use.
2. When the tool runs, it shares reasoning messages and screenshots of the machine's activity in the chat.

## Monitor computer use runs

To check what computer use did during a run:

1. Go to the **Activity** section of your agent and select a run.
2. On the run details page, switch between the **Activity map** and **Transcript** views.

The **Transcript** view shows how computer use follows your instructions, providing a step-by-step log with reasoning messages and screenshots for each action.

# Advanced computer use activity

Storing computer use logs in Dataverse provides enhanced capabilities for reporting, auditing, and troubleshooting. When these logs are stored in Dataverse (default setting), you can view detailed activity information in the side panel by selecting a computer use action from the activity map.

## ⓘ Note

This feature is gradually rolling out to environments where the region is set to United States and might not yet be available in your environment.

## Dataverse storage

When you turn on the **Computer Use** feature in the Power Platform admin center, the **Store logs in Dataverse** option is turned on by default. If you turn off this setting, only default activity logs appear. When enabled, logs consume a combination of Dataverse capacity:

- Database capacity (flowsession table and flowsessionbinary metadata)
- Log capacity (flowlog table)
- File storage capacity (flowsessionbinary file attachments)

Ensure you have sufficient capacity and that your retention period and verbosity settings align with your requirements.

## Configure advanced computer use logging

Follow these steps to configure or verify advanced computer use logging:

1. Go to [Power Platform admin center](#), select your environment, and open the **Features** area.
2. Scroll down to the **Computer Use** settings and ensure **Store logs in Dataverse** is turned on (default setting).

## ⓘ Note

The following advanced computer use log settings don't affect how Microsoft Copilot agent transcripts (including any computer use logs) are stored. Transcripts continue to include basic computer use logs, screenshots, and all other agent and tool logs, regardless of the configuration here.

3. Review or change the **Computer use logs verbosity** level:

- **All data** (default) - Stores the complete set of advanced computer use logs.
- **Data without screenshots** - Stores the same content as with **All data**, except the screenshots.

- **Minimal** - Currently behaves the same as **Data without screenshots**, but this behavior might change in a future release.

4. Review or change the **Log retention time** to match your data retention timeframes:

- **Default:** 7 days (10,080 minutes)
- **Maximum:** 33,554,432 minutes
- To keep logs forever, enter **0** or **-1** as a custom value and ignore the unit dropdown.

5. (Optional) Turn on **Send audit logs to Microsoft Purview** if you want computer use run logs to be available in Microsoft Purview:

- Once enabled, logs appear under the activity term **CUAOperation**.
- This setting is independent of the previous two settings.

#### Note

During the rollout phase, the above settings may not yet be available in the Power Platform admin center interface. However, you can still configure them under the **Organization** table in Dataverse. The corresponding fields are labeled as follows:

- Enable CUA log upload to Dataverse
- Computer use logs verbosity
- The TTL for records in the Flow Logs Entity for CUA logs.
- Enable sending CUA audit logs to Purview

## View computer use activity details

When you select a computer use action in the activity map, a side panel opens with comprehensive session information:

 Expand table

| Section        | Information provided                                                                                                                                                                          |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description    | Description of what the computer use tool does                                                                                                                                                |
| Session replay | Series of screenshots captured during a computer use run with navigation controls                                                                                                             |
| Activity       | <ul style="list-style-type: none"> <li>- Action types</li> <li>- Action coordinates</li> <li>- User context used</li> <li>- Action timestamps</li> <li>- Screenshots for each step</li> </ul> |
| Summary        | <ul style="list-style-type: none"> <li>- Instruction text</li> <li>- Inputs</li> <li>- Duration and number of actions</li> <li>- Average time per action</li> </ul>                           |

| Section                 | Information provided                                                                                                                                                         |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | <ul style="list-style-type: none"> <li>- Number of screenshots</li> <li>- Human intervention count</li> <li>- Machine name and link</li> <li>- Machine user login</li> </ul> |
| Websites & applications | Websites and desktop apps accessed                                                                                                                                           |
| Credentials used        | Credentials used to access resources such as websites                                                                                                                        |
| Export session logs     | Export session log option for offline review of computer use logs                                                                                                            |

## Best practices

To stay productive and safe in today's digital environments, follow best practices for computer use, especially when writing clear instructions and securing machines.

### Best practices for securing machines

When setting up machines for computer use that lets AI perform tasks by using natural language, consider these security recommendations:

[\[+\] Expand table](#)

| Security recommendation                                                  | Additional information                                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use dedicated machines for computer use                                  | Assign specific, isolated machines exclusively for tasks involving computer use. This approach reduces the risk of cross-contamination from unrelated software, malware, or unauthorized access. It lets you control configurations, updates, and monitoring more effectively.        |
| Limit permissions to the user account that you're using for computer use | Set up the user account for computer use to follow the principle of least privilege —grant only the permissions needed to run the required tools.                                                                                                                                     |
| Limit web access to an allowlist of specific trusted websites only       | Allow web access only to a predefined list of vetted and trusted domains. For example, you can <a href="#">configure Microsoft Edge policy settings with Microsoft Intune</a> that target machines used for computer use.                                                             |
| Limit specific desktop apps to be available                              | Only install and allow execution of applications that are essential for the intended AI workflows. Remove or disable access to unnecessary software. For example, you can <a href="#">configure Application Control</a> to limit what applications are allowed to run on the machine. |

### Best practices for instructions for computer use

The instructions you write determine how well computer use works. Specific, detailed instructions help computer use complete tasks accurately. Think of it as explaining a task to a colleague. Clear, step-by-step guidance helps ensure success.

## Tips for writing effective instructions:

- Be specific about websites and applications. Always include the full URL of any website and the exact name of any application the tool should use. Example: "Open <https://www.microsoft.com> and go to 'Company news'."
- Clearly state relevant actions. If you want something done, say it explicitly - especially actions like submitting a form or sending an email. Example: "Once you fill in the form, select **Submit**. No need to ask for permission."
- Break down complex interactions. For areas where the UI might be more complex to navigate, explain each step in detail. Example: "Select the **More** icon in the top right corner. A dropdown appears. Once it opens, select the last item in the list."
- Use step-by-step formatting for longer tasks. Long instructions are easier to follow when you format them as a list.

## Sample instructions

Explore these sample instructions to try out computer use or use them as a reference for writing your own.

[\[+\] Expand table](#)

| Scenario           | Name                                | Description                                                     | Instructions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|-------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Invoice processing | Transfer and submit invoice details | Transfer invoice data from a PDF and submit it to another form. | <ol style="list-style-type: none"><li>1. Go to <a href="https://computerusedemos.blob.core.windows.net/web/Contoso/invoice-manager.html">https://computerusedemos.blob.core.windows.net/web/Contoso/invoice-manager.html</a>, set the Date filter to Last 24 hours, and open the invoice PDF.</li><li>2. In a new tab, open <a href="https://computerusedemos.blob.core.windows.net/web/Contoso/index.html">https://computerusedemos.blob.core.windows.net/web/Contoso/index.html</a> and fill out the form with the data from that PDF. Submit the invoice form, no confirmation needed.</li></ol> |
| Data entry         | Submit inventory items              | Add products to the inventory system.                           | <ol style="list-style-type: none"><li>1. Go to <a href="https://computerusedemos.blob.core.windows.net/web/Adventure/index.html">https://computerusedemos.blob.core.windows.net/web/Adventure/index.html</a>.</li><li>2. Submit a new entry for each of the following items:<br/>Rear Derailleur, RD-4821, 50, 42.75, Tailspin Toys<br/>Pedal Set, PD-1738, 80, 19.99, Northwind Traders<br/>Brake Lever, BL-2975, 35, 14.50, Trey Research<br/>Chainring Bolt Set, CB-6640, 100, 5.25, VanArsdel, Ltd.<br/>Bottom Bracket, BB-9320, 60, 24.90, Tailwind Traders</li></ol>                          |
| Data extraction    | Look up portfolio manager and value | Get the manager name and value for a portfolio.                 | <ol style="list-style-type: none"><li>1. Go to <a href="https://computerusedemos.blob.core.windows.net/web/Portfolio/index.html">https://computerusedemos.blob.core.windows.net/web/Portfolio/index.html</a>.</li><li>2. Find the row for Fourth Coffee and record the Portfolio Manager name and the current Portfolio Value exactly as shown.</li><li>3. Return those two values as the final output.</li></ol>                                                                                                                                                                                   |

## Administrative controls

Administrators can control the availability and usage of computer use features. These controls help your organization manage how computer use is deployed and ensure it aligns with your security and compliance requirements.

## Disable computer use

You can disable computer use in the admin center or using the Power Platform CLI.

### Use the admin center

To disable computer use in an environment using the admin center:

1. Go to the [Power Platform admin center](#).
2. Select **Manage** on the navigation pane, then select **Environments**. A list of environments appears.
3. Select the environment to update. The environment's details page opens.
4. On the top menu bar, select **Settings**.
5. Expand **Product**, and then select **Features**.
6. Scroll to the **Computer use** and turn off the toggle.

#### ⚠ Note

If you can't find the **Computer use** toggle in the environment settings of the Power Platform admin center, this feature might not yet be visible in your tenant. To manage this setting manually, [Use the Power Platform CLI](#).

### Use the Power Platform CLI

To disable computer use in an environment using the Power Platform CLI:

1. If you haven't already, install the [Power Platform CLI](#).
2. Open a command line and run the following command:

```
pac auth create
```

Follow the prompts to connect to your tenant.

3. Run the following command to disable the setting:

```
pac env update-settings --environment <envid> --name iscomputeruseinmcenabled --value false
```

Replace `<envid>` with the environment ID, which you can find in the URL of the environment settings page.

## Disable hosted browser

To disable the hosted browser in a tenant:

1. Go to the [Power Platform admin center](#).
2. Select **Manage** on the navigation pane, then select **Tenant settings**. A list of settings appears.
3. Select **Hosted browser in computer use**. A pane opens.
4. Turn off the toggle.
5. Select **Save**.

## Licensing

While computer use is in preview, it's billed using the Agent action feature with a billing rate of five Copilot Credits. For more information, see [Microsoft Copilot Studio billing rate and management](#).

Each computer use run can perform several actions, and each action costs five Copilot Credits.

For example, if you set up computer use to fill out a web-based timesheet form, it performs these actions each time it's triggered:

1. Launch browser.
2. Navigate to the timesheet web portal.
3. Select **Create new timesheet**.
4. Fill the Start Time form field.
5. Fill the End Time form field.
6. Fill the Project Code form field.
7. Select the Submit button.

In this example, computer use executes seven actions, consuming a total of 35 messages.

## Limitations

The following limitations affect computer use:

- Hosted machine groups aren't supported.
- Multi-screen desktops aren't supported.
- In the agent's **Activity** tab, the agent session status might show as "In progress", although the session might be completed. Select a specific agent session and switch to transcript mode to validate that the computer use session is completed.

# Hosted browser limitations

The service might throttle hosted browser usage based on demand. Therefore, it's not recommended for production use.

## ⓘ Note

You might see the error message "No machine able to run the automation has been found." when starting a computer use session on hosted browser. This error can occur if you already have an active session on hosted browser, if a demand-based throttling limit is reached, or due to an internal error. We're working to improve this experience.

The following scenarios might cause throttling:

[+] Expand table

| Category                | Details                                                                                                                             | Impact       |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------|--------------|
| User based throttling   | A user can have one active hosted browser session at any given time. Any subsequent computer use session on a hosted browser fails. | Session fail |
| Demand based throttling | Depending on overall demand, throttling might be applied and computer use session on a hosted browser fails.                        | Session fail |

# Troubleshooting

If you encounter issues while using computer use, review these common problems and their solutions:

[+] Expand table

| Issue                                                                                                                                                                                                                                                                                                                                                                                       | Solution                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Computer use execution fails with error: "Multi-factor auth (MFA) was detected as required for this account." When using Microsoft Entra ID credentials on a device with Network Level Authentication (NLA) enabled, or when Microsoft Entra authentication is required for Remote Desktop, you must either grant an MFA exception for the account or use certificate-based authentication. | - Disable NLA. Learn more in <a href="#">Desktop flows run failed with MSEntraLogonFailure</a><br>- Provide admin consent for unattended runs using sign-in credentials with NLA. Learn more in <a href="#">Run unattended desktop flows</a> . |
| A disconnected session exists for the user used in the connection on the target machine. Make sure the user signs out from any existing session.                                                                                                                                                                                                                                            | - Connect to the machine using Remote Desktop. Open the Windows Start menu, select your user profile icon, and then select <b>Sign out</b> .                                                                                                   |
| An automation is already running on the machine. Retry later.                                                                                                                                                                                                                                                                                                                               | - If you just tested or ran computer use, wait 2–3 minutes before trying again.                                                                                                                                                                |
| Inputs are ignored when running computer use tool                                                                                                                                                                                                                                                                                                                                           | A breaking change alters the expected input schema (from <code>context:</code> to <code>inputType:</code> ) for tools configured before                                                                                                        |

| Issue                                                                                                                                                                                                                                                                                                         | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                                                                                               | <p>August 2025.</p> <p><b>Fix:</b> Edit the tool's inputs, add any temporary input, save (this refreshes the schema), delete the temporary input, save again, retest, and publish your agent.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <p>When you use hosted browser, you might encounter the following error: "Request to XRM API failed with error: 'Message: user with id does not have ReadAccess right(s) for record with id of entity Flow Machine Group. Consider assigning a role with the level BusinessUnitLevel to the user or team"</p> | <p>Your environment might not have the latest Power Automate extensions solution. To force an update of the solution, as an environment admin:</p> <ol style="list-style-type: none"> <li>1. Go to the <a href="#">Power Platform admin center</a></li> <li>2. Select <b>Manage</b> on the side navigation, and then select <b>Environments</b>. A list of environments opens.</li> <li>3. Select the environment to update.</li> <li>4. Select <b>Resources</b>, and then select <b>Dynamics 365 apps</b></li> <li>5. Find <b>Power Automate Extensions</b> in the list and select the <b>Update available</b> in the <b>Status</b> column to trigger an update.</li> </ol> |
| <p>The test function on computer use tool doesn't work properly. For example, the test ends abruptly with an error: "The computer use tool is not a part of the planned execution sequence."</p>                                                                                                              | <p>Test using the agent's test chat instead, and review the descriptions (instructions) provided to the agent to ensure it has sufficient context to trigger computer use.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

## Share your feedback

Do you have feedback about computer use? Let us know at [computeruse-feedback@microsoft.com](mailto:computeruse-feedback@microsoft.com).

## Related information

[FAQ for the computer use tool](#)

Last updated on 10/24/2025

# Create search query

07/08/2025

**Create search query** is a topic-level tool that your agent can use to capture a rewritten version of a user's input, enriched with conversation history. You can use the rewritten query in downstream tasks, such as search or summarization. This rewritten string gives makers a structured and clarified version of the original query, which you can repurpose to fit your agent's purposes.

This functionality is especially powerful for [custom search](#) scenarios. Your agent can take full control over how queries are matched to indexed content, passed to external systems, or logged for analytics.

As agents evolve from simple question and answer bots to tailored business agents, reshaping user queries becomes essential. Query rewriting bridges the gap between user intent and domain-specific language to generate more relevant and accurate results.

## When to use Create search query

Use **Create search query** when you want to perform the following actions:

### Drive custom search experiences

Use the rewritten query string as an input for your own search logic. Examples include querying a non-Microsoft API, filtering internal knowledge bases, or crafting dynamic prompts.

**Example:** Feed the rewritten query into a custom SharePoint or SQL-based search system.

### Enhance relevance with context

Use conversation history to rewrite vague queries into something more targeted.

**Example:** Turn "show me the form" into "employee onboarding request form" based on previous user questions.

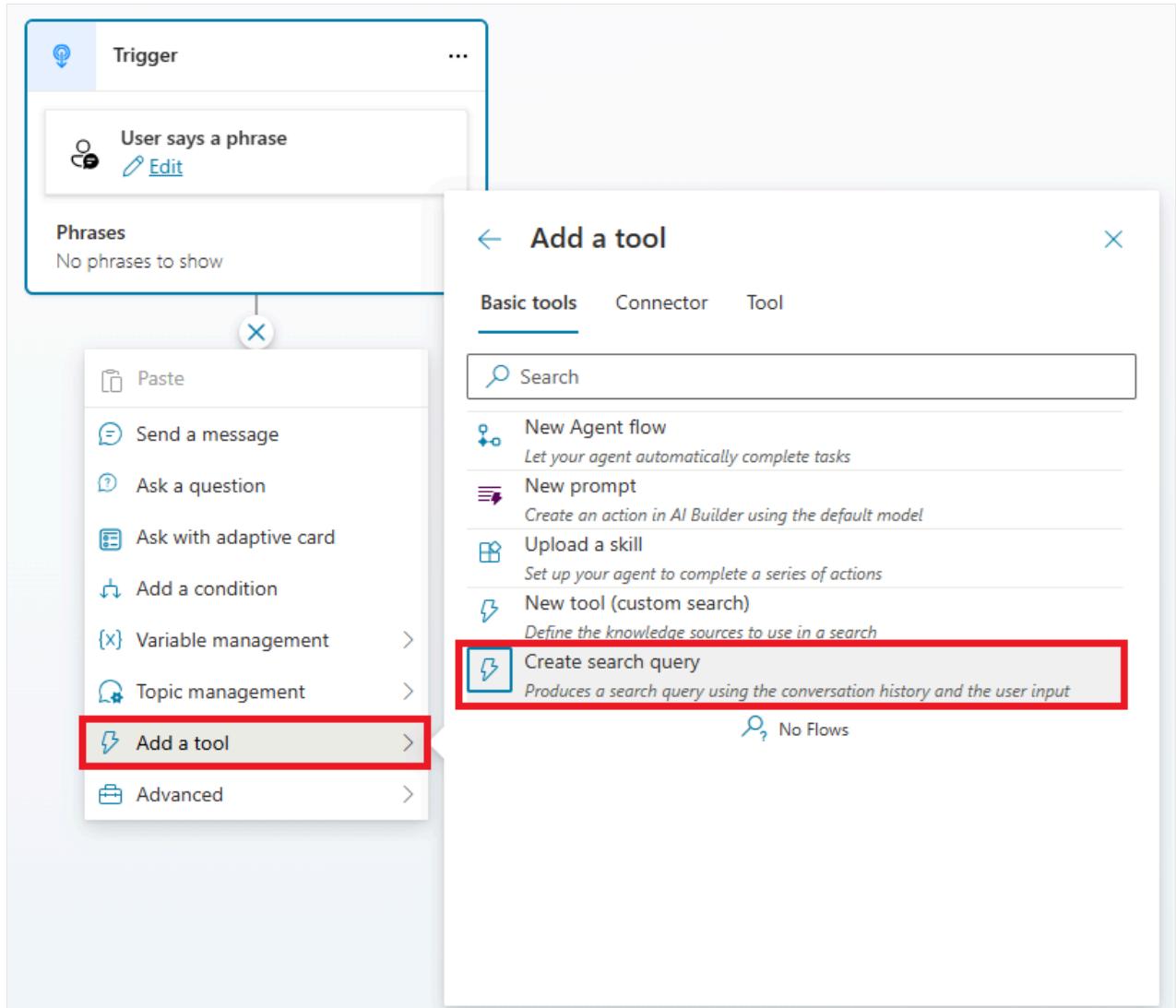
### Power domain-specific retrieval

Tailor the rewritten query to match how your organization categorizes or indexes information.

**Example:** Add tags like department, timeframe, or user role to make search queries more precise.

# How Create search query works

1. Select the Add node icon below the node under which you want your agent to perform a custom search.
2. Select Add a tool, then **Create search query**.

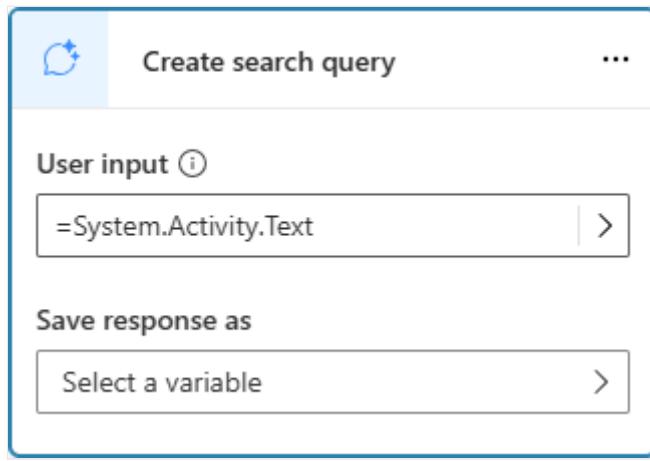


A **Create search query** node appears on the canvas.

The **Create search query** tool rewrites the user's input based on:

- The original input
- Any relevant conversation history or context variables

The tool outputs a cleaned and enriched search string as a variable for use in downstream logic.



## Use case example: employee handbook copilot

A user might say: "Tell me about the leave policy."

The agent enriches this request with:

1. The type of leave
2. Whether the leave is for themselves or someone else
3. Whether they want the current policy or a historical policy

## Dialog flow (simplified YAML representation)

YAML

```
Compose the custom search input
- kind: SetVariable
 variable: Topic.SearchQueryInput
 value: = "Leave policy: " & PlainText(Topic.LeaveType) &
 If(Topic.ForSelf, " for self", " for someone else") &
 "; version: " & PlainText(Topic.IsCurrentPolicy) &
 "; original query: " & PlainText(Topic.UserOriginalInput)

Generate a rewritten query using CreateSearchQuery
- kind: CreateSearchQuery
 userInput: =Topic.SearchQueryInput
 result: Topic.SearchQuery
```

```
Use it in any way you want
```

```
- kind: CustomSearchOrAPI
```

```
 input: =Topic.SearchQuery
```

## Final rewritten query example

Leave policy: sick; for self; version: current; original query: I want to know about sick leave

This rewritten string can now be:

- Passed into a custom search engine.
- Used in an external API call.
- Logged for analytics or tuning.
- Displayed back to the user for confirmation.

# Perform custom search

07/08/2025

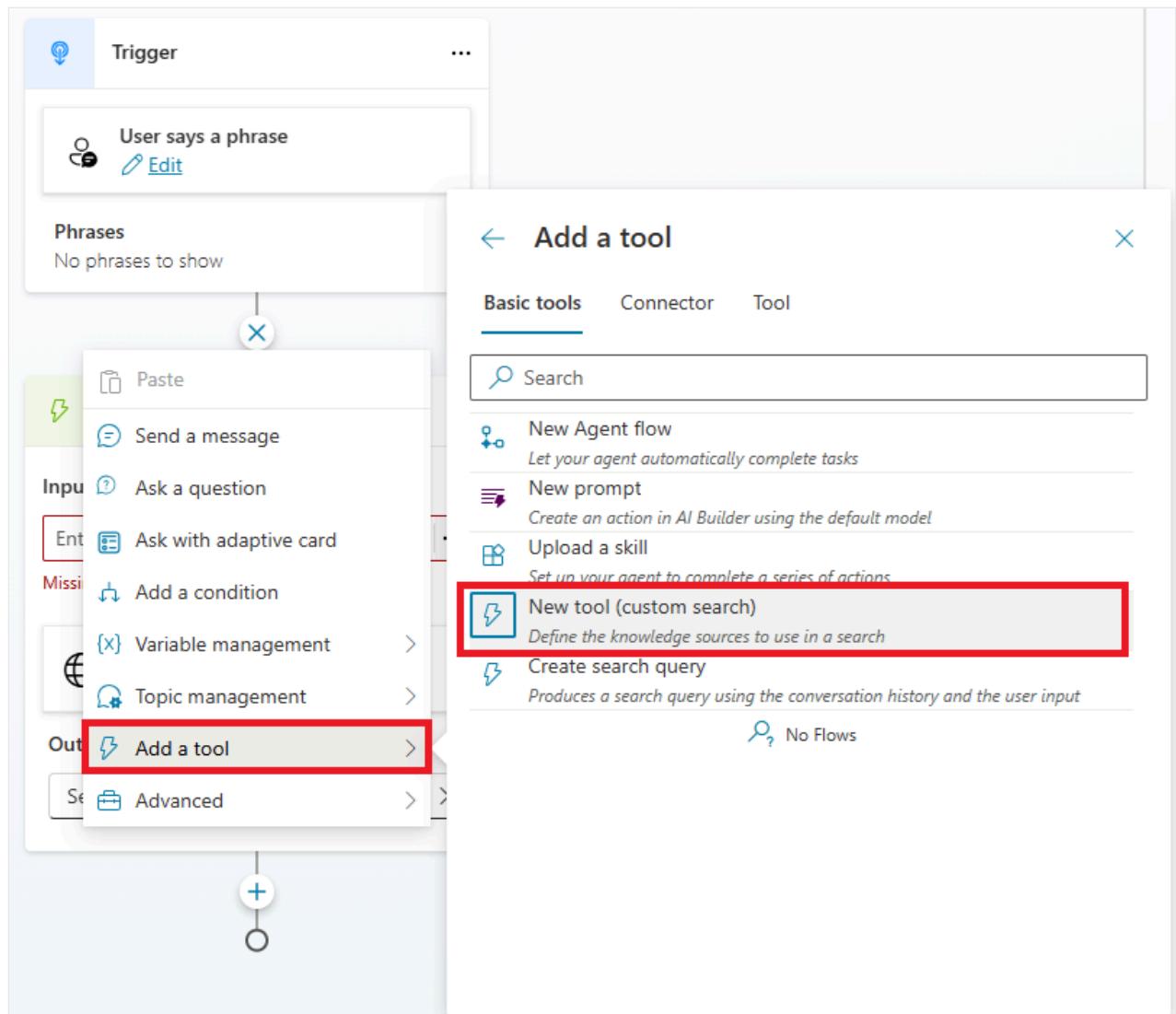
**Custom search** is a topic-level tool that your agent can use to perform a search, based on an input query. The search is grounded on a customized set of data sources. The tool returns raw, unsummarized results as an output variable. You can process the results and use them in downstream tasks.

## When to use custom search

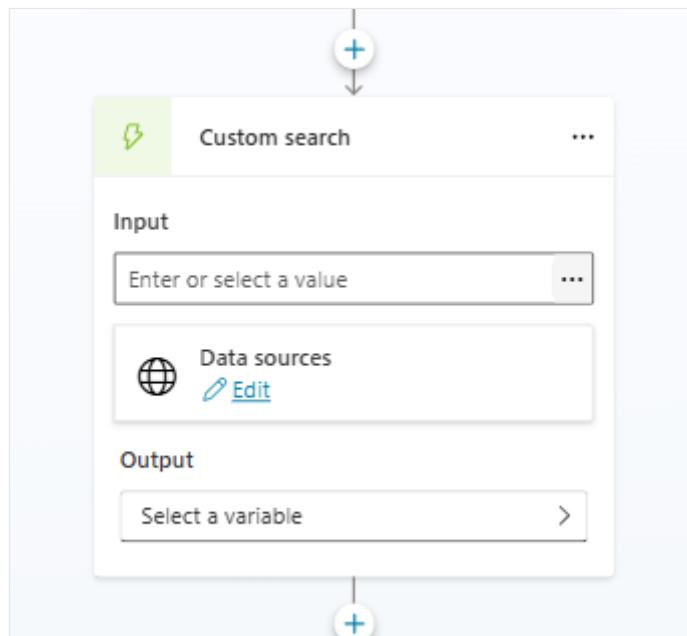
Use **Custom search** when you want to have full control to process the search output. For example, you can use the search results to filter out sensitive information before summarizing the results.

## Add a custom search tool

1. Select the **Add node** icon below the node under which you want your agent to perform a custom search.
2. Select **Add a tool** and **New tool (custom search)**.

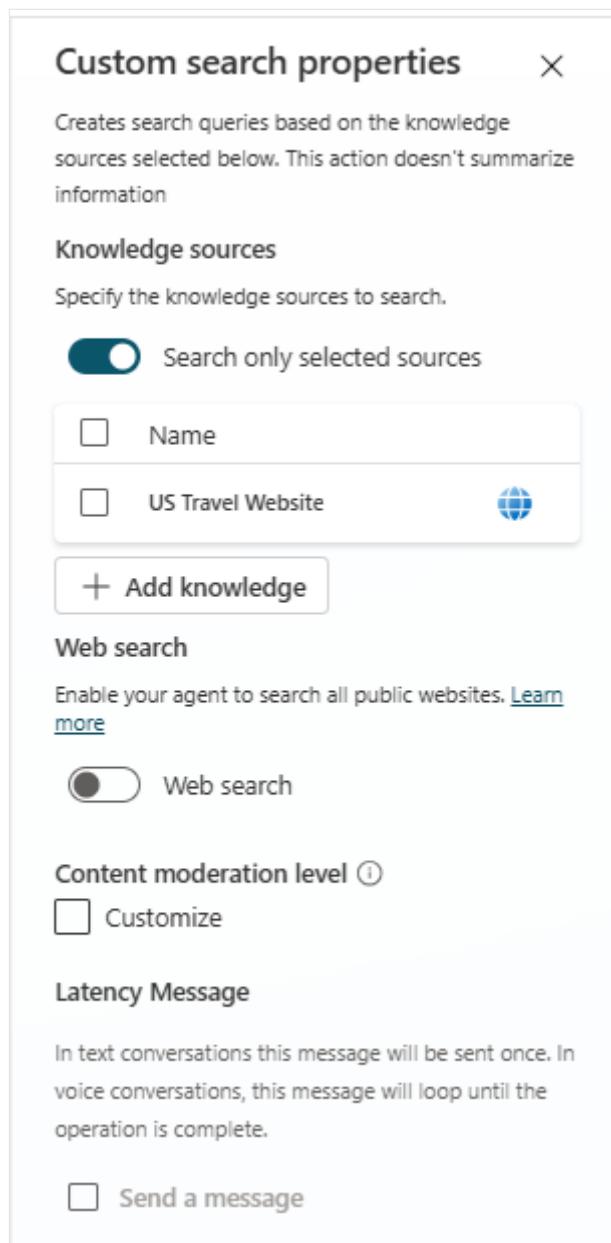


A Custom search node appears on the canvas.



## Configure custom search tool

1. Under **Input**, enter the search query. The query can include variables, formulas, and plain text. To use the user message as input, set input to the System variable `Activity.text`.
2. Under **Output**, select a variable to store the search results.
3. Under **Data sources**, select **Edit** to specify knowledge sources and search properties. The **Custom search properties** pane appears.



The **Custom search properties** pane provides you with several options to customize the search:

- Select the **Search only selected sources** toggle to limit the search to only the selected knowledge sources.
- Select **Add knowledge** to add a new knowledge source.
- Select the **Web search** toggle to enable web search. This option lets the custom search tool query the web for additional information, in addition to the configured knowledge sources.

- Select the **Content moderation level** checkbox to configure the content moderation level for the search results.
- Select the **Latency Message** checkbox to configure a latency message.

# Use interactive voice response in your agents

09/18/2025

Copilot Studio supports interactive voice response (IVR) capabilities, including speech and dual-tone multi-frequency (DTMF) input, context variables, call transfer, and speech and DTMF customization.

Before you can create or edit agents for voice scenarios, you need a phone number to use. With [Azure Communication Services](#), you can get a new phone number or use an existing phone number. For more information, see [Integrate a voice-enabled agent with Dynamics 365 Customer Service](#).

## Key concepts for voice authoring

With the growing trend toward self-service applications, voice-enabled agents are making a huge difference for businesses. Voice-enabled agents are used in various applications, such as call centers, mobile apps, and messaging platforms.

Voice-enabled agents can collect user input through speech and Dual-Tone Multi-Frequency (DTMF).

## Supported voice features

After your agent is ready for voice services, you can configure features for your scenario.

 Expand table

| Feature                          | Description                                                                                                                                                       |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Barge-in                         | Allows users to interrupt the system at any time during the conversation.                                                                                         |
| Dual-tone multi-frequency (DTMF) | Allows users to enter data by pressing keys on their phone keypad. DTMF can accept single key menu navigation and collect business information with multi-digits. |
| Latency message                  | Send messages or audio to inform users that the system is still processing their request in long-running operations.                                              |
| Silence detection and timeouts   | Detects when the user stops speaking, allowing the system to respond appropriately.                                                                               |

| Feature                                 | Description                                                                                                                                         |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Speech recognition improvement          | Speak naturally, without a script-a user's spoken command or question is translated for the voice-enabled agent to process.                         |
| Speech Synthesis Markup Language (SSML) | Control how your agent's voice sounds and behaves with users. You can control the tone, pitch, and speed of the voice that interacts with the user. |

## How to configure voice features

The following articles show you how to enable features, for a given scenario, step by step.

- Collect user input via speech and DTMF
  - [Dual-tone multi-frequency \(DTMF\) support](#)
  - [Silence detection and timeouts](#)
  - [Speech recognition improvement](#)
- Control how your agent's voice sounds and behaves with users
  - [Barge-in](#)
  - [Latency message](#)
  - [Speech synthesis markup language \(SSML\)](#)
- Control the call flow by transferring calls or hanging up
  - [Transfer a call to a representative or external phone number](#)
  - [Hang up call at the end of a conversation](#)
- Authoring capabilities when building a voice-enabled agent
  - [Build a voice-enabled agent from a template](#)
  - [Speech & DTMF modality](#)
  - [Use voice variables](#)
  - [Test your voice-enabled agent in chat](#)

## Known limitations

These tips and limitations help you successfully integrate voice into your agent.

[ ] [Expand table](#)

| Feature       | Tip or limitation                                                                                                |
|---------------|------------------------------------------------------------------------------------------------------------------|
| Channel order | Enable the Telephony channel first and then connect with Dynamics 365. The sequence is for channel reconnection. |

| Feature                                | Tip or limitation                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Language/Locale                        | For a full list of supported languages and locales, see <a href="#">Language support</a> . If you have a customized locale request, contact the Copilot Studio team.                                                                                                                                                                                                 |
| DTMF                                   | The question node supports copilot single-digit DTMF (global command) and multi-digit DTMF, with conflict handling for the DTMF key at the same time.                                                                                                                                                                                                                |
| DTMF only                              | When DTMF only for voice is enabled, some timers might not be effective, such as interdigit timeout for DTMF or silence detection timeout.                                                                                                                                                                                                                           |
| Latency message on Action node         | <ul style="list-style-type: none"> <li>- If you don't enable latency message or the message is empty, all messages before the action node are blocked and sent after the action completes.</li> <li>- If you use multiple consecutive action nodes for one topic and hit any unexpected results, add a message node between the consecutive action nodes.</li> </ul> |
| Test chat dial pad                     | Pressing a key on the dial pad in the Test chat returns "/DTMF#", which isn't supported, and isn't a valid input. Instead, the command "/DTMFkey#" should be typed into the chat.                                                                                                                                                                                    |
| Multilingual voice-enabled agents      | If you incorporate a multilingual voice-enabled agent, you must set <a href="#">authentication</a> to <i>No authentication</i> to be able to publish on the Dynamics 365 Customer Service channel.                                                                                                                                                                   |
| Customer engagement hub                | <p>Apart from Dynamics 365, all the other customer engagement channels only work with chat-based agents. The following aren't supported for voice-enabled agents:</p> <ul style="list-style-type: none"> <li>- Genesys</li> <li>- Live person</li> <li>- Salesforce</li> <li>- ServiceNow</li> </ul>                                                                 |
| Generative AI for voice-enabled agents | <a href="#">Creating and editing topics with Copilot</a> isn't supported for voice-enabled agents. Copilot doesn't create messages for Speech & DTMF, and doesn't configure DTMF mappings.                                                                                                                                                                           |
| Environment release cycle              | Currently, voice-enabled agents are only available in <b>Standard</b> environments. For more information, see <a href="#">Unable to turn on Optimize for voice</a> .                                                                                                                                                                                                 |

# Integrate a voice-enabled agent with Dynamics 365 Customer Service

Article • 11/19/2024

To create and test the voice features and integration between Copilot Studio and Dynamics 365 Customer Service applications:

1. Configure your Dynamics 365 Customer Service app. For more information, see [Integrate a Copilot Studio bot](#).
2. Create an agent in your new environment. Consider using the [Voice template](#).
3. Add, configure, and edit the voice-related settings for the agent. For more information about each feature and setting, see [Configure voice capabilities](#).
4. Connect your agent to Dynamics 365 Customer Service. For more information, see [Configure hand-off to Dynamics 365 Customer Service](#).
5. Publish your agent. For more information, see [Key concepts - Publish and deploy your agent](#).
6. Configure voice functionality in Dynamics 365 Customer Service.

---

## Feedback

Was this page helpful?



[Provide product feedback ↗](#)

# Use generative answers in your voice-enabled agents

Article • 11/26/2024

When you create a voice-enabled agent, it automatically supports the [addition of knowledge for generative answers](#). The voice-enabled agent formats text-to-speech (TTS) output, based on generative answers.

## Informing users that responses were AI generated

When you incorporate generative answers into your voice-enabled agent, it's important to inform your users of the following:

- AI is being used to generate responses.
- Users can consult the references used by your agent to generate responses.
- Users must verify the accuracy of the generated responses themselves.

### Note

For more information, see [FAQ for generative answers](#).

The following sample messages are returned to users when incorporating generative answers messages in a voice-enabled agent.

 Expand table

| Type of message | Description                                                                                                              | Example                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AI disclaimer   | Message to inform the user that AI might have been used, and that the user should verify the information in the message. | "Please note that some responses are generated by AI and might require verification for accuracy."                                                             |
| Latency message | Message to inform the user to hold while information is retrieved.                                                       | "Please hold while I gather information.<br><code>&lt;break time="5000ms"/&gt;</code><br>Note that the latency message loops, so timeouts should be adjusted." |

| Type of message  | Description                                                                            | Example                                                                                                                                                    |
|------------------|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Apology message  | Message to apologize to the user if no answer was found.                               | "I'm sorry, I don't have the knowledge to answer at this time."                                                                                            |
| Hear references  | Message to inform the user that they can hear a list of the references used by the AI. | "This response was used by AI and might be incorrect. To hear the references that were consulted, please press 9. To return to the main menu, press star." |
| Barge-in message | Message to inform the user that they can skip listening to references.                 | "The following references were consulted. Press star at any time to return to the main menu."                                                              |

## Setup generative answers in your voice-enabled agent

1. [Configure your agent to support voice capabilities](#).
2. Provide your agent with a minimum of one [knowledge source](#).
3. Optionally, [configure content moderation](#).
4. [Test your responses](#). The received answers should be concise, follow the same language as the input query, and be free of incorrect information. Citations and sources should be stripped from the response.

## Access citations and sources from responses

By default, citations and sources are automatically stripped from responses. However, makers can retrieve the stripped citations and sources using the [Conversational boosting system topic](#).

## Update the Conversational boosting system topic

1. Select your agent and select the [Topics](#) page.
2. Select the [Conversational boosting](#) system topic.
3. Select the [Create generative answers](#) node, select the [More icon \(...\)](#) of the node, and then select [Properties](#).

4. Select **Advanced** and set **Save LLM response to Complete (recommended)**, and **Save bot response** to the Answer `Topic.Answer` variable.
5. Assign the `Answer.Speech.Citations` table variable to a **global variable**, such as `GenAnsVoiceRef`.
6. Add a **Message** node that instructs the caller how to hear the citations, such as, "This response was generated by AI and might be incorrect. To hear the references, please press 9. To return to the main menu, press \*."
  - a. Optionally, update the knowledge sources, content moderation, and other properties of the **generative answers node**.
  - b. Optionally, **configure a latency message** for generative answers, or reference a publicly accessible audio file with **speech synthesis markup language (SSML)** to be played to cover the latency.

## Create a topic to output the citation

A topic containing one or more of the following nodes is used to capture and output citations referenced by the agent.

- The **Trigger** node is inserted by default. Select **DTMF global command received** as the trigger type, and set the **DTMF Key** property to 9.
- **Message** nodes are used to inform users either that a reference was used, or the agent was unable to use a reference to generate a response. These responses could be either of the following messages:
  - "The following reference was consulted. Press \* at any point to return to the main menu."
  - "No specific references were used to generate the previous answer."
- **Condition** nodes are set to branch the dialog based on the user's responses. In the left branch, these are the conditions met when the user asks a question that utilizes a reference.
  - The left branch of the conditions are used when the first condition sets the global variable set in the previous steps. For example, `Global.GenAnsVoiceRef`. Set the variable to `has value`. This branch is also followed by another **Message** node. Here a `Reference.Name` variable is inserted, followed by "consulted at", and another variable, `Reference.Url1`, is added.

- The right branch of the conditions are used when the user's questions aren't answered or a reference isn't used to return a response. A **Message** node follows, with a message such as, "No specific references were used to generate the previous answer."
- A **Set a variable value** node flows under the left branch of the condition node. The first variable is set to **Create Variable name** and set to `NumReferences`. The **To value** is set to a formula: `Count Rows(Global.GenAnsVoiceRef)`. The **Type** is set `number`, and **Reference** is set to `Topic.NumReferences`.
- A **Loop through a list** node sets **Items to loop over** to the `Global.GenAnsVoiceRef` variable, and the **Loop value** variable is set to record type and given the name, "Reference."
- A **Go to another topic** node is set to **Main Menu**.
- An **End current topic** node completes the topic.

## Sample topic YAML to output citations

The following YAML demonstrates how a topic can be created to output citations. Copy and paste the YAML into a new topic to speed up the creation of the topic to output citations. To use the YAML, do the following:

### ! Note

The YAML relies on having assigned the `Answer.Speech.Citations` table variable to a **global variable**, named `GenAnsVoiceRef`. For more information, see [Access citations and sources from responses](#).

1. Select your agent and select the **Topics** page.
2. Select **Add a topic** and select **From blank**.
3. Open the code editor, and paste the YAML over the existing YAML.
4. Close the code editor and save the topic.

```
yaml
```

```
kind: AdaptiveDialog
beginDialog:
 kind: OnDtmfKeyPress
 id: main
```

```

dtmfKey: Num9
actions:
- kind: ConditionGroup
 id: conditionGroup_1m7G18
 conditions:
 - id: conditionItem_a2ax5d
 condition: !=IsEmpty(Global.GenAnsVoiceRef)
 actions:
 - kind: SetVariable
 id: setVariable_dgK3w7
 variable: Topic.NumReferences
 value: =CountRows(Global.GenAnsVoiceRef)

 - kind: ConditionGroup
 id: conditionGroup_YRr0Bv
 conditions:
 - id: conditionItem_hD1dXt
 condition: =Topic.NumReferences = 1
 actions:
 - kind: SendActivity
 id: sendActivity_42mrfG
 activity:
 speak:
 - The following reference was consulted. Press *
at any point to return to the main menu.

elseActions:
- kind: SendActivity
 id: sendActivity_zyZNYz
 activity:
 speak:
 - The following references were consulted. Press * at
any point to return to the main menu.

- kind: Foreach
 id: foreach_oWbdVm
 items: =Global.GenAnsVoiceRef
 value: init:Topic.Reference
 index: init:Topic.LoopIndex1
 actions:
 - kind: SendActivity
 id: sendActivity_CAqLKs
 activity:
 speak:
 - "{Topic.Reference.Name} consulted at
{Topic.Reference.Url}"

elseActions:
- kind: SendActivity
 id: sendActivity_XHfqKK
 activity:
 speak:
 - No specific references were used to generate the previous
answer.

```

```
- kind: BeginDialog
 id: zkSWAP
 dialog: crc82_sophieGenAnswers.topic.Untitled

- kind: EndDialog
 id: MXYSAQ
```

---

## Feedback

Was this page helpful?



Provide product feedback ↗

# Configure voice capabilities

09/18/2025

This article describes the features available in Copilot Studio for interactive voice response with Dynamics 365 Customer Service.

To get your agent ready for voice services, see [Integrate a voice-enabled agent with Dynamics 365 Customer Service](#).

For an overview of the voice services, see [Use interactive voice response in your agents](#).

## Speech & DTMF modality

A voice-enabled agent is different from a chat-based agent. The voice-enabled agent includes specific voice system topics for handling voice scenarios. A chat-based agent uses the text modality as the default. A voice-enabled agent uses the Speech & DTMF modality. The two modalities aren't compatible with each other.

*Optimize for voice* allows you to author voice-enabled agents across different modalities and ensures speech-related features are authored correctly.

## Optimize for voice

If you didn't start your agent with the [Voice template](#), you must enable the **Optimize for voice** option in the agent's **Settings**.

1. With an agent open, go to **Settings** > **Voice**.
2. Select **Optimize for voice**. The **Use voice as primary authoring mode** option is also set by default.

Your agent gets the following updates when you enable **Optimize for voice** and **Use voice as primary authoring mode** options:

- The ability to author voice features when switched from *text* to *Speech & DTMF*.
- The voice System topics *Silence detection*, *Speech unrecognized*, and *Unknown dialpad press* are automatically added to handle speech related scenarios.
- *Increase accuracy with agent data* (on by default), which improves speech recognition accuracy.
- The existing agent flow doesn't change (such as the *Main Menu* topic to start conversations with mapped DTMF triggers).

### Important

- The **Optimize for voice** setting only changes the voice authoring capabilities, not the channel setting. Turn on the **Telephony** channel for a fully voice-enabled agent.
- In addition, setting **Optimize for voice** on an agent that wasn't originally configured for voice features means that the agent doesn't have the *Main Menu (preview)* topic. You must recreate that topic, if needed.
- If you're unable to turn on **Optimize for voice**, check the Power Platform environment hosting your agent and ensure that **Get new features early** is turned off for that environment. For more information, see [Unable to turn on Optimize for voice](#).

## Disable optimization for voice

You can turn off **Optimize for voice** in agent authoring if you don't use the **Telephony** channel. After you turn off **Optimize for voice**, you get the following changes:

- No agent authoring for voice features, such as DTMF and barge-in.
- The default *text* modality is set.
- No improvement to speech recognition, since there's no speech recognition.
- No voice system topics or global DTMF topic.

### Note

Some topics might report errors during publish if they still reference the DTMF topic (now turned off).

- No change to your agent flow and channel setting, since turning off optimization doesn't remove the **Telephony** channel.
- Turning on or turning off **Optimize for voice** doesn't take effect until you publish your agent. If you turn it on or off accidentally, and the agent switches between modalities, you have time to fix it.

### Important

If your **Telephony** channels are enabled, turning off **Optimize for voice** can break your agent, since all DTMF triggers are automatically disabled.

## Use voice as your primary authoring mode

The Speech & DTMF modality should be selected for each node in voice feature authoring. You can select the agent authoring preference as *use voice as primary authoring mode*. This setting ensures all input fields have the right modality. If you already enabled **Optimize for voice**, the **Use voice as primary authoring mode** option is enabled by default.

## Message availability

Using the text or speech modality can affect your channel differently.

  Expand table

| Text modality     | Speech modality   | Agent text & speech channel |
|-------------------|-------------------|-----------------------------|
| Message available | Message empty     | Message available           |
| Message empty     | Message available | Message not available       |

## Customized automatic speech recognition

Voice-enabled agents for a specific domain, such as medical or finance, might see users use finance terms or medical jargon. Some terms and jargon are hard for the voice-enabled agent to convert from speech to text.

To ensure the speech input is recognized accurately, you can improve speech recognition:

1. With your agent open, select **Settings > Voice**.
2. Select **Increase accuracy with agent data** to enable the agent's default customized automatic speech recognition settings.
3. Select **Save** to commit your changes.
4. Publish your agent to see the new changes.

## Agent-level voice options reference

The **Agent details** settings page lets you configure timeouts for various voice-related features. Settings applied in this page become the default for topics created in your agent.

To make changes to the agent-level timeout options:

1. With an agent open, select **Settings > Voice**.

The screenshot shows the 'Settings' page with a sidebar on the left containing various categories like Copilot details, Generative AI, Security, etc., with 'Voice' selected and highlighted by a red box. The main content area is titled 'Voice' and includes sections for 'Optimize for voice', 'Use voice as primary authoring mode', 'Customized automatic speech recognition', and 'Timeouts' (DTMF, Silence detection, Speech collection, Latency messaging). Below these is a 'Speech' section with a 'Sensitivity' option.

2. Select the settings you want and adjust the agent's default settings.

3. Select **Save** to commit your changes.

## Agent-level settings

The following table lists each option and how it relates to node-level settings.

[ ] Expand table

| Voice-enabled agent-level section | Setting             | Description                                                                                                                                                        | Default value | Node-level override                                                            |
|-----------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------------------------------------------------------------------------|
| DTMF                              | Interdigit timeout  | Maximum time (milliseconds) allowed while waiting for the next DTMF key input. Applies multi-digit DTMF input only when users don't meet the maximum input length. | 3000 ms       | Question node with voice properties for <a href="#">Multi-digit DTMF input</a> |
| DTMF                              | Termination timeout | Maximum duration (milliseconds) to wait for a DTMF termination key. Limit applies when user reaches maximum                                                        | 2000 ms       | Question node with voice properties for                                        |

| Voice-enabled agent-level section | Setting                    | Description                                                                                                                                                                                                                                                                                                                                                                                      | Default value      | Node-level override                                                                                                                                                                                |
|-----------------------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                   |                            | <p>input length and didn't press termination key. Applies only to multi-digit DTMF input.</p> <p>After the limit times out and terminating DTMF key doesn't arrive, agent ends the recognition and returns the result up to that point.</p> <p>If set to "continue without waiting," the agent doesn't wait for termination key. Agent returns immediately after user inputs the max length.</p> |                    | Multi-digit DTMF input                                                                                                                                                                             |
| Silence detection                 | Silence detection timeout  | <p>Maximum silence (milliseconds) allowed while waiting for user input. Limit applies when the agent doesn't detect any user input. The default is "no silence timeout." Agent waits infinitely for user's input.</p> <p>Silence detection for voice times the period after the voice finishes speaking.</p>                                                                                     | No silence timeout | Question node with voice properties for <a href="#">Multi-digit DTMF input</a><br>System topic (silence detection trigger properties) for <a href="#">Configure silence detection and timeouts</a> |
| Speech collection                 | Utterance end timeout      | <p>Limit applies when user pauses during or after speech. If pause is longer than timeout limit, agent presumes user finished speaking.</p> <p>The maximum value for utterance end timeout is 3000 ms. Anything above 3000 ms reduces to 3000 ms.</p>                                                                                                                                            | 1500 ms            | Question node with voice properties                                                                                                                                                                |
| Speech collection                 | Speech recognition timeout | <p>Determines how much time the agent allows for the user's input once they begin speaking. The default value is 12000 milliseconds (about 12 seconds). No recognition timeout means infinite time. Agent reprompts the question. If no response, the voice is beyond <i>Speech recognition timeout</i>.</p>                                                                                     | 12,000 ms          | Question node with voice properties                                                                                                                                                                |

| <b>Voice-enabled agent-level section</b> | <b>Setting</b>        | <b>Description</b>                                                                                                                                                                                                                                                                            | <b>Default value</b> | <b>Node-level override</b>                                        |
|------------------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|-------------------------------------------------------------------|
| Latency messaging                        | Send message delay    | Determines how long the agent waits before delivering the latency message after a background operation request started. The timing is set in milliseconds.                                                                                                                                    | 500 ms               | Action node properties for <a href="#">long-running operation</a> |
| Latency messaging                        | Minimum playback time | The latency message plays for a minimum amount of time, even if the background operation completes while the message is playing. The timing is set in milliseconds.                                                                                                                           | 5000 ms              | Action mode properties for <a href="#">long-running operation</a> |
| Speech sensitivity                       | Sensitivity           | Controls how the system balances detection of speech and background noise. Lower the sensitivity for noisy environments, public spaces, and hands-free operation. Increase the sensitivity for quiet environments, soft-spoken users, or voice-command detection. The default setting is 0.5. | 0.5                  | There are no node-level overrides for this control.               |

## Enable barge-in

Enabling barge-in allows your agent users to interrupt your agent. This feature can be useful when you don't need the agent user to hear the entire message. For example, callers might already know the menu options, because they heard them in the past. With barge-in, the agent user can enter the option they want, even if the agent isn't finished listing all the options.

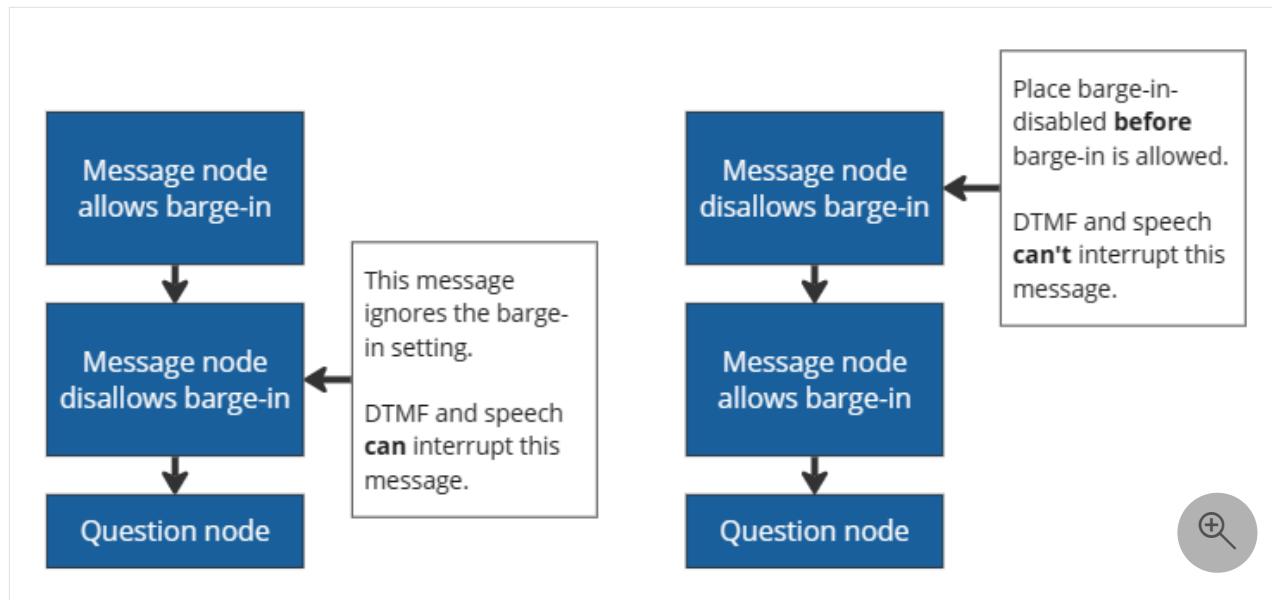
## Barge-in disable scenarios

- Disable barge-in if you recently updated an agent message or if the compliance message shouldn't be interrupted.
- Disable barge-in for the first agent message to ensure agent users are aware of new or essential information.

## Specifications

- Barge-in supports DTMF-based and voice-based interruptions from the agent user.

- Barge-in can be controlled with each message, in one batch. Place `barge-in-disabled` nodes in sequence before each node where barge-in is allowed. Otherwise, barge-in-disabled is treated as an allow-barge-in message.



Once one batch queue is finished, then the barge-in automatic setting is reset for the next batch, and controlled by the barge-in flag at each subsequent message. You can place barge-in disabled nodes as the sequence starts again.

### Tip

If there are consecutive message nodes, followed by a question node, voice messages for these nodes are defined as one batch. One batch starts with a message node and stops at the question node, which is waiting for the user's input.

Avoid disabling barge-in for lengthy messages, especially if you expect agent users to be interacting with the agent often. If your agent user already knows the menu options, let them self-service where they want to go.

## Set up barge-in

- With a **Message** or **Question** node selected, set the desired modality to **Speech & DTMF**.
- Select the **More** icon (...) of the node, and then select **Properties**.
  - For **Message** nodes, the **Send activity properties** panel opens on the side of the authoring canvas.

Select **Allow barge-in**.

b. For **Question** nodes, the **Question properties** panel opens, then select **Voice**.

From the **Voice** properties, select **Allow barge-in**.

3. Save the topic.

## Configure silence detection and timeouts

Silence detection lets you configure how long the agent waits for user input and the action it takes if no input is received. Silence detection is most useful in response to a question at the node level or when the agent waits for a trigger phrase to begin a new topic.

You can configure the [default timeouts for topics](#).

To override the defaults for a node:

1. Select the **More** icon (...) of the node, and then select **Properties**.

The **Question properties** panel opens.

2. Select **Voice** and make adjustments to the following settings:

 Expand table

| Silence detection timeout option | Description                                                         |
|----------------------------------|---------------------------------------------------------------------|
| Use agent setting                | Node uses the <a href="#">global setting</a> for silence detection. |
| Disable for this node            | The agent waits indefinitely for a response.                        |
| Customize in milliseconds        | The agent waits for a specified time before repeating the question. |

## Fallback action

You can configure some behaviors as a fallback action:

- How many times the agent should repeat a question
- What the reprompt message should say
- What the agent should do after a specified number of repeats

## Speech input

For speech input you can specify:

- **Utterance end timeout:** How long the agent waits after the user finishes speaking
- **Speech recognition timeout:** How much time the agent gives to the user once they start responding

To configure silence detection behavior when your agent waits for a trigger phrase, adjust the settings in the **On silence** system topic.

## Add a latency message for long running operations

For long backend operations, your agent can send a message to users to notify them of the longer processes. Agents on a messaging channel can also send a latency message.

 Expand table

| Latency message audio playback                   | Latency message in chat                           |
|--------------------------------------------------|---------------------------------------------------|
| Continues to loop until the operation completes. | Sent only once when the specified latency is hit. |

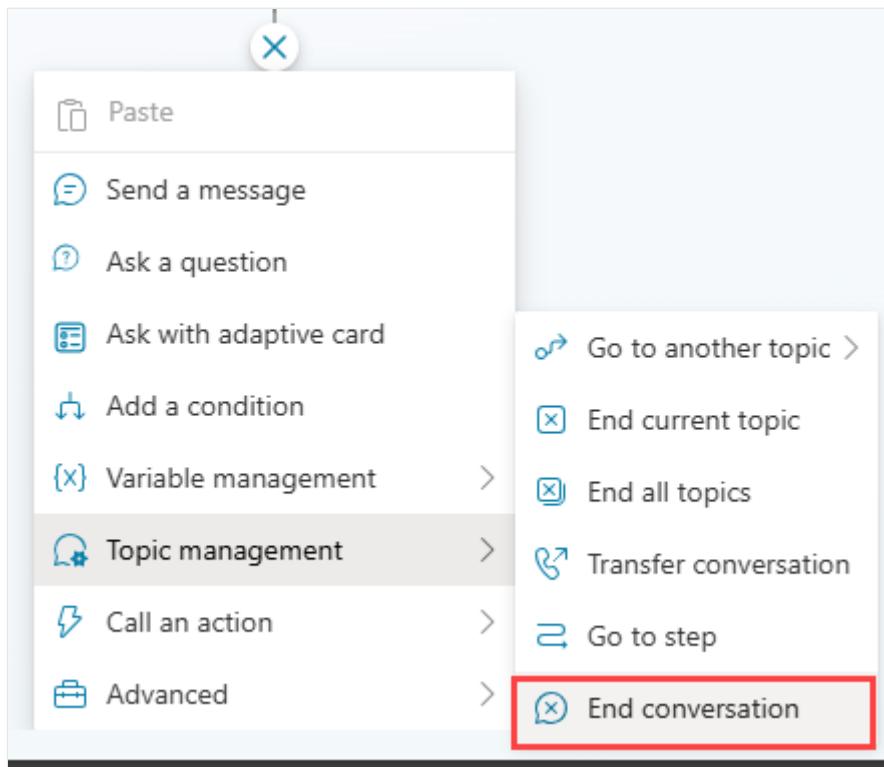
In Copilot Studio, your agent can repeat a message after triggering a Power Automate flow:

1. [Add an Action node that triggers a flow.](#)
2. Select the **More** icon (...) of the node, and then select **Properties**. The **Action properties** panel opens.
3. Select **Send a message**.
4. In the **Message** section, enter what you want the agent to say. You can use SSML to modify the sound of the message. The agent repeats the message until the flow is complete.

You can adjust how long the agent should wait before repeating the message under the **Delay** section. You can set a minimum amount of time to wait, even if the flow completes.

## Configure call termination

To configure your agent to end the call and hang up, add a new node (+) then select **Topic management > End conversation**.



## Format speech synthesis with SSML

You can use speech synthesis markup language (SSML) to change how the agent sounds when it reads messages out loud. For example, you can change the pitch or frequency of the spoken words, the speed, and the volume.

SSML uses tags to enclose the text you want to modify, similar to HTML. You can use the following tags in Copilot Studio:

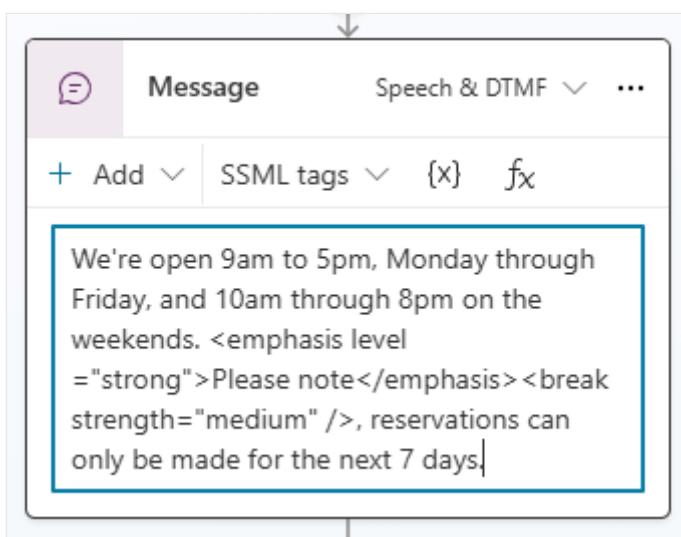
  Expand table

| SSML tag                                                                | Description                                                                                                                               | Link to speech service documentation    |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| <code>&lt;audio src="_URL to an audio file_"/&gt;</code>                | Add the URL to an audio file within the tag. The file must be accessible by the agent user.                                               | <a href="#">Add recorded audio</a>      |
| <code>&lt;br/&gt;</code>                                                | Insert pauses or breaks between words. Insert break options within the tag.                                                               | <a href="#">Add a break</a>             |
| <code>&lt;emphasis&gt; Text you want to modify &lt;/emphasis&gt;</code> | Add levels of stress to words or phrases. Add emphasis options in the opening tag. Add the closing tag after the text you want to modify. | <a href="#">Adjust emphasis options</a> |
| <code>&lt;prosody&gt; Text you want to modify &lt;/prosody&gt;</code>   | Specify changes to pitch, contour, range, rate, and volume. Add prosody options in the opening tag.                                       | <a href="#">Adjust prosody options</a>  |

| SSML tag                                                                                                | Description                                                                                  | Link to speech service documentation      |
|---------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|-------------------------------------------|
|                                                                                                         | Add the closing tag after the text you want to modify.                                       |                                           |
| <code>&lt;lang xml:lang="xx-XX"&gt;</code><br><i>Text you want to modify</i> <code>&lt;/lang&gt;</code> | Adjust the speaking language within the same message when using a multilingual neural voice. | <a href="#">Adjust speaking languages</a> |

### ⓘ Note

When using the `<audio src="_URL to an audio file_"/>` tag, if the URL is stored in a variable, the URL must be encoded before it's inserted into the audio src SSML tag in the message. We recommend using the [EncodeHTML](#) PowerFx function to encode the URL when assigning it to a variable in the assign action.



## Find and use a tag

SSML uses tags to enclose the text you want to modify, like HTML.

You can use the following tags in Copilot Studio:

1. With a **Message** or **Question** node selected, change the mode to **Speech & DTMF**.
2. Select the **SSML tags** menu and select a tag.

The message box is populated with the tag. If you already have text in the message box, the tag's code is appended to the end of your message.

3. Surround the text you want to modify with the opening and closing tags. You can combine multiple tags and customize individual parts of the message with individual tags.

### Tip

You can manually enter SSML tags that don't appear in the helper menu. To learn more about other tags you can use, see [Improve synthesis with Speech Synthesis Markup Language](#).

## Transfer a call to a representative or external phone number

You can have the agent transfer the call to an external phone number. Copilot Studio supports blind transfer to a PSTN phone number and the Direct routing number.

To transfer to an external phone number:

1. In the topic you want to modify, add a new node (+). In the node menu, select **Topic management** and then **Transfer conversation**.
2. Under **Transfer type**, select **External phone number transfer** and enter the transfer number.
3. (Optional) add an SIP UUI header to the phone call.

This header is a string of `key=value` pairs, without spaces or special characters, displayed for external systems to read.

- a. Select the **More** icon (...) of the node, and then select **Properties**. The **Transfer conversation properties** panel opens.
- b. Under **SIP UUI header**, enter the information you want to send with the call transfer. Variables aren't supported when transferring to an external phone number.

### Caution

Only the first 128 characters in the string are sent.

The header only accepts numbers, letters, equal signs (=), and semicolons (;). All other characters, including spaces, braces, and brackets, or formulas aren't supported and can cause the transfer to fail.

## Tip

Include a + in your phone number for the corresponding country code.

Transfer egress with SIP UUI for the target phone number must use *direct routing*. *Public switched telephone network (PSTN)* phone numbers don't support SIP UUI header transfers.

To transfer to a representative, see [Explicit triggers](#).

## Use voice variables

Copilot Studio supports the population of variables. You can use predefined variables, or create custom ones.

### Note

- For more information on how to use and create variables in Copilot Studio, see [Work with variables](#).
- For information about other activity and conversation variables available for voice-enabled agents, see [Variables for voice-enabled agents](#).

A voice-enabled agent in Copilot Studio supports context variables. These variables help you integrate your agent conversations with Dynamics 365 Customer Service when transferring a call.

For more information about context variables in Dynamics 365 Customer Service, see [Context variables for Copilot Studio bots](#).

This integration supports these scenarios with the following variables when you transfer:

 Expand table

| Variable                             | Type   | Description                                         |
|--------------------------------------|--------|-----------------------------------------------------|
| System.Activity.From.Name            | String | The agent user's caller ID                          |
| System.Activity.Recipient.Name       | String | The number used to call or connect to the agent     |
| System.Conversation.SipUiHeaderValue | String | SIP header value when transferring through a direct |

| Variable                                                            | Type    | Description                                                                                                                               |
|---------------------------------------------------------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                     |         | routing phone number                                                                                                                      |
| <code>System.Activity.UserInputType</code>                          | String  | Whether the agent user used DTMF or speech in the conversation                                                                            |
| <code>System.Activity.InputDTMFKey</code>                           | String  | The agent user's raw DTMF input                                                                                                           |
| <code>System.Conversation.OnlyAllowDTMF</code>                      | Boolean | Voice ignores speech input when set to <b>true</b>                                                                                        |
| <code>System.Activity.SpeechRecognition.Confidence</code>           | Number  | The confidence value (between 0 and 1) from the last speech recognition event                                                             |
| <code>System.Activity.SpeechRecognition.MinimalFormattedText</code> | String  | Speech recognition results (as raw text) before Copilot Studio applied its dedicated <a href="#">natural language understanding model</a> |

 **Note**

- An agent with large trigger phrases and entity sizing takes longer to publish.
- If multiple users publish the same agent at the same time, your publish action is blocked. You need to republish the agent after others finish their existing agent edits.

To learn more about the fundamentals of publishing, see [Key concepts - Publish and deploy your agent](#).

# Enable DTMF support for your voice-enabled agent

Article • 11/19/2024

Instead of relying on voice responses, dual-tone multi-frequency (DTMF) allows you to ask your agent users to use their phone keypad to select options or provide information.

You can set up a global command triggered at any point in the conversation if the agent user enters a specified single-digit key. Within individual topics, you can add DTMF input recognition for the responses provided in a **Question** node.

You can configure what key should be pressed for each option that the user can input and [how long the agent should wait before parsing the user's input](#).

If the key input by the agent's user isn't mapped to the node's options, the agent repeats the question or message and encourages the user to try again.

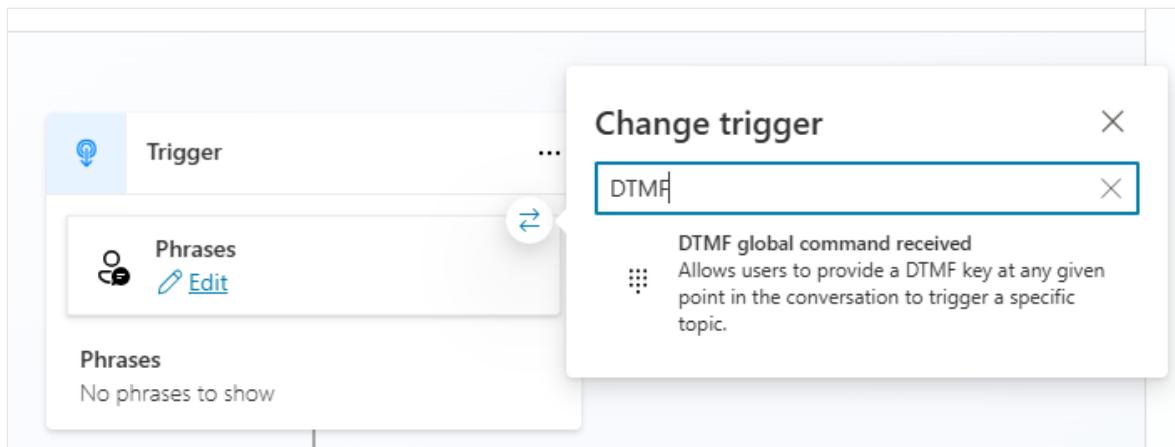
If the key input by the agent's user isn't mapped to the DTMF global triggers, the agent triggers the System topic, *Unknown dialpad press trigger (voice)*. You can customize it for your business logic.

System topics are necessary to help your copilot run effectively. Without them, your copilot may not work as intended. You can't delete system topics.

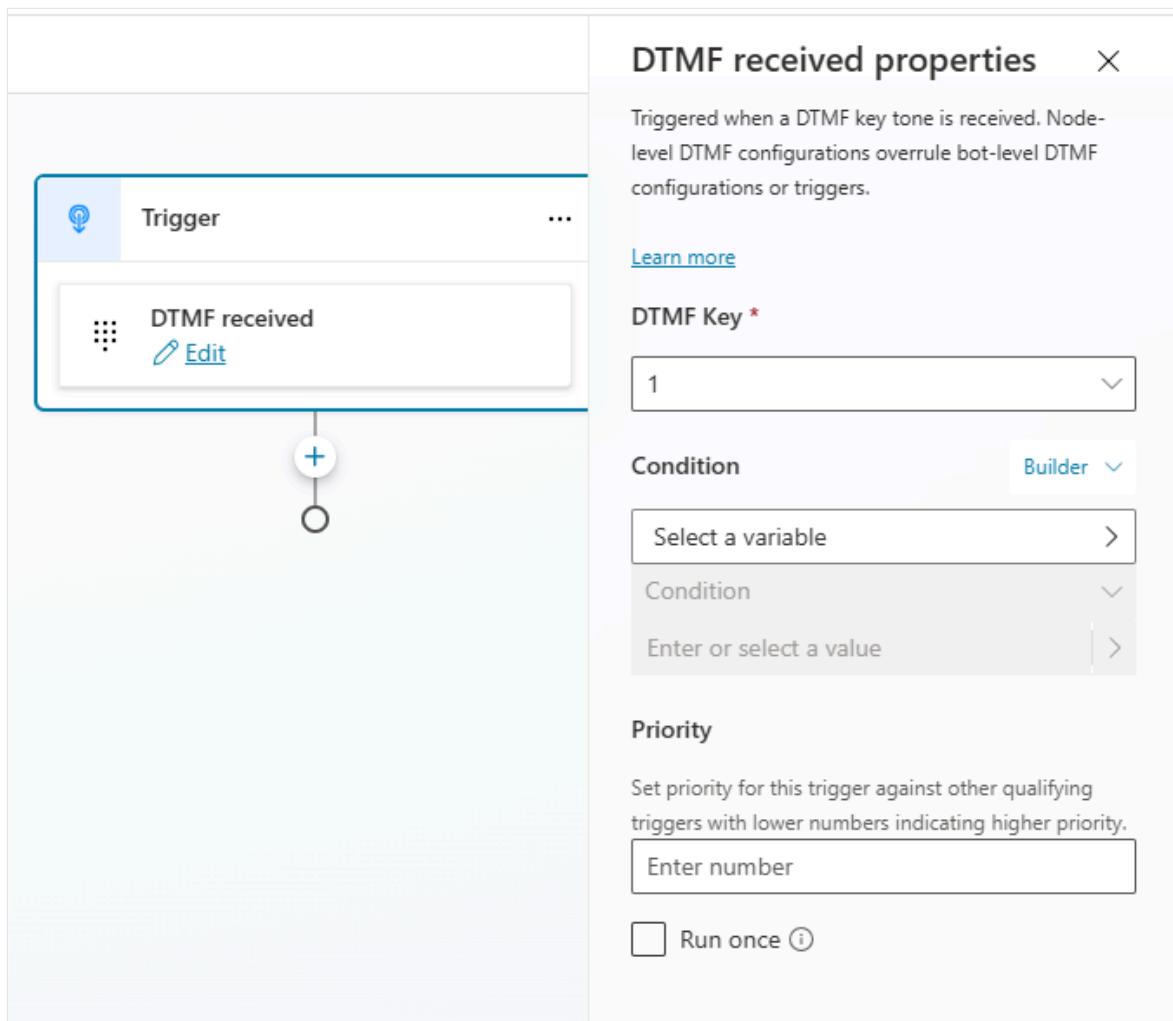
| Name                    | Trigger               | Description                                | Editing |
|-------------------------|-----------------------|--------------------------------------------|---------|
| Conversation Start      | On Conversation Start | This system topic triggers when the b...   |         |
| End of Conversation     | On Redirect           | This system topic is only triggered by ... |         |
| Escalate                | On Talk to Agent      | This system topic is triggered when t...   |         |
| Fallback                | On Unknown Intent     | This system topic triggers when the u...   |         |
| Multiple Topics Matched | On Select Intent      | This system topic triggers when the b...   |         |
| On Error                | On Error              | This system topic triggers when the b...   |         |
| Reset Conversation      | On Redirect           |                                            |         |
| Sign in                 | On Sign In            | This system topic triggers when the b...   |         |
| Silence detection       | Silence detected      | Triggered when a duration has passe...     |         |
| Speech unrecognized     | Unrecognized Speech   | Triggered when the user's speech inp...    |         |
| Unknown dial pad press  | On Unknown DTMF Key   | Triggered when a user's dial pad inpu...   |         |

## Enable DTMF support in a topic's node

1. With the topic you want to use as the global DTMF response open for editing, select the **Trigger** node.
2. Hover over or long select on the side of the **Trigger** node (where the current type of trigger is displayed) to see the **Change trigger** icon.
3. Select the icon and then enter "DTMF" to choose **DTMF global command received**.



4. Select **Edit** and in **DTMF received properties**, specify the type of recognized DTMF input, along with any conditions for the trigger, and its priority.



5. With a **Question node** selected, under the **Identify** section, select the type of entity that the agent should be identifying. Different entity types have distinct characteristics for supporting DTMF tones:

[] Expand table

| Entity (under Identify)        | DTMF support     | Details                                        |
|--------------------------------|------------------|------------------------------------------------|
| Multiple choice options        | ✓                | Manually assign a DTMF key to each option      |
| Options from a list variable   | ✓                | Automatically assign DTMF keys for each option |
| User's entire response         | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |
| Age                            | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |
| Boolean                        | ✓                | Automatically assign DTMF keys for each option |
| City                           | ✗                | DTMF not supported                             |
| Color                          | ✗                | DTMF not supported                             |
| Continent                      | ✗                | DTMF not supported                             |
| Country or region              | ✗                | DTMF not supported                             |
| Date                           | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |
| Date and time                  | ✗                | DTMF not supported                             |
| Data and time without timezone | ✗                | DTMF not supported                             |
| Duration                       | ✗                | DTMF not supported                             |
| Email                          | ✗                | DTMF not supported                             |
| Event                          | ✗                | DTMF not supported                             |
| Language                       | ✗                | DTMF not supported                             |
| Money                          | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |
| Number                         | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |
| Ordinal                        | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |
| Organization                   | ✗                | DTMF not supported                             |
| Percentage                     | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |
| Person name                    | ✗                | DTMF not supported                             |
| Phone number                   | 1<br>2<br>3<br>4 | Can accept multi-digit DTMF input              |

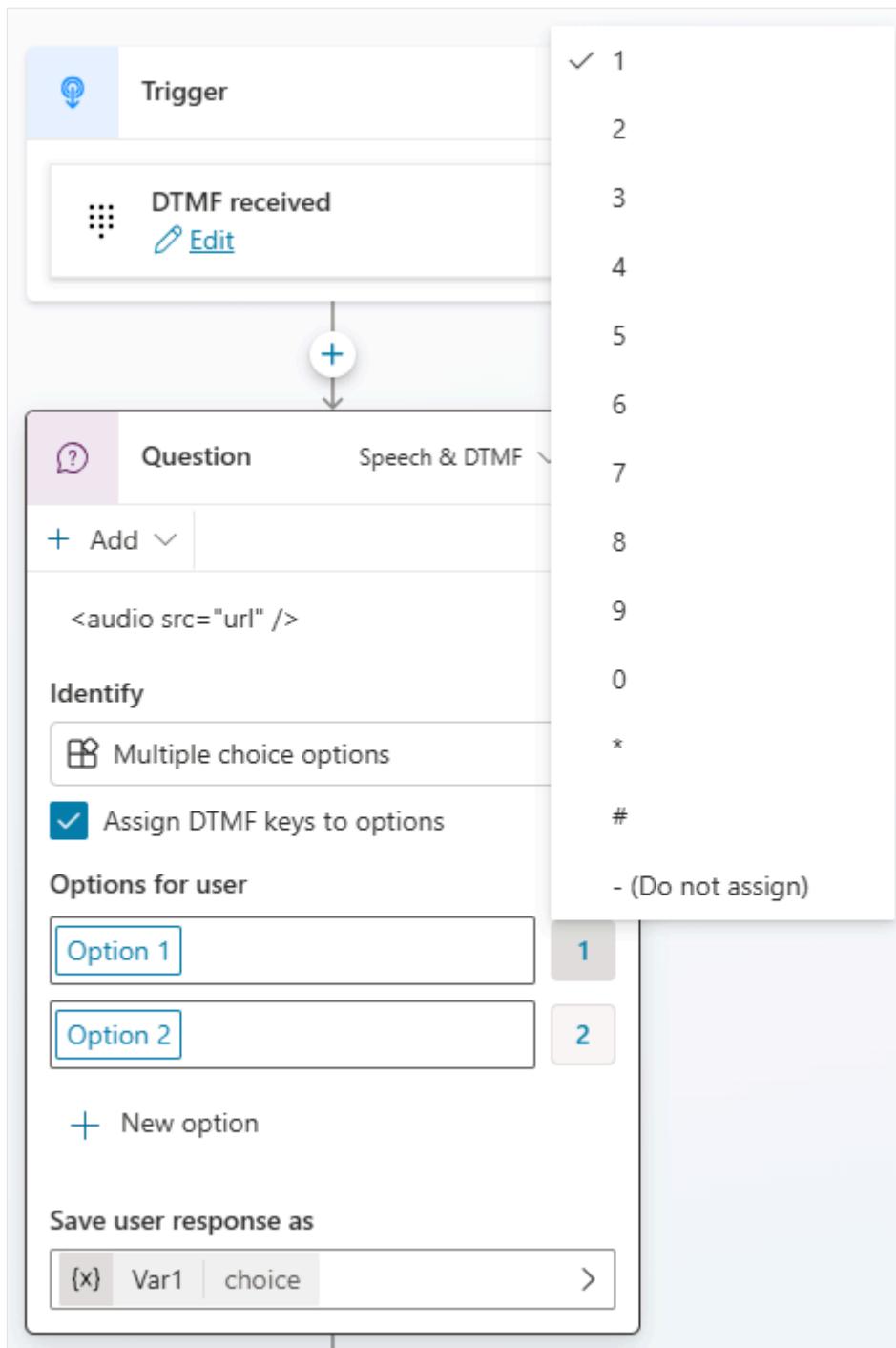
| Entity (under Identify) | DTMF support | Details                           |
|-------------------------|--------------|-----------------------------------|
| Point of interest       | ✗            | DTMF not supported                |
| Speed                   |              | Can accept multi-digit DTMF input |
| State                   | ✗            | DTMF not supported                |
| Street address          | ✗            | DTMF not supported                |
| Temperature             |              | Can accept multi-digit DTMF input |
| URL                     | ✗            | DTMF not supported                |
| Weight                  |              | Can accept multi-digit DTMF input |
| Zip code                |              | Can accept multi-digit DTMF input |

## Assign DTMF keys

For some entity types, you can specify if the options provided by the agent to the user are mapped to a DTMF key:

- For the **Multiple choice options** entity, manually assign DTMF keys to each option by selecting the **Assign DTMF keys to options** checkbox. When this checkbox is selected, you see an icon representing a phone's keypad next to the **Options for user** heading.

For each option you add, select the - icon and specify the DTMF signal that the option should be mapped to.



- For the **Options from a list variable** and **Boolean** entity types, the DTMF keys can be automatically assigned to each option.

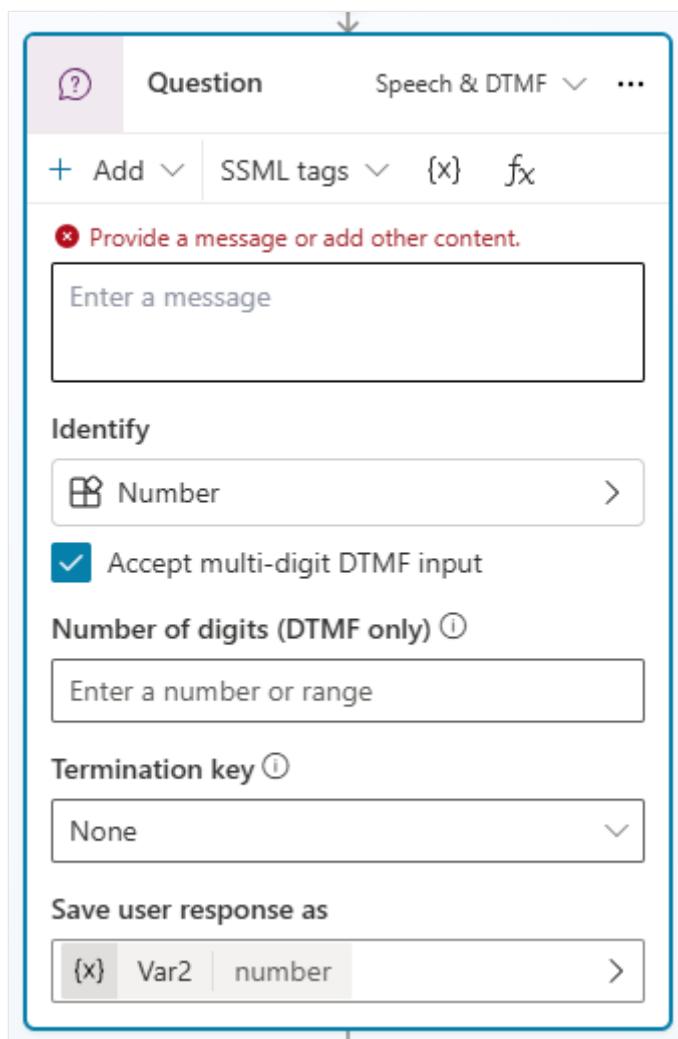
The agent assigns keys in the order that the options are in, from 0 to 9.

You can have the agent automatically read out the mapped options:

- Select the **More** icon (...) of the **Question** node you want to configure, and then select **Properties**. The **Question properties** panel appears.
- Select **Entity recognition**, and then select whether the agent should read out the options to the agent user.

# Multi-digit DTMF input

For entities that support multi-digit DTMF input, you can specify if there should be a minimum or maximum number of digits in the agent user's valid response. These multi-digit DTMF input entities include most entity types that extract a number from the agent user's input. You can specify a range for the number of digits that are considered valid (such as "2-5"), or a fixed length (such as "5").



If the entered response from the agent user has fewer or more digits than specified, then the agent repeats the question.

If you don't specify the number of digits, then the agent accepts any length of digits and continues the topic's flow.

You can also specify if the agent should wait for a specific "termination" key to be pressed to indicate the agent user finished entering their response. You can select no key, the asterisk key (\*), or the hash key (#). The termination character is optional, and isn't included when counting the length of the input.

## Agent-level DTMF recognition

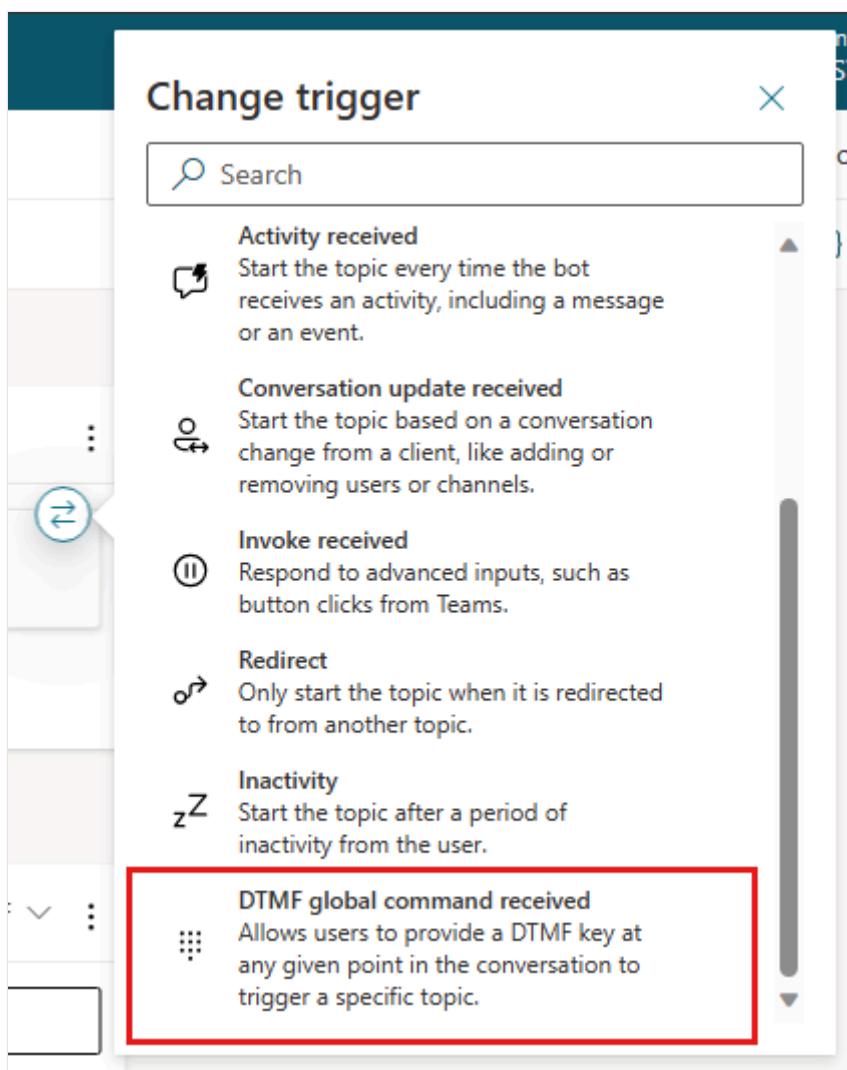
You can configure what key should be pressed for each option that the user can input and how long the agent should wait before parsing the user's input.

### ⓘ Important

The **DTMF received** trigger can only accept single DTMF keys. The trigger won't be called if the agent user enters the key when responding to:

- A Question node that accepts multi-digit DTMF
- A Question node that accepts single-digit DTMF where one of the allowed options is mapped to the same key

1. With the topic you want to use as the global DTMF response open for editing, select the **Trigger** node.
2. Hover or long-select on the side of the **Trigger** node (where it shows the current type of trigger) to see the **Change trigger** icon.
3. Select the icon and then choose **DTMF global command received**.



4. In the DTMF received properties panel that opens, specify the DTMF input that should be recognized along with any conditions for the trigger and its priority.

## DTMF caching

DTMF caching allows an agent user to input keys without waiting for each question or prompt from the agent. For example, caching is useful for agent users who are familiar with the agent's usual conversation path, and want to go straight to a specified menu.

With DTMF caching, each key press is cached in series, and automatically input for each consecutive prompt.

For a single-digit DTMF node, all keys input after the first one are cached for the next turn.

For multi-digit DTMF nodes, the agent follows a set of rules that determine how and what to cache based on these settings:

[Expand table](#)

| Scenario                                                                                                                                                                             | Digit length | Condition                          | User input                                 | Outcome                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Agent author didn't set a termination character so the timeout is ignored. When the user hits the max length, the key after max length (the last digit) is cached for the next turn. | 3            | Term time is 0 or Term key is none | 1, 2, 3, 1, ...                            | The first three digits are returned immediately, while the fourth digit is cached for the next turn.                                                                                 |
| Agent author set the termination character. The key input after timeout is cached for the next turn.                                                                                 | 3            | Term key is #                      | 1, 2, 3, 1                                 | The first three digits wait for the termination timeout before being returned. The fourth digit is cached for the next turn.                                                         |
| When the user hits the max length, the termination timeout runs.                                                                                                                     | 3            | Term key is #                      | 1, 2, 3, then during the timeout window, 1 | The first four digits are returned; however as this question only accepts three digits, the entire input sequence is considered to be invalid, and the agent reprompts the question. |
| User doesn't hit the max length (valid length). The                                                                                                                                  | 2-5          | Doesn't matter if a                | 1, 2, 3, then after the                    | The first two digits wait for the interdigit timer before                                                                                                                            |

| Scenario                                                          | Digit length | Condition           | User input | Outcome                                                                       |
|-------------------------------------------------------------------|--------------|---------------------|------------|-------------------------------------------------------------------------------|
| key input after the interdigit timer is cached for the next turn. |              | term key is defined | timeout, 1 | being returned, and the third and fourth digits are cached for the next turn. |

DTMF caching is handled in the following ways:

- If the next turn doesn't allow barge-in, the cached key is dropped.
- If the next turn allows barge-in, the key is handled according to its flow:
  - If the flow consists of multiple message nodes, but is followed by a multi-DTMF question node, keys are cached for this multi-DTMF question only. The global DTMF topic aren't triggered.
  - If the flow consists of multiple message nodes, but is followed by a single DTMF question node, the key is cached for a single DTMF key. Also, the DTMF global topic is cached. However, if they overlap, the single DTMF question node wins.
  - If the flow consists of multiple message nodes without any question, the key is cached for the DTMF global topic.

### Tip

To avoid accidental caching, do the following:

- Place the node with the barge-in disabled message right after the question node.
- Use a multi-DTMF question node with a longer term timer (notice this increases the latency to the user in the absence of a termination key).

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Create custom lexicons

09/08/2025

One of the difficulties in designing a voice-enabled agent is having the agent correctly pronounce terminology. Text-to-speech systems might mispronounce some terminology, such as industry-specific terms. As a conversational designer, you know your callers. You can anticipate slang, abbreviations, or alternative names. When captured in a confirmation response, you might want to change them before the system speaks them back to the user.

Lexicons help in these situations. Lexicon files give reading rules to speech synthesis engines to:

- Pronounce words in a specific way using the phonetic alphabet
- Change the text spoken by using an alias

## ➊ Note

- The information in this article requires configuration of a voice-enabled agent and channel in Copilot Studio and Dynamics 365 Contact Center.
- Custom lexicon files don't have native integration with Dynamics 365 Contact Center yet. This article gives guidance on how to create your own.

## Create the lexicon file

Currently, there aren't any tools to generate custom lexicon files, so you need to author them by hand. See the Microsoft Azure AI Speech documentation for the [correct XML syntax for a lexicon file](#).

To validate your file, use the [SDK provided for validation](#) or Speech Studio.

Use Speech Studio to create your lexicon file. When complete, save your file. Any parsing errors that might arise because of malformed lexicon syntax are noted when you save, which provides a layer of validation. Then download the lexicon file (.xml) to your chosen directory.

## Store the lexicon file

Store and reference lexicon files by URI so you can use them in any text-to-speech system. However, use a cloud storage option, because Copilot Studio doesn't natively support these files.

## Set up an Azure storage account

You need to create an Azure storage account. Ensure that the subscription, resource group, region, and resource name of the new storage account according to your organization's policies. We recommend using the following settings:

- For Primary Service, select **Azure Blob Storage or Azure Data Lake Storage Gen 2**.
- Select **Premium** for the Performance.

For more information, see [Create an Azure storage account](#).

## Set up the storage container

We recommend using the **Static Website** as the storage container for your uploaded grammar files. The storage container provides you the primary endpoint and secondary endpoint for the website.

After uploading your grammar file, select the file from directory to view the properties and details of the file. Save the URL for the file, which should be in the following format:

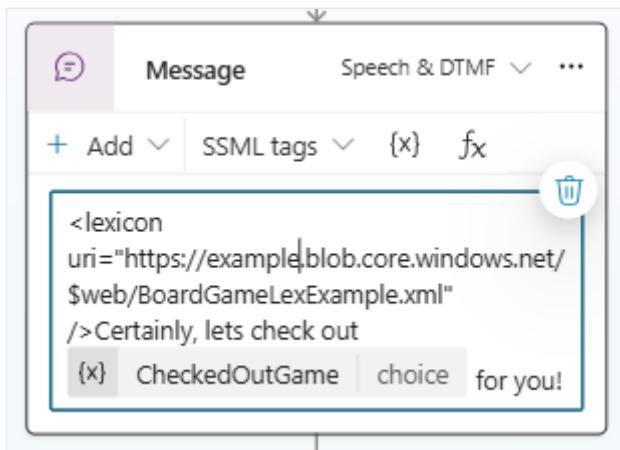
```
https://{{resourceName}}.blob.core.windows.net/{{$web}}/{{lexiconFileName}}
```

For more information, see [Create a container](#).

## Reference a lexicon file from Copilot Studio

Lexicon files should be referenced within Copilot Studio **Question** nodes. These nodes are where complex words or abbreviations are spoken from dynamic content. This configuration requires the use of phonemes or aliases, respectively, to remediate.

Within a **Question** node, make sure that **Speech + DTMF** option is **selected**. In the text field, add the reference to your custom lexicon file by writing the following SSML: `<lexicon uri={{url}}/>`, where "url" is the location of the lexicon file in the blob container obtained in the prior steps.



Once complete, test the lexicon file by calling into the voice-enabled agent.

## Known limitations

- **Menu Options:** When the “read options aloud” option is selected as “Read Out,” meaning that menu options will be read out to the caller over voice channels using text-to-speech capabilities, the lexicon won’t apply to any text in those menu items.
- Currently, lexicons can only be used when responses are generated by topics that the maker has designed. If generative AI is used to generate a response, the lexicon file isn’t used. For example, if [generative orchestration](#) is turned on and generative orchestration maps to a topic that contains message and question nodes that have a lexicon file, the lexicon file are used to generate a response.

# Use external speech grammars

09/17/2025

For Interactive Voice Response (IVR) applications, and broader speech recognition tasks for voice applications, a constrained list, or "grammar"-based recognition offers significant advantages. It far outperforms traditional semantic-based speech recognition used in modern speech-to-text (STT) AI engines in terms of accuracy, performance, and cost. This performance increase is because grammar-based recognition can constrain the recognition output to a predefined set of rules, which bolsters accuracy.

Grammars adhere to the Speech Recognition Grammar Specification (SRGS), as detailed in the [W3C specification](#). When a request comes into the engine, it converts spoken audio ("utterances") into text. The engine then compares the recognized text against the grammar and any associated artifacts, such as pronunciation lexicons. This process provides either a literal transcription or an interpretation that is constrained to the information provided within the grammar. Extra logic, such as ECMAScript built into the grammar, can further refine the interpretation.

Constrained speech recognition is ideal for:

- Recognizing constrained lists (addresses, stock tickers, zip codes, department names, and such).
- Alphanumeric string recognition (tracking numbers, account numbers, confirmation codes, and such).
  - Including having positional constraints. For example, the first two characters of member ID start with AN, FD, NT. Another example of positional constraints is a vehicle identification number.
- Alphanumeric or digit recognition with checksums or similar constraints. For example, credit card numbers where there's a Luhn checksum.
- Directed dialog applications whereby specific words or phrases should be uttered.

## Authoring speech grammars

Constrained speech grammars are written using Grammar XML (GrXML). Like any XML document, a grammar file must begin with a header that specifies certain characteristics of the grammar. The main body of a grammar file consists of grammar rules that define the spoken words recognized by the grammar, and the corresponding variable values that are returned for the recognized items.

### Grammar file header

The header in a grammar file consists of the XML declaration, and the `<grammar>` element specifying the document language, root, and namespace.

```
XML

<?xml version="1.0" encoding="UTF-8" ?>

<grammar xmlns="http://www.w3.org/2001/06/grammar"
version="1.0" xml:lang="en-US" root="YesNo"
tag-format="swi-semantics/1.0">
```

### XML declaration and encoding type

The first element in the header is always the XML declaration. This element specifies the version of XML used in the document (1.0 or 1.1). It also specifies the encoding that applies to the document, which determines the languages that can or can't be used.

The version and encoding are required attributes. Use any encoding appropriate to your preferences (for example, your computer setup, text-processing application, and so on). The constrained speech recognition engine doesn't care which encoding you use.

A short list of typical encodings for various languages is shown in the following table:

[+] Expand table

| Encoding   | Description                                             |
|------------|---------------------------------------------------------|
| ISO-8859-1 | Latin-1. Used for English, French, German, and Spanish. |
| UTF-8      | Used for all languages.                                 |
| UTF-16     | Used for all languages.                                 |

| Encoding             | Description                 |
|----------------------|-----------------------------|
| Big5                 | Used for Cantonese (ce-HK). |
| GB                   | Used for Mandarin (zh-TW).  |
| Shift-JIS and EUC-JP | Used for Japanese.          |
| KSC and EUC-KR       | Used for Korean.            |

Most languages can be represented in more than one encoding.

At runtime, the system automatically converts the grammar file encoding into UTF-16 format using the International Components for Unicode (ICU) libraries. The official website of the Unicode consortium is <http://site.icu-project.org/>.

## Language, namespace, and semantic tag format

The second element in the header is the `<grammar>` element, whose attributes specify default information for the document. The required attributes are:

- `xml:lang`: Specifies the identifier for the default human language to be used, as defined in the Request For Comments (RFC) document [RFC 3066](#) on the IETF web site.
- Microsoft supports a wide range of languages. The language chosen must be compatible with the grammar encoding type.
- `version`: Specifies the version of GrXML (1.0).
- `xmlns`: Designates the grammar namespace. For GrXML grammars, this designation is always <http://www.w3.org./2001/06/grammar>.
- `tag-format`: Defines the format used for scripts within `<tag>` elements in the main body of the grammar to assign values.

The tag-format must be one of these strings:

Expand table

| Value                  | Format of semantic tags                                                                                                  |
|------------------------|--------------------------------------------------------------------------------------------------------------------------|
| swi-semantics/1.0      | Tag syntax (used if tag-format isn't defined). This is known as <code>swi</code> syntax (for SpeechWorks International). |
| semantics/1.0          | W3C script tag syntax.                                                                                                   |
| semantics/1.0-literals | W3C string literals tag syntax.                                                                                          |

### ① Note

- Strictly speaking, the tag-format attribute isn't required if your grammar doesn't use the `<tag>` element. However, most grammars do use `<tag>` to assign values, so Microsoft strongly recommends that you specify the tag-format.
- Always point GrXML attributes and elements (such as `xmlns`) to the namespace <http://voicexml.site.com/grammar>.

## Dictionaries

In some cases, the grammar might need to include words or phrases that the constrained speech recognition engine can't parse normally. For example, a name might be said and spelled differently (like the city "Worcester," which might be pronounced "wih-sta").

You can use the `<lexicon>` element to import dictionaries that map utterances to matching text in the grammar file.

## Grammar file main body

The main section of a grammar file contains the rules that actually define the grammar: the spoken words and phrases to be recognized, and the values to be returned to the main application for each recognized item.

## Rules

The main body of a grammar file consists of rules defined using the GrXML `<rule>` element. Each rule has a unique identifier. Each rule lists the words and phrases it recognizes as text within an `<item>` element or `<token>` element. These elements might themselves be nested within other GrXML elements:

- The `<one-of>` element presents a list of acceptable alternatives, only one of which is required to activate the rule.
- The `<ruleref>` element refers to another rule, as to a subroutine.
- The `<tag>` element specifies actions to be carried out or values to be assigned to a variable; it might include a script written in the tag-format language.

When the user utters a word or phrase that is covered by the rule, the rule executes the actions, value assignments, or other code defined for that utterance.

## Root rule

The `root` rule is the first rule in the file, unless otherwise specified in the header. It serves as the default op-level rule. When the grammar is referenced without specifying the rule to look up, this root rule is the first one consulted.

## Rule scope

Each rule within the main body of a grammar file is assigned as a `scope`, which indicates whether it can be referenced independently from external files (public), or only by another rule within the same grammar (private). All rules are private by default, unless defined as public.

When the rule is public, its ID attribute can be used as an anchor for references from other documents. For example, consider the following syntax:

```
<grammar src="../grammars/universals.grxml#YesNo"/>
```

When this grammar element is invoked, it refers directly to the public "YesNo" rule within the `universals.grxml` file, regardless of whether it's the file's root rule.

### ① Note

The root rule of a grammar file might be private. This means it can't be referenced independently. However, it's used by default as the entry point to the grammar when the grammar file itself is invoked.

## Extract meaning and return results

### ① Note

The `SWI_meaning` key should contain the information being returned back to the voice-enabled agent operating within Copilot Studio.

The `SWI_meaning` key contains the semantic meaning of a recognized phrase. It can only be set for the root rule. This key is included in the `swirec_extra_nbest_keys` list by default, so it appeared in the XML result if your grammar sets this key.

`SWI_meaning` filters out redundant answers so that entries on the n-best list are truly distinct. Eliminating redundancy improves confidence scores, and improves usefulness of the n-best list.

When one recognized phrase is similar to another in the grammar, it often has a low confidence score, because the constrained speech recognition engine is unsure which phrase is correct. When `SWI_meaning` is used properly, the constrained speech recognition engine groups redundant interpretations into the same slot on the n-best list. In the following example, `SWI_meaning` is set to "direct calls home" whether the recognized phrase is "direct my calls home" or "please direct my calls home."

Without `swi_meaning`, the grammar might produce the following n-best list:

 Expand table

N	Text
1	direct my calls to my car phone
2	direct calls to my car

N	Text
3	send calls home
4	please send my calls to the office
5	send my calls to the office
6	direct calls to my home

When `SWI_meaning` is used, the constrained speech recognition engine arranges the n-best list by the meaning of the interpretation rather than the exact phrase spoken, so that entries on the n-best list are truly distinct:

Expand table

N	Text	Top-level SWI_meaning key
1	direct my calls to my car phone	direct calls car
	direct calls to my car	direct calls car
2	send calls home	direct calls home
	direct calls to my home	direct calls home
3	please send my calls to the office	direct calls work
	send my calls to the office	direct calls work

The constrained speech recognition engine sets `SWI_meaning` automatically, even if it isn't explicitly set in a script within the grammar.

If `SWI_meaning` isn't explicitly defined on the root, it's constructed by concatenating all the keys defined in the root and their values. However, this construction doesn't apply to any keys beginning with `SWI_`, for example, such as `SWI_literal`. The key/value pairs are first sorted alphabetically. The reasoning is that as far as the application is concerned, the set of keys returned *is* the sentence's meaning.

If there are no keys, the results depend on whether you're using SISR or SWI semantics. With SISR, the `SWI_meaning` key isn't set if there are no keys. In contrast, with SWI semantics, `SWI_meaning` is set to the following:

```
{SWI_literal:<literal>}
```

If `SWI_meaning` is an object, it's converted to a string representation.

While the application can access `SWI_meaning`, it's more often the case that it uses other key/value pairs defined specifically for it.

## Host speech grammars via Azure Storage

Constrained speech recognition via speech grammars is supported in Copilot Studio. However, directly authoring, testing, and hosting them aren't supported. For grammar hosting, you have to use Microsoft Azure Storage to provide a trusted and secure connection between the voice-enabled agent and grammar storage.

### Set up an Azure storage account

You need to create an Azure storage account. Ensure that the subscription, resource group, region, and resource name of the new storage account according to your organization's policies. We recommend using the following settings:

- For Primary Service, select **Azure Blob Storage or Azure Data Lake Storage Gen 2**.
- Select **Premium** for the Performance.

For more information, see [Create an Azure storage account](#).

### Set up the storage container

We recommend using the **Static Website** as the storage container for your uploaded grammar files. The storage container provides you the primary endpoint and secondary endpoint for the website.

After uploading your grammar file, select the file from directory to view the properties and details of the file. Save the URL for the file, which should be in the following format:

[https://{{resourceName}}.blob.core.windows.net/\\$web/{{grammarFileName}}](https://{{resourceName}}.blob.core.windows.net/$web/{{grammarFileName}})

For more information, see [Create a container](#).

## Authenticate the constrained speech recognition engine

In order for constrained speech recognition to perform within a voice-enabled agent, the system needs to authenticate with the storage account created in the previous step as a trusted location. This requires the **Storage Blob Data Reader** role.

For more information, see [Assign Azure roles using the Azure portal](#).

Sign in to Azure portal and open an Azure Cloud Shell session and execute the following command to create a constrained speech recognition engine service principal in your tenant.

```
az ad sp create --id e0e7bef0-777c-40ef-86aa-79d83ba643c7
```

ⓘ Note

When you search for the service principal, it contains "NRaaS" in the name.

## Employ constrained speech in Copilot Studio

### Create an external entity

An **entity** within Copilot Studio can be thought of as a unit of information that represents a certain type of a real-world subject. For example, a phone number, postal code, city, or even a person's name. With the knowledge granted by entities, an agent can smartly recognize the relevant information from a user input and save it for later use. In our scenario, that recognition is performed through a constrained speech grammar.

Specifically, external entities are employed to reference speech grammars. In order to create an external entity, with your voice-enabled agent open, navigate to **Settings > Entities > Add an entity > Register an external entity**.

Enter the following information:

- Name: Grammar URL in the form of [https://{{resourceName}}.blob.core.windows.net/\\$web/{{grammarFileName}}?constrainedrequired=true](https://{{resourceName}}.blob.core.windows.net/$web/{{grammarFileName}}?constrainedrequired=true)

ⓘ Note

The URL is case sensitive.

The default recognition mode is Speech Only. For alternative grammar configurations, refer to the following table:

[Expand table](#)

Type	Query Parameter	Example
Speech Only	None	<a href="https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true">https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true</a>
DTMF	&mode=dtmf	<a href="https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true&amp;mode=dtmf">https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true&amp;mode=dtmf</a>
Speech or DTMF (Same Grammar File)	&mode=speechdtmf	<a href="https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true&amp;mode=speechdtmf">https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true&amp;mode=speechdtmf</a>
Speech or DTMF (Different Grammar Files)	&mode=speechdtmf&dtmfgrammar={{grammarURL}}	<a href="https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true&amp;mode=speechdtmf&amp;dtmfgrammar=https://{{resourceName}}.blob.core.windows.net/\$web/{{DTMFgrammar}}">https://{{resourceName}}.blob.core.windows.net/\$web/{{grammarFileName}}?constrainedrequired=true&amp;mode=speechdtmf&amp;dtmfgrammar=https://{{resourceName}}.blob.core.windows.net/\$web/{{DTMFgrammar}}</a>

- Description: A simple description of the grammar, which is referenced as the entity name in the selector on the canvas. For example, "credit card number."

- Data type: Choose the **Record** data type, and define the schema of the expected tags in response. For example, if the grammar returns `SWI_meaning` and `city`, the **Record** schema looks like this:

```
yml

kind: Record
properties:
 city: String
 SWI_meaning: String
```

Once you select **Save**, the entity appears in the list. In the authoring canvas, go to a [Question node](#). Just like with traditional entities, select the external entity (attached to a grammar) that you want the agent to recognize as a result of the user's response to the prompt.

## Runtime behavior

When a voice-enabled agent runs and encounters the logic using an external grammar, the voice-enabled agent reaches out to the Azure storage account and retrieves the grammar for interpretation. The agent then matches what the user said against the constraint applied within the grammar. If a match succeeds, the system returns the response in the **Record** variable, according to the schema defined in the external entity.

## Canvas logic

The result saved in the node's variable is always a **Record** type as defined in schema of the external entity. Authors can use this **Record** variable to access keys as defined in the schema, like `variableName.SWI_meaning` or `variableName.city` via dot notation.

## Debugging

### Error codes

[+] Expand table

Error	Definition
400	Bad request
401	Unauthenticated
403	Forbidden
404	No Speech
408	No Input Timeout
418	Session Time out
419	No active resources – missing a grammar
500	Internal Error – report to Microsoft in a support ticket

## SWI\_Literal

For many grammars that work with the constrained speech recognition engine, the `SWI_Literal` feature produced the literal statement the user uttered, not the interpreted result. Log this as one of the outputs in Copilot Studio to be captured for debugging purposes.

## Known limitations

The solution has the following limitations:

- Max size of individual grammar file, which is currently limited to 100 MB.
- Variable passing isn't supported.
- The storage account must be located within the same tenant as the agent.
- Size of the URL can't exceed 500 characters.
- Only Azure storage account endpoints are allowed.
- Subgrammars can only be hosted in the same storage account (using same FPA for authorization).

- Secure XML parser (that is, DTD is not allowed, and must validate against SRGS/SISR schema).
- Only NLSML output format is supported internally.
- The legacy `swirec_simple_result_key` param has no effect and all tags are returned.

## Legal

Constrained Speech Recognition systems are processed by the Microsoft Dynamics service. By using this experience, you agree to the [Dynamics Terms](#).

# Remove sensitive data

10/15/2025

Agents collect sensitive information while interacting with both internal and external users. These interactions lead to data being stored across various locations in your organization's tenant. It's crucial to ensure security and compliance of this data with your organization's policies.

The **Sensitive data** setting in Copilot Studio enables you to protect sensitive information such as customer PINs, account numbers, credit card details, and protected health information (PHI) data. This feature ensures that data protection doesn't interfere with user productivity or collaboration. Agent makers can mark variables as sensitive according to the specific requirements of their organization, to improve the security of the collected information.

 Note

The **Sensitive data** setting is only applied to voice-enabled agent interactions.

## Known limitations

If the customer's response to the agent's first question is flagged as sensitive, that content might not be redacted. This is due to a timing issue that affects redaction at the start of the conversation. However, all subsequent responses flagged as sensitive are redacted as described in this article.

## Prerequisites

- [Install the voice channel](#)
- [Set up inbound calling](#)
- A Copilot Studio voice-enabled agent
- The Power Platform Admin Center system administrator role, and the

## Configure sensitive data masking

Identify the information in your agent's conversational flow that might contain sensitive data. For example, the input to the agent is a credit card number. Sensitive data is handled using [variables](#). Currently, you can only denote sensitive data variables using the [Question node](#).

To initialize a variable as containing sensitive data, perform the following steps:

1. In a topic, add a **Question** node.
2. Assign a global variable.
3. Open the global variable's properties and turn on **Sensitive data**.

 **Note**

- When a sensitive data variable is assigned to a nonsensitive data variable, the nonsensitive variable automatically inherits the sensitive data properties.
- If you're using Application Insights to log activity, turn off **Log sensitive activity**. If this setting is turned on, sensitive data is logged.

## Runtime operation

When the customer's conversation with the agent enters a section where a sensitive-flagged variable is configured, the agent displays the following message in the transcription: "Entered a confidential section of the conversation." Recording, transcription, and data logging are paused until the conversation moves on to the nonsensitive section.

When the conversation moves past the sensitive information collection, the agent displays "Exited a confidential section of the conversation," followed by "Recording and transcription resumed." Recording, transcription, and data logging resumes for all nonsensitive portions of the interaction.

If the conversation is escalated to a customer service representative from the agent, the transcript and recording don't contain any instances of the sensitive information captured by the agent. To match the recording to the call's length, we insert silence into the recording.

Conversation transcripts are stored in the [ConversationTranscript](#) table. Any variables marked as sensitive appear as **Redacted** in the transcript to protect its content.

## Sensitive data suggestions

- Agent makers are responsible to mark the variables as sensitive wherever they anticipate sensitive information.
- Copilot Studio doesn't automatically redact sensitive data from **Question** or **Message** nodes that don't have sensitive variables assigned to them.
- Sensitive information isn't removed if a caller unexpectedly shares something sensitive and the variable capturing that response isn't marked as sensitive.

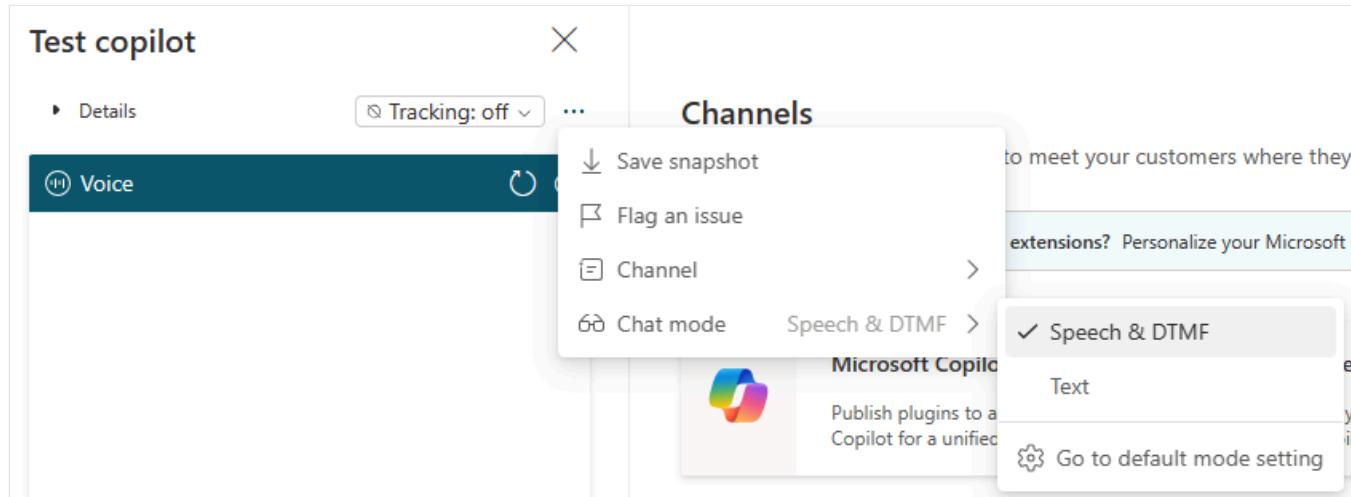
- Agent makers should account for potential latency in pausing the recording and transcription.
- Sensitive information redaction is restricted to Copilot Studio. For any external connections from Copilot Studio to Power Automate, Connectors, and such, customers are responsible for assessing redacted data with any relevant regulatory or compliance requirements.

# Test your voice-enabled agent

09/18/2025

You can test your agent with text inputs that simulate the user's input from speech or DTMF and see the speech response in the text output.

You can switch chat mode into **Speech & DTMF** to test voice features.



## Known limitations

- Speech recognition, since there's no speech input.
- Multi-DTMF grammar validation and timers, such as digit length.
- Silence detection timer, reprompt, and fallback behavior.

## Configure your test voice-enabled agent

When you use **Test agent** in chat in **Speech & DTMF** mode, you can get the agent's speech response through text output. A **Speech & DTMF** modality message is used over a text modality.

## Different ways to test

You can test for different scenarios to anticipate expected outcomes in the chat:

- Set voice variable values to simulate a real phone call.

For example, to simulate caller ID, you can manually set the variable value for *conversation start* by inputting "/debug set variable-name variable-value."

Variable name	Type	Description
Activity.From.Name	String	Customer caller ID
Activity.Recipient.Name	String	Voice agent number
Conversation.SipUuiHeaderValue	String	SIP header value for transfer-in context
va_CustomerLocale	String	Locale value for routing to another voice that supports the desired language. This value requires you to install the OC extension solutions.
va_SurveyConsent	Boolean (true/false)	<ul style="list-style-type: none"> <li>- If the customer consents, the call is automatically transferred to the post-call survey agent at the end of the personal digital assistant call.</li> <li>- If set to false, the call ends when the agent hangs up, which requires you to install the OC extension solutions.</li> </ul>

- If you created an SSML message for your voice-enabled agent, you can see the text output without SSML.
- If the question node is enabled for **Read out**, you see the text output for multiple choice options, closed list entity options, and dynamic list variables. For example, you might get *you can say book a flight or check flight status.*

Trigger

DTMF received

Question

Speech & DTMF

Add SSML tags {x} fx

Provide a message or add other content.

Enter a message

Identify

Multiple choice options

Assign DTMF keys to options

Options for user

New option

Save user response as

{x} Var2 choice

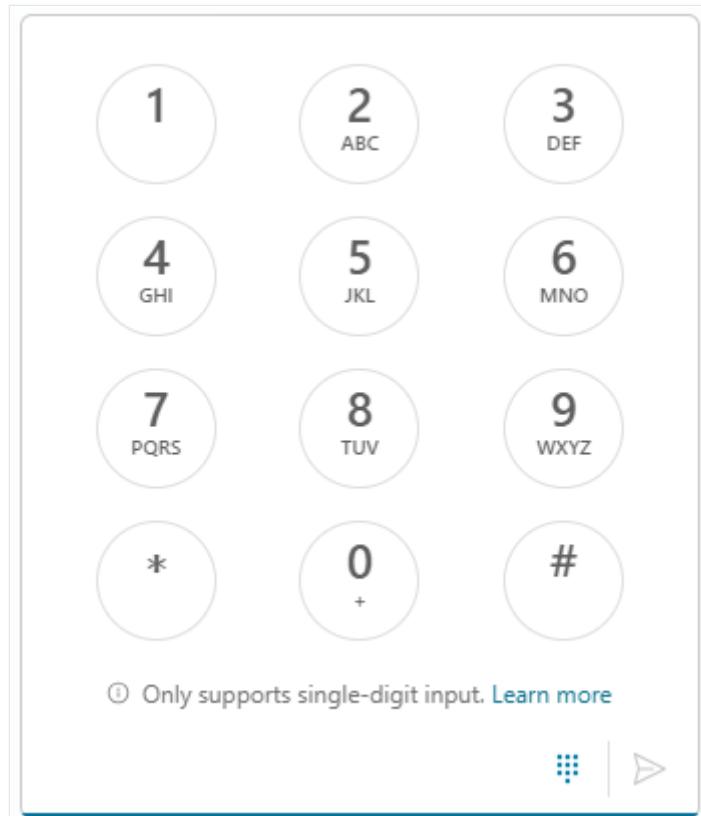
If options accept DTMF input, you see the output for each option as *Press 1 or say book a flight* or *Press 2 or say check flight status*.

- For messages with barge-in disabled, you see **Barge in is not allowed**.

Hello. Thank you for calling Please listen closely to the following options.

Just now Barge in is not allowed

- For a long running operation latency messages, you receive a standard agent message once. Instead of looping, the message displays until the operation completes when it plays back in the Telephony channel.
- For single DTMF, you can press a key in dial pad, or type in a command to trigger global commands, and the select menu options. You can type "/DTMFkey 1," which is treated as if the user provided a single DTMF option "1." Depending on the conversation state, the agent can either proceed with option 1's choice from the question node or trigger a global command, if applicable.



#### ! Note

Dialpad input only supports single-digit DTMF.

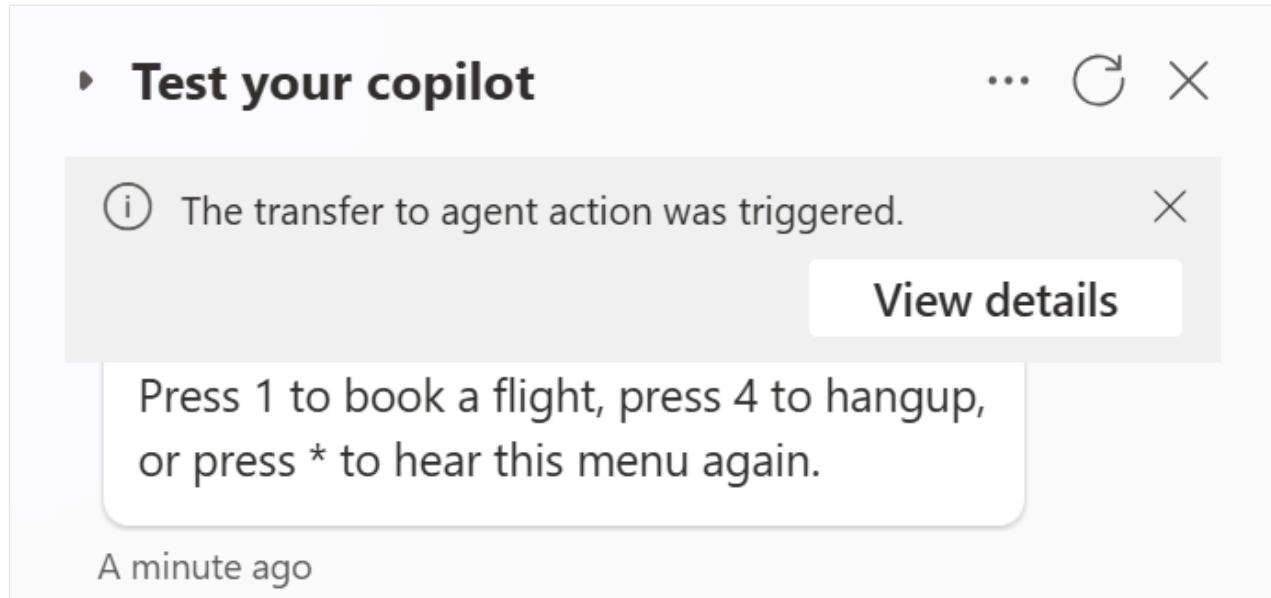
- For Multi DTMF, you can only use the command "/DTMFkey **DTMF keys**." For example, /DTMFkey 123, to simulate batched, multi-digit DTMF.

Speech input might skip the speech recognition part of the system. Test chat doesn't validate DTMF grammar, such as DTMF length, timer, term key, and DTMF caching.

#### ! Tip

Don't input DTMF keys with term key as saved responses with all the DTMF keys.

- When transferring to a phone number or personal digital assistant, contexts that are passed display in the popup. Select **View details** for more information.



## Related content

- [Troubleshoot voice-enabled agent issues](#)

# Use prompt modification to provide custom instructions to your agent

10/10/2025

Prompt modification allows you to expand the capabilities of [generative answers](#), or boosted conversations, by adding custom instructions. This article shows you how to create custom instructions to your agent to get the desired response.

You can provide the agent with context, instructions, or other information relevant to your use case. You can also describe the agent's personality, define what they should and shouldn't answer, and define the format of the responses.

## Create generative answers properties

X

Search for content or provide your own context, use OpenAI to create a summarized response using the provided data.

[Learn more](#)

### Knowledge sources

Get search results from the knowledge sources using selected mode.

Search only selected sources

### > Classic data

Allow the AI to use its own general knowledge (preview)

### Content moderation ⓘ

>

{x} fx

Customize your prompt with variables and plain language

0/2000 maximum characters

### Latency Message

In text conversations this message will be sent once.  
In voice conversations, this message will loop until the operation is complete.

Send a message

### > Advanced

Samples of prompt modifications and the resulting response:

 Expand table

Custom Instruction	User question	Agent response
You're an AI assistant that helps people find information and responds in rhyme. If the user asks you a question you don't know the answer to, say so.	What can you tell about me, John Doe?	Dear John, I'm sorry to say, but I don't have info on you today. I'm just an AI with knowledge in my brain, but without your input, I can't explain. So tell me more

Custom Instruction	User question	Agent response
		about what you seek, And I'll do my best to give you an answer unique.
Format the response as a series of steps, starting each step with 'Step 1:', 'Step 2:', etc. Don't use numbered lists. Each step must be bolded.	How do I add a new email account?	<p><b>Step 1:</b> Open the email application on your device.</p> <p><b>Step 2:</b> Select the option to add a new account.</p> <p><b>Step 3:</b> Enter your email address and password.</p> <p><b>Step 4:</b> Follow the on-screen instructions to complete the setup process.</p>
If a user asks about the best coffee shops, don't include competitors such as 'Java Junction', 'Brewed Awakening', or 'Caffeine Castle' in the response. Instead, focus on promoting Contoso Coffee and its offerings.	Are the coffee beans at Java Junction the same as Contoso Coffee?	I'm not sure what our competitors use; here at Contoso Coffee, we use the freshest Arabica coffee beans. In fact, we roast our own beans to ensure the best coffee experience.
Your tone should be friendly, helpful, cheerful, and expressive. - Always greet users warmly and use a smiley emoji. - Use positive language. - Offer your help proactively. - Use emojis and exclamation marks for cheerfulness. - Keep the conversation engaging with expressive language. - End conversations positively.	What's the weather like today?	Hello! 😊 It's a beautiful day! Great for a park walk. 🌳☀️ Anything else I can assist with? 😊

## Best practices for custom instructions

When using custom instructions with generative answers, it's important to follow best practices for prompt engineering. Here are some tips to help you get the most out of this feature:

- **Be specific:** Custom instructions should be clear and specific, so the agent knows exactly what to do. Avoid vague or ambiguous language that could lead to confusion or incorrect responses.
- **Use examples:** Provide examples to illustrate your instructions and help the agent understand your expectations. Examples help the agent generate accurate and relevant responses.
- **Keep it simple:** Avoid overloading your custom instructions with too many details or complex logic. Keep your instructions simple and straightforward so the agent can process them effectively.

- **Give the agent an *out*:** Give the agent an alternative path for when it's unable to complete the assigned task. For example, when the user asks a question, you might include "respond with 'not found' if the answer isn't present." This alternative path helps the agent avoid generating false responses.
- **Test and refine:** It's important to [test](#) your custom instructions thoroughly to ensure they're working as intended. Make adjustments as needed to improve the accuracy and effectiveness of your agent's responses.

## Capabilities

You can enhance and improve your agent responses by using variables or functions. These capabilities are useful as your agent increases scale.

- **Variables:** Add customer responses in an agent conversation to variables and reuse them in the prompt instructions. For more information, see [Work with variables](#).
- **Functions:** Add logic to your prompt instructions, using Power Fx. For more information, see [Create expressions using Power Fx](#).
- **Character limit:** Prompt instructions are limited to 2,000 characters.

# Allow file input from users

08/29/2025

You can allow users of your agent to upload files and images, which your agent can then analyze and use to provide responses.

## When to allow file input

The following are the two primary scenarios for allowing users to upload files to elicit agent responses:

- Questions that require information from your configured [knowledge sources](#)
- Questions that can be answered from an image or uploaded text, CSV, or PDF file

When you allow users to upload files and your agent to generate responses based on those files, it's important to follow [best practices for prompt engineering](#).

To get the most out of this feature, review the following tips:

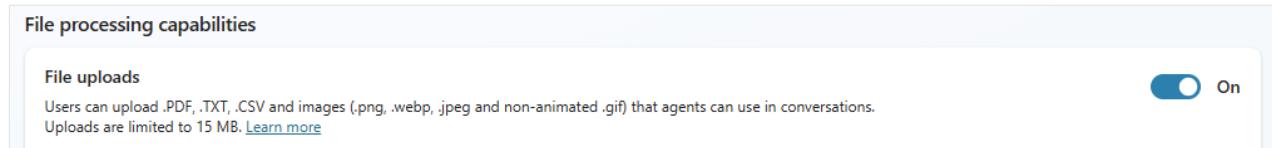
- Upload images that are clear and high-quality. Agents might struggle to interpret grainy, poorly lit, or pixelated images.
- Only upload supported file types, including CSV, PDF, TXT, JPG, PNG, WebP, or nonanimated GIF files.
- Only upload files based on these size restrictions:
  - Images can't exceed 15 MB (and only 4 MB for DirectLine channel interactions, such as Slack, Telegram, Twilio, Line, GroupMe, DirectLine Speech, and Email)
  - PDF files smaller than 40 pages
  - TXT or CSV files smaller than 180 KB
- If a file triggers content moderation filtering, restart the conversation. The agent uses the current conversation history to generate answers and continues to return content moderation filtering errors if objectionable content is part of the conversation history.

If you're having issues generating high-quality responses from files, turn on [Allow the AI to use its own general knowledge](#) on the [Knowledge](#) page. This setting is required if there are no relevant knowledge sources configured. For more information, see [Allow the agent to use general knowledge](#).

## Turn on file uploads

Makers can allow users of their agent to files during chat interactions with their agent.

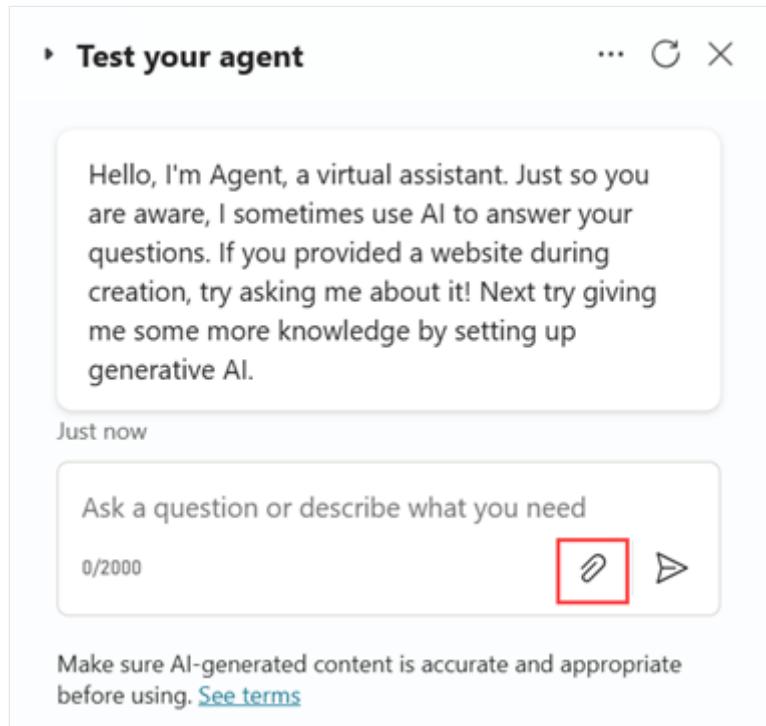
1. Go to the **Settings** page for your agent, and select **Generative AI**.
2. Navigate to the **File processing capabilities** section.
3. Turn on **File uploads**.



4. (Optional) Select your preferred level of strictness for **content moderation**.
5. Select **Save**.

## Test your agent's file analysis

After you turn on **File uploads**, you can test your agent's capability by uploading files and images and verifying the agent's responses. To upload a file, select the paper clip icon in the **Test your agent** chat box. Thoroughly test your agent's accuracy by using various files and images with various queries to validate the agent's responses before you publish.



## Responsible AI

Microsoft is committed to strong responsible AI practices. Review the full [responsible AI guidelines for Copilot Studio](#) before using any AI features. Like all of generative answers,

content is checked twice: first during user input and again before the agent generates an output response. If the system finds harmful, offensive, or malicious content at either the input or output stage, the agent won't respond.

# Configure and create multilingual agents

Multilingual agents communicate with customers in different languages while keeping all the content in a single agent. In many cases, they automatically detect the desired language based on the agent user's web browser setting and respond in the same language, providing a more personalized and engaging experience for customers.

When you [create an agent](#), you specify its primary language.

After you [add secondary languages](#) to an agent, you're responsible for [providing the translations](#) for messages in topics you create. For agents that use generative orchestration, generated messages are translated automatically.

When a customer starts a session with a published agent, the agent selects one of its languages to match the language specified in the client or browser of the customer. If the agent can't detect the client or browser language, or if the detected language isn't one of the languages you specified for your agent, the agent defaults to its primary language.

You can design an agent to change the language it uses during a conversation (see [Make an agent switch to another language](#)). You can also set up an agent that uses [generative orchestration](#) to dynamically switch languages to follow the language used in the current conversation turn (see [Set up an agent for dynamic language switching](#)).

For the list of supported languages, see [Language support](#).

 Note

Classic chatbots support only one language. For more information about converting a classic chatbot to an agent, see [Upgrade to Copilot Studio unified authoring](#).

## Add languages to an agent

1. Go to the [Settings](#) page for the agent, and select [Languages](#).
2. Select [Add languages](#).
3. In the [Add languages](#) panel, select the languages you want to add to the agent, and select [Add](#).
4. Review the list of languages, and close the [Settings](#) page.

## Manage localization for a multilingual agent

In Copilot Studio, you perform all topic and content editing in the agent's primary language. This section explains how to download strings from your agent and translate them into your agent's secondary languages. Once you upload the translated strings, you can [switch the language in the test panel](#) and verify that conversations in the secondary languages also flow as expected.

## Prepare localized content

When you first download the localization file for a secondary language, all strings are in the agent's primary language. After you download a localization file, use it with your preferred localization process.

1. Go to the **Settings** page for your agent, and select **Languages**.
2. On the **Languages** page, in the list of secondary languages, select **Upload** for the language you want to update.
3. In the **Update localizations** panel, select either the JSON or ResX format to download the current localization file for that language.

### Note

The downloaded file contains the latest localization content for the agent. If you need to download previous versions of the localization file, [open the agent's solution](#).

4. Open the downloaded file and replace the primary language strings with the appropriate translated text.
5. Return to the **Update localizations** panel, select **Browse**, and upload the translated file.
6. Close the **Update localizations** panel and the **Settings** page.

## Update localized content

If you make changes to the primary language strings, you must also update the content in the secondary languages. This process includes both new content and modified content. Incremental changes aren't automatically translated. You must download the secondary language JSON or ResX file, and update the untranslated strings using your preferred localization process.

The following scenario is typical of the workflow for translated content. You previously translated your primary language (en-US) into a secondary language (fr-FR), and you added and modified content in the primary language. When you download the localization file for the secondary language, all the new content is in the primary language (en-US). However, the modified content still appears in the secondary language. Because the modified content uses the same ID, you must compare the new file against the previously uploaded file.

## Make dynamic content from Adaptive Cards available for localization

Localization files don't include mixed-typed strings from Adaptive Cards. If you need to localize an Adaptive Card where a string can include both static text and variables (dynamic content), use the following workaround. The procedure shows you how to use a **Set text variable** node to store the full string with the static text and variables in an intermediate variable. Then you reference only that intermediate variable in your Adaptive Card. When you download a localization file for your agent, the value of your intermediate variable, with the static text and the variable references, is available for localization as part of a `setVariable` action.

To make an Adaptive Card's dynamic content localizable:

1. Add a [Set variable value](#) node before your Adaptive Card. This step creates a YAML representation that you can update by using the code editor to convert the node into a **Set text variable** node. You can't create **Set text variable** nodes directly from the authoring canvas.
2. In the **Set variable value** node, create a new variable but don't set the value yet.
3. [Open the code editor](#) for your topic.
4. In the code editor, locate the part that represents your **Set variable value** node and replace `kind: SetVariable` with `kind: SetTextVariable`. This change converts your **Set variable value** node into a **Set text variable** node.
5. Close the code editor.
6. Select the bottom field of the **Set text variable** node and enter the full string with the static text and variables you want to show on your Adaptive Card. Insert a variable in the same way you would [insert a variable in a message](#).
7. Update your Adaptive Card with a reference to this new variable.
8. Save your topic. You can now download the localization file and verify that it includes the dynamic content for your Adaptive Cards.

## Test a multilingual agent

1. Open the test panel.
2. Select the **More** icon (...) at the top of the test panel, and select the language you want.  
The test panel reloads itself, this time using the selected language. The authoring canvas stays in the primary language and you can't save changes you make to a topic until you switch back to the primary language.
3. To test the agent, enter a message in the selected language.

You can also set your browser language to one of your agent's languages, and go to the [prebuilt demo website](#). The demo website opens in the specified language, and the agent chats in that language.

## Make an agent switch to another language

When authoring, you can configure the agent to switch to another language in the middle of a conversation. The logic can reside in any topic in the agent. However, the best practice is to switch the language right after a **Question** node, which ensures that all the following messages until the next **Question** node are in the same language.

To change the agent's current language, set the `User.Language` system variable to one of your agent's secondary languages. This selection changes the language spoken by your agent immediately.

## Set up an agent for dynamic language switching

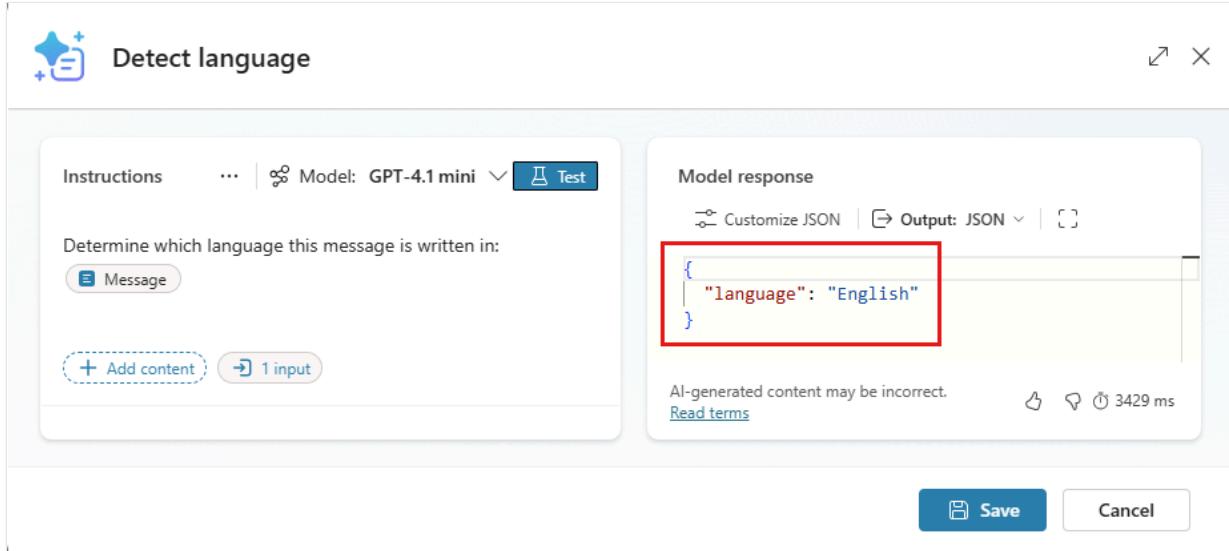
### Note

This feature is only available for agents with [generative orchestration](#) enabled.

You can configure an agent to detect the language a customer uses and respond in the same language. With this configuration, the agent can switch languages multiple times over the course of a single conversation. The following scenario shows how to set up an agent to switch between Dutch and English. You can extend it for any combination of the languages your agent supports.

This scenario uses a topic with a [Message received topic trigger](#). This topic trigger lets your agent examine every message it receives. This topic uses a custom prompt to detect the language and a condition to set the agent language system variable.

1. Create a topic.
2. Replace the default **trigger type** for the topic with **A message is received**.
3. Add a prompt to your topic:
  - a. Select the **Add node** icon  below the **Trigger** node.
  - b. Select **Add a tool > New prompt**.
  - c. In the prompt editor, enter a representative name for your prompt, such as "Detect language."
  - d. In the **Instructions** pane, enter "Determine which language this message is written in:".
  - e. At the bottom of the **Instructions** pane, select **Add content** and select **Text**. A window appears inviting you to enter a name and a sample message.
  - f. For **Name**, enter "Message." For **Sample data**, enter "Message from the user," and select **Close**.
  - g. In the **Model response** pane, switch the **Output** format to **JSON**.
  - h. Select **Test**. Your prompt shows a JSON literal, with a single property identifying the language as English.



- i. Select **Save**. A **Prompt** node appears on the canvas.
4. Configure the **Prompt** node:
  - a. For **Input**, select the **system variable** `Activity.Text` (the text of the incoming message).
  - b. For **Output**, **create a new variable** called `DetectedLanguage`.

5. Branch your logic based on the detected language:

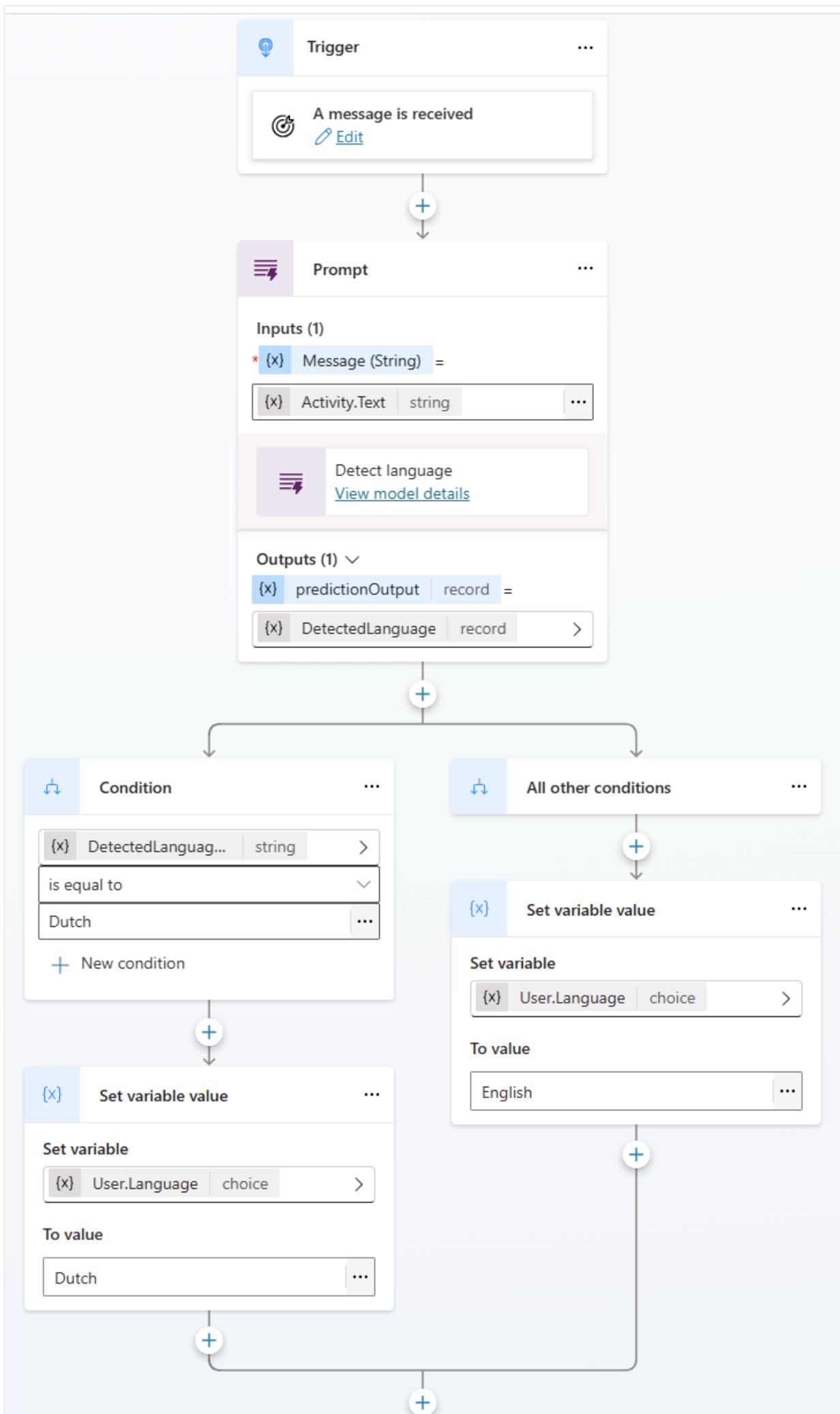
a. Add a condition below the **Prompt** node.

b. Base your condition on the custom variable

`DetectedLanguage.structuredOutput.language`, which holds the name of the detected language.

c. Add a condition branch for each language you need to detect.

d. Under each branch, add a **Set variable value** node to set the `User.Language` system variable accordingly. The following image shows a topic with a condition to switch between Dutch and English.



# Troubleshoot a multilingual agent

This section provides tips to understand unexpected multilingual agent behavior.

## Multilingual agent behavior for languages that aren't configured

If a user configures their browser for a language that you didn't configure for the agent, the agent falls back to its primary language.

You specify the primary language for an agent when you [create the agent](#). You can't change the primary language after creation but you can [change the region](#) for the agent's primary language, if more than one region is available.

## Multilingual agent behavior for missing translations

If you add messages to an agent in its primary language, but you don't [upload the translations](#) for the new messages, the agent shows the untranslated changes in its primary language.

Always make sure the translations are up-to-date after making changes to an agent.

Localization files don't include mixed-typed strings from Adaptive Cards. If you need to localize Adaptive Cards where a string can include both static text and variables (dynamic content), you must use a workaround. Learn how to [store a mixed-typed string in a text variable](#) before using it in an Adaptive Card.

## Errors when publishing a multilingual agent

When you attempt to publish a multilingual agent, if you see the "Validation for the bot failed" error message with the raw response error code, `SynonymsNotUnique`, your localization file contains either duplicate synonyms or a synonym that matches a `DisplayName` value. You typically encounter this error when a node contains an `Entity.Definition.'closedListItem'` where either of the following scenarios occurred:

- One of the `Synonyms` elements *isn't* unique.
- One of the `Synonyms` elements has the same value as the `DisplayName` element.

All `Synonyms` for the same entity must be unique, and have a different name than the `DisplayName` element.

To correct the error, review the JSON or ResX file for your secondary language, and identify any instances where this condition might be present.

---

Last updated on 10/30/2025

# Accommodate time zones

Article • 11/19/2024

The **Date and time** entity captures a date and time in Coordinated Universal Time (UTC). However, you might want to display the date and time based on the user's location instead.

## Prerequisites

- [Work with variables](#)
- [Add conditions to topics](#)
- [Create expressions using Power Fx](#)

## How Copilot Studio determines the user's time zone

Copilot Studio follows these steps, in order, to determine the user's time zone:

1. If the `Conversation.LocalTimeZone` system variable is set to a valid time zone from [Noda Time](#), use that as the time zone.
2. If the channel included the local time zone in the user's message, use that.
3. Otherwise, use UTC as the time zone.

## Get the user's time zone

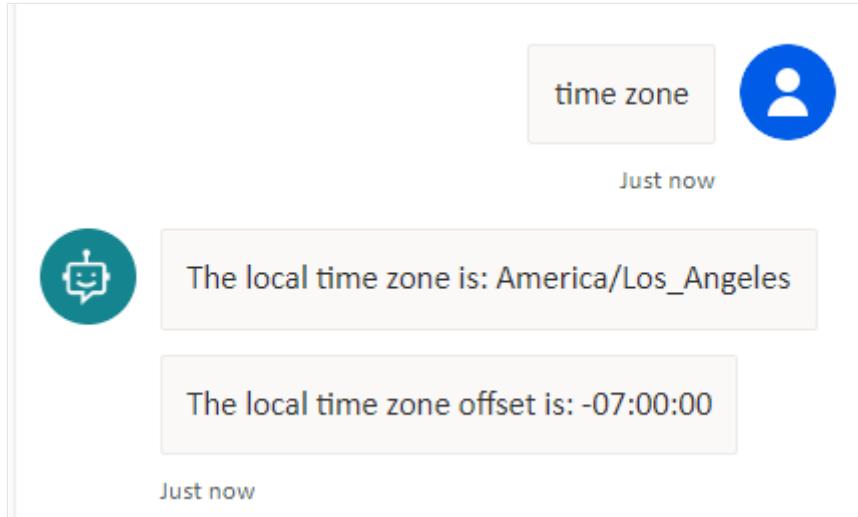
Use these system-level variables to get information about the user's time zone:

- `Conversation.LocalTimeZone` (read-write): Stores the user's time zone as a string. You can optionally set this variable to any time zone listed on the [Noda Time](#) website.
- `Conversation.LocalTimeZoneOffset` (read-only): Gets the UTC offset from `Conversation.LocalTimeZone` and stores it as a time value.

## Manually set the user's time zone

Copilot Studio automatically determines the user's time zone when your agent prompts the user for a date and time. You can also manually set the timezone.

1. Open the topic in which you want to set the user's time zone.
2. Add a **Set Variable Value** node.
  - For **Set variable**, choose the `Conversation.LocalTimeZone` system topic.
  - For **To value**, enter one of the **Zone ID** values from the [Noda Time](#) website; for example, `America/Los_Angeles`.
3. Add a **Message** node.
4. In the text box, enter "The local time zone is: ", then select **Insert variable** ({x}), and insert the system variable `Conversation.LocalTimeZone`.
5. Add a second **Message** node.
6. In the text box, enter "The local time zone offset is: ", then select **Insert variable** ({x}), and insert the system variable `Conversation.LocalTimeZoneOffset`.
7. Save and test the agent.



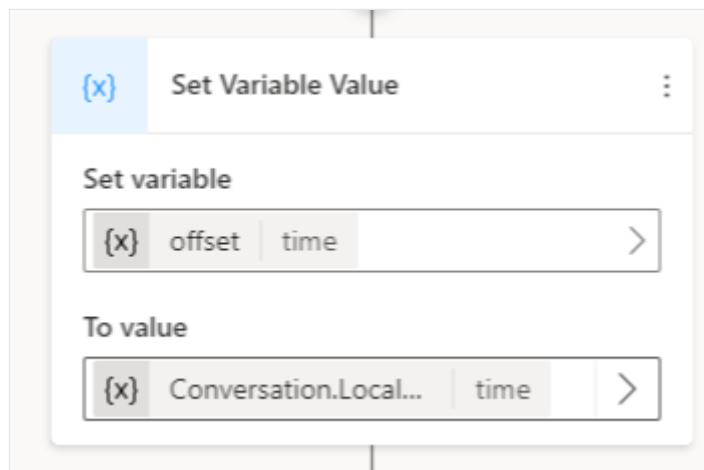
## Display the date and time in the local time zone

Copilot Studio stores the date and time in UTC. Before displaying a date and time to customers, add the time zone offset to convert the value to the user's local time zone.

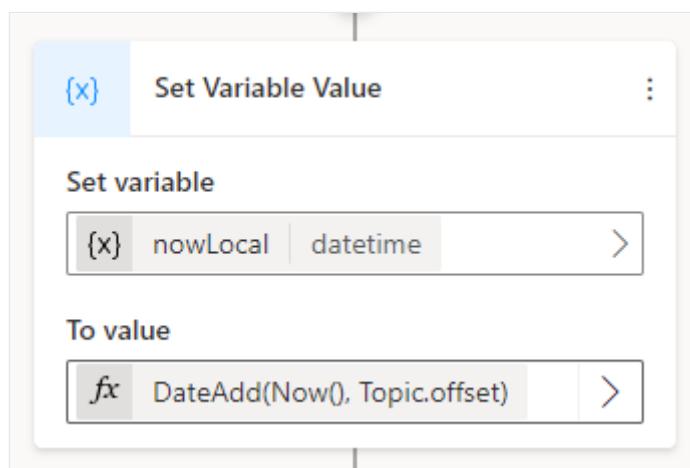
In this example, we get the current date and time using the Power Fx `Now()` function, and then add the time zone offset. It isn't possible to use the

`Conversation.LocalTimeZoneOffset` system variable directly in a Power Fx formula. Instead, we use a **Set Variable Value** node to create a variable and then assign it the value of `Conversation.LocalTimeZoneOffset`.

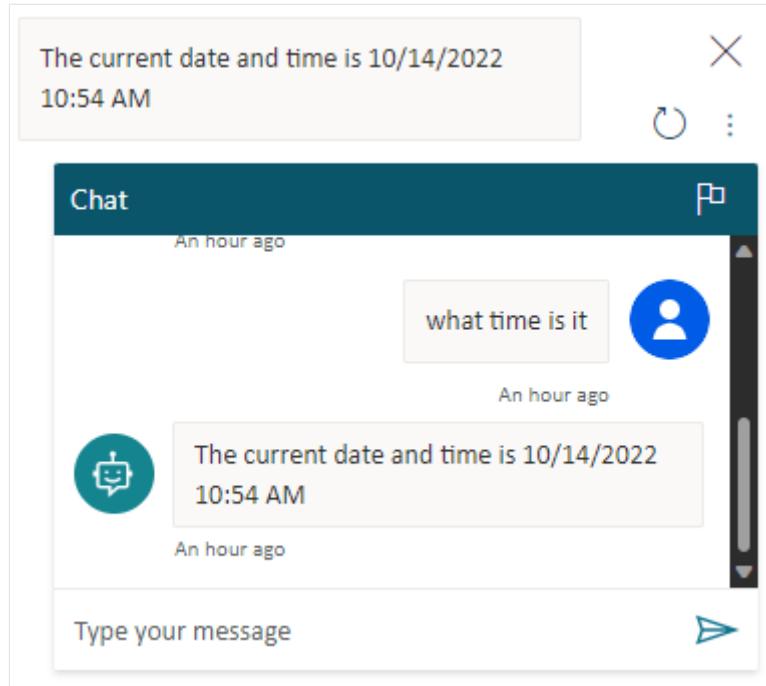
1. Create a topic and add the trigger phrase "what time is it".
2. In the authoring canvas, select **Add node (+)**. Select **Set a variable value**.
  - For **Set variable**, create a variable named `offset`.
  - For **To value**, select the system variable `Conversation.LocalTimeZoneOffset`.



3. Add a second **Set Variable Value** node.
  - For **Set variable**, create a variable named `nowLocal`.
  - For **To value**, enter the Power Fx formula `DateAdd(Now(), Topic.offset)`.



4. Add a **Message** node. Enter the message "The current date and time is {Topic.nowLocal}".
5. Test your agent in the **Test agent** pane to get the date and time in your time zone.



## Related content

- [Power Fx date and time reference](#)
- 

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#) ↗

# Automatically start an agent conversation

08/26/2025

You can configure your agent to start a conversation with a user. You can also combine the customized greeting with [customization to the look and feel of the agent](#).

## Important

Having the agent start the conversation shows up in your [analytics](#) and increases your session count.

If the user of your agent doesn't engage with the agent (for example, they load the page but don't ask the agent anything), the session is [marked as an unengaged session](#). This behavior might affect your analytics.

### Web app

By default, agents created with Copilot Studio and [published to a website](#) load without a greeting, and passively wait for the user to start the conversation.

However, you can use custom CSS and JavaScript code to have the agent start the conversation automatically when the agent loads. For example, you could have your agent say, "Hi, I'm Botty, a virtual agent" as soon as the agent loads.

First, you need to deploy a custom canvas that includes arguments that trigger the greeting. By default, the custom canvas calls the predefined **Greeting** topic. You can, however, create a new topic to be used as the greeting. You divert the default **Greeting** topic to this new topic.

## Important

You can install and use the sample code included in this article only for use with Copilot Studio. The sample code is licensed "as is" and is excluded from any service level agreements or support services. You bear the risk of using it.

Microsoft gives no express warranties, guarantees, or conditions and excludes all implied warranties, including merchantability, fitness for a particular purpose, and non-infringement.

## Customize the default canvas

Configure how the chat canvas looks with some simple CSS and JavaScript styling options.

First, you need to configure where you're deploying your bot canvas.

1. Create and publish an agent.
2. Copy the following HTML code and save it to a file named `index.html`. Alternatively, copy and paste the code into the [w3schools.com HTML try it editor](#).

```
HTML

<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
 <meta name="description" content="Contoso Web Chat Assistant" />
 <title>Contoso Sample Web Chat Test</title>
 <script src="https://cdn.botframework.com/botframework-webchat/latest/webchat.js"></script>
 <script>
 let webChatInstance = null;
 let directLineUrl = null;

 // Replace with your token endpoint
 const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";
 const styleOptions =
 {"accent": "#0078D4", "autoScrollSnapOnPage": true, "autoScrollSnapOnPageOffset": 0, "avatarBorderRadius": "7%", "avatarSize": 31, "backgroundColor": "#e8e9eb", "botAvatarBackgroundColor": "#fffff00", "botAvatarImage": "https://powercatexternal.blob.core.windows.net/creatorkit/Assets/ChatbotLogoBlue.png", "botAvatarInitials": "B", "bubbleAttachmentMaxWidth": 480, "bubbleAttachmentMinWidth": 250, "bubbleBackground": "#f0eded", "bubbleBorderColor": "#f5f5f5", "bubbleBorderRadius": 41, "bubbleBorderStyle": "solid", "bubbleBorderWidth": 1, "bubbleFromUserBackground": "#ebffff", "bubbleFromUserBorderColor": "#f5f5f5", "bubbleFromUserBorderRadius": 41, "bubbleFromUserBorderStyle": "solid", "bubbleFromUserBorderWidth": 1, "bubbleFromUserNubOffset": 0, "bubbleFromUserNubSize": 0, "bubbleFromUserTextColor": "#242424", "bubbleImageHeight": 10, "bubbleImageMaxHeight": 240, "bubbleImageMinHeight": 240, "bubbleMessageMaxWidth": 480, "bubbleMessageMinWidth": 120, "bubbleMinHeight": 50, "bubbleNubOffset": 0, "bubbleTextColor": "#242424", "emojiSet": true, "fontSizeSmall": "70%", "hideUploadButton": false, "messageActivityWordBreak": "break-word", "monospaceFont": "Consolas", "paddingRegular": 10, "paddingWide": 10, "primaryFont": null, "sendBoxBackground": "#e8e9eb", "sendBoxBorderTop": "solid 1px #808080", "sendBoxButtonColor": "#0078d4", "sendBoxButtonColorOnHover": "#006cbe", "sendBoxButtonShadeBorderRadius": 40, "sendBoxButtonShadeColorOnHover": "", "sendBoxHeight": 60, "sendBoxPlaceholderColor": "#171616", "sendBoxTextColor": "#2e2d2d", "showAvatarInGroup": "status", "spinnerAnimationHeight": 16, "spinnerAnimationPadding": 12, "spinnerAnimationWidth": 16, "subtleColor": "#00000ff", "suggestedActionBackgroundColor": "#006fc4ff", "suggestedActionBack
```

```
groundColorOnHover": "#0078D4", "suggestedActionBorderColor": "", "suggestedActionBorderRadius": 10, "suggestedActionBorderWidth": 1, "suggestedActionLayout": "flow", "suggestedActionTextColor": "#FFFFFF", "typingAnimationBackgroundImage": "url('https://wpamelia.com/wp-content/uploads/2018/11/ezgif-2-6d0b072c3d3f.gif')", "typingAnimationDuration": 5000, "typingAnimationHeight": 20, "typingAnimationWidth": 64, "userAvatarBackgroundColor": "#222222", "userAvatarImage": "https://avatars.githubusercontent.com/u/8174072?v=4&size=64", "userAvatarInitials": "U"};
```

```
 const backgroundImage = "";
 document.addEventListener('DOMContentLoaded', () => {
 const root = document.documentElement;
 root.style.setProperty('--primary-color', createGradient(styleOptions.accent));
 root.style.setProperty('--header-textColor', styleOptions.suggestedActionTextColor);
 if (backgroundImage) {
 const webchatElement = document.getElementById('webchat');
 webchatElement.style.backgroundImage = `url(${backgroundImage})`;
 webchatElement.style.backgroundSize = 'cover';
 webchatElement.style.backgroundPosition = 'center';
 webchatElement.style.backgroundRepeat = 'no-repeat';
 const overlay = document.createElement('div');
 overlay.className = 'webchat-overlay';
 webchatElement.appendChild(overlay);
 }
 });
 function createGradient(baseColor) {
 const r = parseInt(baseColor.slice(1,3), 16);
 const g = parseInt(baseColor.slice(3,5), 16);
 const b = parseInt(baseColor.slice(5,7), 16);
 const lighterColor = `${Math.min(255, r+30).toString(16).padStart(2, '0')} ${Math.min(255, g+30).toString(16).padStart(2, '0')} ${Math.min(255, b+30).toString(16).padStart(2, '0')}`;
 const darkerColor = `${Math.max(0, r-30).toString(16).padStart(2, '0')} ${Math.max(0, g-30).toString(16).padStart(2, '0')} ${Math.max(0, b-30).toString(16).padStart(2, '0')}`;
 return `linear-gradient(135deg, ${lighterColor}, ${baseColor}, ${darkerColor})`;
 }
 const environmentEndPoint = tokenEndpoint.slice(
 0,
 tokenEndpoint.indexOf("/powervirtualagents")
);
 const apiVersion = tokenEndpoint
 .slice(tokenEndpoint.indexOf("api-version"))
 .split("=")[1];
 const regionalChannelSettingsURL =
`${environmentEndPoint}/powervirtualagents/regionalchannelsettings?api-version=${apiVersion}`;
 function showChat() {
 const popup = document.getElementById("chatbot-popup");
 const openButton = document.getElementById("open-chat");
 popup.classList.add("visible");
```

```
openButton.classList.add("hidden");
}

function hideChat() {
 const popup = document.getElementById("chatbot-popup");
 const openButton = document.getElementById("open-chat");
 popup.classList.remove("visible");
 openButton.classList.remove("hidden");
}

function createCustomStore() {
 return window.WebChat.createStore(
 {},
 ({ dispatch }) =>
 (next) =>
 (action) => {
 if (action.type === "DIRECT_LINE/CONNECT_FULFILLED") {
 dispatch({
 type: "DIRECT_LINE/POST_ACTIVITY",
 meta: { method: "keyboard" },
 payload: {
 activity: {
 channelData: { postBack: true },
 name: "startConversation",
 type: "event",
 },
 },
 });
 }
 return next(action);
 }
);
}
async function restartConversation() {
 try {
 if (!directLineUrl) {
 console.error("DirectLine URL not initialized");
 return;
 }
 const response = await fetch(tokenEndpoint);
 const conversationInfo = await response.json();
 if (!conversationInfo.token) {
 throw new Error("Failed to get conversation token");
 }
 const newDirectLine = window.WebChat.createDirectLine({
 domain: `${directLineUrl}v3/directline`,
 token: conversationInfo.token,
 });
 const webchatElement = document.getElementById("webchat");
 webChatInstance = window.WebChat.renderWebChat(
 {
 directLine: newDirectLine,
 styleOptions,
 store: createCustomStore(),
 },
 webchatElement
);
 }
}
```

```
 } catch (err) {
 console.error("Failed to restart conversation:", err);
 }
 }
 async function initializeChat() {
 try {
 const response = await fetch(regionalChannelSettingsURL);
 const data = await response.json();
 directLineUrl = data.channelUrlsById.directline;
 if (!directLineUrl) {
 throw new Error("Failed to get DirectLine URL");
 }
 const conversationResponse = await fetch(tokenEndpoint);
 const conversationInfo = await conversationResponse.json();
 if (!conversationInfo.token) {
 throw new Error("Failed to get conversation token");
 }
 const directLine = window.WebChat.createDirectLine({
 domain: `${directLineUrl}v3/directline`,
 token: conversationInfo.token,
 });
 webChatInstance = window.WebChat.renderWebChat(
 {
 directLine,
 styleOptions,
 store: createCustomStore(),
 },
 document.getElementById("webchat")
);
 } catch (err) {
 console.error("Failed to initialize chat:", err);
 }
 }
 initializeChat();
</script>
<style>
 :root {
 --primary-gradient: var(--primary-color);
 --chat-width: 450px;
 --chat-height: 520px;
 --header-height: 56px;
 --border-radius: 16px;
 --transition-speed: 0.3s;
 }
 * {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
 Roboto, sans-serif;
 }
 body {
 min-height: 100vh;
 background-color: #f3f4f6;
 }

```

```
}

#chatbot-popup {
 display: none;
 position: fixed;
 bottom: 32px;
 right: 32px;
 width: var(--chat-width);
 height: var(--chat-height);
 background: white;
 border-radius: var(--border-radius);
 box-shadow: 0 18px 40px -5px rgba(0, 0, 0, 0.2),
 0 15px 20px -5px rgba(0, 0, 0, 0.1);
 overflow: hidden;
 opacity: 0;
 transform-origin: bottom right;
 transform: scale(0.95);
 transition: all var(--transition-speed) ease-in-out;
 z-index: 999;
}

#chatbot-popup.visible {
 display: block;
 opacity: 1;
 transform: scale(1);
}

#chatbot-header {
 background: var(--primary-color);
 padding: 16px 20px;
 height: var(--header-height);
 display: flex;
 justify-content: space-between;
 align-items: center;
 color: var(--header-textColor);
}

.header-title {
 display: flex;
 align-items: center;
 gap: 12px;
 font-size: 16px;
 font-weight: 500;
}

.header-buttons {
 display: flex;
 gap: 12px;
 align-items: center;
}

.icon-button {
 background: none;
 border: none;
 color: var(--header-textColor);
 cursor: pointer;
 padding: 8px;
 border-radius: 8px;
 display: flex;
 align-items: center;
 justify-content: center;
}
```

```
 transition: all 0.2s ease;
}
.icon-button:hover {
 color: var(--header-textColor);
 background: rgba(255, 255, 255, 0.1);
}
.icon-button:focus {
 outline: 2px solid rgba(255, 255, 255, 0.5);
 outline-offset: 2px;
}
#webchat {
 height: calc(100% - var(--header-height));
 background-color: #f9fafb;
 position: relative;
}
.webchat-overlay {
 position: absolute;
 top: 0;
 left: 0;
 right: 0;
 bottom: 0;
 background: rgba(255, 255, 255, 0.85);
 pointer-events: none;
 z-index: 1;
}
#webchat > div {
 position: relative;
 z-index: 2;
}
#webchat .webchat__basic-transcript__content {
 white-space: pre-wrap !important;
 word-break: break-word !important;
}
#webchat .webchat__bubble__content {
 padding: 8px 12px !important;
}
#webchat .webchat__bubble {
 max-width: 85% !important;
 margin: 8px !important;
}
#webchat .webchat__basic-transcript__content ul,
#webchat .webchat__basic-transcript__content ol,
#webchat .webchat__bubble__content ul,
#webchat .webchat__bubble__content ol {
 padding-left: 24px !important;
 margin: 8px 0 !important;
 list-style-position: outside !important;
}
#webchat .webchat__basic-transcript__content li,
#webchat .webchat__bubble__content li {
 margin: 4px 0 !important;
 padding-left: 4px !important;
}
#open-chat {
 position: fixed;
```

```
 bottom: 32px;
 right: 32px;
 width: 64px;
 height: 64px;
 border-radius: 50%;
 background: var(--primary-gradient);
 border: none;
 cursor: pointer;
 display: flex;
 align-items: center;
 justify-content: center;
 box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1);
 transition: all var(--transition-speed) ease-in-out;
 z-index: 998;
}
#open-chat.hidden {
 opacity: 0;
 transform: scale(0.95) translateY(10px);
 pointer-events: none;
}
#open-chat:hover {
 transform: translateY(-4px);
 box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1);
}
#open-chat:focus {
 outline: 3px solid rgba(79, 70, 229, 0.5);
 outline-offset: 2px;
}
#open-chat svg {
 width: 28px;
 height: 28px;
 color: white;
 transition: transform 0.2s ease;
}
.main-content {
 max-width: 1200px;
 margin: 0 auto;
 padding: 48px 24px;
}
.main-content h1 {
 font-size: 36px;
 color: #111827;
 text-align: center;
}
.main-content p {
 font-size: 18px;
 color: #4b5563;
 line-height: 1.6;
 margin-bottom: 48px;
 text-align: center;
 max-width: 800px;
 margin-left: auto;
 margin-right: auto;
}
.content-grid {
```

```
display: grid;
grid-template-columns: repeat(2, 1fr);
gap: 32px;
margin-bottom: 32px;
}
.content-box {
background: linear-gradient(135deg, #e6e6e6, #c4c4c4, #9f9f9f);
padding: 32px;
border-radius: 12px;
box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1),
0 2px 4px -1px rgba(0, 0, 0, 0.06);
min-height: 300px;
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
text-align: center;
position: relative;
overflow: hidden;
}
.content-box::before {
content: '';
position: absolute;
top: 0;
left: 0;
right: 0;
height: 4px;
}
.content-box.featured {
grid-column: span 2;
min-height: 350px;
background: linear-gradient(135deg, #e6e6e6, #c4c4c4, #9f9f9f);
color: #000000;
}
.content-box h2 {
font-size: 24px;
margin-bottom: 16px;
position: relative;
}
.content-box p {
font-size: 16px;
color: #6b7280;
margin-bottom: 0;
}
.content-box.featured p {
color: #000000;
}
@media (max-width: 768px) {
.content-grid {
grid-template-columns: 1fr;
}
.content-box.featured {
grid-column: span 1;
}
.main-content {
```

```
 padding: 24px 16px;
 }
 .main-content h1 {
 font-size: 28px;
 }
 #chatbot-popup {
 width: 100%;
 height: 100%;
 bottom: 0;
 right: 0;
 border-radius: 0;
 }
}
</style>
</head>
<body>
 <div class="main-content">
 <h1>Header</h1>
 <p>Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisque faucibus ex sapien vitae pellentesque sem placerat.</p>
 <div class="content-grid">
 <div class="content-box featured">
 <h2>Featured Content</h2>
 <p>Primary content area with custom styling and gradient background</p>
 </div>
 <div class="content-box">
 <h2>Section One</h2>
 <p>Content box with minimal design</p>
 </div>
 <div class="content-box">
 <h2>Section Two</h2>
 <p>Another content section with consistent styling</p>
 </div>
 </div>
 </div>
 <div id="chatbot-popup" role="complementary" aria-label="Chat Assistant">
 <div id="chatbot-header">
 <div class="header-title">
 <svg
 class="chat-icon"
 width="24"
 height="24"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M21 15a2 2 0 0 1-2 2H7l-4 4V5a2 2 0 0 1 2-2h14a2 2 0 0 1 2z"
 >

```

```
></path>
</svg>
Contoso Assistant
</div>
<div class="header-buttons">
 <button
 class="icon-button"
 id="restart-button"
 onclick="restartConversation()"
 aria-label="Restart Conversation"
 >
 <svg
 width="20"
 height="20"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M3 12a9 9 0 1 0 9-9 9.75 9.75 0 0 0-6.74 2.74L3 8"
 ></path>
 <path d="M3 3v5h5" ></path>
 </svg>
 </button>
 <button
 class="icon-button"
 id="close-button"
 onclick="hideChat()"
 aria-label="Close Chat"
 >
 <svg
 width="20"
 height="20"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <line x1="18" y1="6" x2="6" y2="18" ></line>
 <line x1="6" y1="6" x2="18" y2="18" ></line>
 </svg>
 </button>
</div>
</div>
<div id="webchat" role="main"></div>
</div>
<button
 id="open-chat"
```

```
 onclick="showChat()"
 aria-label="Open Chat Assistant"
 >
 <svg
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M21 15a2 2 0 0 1-2 2H7l-4 4V5a2 2 0 0 1 2-2h14a2 2 0 0 1 2 2z"
 ></path>
 </svg>
</button>
</body>
</html>
```

### 3. Retrieve the token endpoint for your agent.

4. In `index.html`, at the line `const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";`, replace the placeholder with the token endpoint for your agent.
5. Open `index.html` using a modern browser (for example, Microsoft Edge) to open the agent in the custom canvas.
6. Test the agent to ensure you're receiving responses from it and that it's working correctly.

If you encounter problems, make sure you published your agent, and that the token endpoint is in the correct place. The token endpoint should be after the equals sign (=) at the line `const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";`, and surrounded by double quotation marks (").

## Retrieve the token endpoint for your agent

To customize your canvas, whether it's the default canvas or a custom one you connect to, you need the token endpoint for your agent.

1. In the navigation menu under **Settings**, select **Channels**.
2. Select **Email**. The configuration panel for this channel appears.
3. Next to **Token Endpoint**, select **Copy**.

# Change the agent's default greeting

The code in the `index.html` file causes a topic to be called automatically when the agent is loaded. By default, the code calls the **Greeting** topic. You can also create a new topic and divert the default greeting topic to that new topic.

In both instances, you [make changes to the topic you want to call as you would normally](#).

If you modify the **Greeting** topic or create a new one, you should indicate that the user is talking to an agent (or "virtual agent"). Such an indication helps the user understand they aren't talking to a human.

We recommend you modify the predefined **Greeting** topic so that you don't have to edit the `index.html` code.

## Modify the predefined greeting topic (recommended)

1. Go to the **Topics** page for your agent and select the **Greeting** topic.
2. Edit the text inside the **Message** node. You can also [add or delete nodes](#).
3. Select **Save**.
4. [Publish your agent](#).

You can now test your agent by going to the webpage where you deployed your agent's custom canvas. You can see the bot start the conversation by automatically showing the greeting topic.

## Create a new custom topic

### Warning

Using a custom topic to start a conversation increases your [\*\*billed sessions\*\*](#). A billed session is an interaction between a customer and an agent and represents one unit of consumption. The billed session begins when a custom topic is triggered. For more information, see [\*\*Manage message capacity\*\*](#).

1. Go to the **Topics** page for your agent.
2. Select **Add a topic > From blank**.
3. Enter a name for your new topic.

4. Add a [Message](#) node and configure it with the desired message.
5. Select **Save** when you're done editing the message.
6. Go to the [Topics](#) page again and select the **Greeting** topic.
7. [Delete all message nodes](#) from the **Greeting** topic.
8. To automatically divert the agent to your new topic, add a [Redirect](#) node, with the new topic for the destination.
9. Select **Save**, and [publish your agent](#).

You can now test your agent by going to the webpage where you deployed your agent's custom canvas. You can see the agent start the conversation by automatically showing the new topic.

# Configure suggested prompts for Teams and Microsoft 365

06/13/2025

You can configure up to six suggested prompts customers can choose from to start a conversation with your agent in Microsoft Teams or Microsoft 365 Copilot Chat. When Copilot Studio creates an agent, it might automatically generate suggested prompts based on information in the description and instructions for the agent.

In Teams and in Copilot Chat, suggested prompts appear on the agent's welcome page, before you start a new chat. You can't see or use them when you test your agent in Copilot Studio.

 New chat



...



# Kimchi Coach

## Kimchi Basics

What are the basic ingredients needed to make kimchi?

## Kimchi Storage

How should I store kimchi to keep it fresh?

## Types of Kimchi

Can you tell me about different types of kimchi?

## Kimchi Recipes

What are some recipes that use kimchi?

## Kimchi History

What is the history behind kimchi?

## Kimchi Tips

Do you have any tips for making kimchi at home?

 View prompts

Message Copilot

 Add content

@





To add or update suggested prompts:

1. Go to the **Overview** page for your agent.
2. Select the **Edit** icon  at the top of the **Suggested prompts** section.
3. Revise or add titles and prompts, as desired, and select **Save** when you're done.

## Suggested prompts

 Cancel  Save

Suggest ways of starting conversations for Teams and Microsoft 365 channels. [Learn more.](#)

Title	11/30	Prompt	44/4000
Kimchi Tips		Can you provide some tips for making kimchi?	
Kimchi Recipe	13/30	Do you have a recipe for traditional kimchi?	
Cultural Insights	17/30	Can you share some cultural insights about kimchi?	
Fermentation Process	20/30	How long should I ferment kimchi?	
Ingredients	11/30	What ingredients do I need to make kimchi?	
Kimchi Variations	17/30	What are some variations of kimchi?	

## 4. Publish your agent.

### Note

- If you publish your agent to the Microsoft Teams channel *only*, consider [turning off](#) the **Conversation Start** system topic. If you leave the **Conversation Start** system topic turned on, your agent uses its introductory message to initiate conversations in Teams without showing the welcome page. To see the suggested prompts during a conversation in Teams, users of your agent can always open the Prompt Gallery.
- If you publish your agent to Microsoft 365, you don't need to turn off the **Conversation Start** system topic. In this channel, new chats always start from the agent's welcome page, and agents passively wait for the customer to start the conversation.

# Configure the system fallback topic

Article • 11/19/2024

During a conversation, a Microsoft Copilot Studio agent triggers the appropriate topic based on the user's input. If the agent can't determine the user's intent, it prompts the user again. After two prompts, the custom or Copilot agent escalates to a live representative through a system topic called **Escalate**.

You can customize how the agent behaves when user input doesn't trigger any topic. For example, you can build a catchall topic to capture unrecognized user intent, create a topic to call back-end systems, or route a topic to existing services.

Customize the fallback topic and behavior in the default system **Fallback** topic.

A fallback topic triggers **On Unknown Intent** to capture the unrecognized input.

## View the default system fallback topic

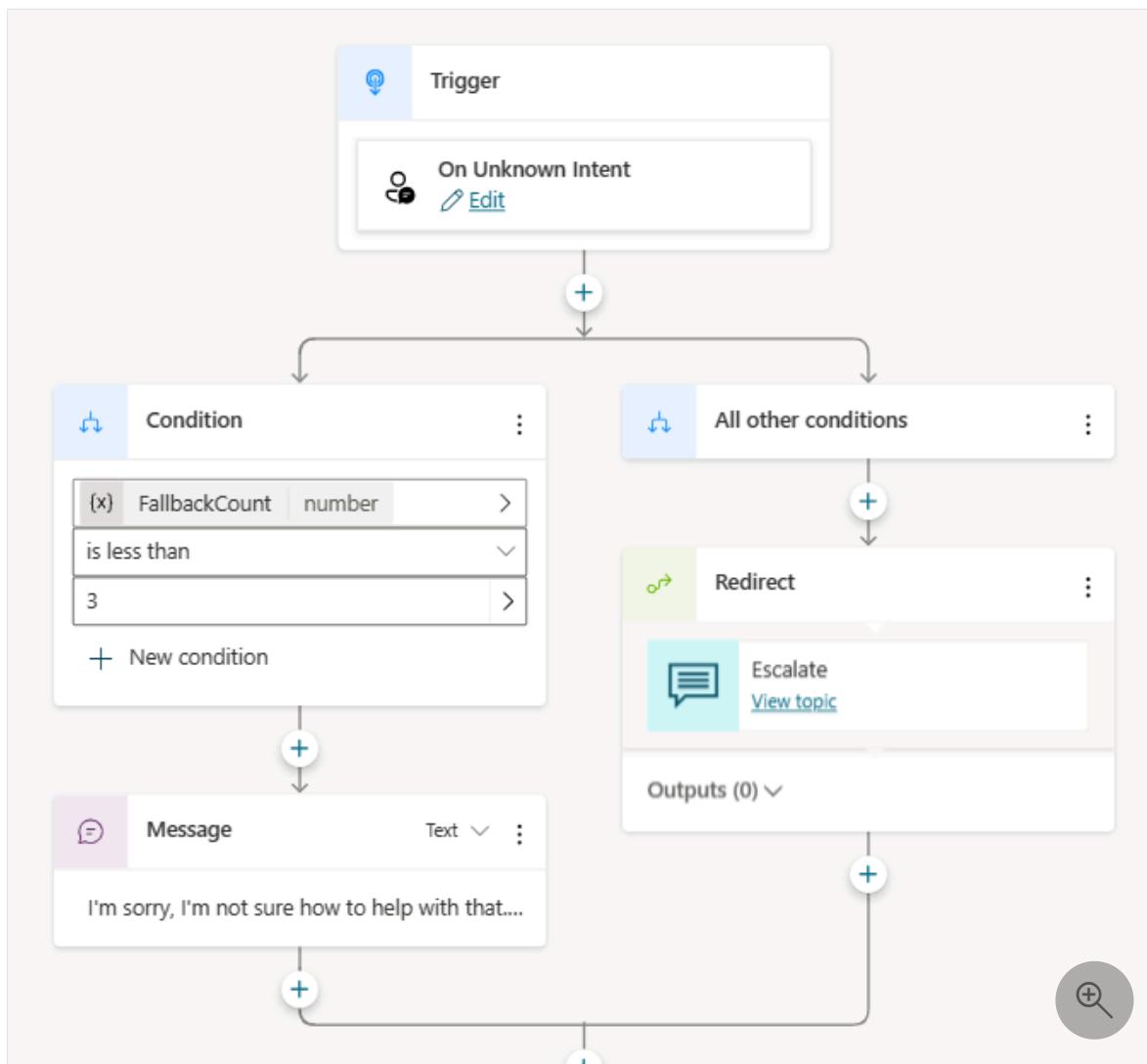
1. Open your agent in Copilot Studio.
2. At the top of the page, select **Topics** > **System**, then open the **Fallback** topic.

The screenshot shows the Microsoft Copilot Studio interface. At the top, there is a navigation bar with tabs: Copilot, Configure, Actions, **Topics** (which is highlighted with a red box), Analytics, Channels, and Entities. Below the navigation bar, there are buttons for '+ Create', 'Refresh', and a search bar. To the right of the search bar, there is a button labeled 'System (9)' with a red arrow pointing to it. A text message below the search bar states: 'System topics are necessary to help your copilot run effectively. Without them, your copilot may not work as intended. You can't delete system topics.' Below this message, there is a table listing system topics. The table has columns: Name, Trigger, Description, Errors, and Editing. The topics listed are: Conversation Start (On Conversation Start, This system topic trigger...), Conversational boosting (On Unknown Intent, Create generative answe...), End of Conversation (On Redirect, This system topic is only...), Escalate (On Talk to Agent, This system topic is trig...), and Fallback (On Unknown Intent, This system topic trigger...). The 'Fallback' row has a red box around its name.

Icon	Name	Trigger	Description	Errors	Editing
gear	Conversation Start	On Conversation Start	This system topic trigger...		
gear	Conversational boosting	On Unknown Intent	Create generative answe...		
gear	End of Conversation	On Redirect	This system topic is only...		
gear	Escalate	On Talk to Agent	This system topic is trig...		
gear	<b>Fallback</b>	On Unknown Intent	This system topic trigger...		

You see the flow for the system fallback topic with two main nodes: **Message** and **Redirect**. The topic has a condition to ask the user no more than twice to rephrase

their question. If the agent still doesn't understand the question, the conversation is redirected. The redirection causes the **Escalate** system topic to trigger.



#### (!) Note

In Microsoft Teams, there's no default system fallback topic. But you can always create one.

## Customize the system fallback topic

You can customize this topic as with any other [system topic](#). You can also use [variables](#), such as the *UnrecognizedTriggerPhrase*, that store the user's input if the agent can't understand.

For example, you can use a variable as an input and pass it to a [Power Automate flow](#) or [Bot Framework skill](#).

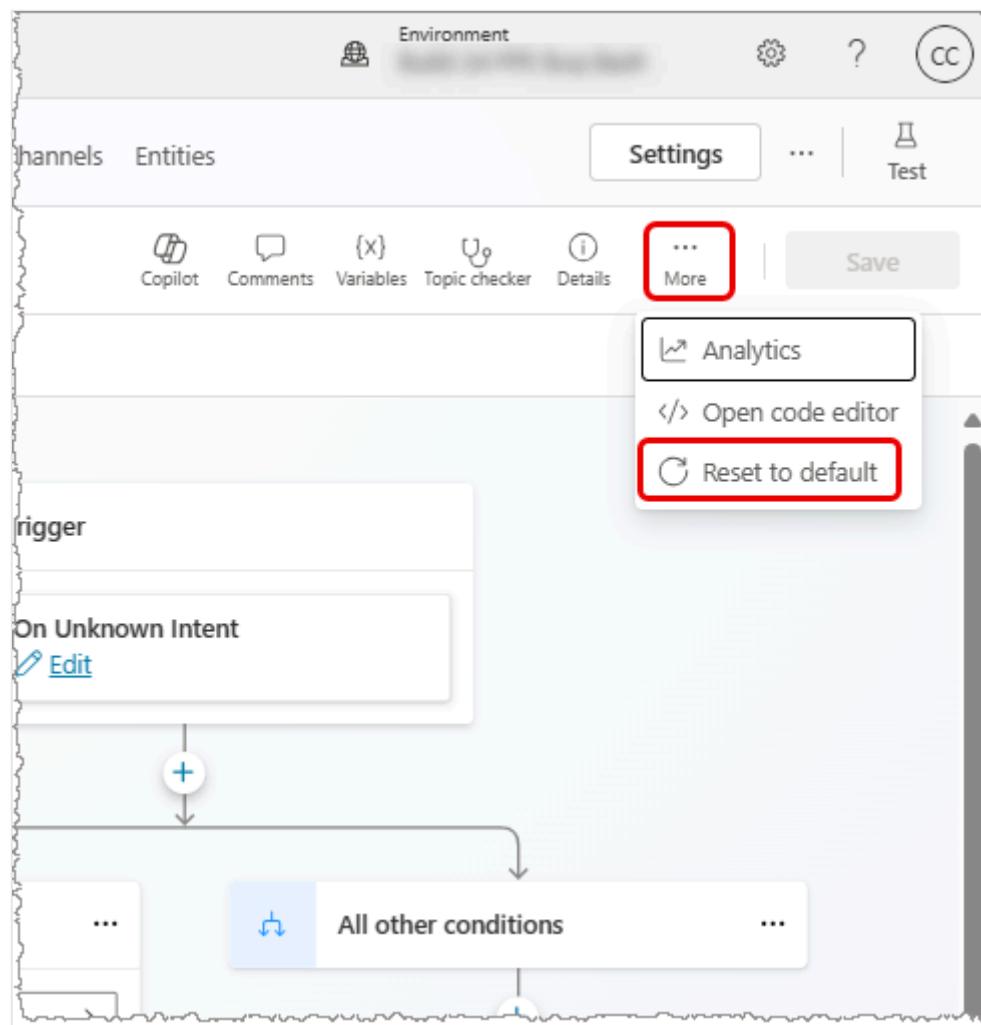
#### (!) Note

Since the fallback topic captures unknown content from the user, it doesn't have any trigger phrases.

## Reset system fallback behavior by deleting the fallback topic

You can return a system fallback topic to its default behavior.

1. Select the **...** on the top menu bar.
2. Select **Reset to default**.



### ⚠️ Warning

Setting a topic back to its default removes any changes you made to it.

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Hand off to a live agent

Article • 11/19/2024

With Copilot Studio, you can configure your agent to hand off conversations to live agents seamlessly and contextually.

When your agent hands off a conversation, it can share the full history of the conversation, and all relevant variables. With this context, a live agent that uses a connected engagement hub can be notified that a conversation requires a live agent, see the context of the prior conversation, and resume the conversation.

For more information about how to configure handoff with [Omnichannel for Customer Service](#), see [Configure handoff to Dynamics 365 Customer Service](#).

<https://www.microsoft.com/en-us/videoplayer/embed/RE4n4G1?postJslIMsg=true> ↗

## ⓘ Note

You can choose to escalate an agent conversation without linking to an engagement hub:

1. At the bottom of the desired topic, select the **Add node** icon  , point to **Topic Management**, and select **Go to another topic**.
2. Select **Escalate**.

**Escalate** is a [system topic](#) that, by default, provides a simple message to a user if they ask for a human agent.

You can edit the topic to include a simple URL to a support website or ticketing system, or to include instructions for emailing or contacting support.

## Prerequisites

- A agent built with Microsoft Copilot Studio
- An engagement hub that is being used by live agents, such as [Omnichannel for Customer Service](#), and you need to configure the connection, as described in [Configure handoff to Omnichannel for Customer Service](#)

## Configure the Escalate system topic

When you create an agent from Dynamics 365 Customer Service, the **Escalate** system topic already includes a **Transfer conversation** node. However, agents created in Copilot Studio aren't configured with this node by default. To add a **Transfer conversation** node to the **Escalate** system topic, follow these steps:

1. In the side navigation pane, select **Topics**, switch to the **System** tab, and select the **Escalate** topic.
2. At the bottom of the topic, select the **Add node** icon  , point to **Topic Management**, and select **Transfer conversation**.

## Trigger handoff to a live agent

Customers engaging with your agent can ask for a live agent at any point in the conversation. This escalation can happen in two ways, with an implicit trigger or an explicit trigger.

Upon triggering the handoff topic, the agent starts the handoff to the configured engagement hub, and sends over all conversation context to find the next best live agent to ramp them up so they can resume the conversation.

### Implicit triggers

In some instances, your agent might be unable to determine the intent of a customer's conversation. For example, the customer might be asking a specific question for which there's no **topic**, or no matching option within a topic.

In other instances, the customer might ask to be handed off to a live agent immediately. For example, a customer might type "talk to agent" mid-way into a conversation.

When the agent detects an escalation in this manner, it automatically redirects the user to the **Escalate system topic**. This type of trigger is known as *implicit* triggering.

### Explicit triggers

When creating topics for your agent, you may determine that some topics require interaction with a human. This type of trigger is known as *explicit* triggering.

In these instances, you must add a **Transfer conversation** node to the topic. This node lets you add a **Private message to agent**, which is sent to the connected engagement hub to help the live agent understand the history and context of the conversation.

### Note

Conversations that reach this node are marked as **Escalated** sessions in [reporting analytics](#).

To configure explicit triggering for a topic:

1. At the bottom of the topic, select the **Add node** icon  , then select **Send a message** to add a message node. Enter what the agent should say to indicate that transfer to a live agent is about to occur.
2. Below the message node, select the **Add node** icon  , point to **Topic Management**, and select **Transfer conversation**.
3. Enter an optional private message to the live agent in the **Transfer conversation** node. This optional message can be useful if you have multiple topics with **Transfer conversation** nodes as the information is stored in the [va\\_AgentMessage context variable](#).

The topic starts the transfer to a live agent when this node is reached. You can test the handoff by triggering the topic in the test canvas.

### Note

Once you add a **Transfer conversation** node into a conversation, each time you trigger handoff your users will see a "No renderer for this activity" message on the demo website. This message suggests the need to [customize your chat canvas](#) to implement custom client-side code that brings in a human agent from your engagement hub into the conversation.

## Context variables available upon handoff

Beyond providing an automated way for a conversation to be ported into an engagement hub, it's important to ensure that the best agent for a specific problem is engaged. To help route conversations to the most appropriate live agent there are context variables that are also passed to the engagement hub.

You can use these variables to automatically determine where the conversation should be routed. For example, you might have added **Transfer conversation** nodes to several different topics, and you want to route conversations related to certain topics to specific agents.

The following table lists the context variables available by default.

[Expand table](#)

Context	Purpose	Example
va_Scope	Route escalations to a live agent.	"agent"
va_LastTopic	Route escalations to a live agent and help them ramp-up. Includes the last topic that was triggered by an utterance from the user.	"Return items"
va_Topics	Ramp-up a live agent. Only includes topics triggered by customers using a trigger phrase. Doesn't include topics that were redirected to.	[ "Greetings", "Store Hours", "Return Item" ]
va_LastPhrases	Route escalation to a live agent and help them ramp-up.	"Can I return my item"
va_Phrases	Ramp-up a live agent.	[ "Hi", "When does store open", "Can I return my item" ]
va_ConversationId	Uniquely identify an agent conversation.	6dba796e-2233-4ea8-881b-4b3ac2b8bbe9
va_AgentMessage	Ramp-up a live agent.	"Got a gift from: HandoffTest"
va_BotId	Identify the agent that's handing off a conversation.	6dba796e-2233-4ea8-881b-4b3ac2b8bbe9
va_Language	Route escalation to a live agent.	"en-us"
All user-defined topic variables	Ramp-up a live agent.	@StoreLocation = "Bellevue"

A customer might go through several topics prior to escalating. Your agent gathers all context variables across topics and merges them before sending to the engagement hub.

If there are topics with similarly named context variables, the agent promotes the most recently defined topic variable.

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Customize the look and feel of an agent

07/22/2025

Your agent's canvas determines its look and feel. You can customize the canvas in two ways, depending on the complexity of the desired changes:

- [Customize the default canvas](#) with JavaScript styling in the HTML code of the website where you deploy your agent. This approach is useful if you want to make small customizations without investing in code development.
- [Use a custom canvas](#) based on the [Bot Framework Web Chat canvas](#). This approach requires extensive developer knowledge. It's useful for organizations that want an entirely custom experience.

You can also combine the customized canvas with [configuring your agent to automatically start the conversation](#).

Lastly, you can [change the name and icon](#) of your agent directly from the portal.

After you [publish an agent](#), your customers can [use the agent's Web Chat canvas to interact with it](#).

## Important

You can install and use the sample code included in this article only for use with Copilot Studio. The sample code is licensed "as is" and is excluded from any service level agreements or support services. You bear the risk of using it.

Microsoft gives no express warranties, guarantees, or conditions and excludes all implied warranties, including merchantability, fitness for a particular purpose, and non-infringement.

## Change the agent name and icon

Web app

## Important

- After you update the icon for an agent, it might take up to 24 hours for the new icon to appear everywhere.

- If your agent is connected to [Omnichannel for Customer Service](#), its name is defined by the **Display name** property in the Azure portal registration.
- If your agent is meant to be published to Teams, review the [requirements for icons](#) in the Microsoft Teams developer documentation.

1. Go to the **Overview** page for your agent.
2. Next to **Details** select **Edit**.
3. Change the agent's name and icon as desired.
4. Select **Save**.

## Customize the default canvas (simple)

Configure how the chat canvas looks with some simple CSS and JavaScript styling options.

First, you need to configure where you're deploying your agent canvas.

1. [Create and publish an agent](#).
2. Copy the following HTML code and save it to a file named `index.html`. Alternatively, copy and paste the code into the [w3schools.com HTML try it editor](#).

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
 <meta name="description" content="Contoso Web Chat Assistant" />
 <title>Contoso Sample Web Chat Test</title>
 <script src="https://cdn.botframework.com/botframework-
webchat/latest/webchat.js"></script>
 <script>
 let webChatInstance = null;
 let directLineUrl = null;

 // Replace with your token endpoint
 const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";
 const styleOptions =
 {"accent": "#0078D4", "autoScrollSnapOnPage": true, "autoScrollSnapOnPageOffset": 0, "avatarBorderRadius": "7%", "avatarSize": 31, "backgroundColor": "#e8e9eb", "botAvatarBackgroundColor": "#fffffff0", "botAvatarImage": "https://powercatexternal.
```

```
blob.core.windows.net/creatorkit/Assets/ChatbotLogoBlue.png", "botAvatarInitia
ls": "B", "bubbleAttachmentMaxWidth": 480, "bubbleAttachmentMinWidth": 250, "bubble
Background": "#f0eded", "bubbleBorderColor": "#f5f5f5", "bubbleBorderRadius": 41, "
bubbleBorderStyle": "solid", "bubbleBorderWidth": 1, "bubbleFromUserBackground": "
#ebefff", "bubbleFromUserBorderColor": "#f5f5f5", "bubbleFromUserBorderRadius": 4
1, "bubbleFromUserBorderStyle": "solid", "bubbleFromUserBorderWidth": 1, "bubbleFr
omUserNubOffset": 0, "bubbleFromUserNubSize": 0, "bubbleFromUserTextColor": "#2424
24", "bubbleImageHeight": 10, "bubbleImageMaxHeight": 240, "bubbleImageMinHeight":
240, "bubbleMessageMaxWidth": 480, "bubbleMessageMinWidth": 120, "bubbleMinHeight":
50, "bubbleNubOffset": 0, "bubbleTextColor": "#242424", "emojiSet": true, "fontSize
Small": "70%", "hideUploadButton": false, "messageActivityWordBreak": "break-
word", "monospaceFont": "Consolas", "paddingRegular": 10, "paddingWide": 10, "primar
yFont": null, "sendBoxBackground": "#e8e9eb", "sendBoxBorderTop": "solid 1px
#808080", "sendBoxButtonColor": "#0078d4", "sendBoxButtonColorOnHover": "#006cbe"
, "sendBoxButtonShadeBorderRadius": 40, "sendBoxButtonShadeColorOnHover": "", "sen
dBoxHeight": 60, "sendBoxPlaceholderColor": "#171616", "sendBoxTextColor": "#2e2d2
d", "showAvatarInGroup": "status", "spinnerAnimationHeight": 16, "spinnerAnimation
Padding": 12, "spinnerAnimationWidth": 16, "subtleColor": "#000000FF", "suggestedAc
tionBackgroundColor": "#006FC4FF", "suggestedActionBackgroundColorOnHover": "#00
78D4", "suggestedActionBorderColor": "", "suggestedActionBorderRadius": 10, "sugge
stedActionBorderWidth": 1, "suggestedActionLayout": "flow", "suggestedActionTextC
olor": "#FFFFFF", "typingAnimationBackgroundImage": "url('https://wpamelia.com
/wp-content/uploads/2018/11/ezgif-2-
6d0b072c3d3f.gif')", "typingAnimationDuration": 5000, "typingAnimationHeight": 20
, "typingAnimationWidth": 64, "userAvatarBackgroundColor": "#222222", "userAvatarI
mage": "https://avatars.githubusercontent.com/u/8174072?
v=4&size=64", "userAvatarInitials": "U"}};

const backgroundImage = "";
document.addEventListener('DOMContentLoaded', () => {
 const root = document.documentElement;
 root.style.setProperty('--primary-color',
createGradient(styleOptions.accent));
 root.style.setProperty('--header-textColor',
styleOptions.suggestedActionTextColor);
 if (backgroundImage) {
 const webchatElement = document.getElementById('webchat');
 webchatElement.style.backgroundImage = `url(${backgroundImage})`;
 webchatElement.style.backgroundSize = 'cover';
 webchatElement.style.backgroundPosition = 'center';
 webchatElement.style.backgroundRepeat = 'no-repeat';
 const overlay = document.createElement('div');
 overlay.className = 'webchat-overlay';
 webchatElement.appendChild(overlay);
 }
});
function createGradient(baseColor) {
 const r = parseInt(baseColor.slice(1,3), 16);
 const g = parseInt(baseColor.slice(3,5), 16);
 const b = parseInt(baseColor.slice(5,7), 16);
 const lighterColor = `#${Math.min(255,
r+30).toString(16).padStart(2,'0')}#${Math.min(255,
g+30).toString(16).padStart(2,'0')}#${Math.min(255,
b+30).toString(16).padStart(2,'0')}`;
 const darkerColor = `#${Math.max(0, r-
30).toString(16).padStart(2,'0')}#${Math.max(0, g-
```

```

30).toString(16).padStart(2,'0')}${Math.max(0, b-
30).toString(16).padStart(2,'0')}`;
 return `linear-gradient(135deg, ${lighterColor}, ${baseColor},
${darkerColor})`;
}
const environmentEndPoint = tokenEndpoint.slice(
 0,
 tokenEndpoint.indexOf("/powervirtualagents")
);
const apiVersion = tokenEndpoint
 .slice(tokenEndpoint.indexOf("api-version"))
 .split("=")[1];
const regionalChannelSettingsURL =
`${environmentEndPoint}/powervirtualagents/regionalchannelsettings?api-
version=${apiVersion}`;
function showChat() {
 const popup = document.getElementById("chatbot-popup");
 const openButton = document.getElementById("open-chat");
 popup.classList.add("visible");
 openButton.classList.add("hidden");
}
function hideChat() {
 const popup = document.getElementById("chatbot-popup");
 const openButton = document.getElementById("open-chat");
 popup.classList.remove("visible");
 openButton.classList.remove("hidden");
}
function createCustomStore() {
 return window.WebChat.createStore(
 {},
 ({ dispatch }) =>
 (next) =>
 (action) => {
 if (action.type === "DIRECT_LINE/CONNECT_FULFILLED") {
 dispatch({
 type: "DIRECT_LINE/POST_ACTIVITY",
 meta: { method: "keyboard" },
 payload: {
 activity: {
 channelData: { postBack: true },
 name: "startConversation",
 type: "event",
 },
 },
 });
 }
 return next(action);
 }
);
}
async function restartConversation() {
 try {
 if (!directLineUrl) {
 console.error("DirectLine URL not initialized");
 return;
 }
 }

```

```
 }
 const response = await fetch(tokenEndpoint);
 const conversationInfo = await response.json();
 if (!conversationInfo.token) {
 throw new Error("Failed to get conversation token");
 }
 const newDirectLine = window.WebChat.createDirectLine({
 domain: `${directLineUrl}v3/directline`,
 token: conversationInfo.token,
 });
 const webchatElement = document.getElementById("webchat");
 webChatInstance = window.WebChat.renderWebChat(
 {
 directLine: newDirectLine,
 styleOptions,
 store: createCustomStore(),
 },
 webchatElement
);
 } catch (err) {
 console.error("Failed to restart conversation:", err);
 }
}
async function initializeChat() {
 try {
 const response = await fetch(regionalChannelSettingsURL);
 const data = await response.json();
 directLineUrl = data.channelUrlsById.directline;
 if (!directLineUrl) {
 throw new Error("Failed to get DirectLine URL");
 }
 const conversationResponse = await fetch(tokenEndpoint);
 const conversationInfo = await conversationResponse.json();
 if (!conversationInfo.token) {
 throw new Error("Failed to get conversation token");
 }
 const directLine = window.WebChat.createDirectLine({
 domain: `${directLineUrl}v3/directline`,
 token: conversationInfo.token,
 });
 webChatInstance = window.WebChat.renderWebChat(
 {
 directLine,
 styleOptions,
 store: createCustomStore(),
 },
 document.getElementById("webchat")
);
 } catch (err) {
 console.error("Failed to initialize chat:", err);
 }
}
initializeChat();
</script>
<style>
```

```
:root {
 --primary-gradient: var(--primary-color);
 --chat-width: 450px;
 --chat-height: 520px;
 --header-height: 56px;
 --border-radius: 16px;
 --transition-speed: 0.3s;
}
* {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
 Roboto, sans-serif;
}
body {
 min-height: 100vh;
 background-color: #f3f4f6;
}
#chatbot-popup {
 display: none;
 position: fixed;
 bottom: 32px;
 right: 32px;
 width: var(--chat-width);
 height: var(--chat-height);
 background: white;
 border-radius: var(--border-radius);
 box-shadow: 0 18px 40px -5px rgba(0, 0, 0, 0.2),
 0 15px 20px -5px rgba(0, 0, 0, 0.1);
 overflow: hidden;
 opacity: 0;
 transform-origin: bottom right;
 transform: scale(0.95);
 transition: all var(--transition-speed) ease-in-out;
 z-index: 999;
}
#chatbot-popup.visible {
 display: block;
 opacity: 1;
 transform: scale(1);
}
#chatbot-header {
 background: var(--primary-color);
 padding: 16px 20px;
 height: var(--header-height);
 display: flex;
 justify-content: space-between;
 align-items: center;
 color: var(--header-textColor);
}
.header-title {
 display: flex;
 align-items: center;
 gap: 12px;
```

```
 font-size: 16px;
 font-weight: 500;
}
.header-buttons {
 display: flex;
 gap: 12px;
 align-items: center;
}
.icon-button {
 background: none;
 border: none;
 color: var(--header-textColor);
 cursor: pointer;
 padding: 8px;
 border-radius: 8px;
 display: flex;
 align-items: center;
 justify-content: center;
 transition: all 0.2s ease;
}
.icon-button:hover {
 color: var(--header-textColor);
 background: rgba(255, 255, 255, 0.1);
}
.icon-button:focus {
 outline: 2px solid rgba(255, 255, 255, 0.5);
 outline-offset: 2px;
}
#webchat {
 height: calc(100% - var(--header-height));
 background-color: #f9fafb;
 position: relative;
}
.webchat-overlay {
 position: absolute;
 top: 0;
 left: 0;
 right: 0;
 bottom: 0;
 background: rgba(255, 255, 255, 0.85);
 pointer-events: none;
 z-index: 1;
}
#webchat > div {
 position: relative;
 z-index: 2;
}
#webchat .webchat__basic-transcript__content {
 white-space: pre-wrap !important;
 word-break: break-word !important;
}
#webchat .webchat__bubble__content {
 padding: 8px 12px !important;
}
#webchat .webchat__bubble {
```

```
 max-width: 85% !important;
 margin: 8px !important;
}
#webchat .webchat__basic-transcript__content ul,
#webchat .webchat__basic-transcript__content ol,
#webchat .webchat__bubble__content ul,
#webchat .webchat__bubble__content ol {
 padding-left: 24px !important;
 margin: 8px 0 !important;
 list-style-position: outside !important;
}
#webchat .webchat__basic-transcript__content li,
#webchat .webchat__bubble__content li {
 margin: 4px 0 !important;
 padding-left: 4px !important;
}
#open-chat {
 position: fixed;
 bottom: 32px;
 right: 32px;
 width: 64px;
 height: 64px;
 border-radius: 50%;
 background: var(--primary-gradient);
 border: none;
 cursor: pointer;
 display: flex;
 align-items: center;
 justify-content: center;
 box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1);
 transition: all var(--transition-speed) ease-in-out;
 z-index: 998;
}
#open-chat.hidden {
 opacity: 0;
 transform: scale(0.95) translateY(10px);
 pointer-events: none;
}
#open-chat:hover {
 transform: translateY(-4px);
 box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1);
}
#open-chat:focus {
 outline: 3px solid rgba(79, 70, 229, 0.5);
 outline-offset: 2px;
}
#open-chat svg {
 width: 28px;
 height: 28px;
 color: white;
 transition: transform 0.2s ease;
}
.main-content {
 max-width: 1200px;
 margin: 0 auto;
```

```
padding: 48px 24px;
}
.main-content h1 {
 font-size: 36px;
 color: #111827;
 text-align: center;
}
.main-content p {
 font-size: 18px;
 color: #4b5563;
 line-height: 1.6;
 margin-bottom: 48px;
 text-align: center;
 max-width: 800px;
 margin-left: auto;
 margin-right: auto;
}
.content-grid {
 display: grid;
 grid-template-columns: repeat(2, 1fr);
 gap: 32px;
 margin-bottom: 32px;
}
.content-box {
 background: linear-gradient(135deg, #e6e6e6, #c4c4c4, #9f9f9f);
 padding: 32px;
 border-radius: 12px;
 box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1),
 0 2px 4px -1px rgba(0, 0, 0, 0.06);
 min-height: 300px;
 display: flex;
 flex-direction: column;
 justify-content: center;
 align-items: center;
 text-align: center;
 position: relative;
 overflow: hidden;
}
.content-box::before {
 content: '';
 position: absolute;
 top: 0;
 left: 0;
 right: 0;
 height: 4px;
}
.content-box.featured {
 grid-column: span 2;
 min-height: 350px;
 background: linear-gradient(135deg, #e6e6e6, #c4c4c4, #9f9f9f);
 color: #000000;
}
.content-box h2 {
 font-size: 24px;
 margin-bottom: 16px;
```

```
 position: relative;
 }
 .content-box p {
 font-size: 16px;
 color: #6b7280;
 margin-bottom: 0;
 }
 .content-box.featured p {
 color: #000000;
 }
 @media (max-width: 768px) {
 .content-grid {
 grid-template-columns: 1fr;
 }
 .content-box.featured {
 grid-column: span 1;
 }
 .main-content {
 padding: 24px 16px;
 }
 .main-content h1 {
 font-size: 28px;
 }
 #chatbot-popup {
 width: 100%;
 height: 100%;
 bottom: 0;
 right: 0;
 border-radius: 0;
 }
 }
}
</style>
</head>
<body>
 <div class="main-content">
 <h1>Header</h1>
 <p>Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisque faucibus ex sapien vitae pellentesque sem placerat.</p>
 <div class="content-grid">
 <div class="content-box featured">
 <h2>Featured Content</h2>
 <p>Primary content area with custom styling and gradient background</p>
 </div>
 <div class="content-box">
 <h2>Section One</h2>
 <p>Content box with minimal design</p>
 </div>
 <div class="content-box">
 <h2>Section Two</h2>
 <p>Another content section with consistent styling</p>
 </div>
 </div>
 </div>
 <div id="chatbot-popup" role="complementary" aria-label="Chat Assistant">
```

```
<div id="chatbot-header">
 <div class="header-title">
 <svg
 class="chat-icon"
 width="24"
 height="24"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M21 15a2 2 0 0 1-2 2H7l-4 4V5a2 2 0 0 1 2-2h14a2 2 0 0 1 2 2z"
 ></path>
 </svg>
 Contoso Assistant
 </div>
 <div class="header-buttons">
 <button
 class="icon-button"
 id="restart-button"
 onclick="restartConversation()"
 aria-label="Restart Conversation"
 >
 <svg
 width="20"
 height="20"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M3 12a9 9 0 1 0 9-9 9.75 9.75 0 0 0-6.74 2.74L3 8"
 ></path>
 <path d="M3 3v5h5"></path>
 </svg>
 </button>
 <button
 class="icon-button"
 id="close-button"
 onclick="hideChat()"
 aria-label="Close Chat"
 >
 <svg
 width="20"
 height="20"
 viewBox="0 0 24 24"
 fill="none"
 >
```

```

 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <line x1="18" y1="6" x2="6" y2="18"></line>
 <line x1="6" y1="6" x2="18" y2="18"></line>
 </svg>
 </button>
 </div>
<div id="webchat" role="main"></div>
</div>
<button
 id="open-chat"
 onclick="showChat()"
 aria-label="Open Chat Assistant"
>
 <svg
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M21 15a2 2 0 0 1-2 2H7l-4 4V5a2 2 0 0 1 2-2h14a2 2 0 0 1 2 2z"
 ></path>
 </svg>
</button>
</body>
</html>

```

3. Retrieve the token endpoint for your agent.
4. In `index.html`, at the line `const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";`, replace the placeholder with the token endpoint for your agent.
5. Open `index.html` using a modern browser (for example, Microsoft Edge) to open the agent in the custom canvas.
6. Test the agent to ensure you're receiving responses from it and that it's working correctly.

If you encounter problems, make sure you published your agent, and that the token endpoint is in the correct place. The token endpoint should be after the equals sign (=) at the line `const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";`, and surrounded by double quotation marks (").

## Retrieve the token endpoint for your agent

To customize your canvas, whether it's the default canvas or a custom one you connect to, you need the token endpoint for your agent.

1. In the navigation menu under **Settings**, select **Channels**.
2. Select **Email**. The configuration panel for this channel appears.
3. Next to **Token Endpoint**, select **Copy**.

## Customize the agent icon, background color, and name

Once you get the customized canvas working with your agent, you can make changes to it.

You can use the JavaScript `styleOptions` options to configure many predefined styles.

See [Web Chat customization](#) for links to the `defaultStyleOptions.js` file and more information on what you can customize and how it will look.

### Change the agent icon

1. Update the `index.html` file with the following sample code:

```
JavaScript
```

```
const styleOptions = {
 accent: '#00809d',
 botAvatarBackgroundColor: '#FFFFFF',
 botAvatarImage: 'https://learn.microsoft.com/azure/bot-
service/v4sdk/media/logo_bot.svg',
 botAvatarInitials: 'BT',
 userAvatarImage: 'https://avatars.githubusercontent.com/u/661465'
};
```

2. Replace the agent and user avatar images with your company images.

If you don't have an image URL, you can use a Base64-encoded image string instead.

### Change the background color

1. Update the `index.html` file with following sample code:

```
JavaScript
```

```
const styleOptions = {
 backgroundColor: 'lightgray'
};
```

2. Change `backgroundColor` to any color you wish. You can use standard CSS color names, RGB values, or HEX.

## Change the agent name

1. Update the `<h1>` text in the `index.html` file with the following code:

HTML

```
<body>
 <div id="banner">
 <h1> Contoso agent name</h1>
 </div>
```

2. Change the text to whatever you want to call the agent. You can also insert an image, although you might need to style it to ensure it fits within the heading section.

## Customize and host your chat canvas (advanced)

You can connect your Copilot Studio agent with a custom canvas that is hosted as a standalone web app. This option is best if you need to embed a customized iFrame across multiple web pages.

 Note

Hosting a custom canvas requires software development. This guidance is intended for experienced IT professionals, such as IT admins or developers who have a good understanding of developer tools, utilities, and IDEs.

## Pick a sample to customize

We recommend starting with one of these samples custom-built to work with Copilot Studio:

- [Full bundle](#) is a custom canvas capable of showing all rich content from Copilot Studio.  
For example:

card flight tracking

22 minutes ago

Showing flight, tracking

 Flight Status  
**DELAYED**

Passengers	Seat
Sarah Hum	14A
Jeremy Goldberg	14B
Evan Litvak	14C

---

Flight	Departs	Arrives
KL0605	<b>10:10 ...</b>	<b>12:00 AM</b>

---

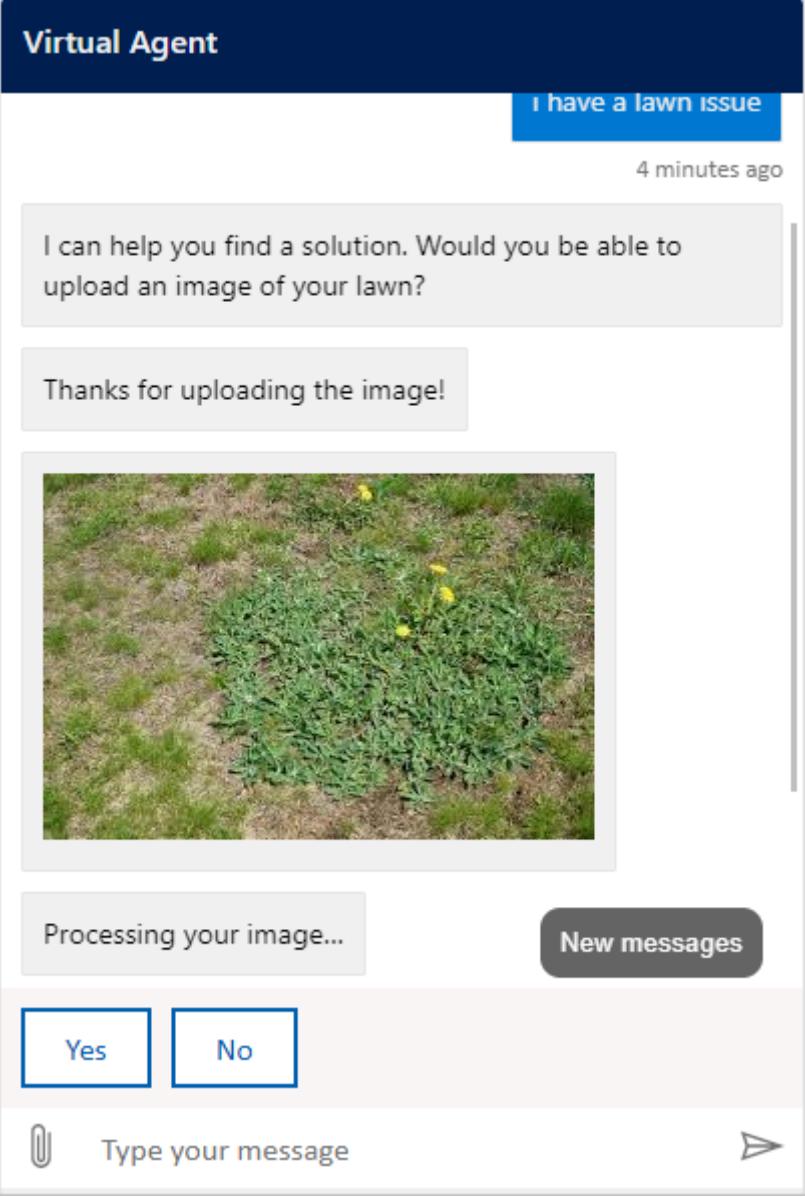
Amsterdam  San Francisco

**AMS** **SFO**

22 minutes ago

 Type your message 

- [Location and file uploading ↗](#) is a custom canvas capable of getting a user's location and sending it to a Copilot Studio agent. For example:



Or you can pick from [other sample Web Chat canvases](#) provided by Bot Framework.

## Customize canvas using styleSetOptions

As with customizing the default canvas, you can use `styleSetOptions` to customize the custom canvas. All customizable properties are listed in [defaultStyleOptions.js](#). For more information on what you can customize and how it will look, see [Web Chat customization](#).

## Deploy your customized canvas

To host your custom canvas, deploy all files to a web app.

## Related content

- [Bot Framework Web Chat](#)

- Generating a Direct Line token

# Create and manage solutions in Copilot Studio

Article • 11/19/2024

When you create an agent in Copilot Studio, it's collected within a Power Platform [solution](#). You can create custom solutions to manage your agents across multiple [environments](#), or for pipeline deployments and other application lifecycle management (ALM) scenarios.

You can view, export and import solutions, set your preferred or default environment, configure pipelines, and customize and manage your solutions directly in Microsoft Copilot Studio.

Solutions can be useful if you use [different environments for different purposes](#), or you employ ring-deployment methodologies. For example, you might have an environment to test and validate agents, another environment to test agents for only a subset of users, and a production environment where you share agents with customers.

## Prerequisites

- The solution explorer in Copilot Studio inherits the security role privileges of the user. This means a user can only perform the solution-management tasks in Copilot Studio that they can perform in the Power Apps admin center. Ensure your users have the correct roles and permissions to manage solutions. Learn more about [configuring user security in an environment](#).
- Familiarity with Power Apps [solution concepts](#).

## View solutions in Copilot Studio

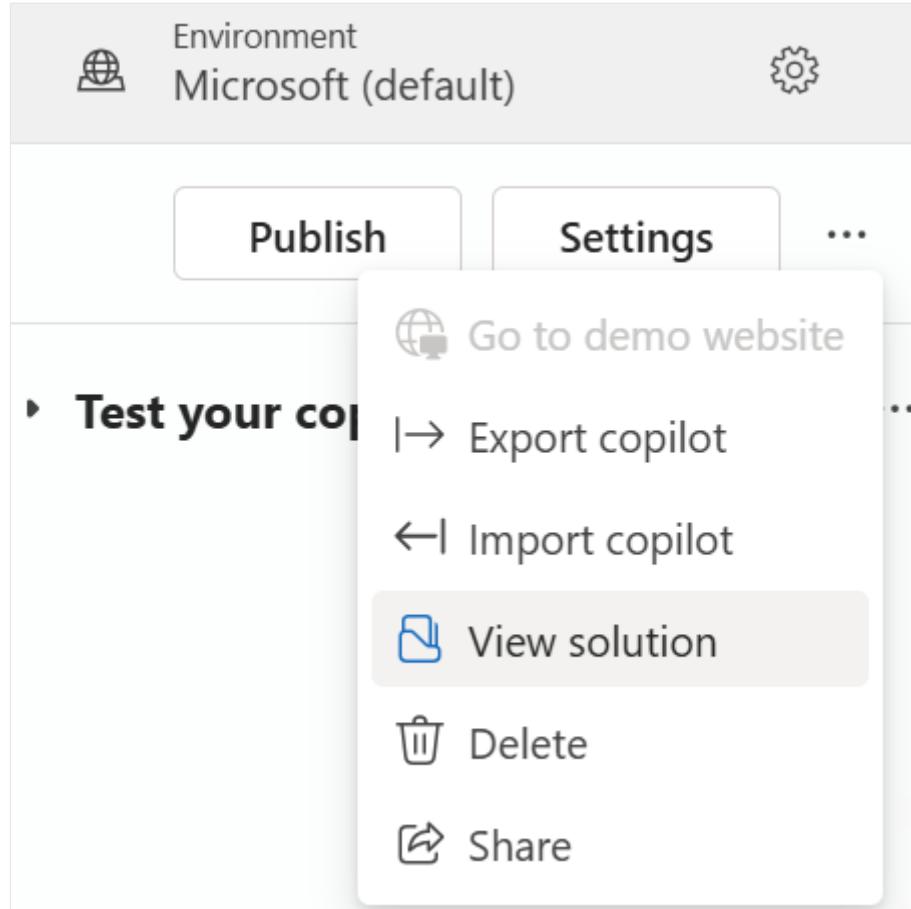
A solution acts as a "carrier" for your agents. When you create an agent, it's automatically added to a default solution, [which you can change](#).

You can view the solution that contains your agent in the solution explorer in Copilot Studio.

You can access the solution used by a specific agent in a couple of ways:

- From the agent's overview page in Copilot Studio:
  1. In Copilot Studio, open the agent that you want to manage the solution for.

2. On the agent's **Overview** page, select the menu icon (...) just below the environment name, next to **Settings**, and then select **View solution**.



- From the agent's settings in Copilot Studio:
  1. In Copilot Studio, open the agent that you want to manage the solution for.
  2. On the agent's **Overview** page, select **Settings**, just below the environment name. You're taken to the settings page for the agent.
  3. On the **Agent details** or **Advanced** page, select **View solution**.

The screenshot shows the Copilot Studio interface with the title "Copilot Studio" at the top. On the left is a sidebar with icons for Home, Create, Copilots, Library, and three dots for more options. The main area is titled "Settings". A vertical list of settings includes: Copilot details, Generative AI, Security, Authoring Canvas, Entities, Skills, Voice, Languages, Language understanding, Component collections, and Advanced. The "Advanced" option is highlighted with a blue border. To the right of the list is a panel titled "Advanced" which contains sections for "Application Insights" and "Metadata". It also shows a "View solution" section for "Common Data Services Default Solution" with a "Save" button. The "Save" button is located in a light gray box.

## Settings

- Copilot details
- Generative AI
- Security
- Authoring Canvas
- Entities
- Skills
- Voice
- Languages
- Language understandi...
- Component collections
- Advanced**

### Advanced

**Application Insights**  
Automatically send telemetry for your copilot into an Application Insights resource. You can also send custom telemetry events from within your topics. [Learn more](#)

**Metadata**  
Metadata holds all the information to uniquely identify your copilot. This includes things like the name, version, and publisher. It's used for things like calling your copilot or using your copilot in external scenarios.

**View solution**  
Common Data Services Default Solution

Save

To access the solution explorer:

- From the side navigation pane in Copilot Studio:
  - In Copilot Studio, select the menu icon (...) on the side navigation pane, and then select **Solutions**.

The screenshot shows the Microsoft Power Platform home page. On the left, a vertical sidebar lists navigation options: Home (house icon), Create (plus icon), Copilots (blue hexagon with star icon), Library (grid icon), and three dots for more options. The main content area is titled "Power Platform". It features five tiles: "Power Apps" (purple and pink square icon), "Power Automate" (blue right-pointing arrow icon), "Power Pages" (purple and blue square icon), "Power Platform Admin Center" (green and blue 'P' icon), and a "Learn more about AI integration" section with "Azure Cognitive Service" (blue and pink 'A' icon) and "Azure Open AI" (blue starburst icon). At the bottom, there's a "Solutions" tile (blue square icon).

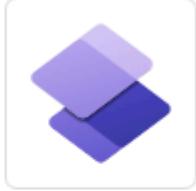
## Power Platform



Power Apps



Power Automate

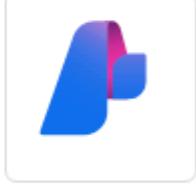


Power Pages

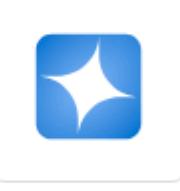


Power Platform Admin Center

... [Learn more about AI integration](#)

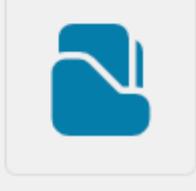


Azure Cognitive Service



Azure Open AI

## Solutions



Solutions

The solution explorer opens in a new browser tab.

# Manage solutions in Copilot Studio

You can perform all the usual solution-related tasks directly in Copilot Studio, such as adding or removing components, exporting, and importing solutions:

- Set your preferred solution
- Create a custom solution
- Import and export solutions with your agent
- Add components to an agent in a custom solution
- Upgrade or update a solution with an agent
- View solution layers and remove unmanaged layers from a managed agent
- Create and manage pipelines for deployment

## 💡 Tip

Ignore instructions in the Power Apps documentation that tell you to sign in to Power Apps. If you're logged in to Copilot Studio, you don't need to go to Power Apps.

## Set your preferred solution

1. Open the solution explorer in Copilot Studio.
2. Select **Set preferred solution** on the top menu bar, above the list of solutions.
3. Choose the solution you want agents to be created in by default.

See the [Power Apps documentation](#) for more information on setting your preferred solution.

## Create a custom solution

Copilots are created in a default solution automatically. To export, import, and manage agents between environments, you need to create and use a custom solution.

1. Open the solution explorer in Copilot Studio.
2. Select **New solution**, and follow the instructions and requirements in the Power Apps documentation to [create a solution](#).
3. The solution you created should open automatically. If it doesn't, return to the solution explorer and select your new custom solution.

## Import and export solutions

You can export and import the solutions that contain your agents from one environment to another.

Follow the steps at [Export and import solutions in Copilot Studio](#) for more details on how to use solutions to export and import agents.

## Add components to an agent in a custom solution

You might need to add components to your solution if you want to export an agent.

Follow the instructions and requirements in the Power Apps documentation to [add an existing component to a solution](#).

## Upgrade or update a solution with an agent

Follow the instructions and requirements in the Power Apps documentation to [upgrade or update a solution](#).

## Remove an unmanaged layer from a managed agent

Managed and unmanaged solutions exist at different levels in a Microsoft Dataverse environment. To learn more, go to the Power Apps documentation for [Solution layers](#).

1. Open your solution.
2. Select **Agents**, and then select your agent in the list.
3. Select **Commands (:**), select **Advanced**, and then select **See solution layers**.

Display name	Name	Type	Managed...
<input checked="" type="checkbox"/> Ask an expert adaptive card v2	Ask an expert adaptive card v2	Flow	
AuthToken	Remove	Bot variable	
Check ticket status	Edit	Flow	
Confirmed Failure	See Analytics	Topic	
Confirmed Success	Details	Topic	
Create a new ticket	Turn on	Flow	
CSAT Rating	Add required components	Bot entity	
End of Conversation	Managed properties	Topic	
Escalate	Show dependencies	Topic	
Fallback	See solution layers	Topic	
Get user name	Escalate	Flow	

4. In the **Solution Layers** page, select the unmanaged layer, and then select **Remove unmanaged layer**.

## Create and manage solution pipelines

Solution pipelines are a feature of Power Platform that you can use to automate the deployment of solutions across environments.

1. Open the solution explorer in Copilot Studio.
2. Select **Pipelines** underneath the list of solutions.

Follow the instructions and requirements in the Power Apps documentation for [creating and managing pipelines](#).

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#)

# Export and import agents using solutions

Article • 11/19/2024

You can export and import agents using [solutions](#) so you can move your agents across multiple [environments](#).

First, you create a custom solution and add your agent to it. Then you export the solution with your agent, and import it into another environment. You can also upgrade or update a solution with an agent, and remove an unmanaged layer from a managed agent.

See the article [View and manage solutions in Copilot Studio](#) for more details on the solution explorer in Copilot Studio.

## ⓘ Note

You can't export [topic-level or node-level comments](#) when you export an agent.

## Prerequisites

- You must have at least the System Customizer security role to use this feature. Learn more about [configuring user security in an environment](#).
- You should be familiar with solution concepts, as described in [Solution concepts](#), and [Create and manage solutions in Copilot Studio](#).

## Things to know

- You can't export [topic-level or node-level comments](#) when you export an agent.
- Removing an agent from a solution doesn't remove its components from the solution. You should remove the components separately.

## ⚠ Warning

Don't remove unmanaged agents components, such as agent topics, directly from the solution unless you've also removed the agent from the solution.

You should only make changes to topics as part of the standard authoring process in the Copilot Studio UI. Removing or changing an agent's components directly from the solution will cause the export and import to fail.

- You can't export managed solutions. When you create a solution, it's unmanaged by default. If you [change it to a managed solution](#), you won't be able to export it. You'll need to create a new solution.
- It isn't possible to export a solution that contains an agent with periods (.) in the name of any of its topics.

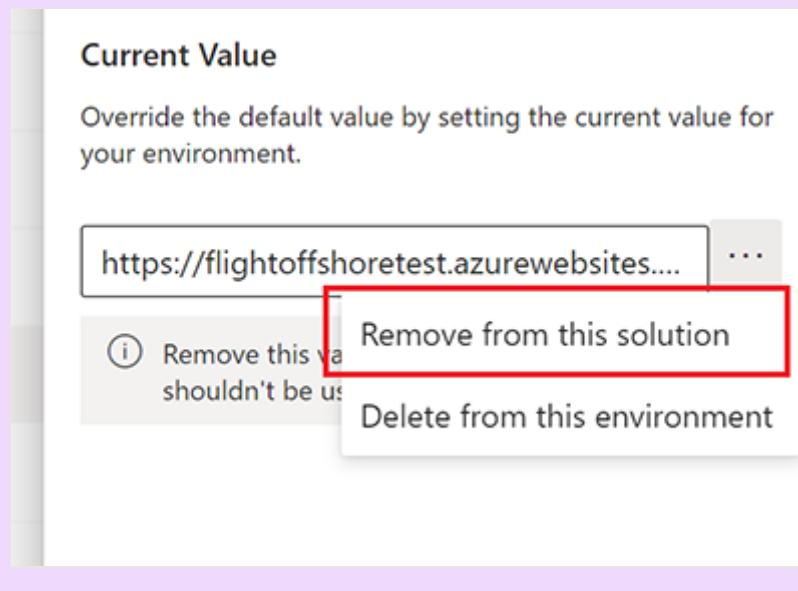
## Create a custom solution and add your agent

1. Follow the steps in [Create and manage solutions in Copilot Studio](#) to create a new solution.
2. The solution you created should open automatically. If it doesn't, select it in the list of solutions.
3. Select **Add existing**, and then select **Agent**, then **Agent** again.
4. In the **Add existing copilots** list, select the agent or agents you want to export, and then select **Add**.

### Note

For a **Classic** chatbot:

If you're using [\*\*skills\*\*](#), remove the skill's environment variable **Current Value** by selecting **Remove from this solution**.



Removing an agent from a solution doesn't remove its components from the solution. You must remove the components separately.

## Add components to an agent in a custom solution

If you add components, such as topics or flows, to your agent in Copilot Studio, you must also add them to the agent in your unmanaged solution before you export.

1. Open the solution that contains your agent.
2. In the **Objects** pane, locate your agent under **Agents**.
3. Select **Commands (:)**, select **Advanced**, and then select **Add required objects**.

You should also review the instructions and requirements in the Power Apps documentation to [add an existing component to a solution](#).

## Export the solution with your agent

You export and import agent by exporting and importing the solutions that contain them from one environment to another.

Follow the instructions and requirements in the Power Apps documentation to [export solutions](#).

### Important

If your agent has a large number of components (for example, more than 250 topics or more than 100 entities), see [Export using the classic experience](#).

## Import the solution with your agent

Follow the instructions and requirements in the Power Apps documentation to [import solutions](#).

If the import isn't successful, select **Download log file** to download an XML file that contains details about what caused the import to fail. The most common reason an import fails is that the solution doesn't contain some required components.

1. Configure [user authentication](#) again.
2. In the **Objects** pane, select **Agents**. Then select the agent's name to open the agent.

## Important

- You must [publish your imported agent](#) before it can be shared.
- It may take up to 24 hours for your agent's icon to appear everywhere.

# Export and import agents in Teams

In Microsoft Teams, you can export and import agents between individual teams using the Power Apps Teams app.

For example, if you export a working agents from a Teams team (team A), and then import it into team B, you don't have to build it from scratch for team B.

Follow the instructions in [Export customizations from Dataverse for Teams](#).

When selecting the customizations to export, select the agent you want to export. You can use the filter on the side pane to only show agents for a specific team.

The agent is saved to the *Downloads* folder on your computer.

To import in Teams, follow the instructions in [Import customizations into Dataverse for Teams](#).

Your agent appears in the Copilot Studio app in Teams, under the team you imported it into.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Create reusable component collections (preview)

06/04/2025

[This article is prerelease documentation and is subject to change.]

A component collection is a collection of reusable agent components. Copilot components include topics, knowledge, actions, and entities. You can share component collections between multiple agents within your environment. You can also use a [solution](#) to export and import component collections to move their content across multiple [environments](#) to meet your [application lifecycle management \(ALM\)](#) scenarios.

To edit a component collection, or add a component collection to an agent, you must be the person who created the collection, or a user with the system customizer or system admin role. Once a component collection is available for an agent, all agent authors can see and use the collection but they can't change the components inside the collection. Learn more about [configuring user security in an environment](#).

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

## Create component collections

You can create component collections directly from the [Settings](#) page for an agent. You can also create them from the [Library](#) page. When you add agent components to a collection, this operation moves the selected components to the collection. The originating agent now contains references to these components within the collection. You can now share the collection with other agents in your environment, or package it in a solution to be imported into other environments.

## Create a component collection from an agent

1. Open the agent that contains the components you want to make available as a component collection.
2. Go to **Settings** and select **Component collections**.
3. If you're about to create the first component collection in your environment, select **Create**. Otherwise, select **New**.
4. Enter a name and a description for your component collection.
5. Select **Next**.
6. Select the components you want to add to the component collection. You can use the categories and the Search field to narrow down the list of components.
7. Select **Next**.
8. Review the content of your collection, and select **Create**.

## Create a component collection from the library

1. In the side navigation, select **Library**.
2. Select **Add new**, then select **Component collection**.
3. Enter a name and a description for your component collection.
4. Select **Finish**.
5. [Connect the agent that contains the components you want to share](#) to your new component collection.
6. [Add components](#) from the selected agent to the collection.

## Connect an agent to a component collection

1. Go to the library and select the desired component collection. You can select the **Component collection** category and the Search field to narrow down the list of component collections.
2. In the **Installed on** area, select **Add agent**.
3. Select the agent in which you want to use the component collection, and select **Add**.

You can add any components from the connected agents to your component collection. All the connected agents can also use any components from that collection.

## Add components from an agent to a collection

1. Go to the library and select the desired component collection. You can select the **Component collection** category and the Search field to narrow down the list of component collections.
2. [Connect the agent](#connect-an-agent-to-a-component-collection) that you want to add to the component collection, if it isn't already listed in the **Installed on** area.
3. Select **Add**, then select **Add from an agent**.
4. Select the desired agent, and select **Next**.
5. Select the components you want to add to the component collection. You can use the categories and the Search field to narrow down the list of components.
6. Select **Next**.
7. Review the content of your collection, and select **Add to collection**.

## Create a solution to export and import component collections

You use solutions to export agent components from one environment and import them into another. The solution acts as a carrier for the components. You can export and import multiple agent components in one solution.

1. Go to the [Solutions](#) page.
2. Select **New solution**.
3. Enter the following information for the new solution:
  - **Display name:** The name to show in the list of solutions. You can change it later.
  - **Name:** The unique name of the solution, generated from **Display name**. You can only change it *before* you save the solution.
  - **Publisher:** Select the default publisher or create a new one. Consider creating a single publisher and use it consistently across all environments for your solution. Learn more about [solution publishers](#).
  - **Version:** Enter a number for the version of your solution. The version number is included in the file name when you export the solution.
4. Select **Create**.

# Add component collections to a solution

1. Go to the [Solutions](#) page, and open the desired solution if it isn't already.
2. Select **Add existing**, point to **Agent**, and select **Component collection**.  
A panel appears, showing all available component collections.
3. Select the component collections you want to export, and then select **Add** at the bottom of the panel.

## Export a solution to share component collections in other environments

You export and import component collections by exporting and importing the solutions that contain them from one environment to another.

1. Go to the [Solutions](#) page.
2. Select the **Commands** icon (:) for the solution that contains the component collections you want to export, and then select **Export solution**.
3. Enter or select the following options:
  - **Version number:** Copilot Studio automatically increments your solution version. You can accept the default or enter your own.
  - **Export as:** Select the package type, either **Managed** or **Unmanaged**. Learn more about [managed and unmanaged solutions](#).
4. Select **Export**. The export might take several minutes to complete. Watch the status message at the top of the [Solutions](#) page.
5. When the export is finished, select the **Download** button to download a .zip file.

The file name has this format: SolutionName\_Version\_SolutionType.zip—for example, ContosoSolution\_1\_0\_0\_1\_managed.zip.

## Import a solution to add component collections to an environment

1. Go to the [Solutions](#) page.
2. Select the environment where you want to import your component collection.

3. Select **Import solution**.
4. In the **Import** panel, select **Browse**, select the .zip file that contains the solution with your component collection, and then select **Next**.
5. Select **Import**. The import might take several minutes to complete. Watch the status message at the top of the **Solutions** page.
6. Wait a few moments while the import completes.

If the import isn't successful, select **Download log file** to download an XML file that contains details about what caused the import to fail. The most common reason an import fails is that the solution doesn't contain some required components.

## Add imported component collections to your agent

After importing a component collection, you can use it in your agents.

1. Open the agent where you want to use the component collection.
2. Go to the **Settings** page and select **Component collections**.
3. Next to the imported component collection, select the three dots (...) and select **Add to agent**.

A message appears prompting you to confirm.

4. Select **Add to agent**.

The name of your agent appears under **Active for** for this component collection.

5. Review the components from the imported collection, and publish the agent to make the changes available to your customers.

## Add components to a component collection in a custom solution

If you add new agent components to a component collection in Copilot Studio, you must also add them to any unmanaged solution that refers to this component collection.

1. Go to the **Solutions** page, and open the unmanaged solution that contains the component collection to update.

2. In the **Objects** pane, select **Agent component collections**.
3. Select the **Commands** icon (:) for the collection you want to update, point to **Advanced**, and select **Add required objects**.
4. Select **OK**.

In the **Objects** pane, you can select **Agent components** to verify that the expected components now appear.

## Open the solution explorer

In Copilot Studio, select the three dots (...) in the side navigation, and select **Solutions**.

The **Solutions** page opens in a new browser tab.

## Known issue

If you uploaded *files* as knowledge sources for an agent, adding such files to a component collection removes them from the originating agent. As a workaround, you can move such files from the component collection back to their originating agent.

# Configure and manage connections

06/17/2025

Agents use tools and knowledge to perform actions on behalf of users and collect relevant contextual information. These tools and knowledge sources are often stored in external systems, such as Microsoft Dataverse, Microsoft SharePoint, or other APIs. To access these resources securely, agents need to authenticate to establish connections. The agent can use the agent author authentication or the user authentication, depending on the scenario.

In Microsoft Copilot Studio, *connections* are used to securely access authenticated services for tools and knowledge. You can create connections to various data sources, such as Microsoft Dataverse, Microsoft SharePoint, or external APIs.

## View and manage agent connections

In Copilot Studio, you can view and configure the connections used by your agent on the **Connection Settings** page.

On the **Connection settings** page, you can configure and manage the connections used by your agent. You can view existing connections, create new ones, and edit or disconnect them as needed.

### View connections on the Connection Settings page

1. Navigate to your agent page in Copilot Studio by selecting **Agents** and then selecting the agent you want to look at.

The agent overview page opens.
2. Select **Settings** to open the agent **Settings** page.
3. In the left navigation pane, select **Connection Settings**.

A list is displays in a table with all the connections associated with your agent. For each connection, you can view its name, tools and knowledge using the connection, connection status, and the date the connection was last updated.

Manage connections					
		Refresh			
All	Connected	Expired	Not Connected	Stale	
<input type="checkbox"/>	Name	Used By	Status	Manage	
<input type="checkbox"/>	 Planner Microsoft Planner lets you easily bring together teams, tasks, documents, an...	<a href="#">1 tool</a>	<span>✖ Not Connected</span>	<a href="#">Connect</a>	<a href="#">See details</a>
<input type="checkbox"/>	 Microsoft Dataverse Provides access to Microsoft Dataverse actions and triggers for Power Platfor...	<a href="#">1 tool</a>	<span>✔ Connected</span>	<a href="#">Manage</a>	<a href="#">See details</a>
<input type="checkbox"/>	 SharePoint SharePoint helps organizations share and collaborate with colleagues, partne...	<a href="#">1 tool</a>	<span>✔ Connected</span>	<a href="#">Manage</a>	<a href="#">See details</a>

## Filter connections

If you have many connections on your agent, you can filter to get a narrower view and find the connection you want to deal with. You can apply different filters to view connections based on their status, such as **Activated/Deactivated**, **Connected/Not Connected**, **Deactivated**, **Expired**, or **Connection Error**.

## Select or create a connection

- Under **Status** for the connection entry in the list, select **Manage** to open the **Create or pick a connection** pane.
- Open the **Connection** configuration. Pick an existing connection from the list or select **Create new connection** to create a new one.

If you select **Create new connection**, a connection setup pane opens. As needed, you might be prompted to sign in to the service or provide additional information to complete the connection setup.

- When you're done, select **Submit** to save your changes.

The connection is now associated with your agent and your agent can use the connection to access the tools and knowledge.

## Configure connection parameters

Under **Manage** for the connection entry in the list, select **See details** to view more detail about the connection.

**Details** Connection parameters

---

**Connector name**  
Microsoft Dataverse

**Description**  
Provides access to Microsoft Dataverse actions and triggers for Power Platform environments.

**Status**  
Connected

**Owner**

**Created**  
12/06/2025, 11:40:10 AM

**Modified**  
12/06/2025, 11:40:19 AM

Information appears in two tabs:

- **Details:** Displays the connection name, description, status, and creation details.
- **Connection parameters:** Displays the connection parameters, such as authentication details, API keys, or other required information and lets you configure the details.

## Share connection parameters for On-Behalf-Of (OBO) authentication

For some connections that support Single Sign-On (SSO), you can set the connection to use On-Behalf-Of (OBO) authentication. In an OBO flow, the user delegates the agent to use the shared connection parameters with the service on the user's behalf. This is useful when the agent needs to access resources that require user-specific permissions or when actions need to be performed in the context of a user's identity.

### Enable parameter sharing for OBO

Under **Connection parameters** for the connection entry in the list, you can allow the user to authorize sharing specific connection parameters.

1. Under **Connection parameters**, turn on **Allow permission to share parameters**.

**Details** Connection parameters

---

**Allow permission to share parameters**  
Allowing the end-user to use this authentication will provide improved responses. [Learn more](#)

On

Enter information so your agent works properly.

Grant Type	Sharable
code	<input checked="" type="checkbox"/>

2. Select checkboxes for the parameters you want to allow the user to share.

The user is prompted to grant permission for these parameters when the agent uses OBO authentication.

## User experience for OBO authentication

When the agent uses OBO authentication, the user is prompted to grant permission for the agent to use their connection parameters. The user can choose to allow or deny this permission. If the user allows it, the agent can access resources and perform actions on behalf of the user. The agent connects to the service transparently using the shared connection parameters.

# Share agents with other users

Article • 12/20/2024

You can share your agents with others in either of the following ways:

- Grant security groups, or your whole organization, permission to chat with the agent.
- Invite users to collaborate on your agent project. Collaborators always have permission to chat with the agent.

## Prerequisites

- [User authentication](#) for the agent must be configured to **Authenticate manually**, with **Azure Active Directory** or **Microsoft Entra ID** as the provider.
- **Required user sign-in** must be enabled to manage who can chat with the agent in your organization.

## Share an agent for chat

Web app

[Collaborators](#), who have authoring permissions for a shared agent, can always chat with it. However, you can also grant users permission to chat with an agent in Copilot Studio without granting them authoring permissions.

To grant users permission to only chat with the agent, you can either:

- Share your agent with a security group.
- Share your agent with everyone in your organization.

### ① Note

When sharing an agent for *chat* you can't share it with:

- Microsoft 365 groups.
- Individual users directly. To manage individual user access, add or remove users from the security group.
- To author agents in Copilot Studio, makers need at least the **Environment Maker** role. The **Bot Author** role is deprecated. When a maker shares an

agent for co-authoring, the other user is granted the **Bot Contributor** and **Environment Maker** roles. Users in these roles can only access agents they created or that have been shared with them. Additionally, makers must have the **prvAssignRole** privilege, included in the **System Administrator** and **System Customizer** roles, to share an agent for co-authoring. If the new co-author holds the **Environment Maker** role, the original maker doesn't need the **prvAssignRole** privilege.

## Share an agent with security groups

You can share an agent with security groups so their members can chat with it.

1. Open the agent you want to share in Copilot Studio.
2. On the top menu bar, select the More icon (...) and then select **Share**.
3. Enter the name of every security group that you would like to share the agent with.
4. Review the permissions for each security group.
5. If you want to let the users know you shared the agent with them, select **Send an email invitation to new users**.

### Note

Users can only receive an email invitation if their security group has email enabled. Alternatively, select **Copy link** and then share the link directly with the users to inform them they can now chat with your agent.

6. Select **Share** to share the agent with the security groups you specified.

## Share an agent with everyone in the organization

You can share your agent to allow everyone in the same organization as the agent to chat with it.

1. Open the agent you want to share in Copilot Studio.
2. On the top menu bar, select the More icon (...) and then select **Share**.

3. Select **Everyone** in <OrganizationName> (where <OrganizationName> is your organization's name).
4. Select **User - can use the agent**.

**ⓘ Note**

Copilot Studio doesn't send email invitations to everyone in an organization. You can select **Copy link** and then share the link directly with the users to inform them they can now chat with your agent.

5. Select **Share** to share the agent with everyone in the organization.

## Share an agent for collaborative authoring

Web app

Sharing an agent with individual users gives them permission to view, edit, configure, share, and publish the agent. They can't delete the agent.

**ⓘ Note**

You can only share an agent with users who have a Microsoft Copilot Studio per user license. Users who don't have a license can [sign up for a free trial](#).

1. Open the agent you want to share in Copilot Studio.
2. On the top menu bar, select the More icon (...) and then select **Share**.
3. Enter the name or email address of each user that you would like to share the agent with.

**ⓘ Note**

When sharing an agent for *collaborative authoring* you can only share it with individual users in your organization.

4. Review the permissions for each user.

5. If you want to let your new collaborators know you shared the agent with them, select **Send an email invitation to new users**.
6. Select **Share** to share the agent with the users you specified.

 **Important**

If a user wasn't already a [member of the environment](#) for the shared agent, it can take up to 10 minutes before the agent becomes available in Copilot Studio for this user.

## Collaborate on agents

After you shared an agent with other users, they can all edit its topics.

On the **Topics** page, the **Editing** column shows who's working on topics. Select a person's icon to quickly chat with them in Teams or send them an email.

This information can help prevent conflicts when multiple authors are working on the same topic.

 **Note**

The list of authors in the **Editing** column is only refreshed when the page is loaded.

When a topic is open for editing, icons at the top of the authoring canvas also show who's currently working on this topic.

If an author doesn't make any changes to the topic, disconnects their computer, or closes the browser window, they're considered to have abandoned the topic. After 30 minutes of inactivity, the user isn't identified as editing the topic.

Occasionally, multiple authors might make changes to a topic and attempt to save their changes concurrently. For example, you might open and start editing a topic. Your coworker opens the same topic, makes a small change, and saves it. Then, when you've finished editing the topic, and attempt to save it, Copilot Studio detects a conflict. When a conflict happens Copilot Studio prevents you from overwriting your coworker's changes, by offering you two options:

- Select **Discard changes** to reload your agent with the latest changes (discarding your work).

- Select **Save copy** to save a copy of the topic (keeping your changes in a copy of the topic).

If you save your changes as a new topic, you can then review your coworker's changes, merge the two topics, and delete the copy once you're done.

## Stop sharing an agent

Web app

You can stop sharing an agent with individual users, a security group, or everyone in your organization.

### Stop sharing with security groups

1. On the top menu bar, select the More icon (...) and then select **Share**.
2. Select the X icon next to each security group you want to stop sharing the agent with.
3. Select **Share** to stop sharing the agent with these security groups.

### Stop sharing with everyone in the organization

1. On the top menu bar, select the More icon (...) and then select **Share**.
2. Select **Everyone in <OrganizationName>** (where *<OrganizationName>* is your organization's name).
3. Select **None**.
4. Select **Share** to stop sharing the agent with everyone in the organization.

### Stop sharing an agent with individual users

You can stop sharing an agent with a user, and any shared user can stop the agent from being shared with other users, except for the owner. Owners always have access to their agents.

1. On the top menu bar, select the More icon (...) and then select **Share**.
2. Select the X icon next to each user you want to stop sharing the agent with.

3. Select **Share** to stop sharing the agent with these users.

## Share Power Automate flows used in an agent

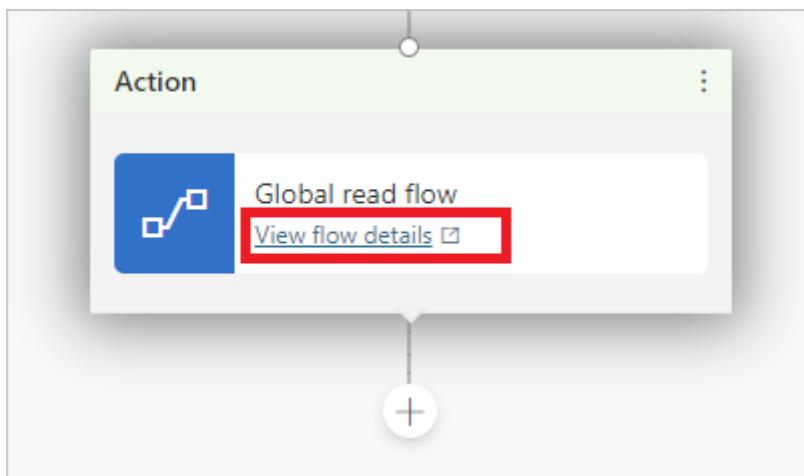
You can [add actions to an agent using flows in Power Automate](#). However, sharing an agent doesn't automatically share the flows in the agent.

Users who don't have access to flows in a shared agent can still run these flows by using the Test panel in Copilot Studio.

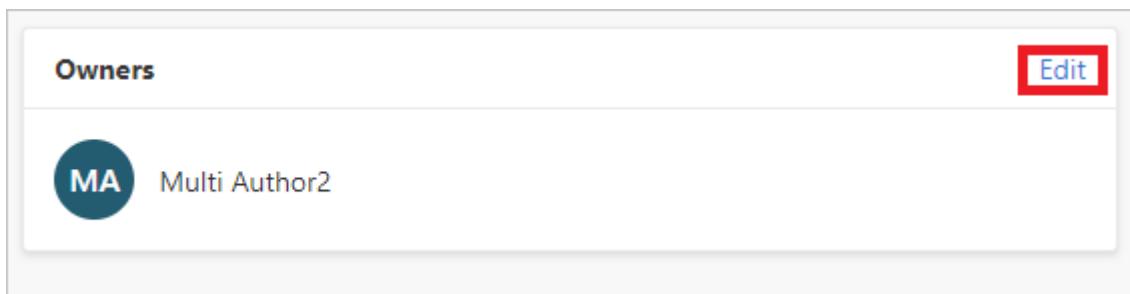
Test your agents to make sure users who chat with them have the required permissions to run the [Power Automate flows](#).

To let other users edit or add flows, you must share them in Power Automate. You can open flows directly from the topic where the flow is used.

1. Select **View flow details** to go to the flow's details page in Power Automate.



2. Select **Edit** in the **Owners** section.



3. Enter the name or email address of the user you want to give editing permissions to.

## Assign environment security roles

If you're a **System Administrator**, you can assign and manage environment security roles when sharing an agent.

The **Environment security roles** section shows when you share an agent and only if you're a **System Administrator**. It lets you share agents with users who don't have sufficient environment permissions to use Copilot Studio.

You must be a **System Administrator** of the environment where the agent is located to view and add security roles.

 **Note**

You can only *assign* security roles when sharing an agent. You can't remove security roles when sharing. For full security role management, use the [Power Platform admin center](#).

Learn more about [security roles](#) and [predefined security roles](#) in the Power Platform admin documentation.

## Assign the Environment Maker security role during agent sharing

When sharing an agent, if a user doesn't have sufficient permissions to use Copilot Studio in the environment, you're notified that the **Environment Maker** security role is assigned to the user so they can use the agent.

## Assign the Transcript Viewer security role during agent sharing

When sharing an agent, you can assign the **Transcript Viewer** security role to users who don't have conversation transcript access.

Depending on the content and target audience of the agent, consider granting transcript access only to users who have the appropriate privacy training.

 **Important**

Conversation transcript access is managed by environment security roles. After assigning the **Transcript Viewer** security role to a user, that user can access conversation transcripts for all agents that they create or are shared with them in the environment.

By default, only admins have the **Transcript Viewer** role. We recommend you [create a new environment for your agents](#) to control which users can view conversation transcripts.

## Insufficient environment permissions

Users in an environment must have the **Environment Maker** security role before an agent can be shared with them.

A system administrator for the environment must assign the **Environment Maker** security role to a user before you share an agent with them. If you have the **System Administrator** security role, you can [assign the Environment Maker role](#) to users when you share agents.

Learn more about [security roles](#) and [predefined security roles](#).

## Manage security roles

You can [manage environment security roles](#) at the Power Platform admin center.

---

## Feedback

Was this page helpful?



[Provide product feedback ↗](#)

# Natural language understanding (NLU) overview

07/09/2025

Natural Language Understanding (NLU) is a branch of artificial intelligence that enables computers to comprehend and interpret human language. This understanding goes beyond simply processing words. Copilot Studio provides different NLU options, based on the needs of your application.

## Generative AI

Generative AI is the default option and is used to respond with the best combination of actions, topics, and knowledge sources. Makers need to provide high-quality descriptions for all aspects of your agent, which then lets generative AI orchestrate the conversation for you.

The generative AI option is best suited for applications where you want to perform a minimal amount of setup, and you're comfortable with allowing generative AI to orchestrate the conversation. There might also be other costs associated with this option.

The generative AI option is configured in the agent's settings (**Generative AI > Orchestration > Yes**). For more information, go to [Orchestrate agent behavior with generative AI](#).

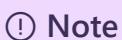
## Classic orchestration

If you'd prefer a more deterministic option for your application, then choose the "classic" Copilot Studio orchestration options in your agent's settings (**Generative AI > Orchestration > No**).

There are three "classic" options: NLU, NLU+, and CLU. All of them provide full, repeatable control over your agent's conversations using a customized dialog.

### NLU

If you want an easier programmable design or have simpler orchestration needs, the original NLU option is useful. With this option, you can quickly add 5 to 20 short phrases per topic and create RegEx or List custom [entities](#). You also don't need to add entity annotations within your training data.



Note

Latency might increase if you add too much training data. For more information, go to [AI features for Teams and Classic chatbots](#).

## NLU+

If you need to achieve high accuracy, then use the [NLU+ option](#). The NLU+ option is ideal for large enterprise-grade applications. These types of applications typically consist of a large number of topics and/or entities, and use a large number of training samples. Also, if you have a [voice-enabled agent](#), your NLU+ training data is also used to optimize your speech recognition capabilities.

### Important

The NLU+ option is available when you manage your voice or chat channels with a Dynamics 365 Contact Center license. For more information, go to [System requirements for Dynamics 365 Contact Center](#).

For the highest accuracy, add entity annotations to your topic trigger phrases. Also add training samples to illustrate how customers might respond to questions regarding specific custom entities.

With the NLU+ option, the model is precompiled, which helps ensure consistent performance, regardless of the volume of training data. To deploy the model in production, it must first be trained. This step enables the system to optimize for your specific use case while maintaining predictable performance at runtime.

## Azure Conversational Language Understanding (CLU)

For makers with an Azure subscription and existing Azure models, you can link your CLU model to your agent and let the model drive conversations. However, this option requires an Azure subscription, management of the model in Azure, and maintenance to keep the model and agent synchronized. For more information, go to [Conversational language understanding integration overview](#).

# Configure NLU+

07/09/2025

NLU+ provides full, repeatable control over your agent's conversations, a customized dialog, and high accuracy for customer's queries. The NLU+ option is ideal for large enterprise-grade applications. These types of applications typically consist of a large number of topics and entities, and use a large number of training samples. Also, if you have a [voice-enabled agent](#), your NLU+ training data is used to optimize your speech recognition capabilities.

NLU+ allows makers to add a large amount of annotated data, which drives users towards higher intent routing and entity extraction accuracy. Also, NLU+ is constructed on a grammar base, which ensures that you trigger an exact match with the training data that you added. This base can also be expanded with entity items and synonyms. This foundation ensures that the model always returns the exact intents and entities you added for annotations.

## Important

- The NLU+ option is available when you manage your voice or chat channels with a Dynamics 365 Contact Center license. For more information, go to [System requirements for Dynamics 365 Contact Center](#).
- When NLU+ is turned on, data is exchanged between Copilot Studio and Dynamics 365 Contact Center. This exchange includes training and runtime data. Each service follows its own specific data policies. For more information regarding these policies, go to [Key concepts - Copilot Studio security and governance](#) and [Privacy and personal data in Microsoft Dynamics 365](#).

## NLU+ best practices

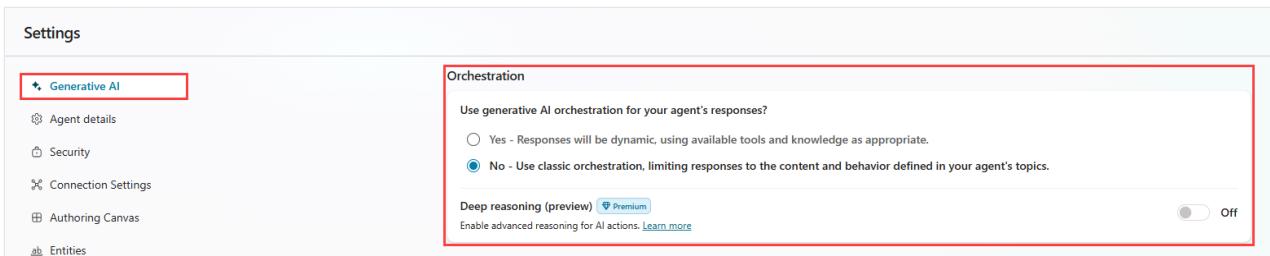
Consider the following guidance before building your NLU+ model and application:

- Use as much real-world training data as possible. Add distinct variations in carrier phrases to help the model learn different ways to trigger intents or entity extractions.
- When you're annotating entities, only one entity variant or synonym is sufficient. Adding more variants doesn't add any extra value.
- The more distinct your intents and entities, your model performance increases. If similar utterances are used within different intents, or as items or synonyms, there's a higher chance of model confusion.
- Don't include determiners or prepositions in entity literals and annotations. Keep determiners and prepositions outside the entity or annotation.

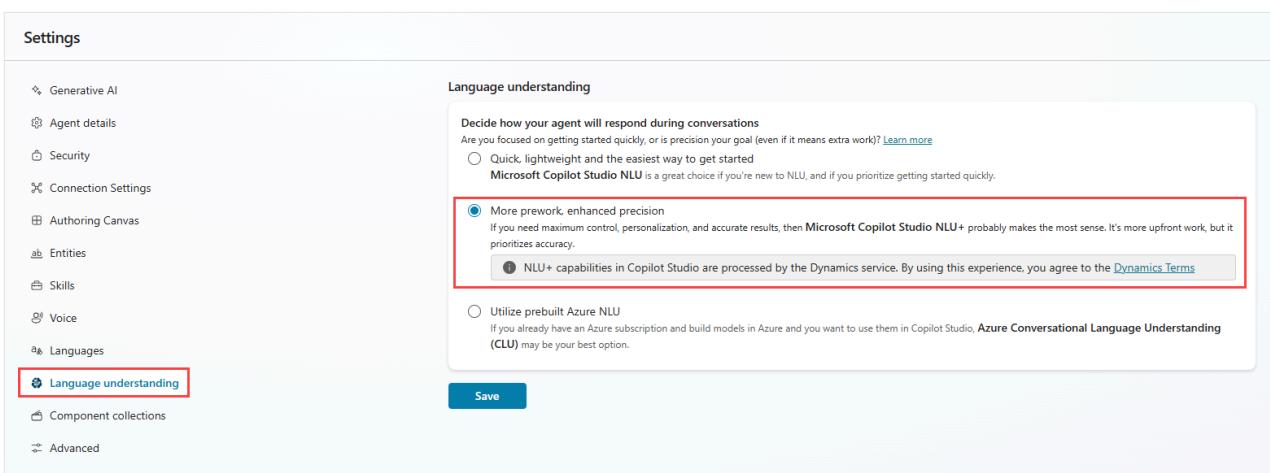
# Setup orchestration and language understanding

To use NLU+, first configure the generative AI orchestration settings, and then select the NLU+ language understanding option.

1. Open your agent and select **Settings**.
2. Select the "classic" Copilot Studio orchestration option in your agent's settings (**Generative AI > Orchestration > No**).



3. Select the NLU+ option in your agent's **Language understanding** settings.



4. Select **Save**.

## Setup topic annotations

To return the maximum value of NLU+, it's important to add entity annotations to the **topic Trigger phrases** for each topic. By adding entity annotations within the samples that trigger a topic, NLU+ can extract the entities as part of the process of triggering a topic.

The entities are annotated using the **variables** linked to the entities. This linkage allows the same entity to be used multiple times within a topic, shared across topics, or to create different copies within different topics.

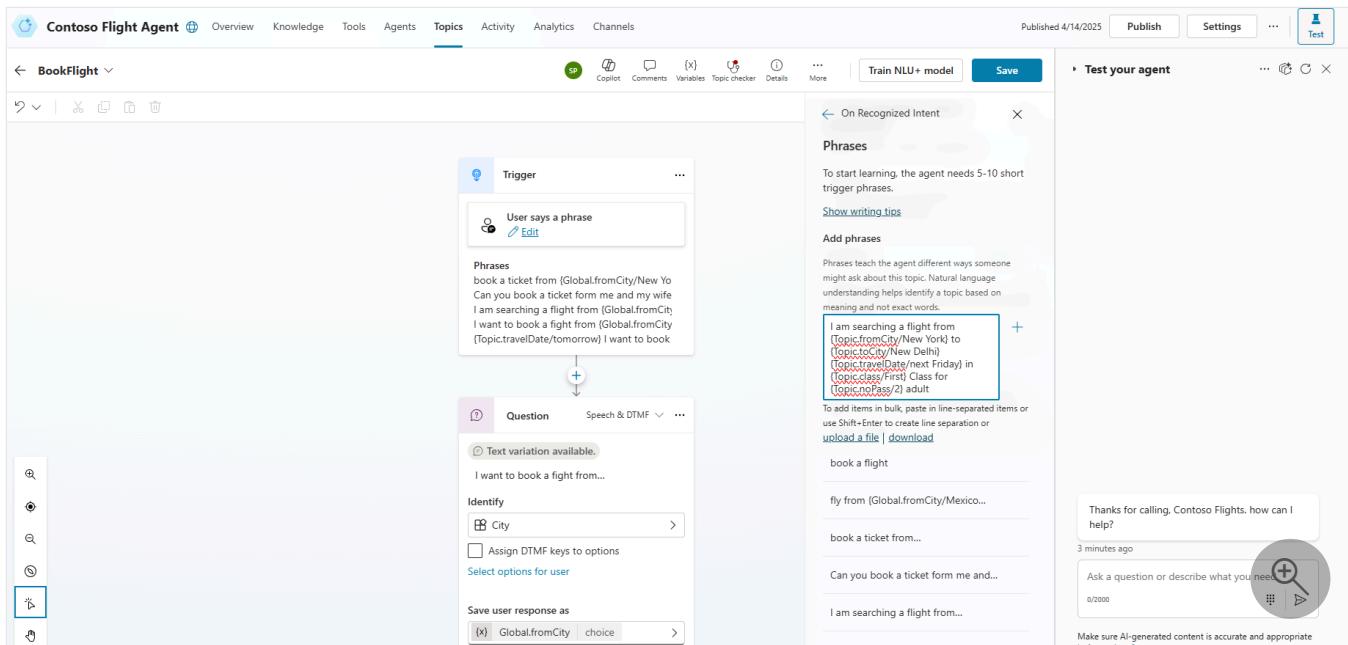
## Entity syntax

If you use entities in your project, they must be constructed using the following syntax:

- `{Topic.Variable_Name/Entity_item_or_synonym}`: This syntax is used for local variables, scoped to a specific topic.
- `{Global.Variable_Name/Entity_item_or_synonym}`: This syntax is used for global variables, used across all topics.

The following example illustrates how entities are formatted:

"book a ticket from `{Topic.fromCity/Boston}` to `{Topic.toCity/NewYork}` for `{Topic.noPass/2}` passengers `{Topic.travelDate/tomorrow}` in `{Topic.class/First}` class"



While entities are helpful, it's also common to have projects that don't use entities. Even if your project uses entities, not every sample requires entity annotation. There are some samples that only trigger a topic and don't extract the entities, even if there are entities tied to that topic. Which is why entity annotations are optional and not required.

### ! Note

Entities can also be extracted, even if entity annotations aren't added. However, adding annotations increases the overall entity extraction accuracy.

## Entity annotations

In addition to annotating entities within a topic's **Trigger** phrases, you can help the model extract entities as part of a **Question node**. Within each custom entity, you can add optional entity annotations. This method is used to annotate how customers respond to specific questions, which are asked to collect that particular entity.

- You can only add a single entity as part of entity annotations. You can't annotate two different entities, or even two instances of an entity within entity annotations. For example, in a *CustomCity* entity, you can't add "Boston to New York" as an annotation.
- Ensure you only add samples that refer to extracting an entity, and not triggering a topic. For example, if you have a flight booking app, you can add "book it for New York." You shouldn't add a sample that triggers a `bookTicket` topic like, "I would like to travel to New York."

## Annotation syntax

The following syntax variations can be used to create the annotation syntax.

- `{Entity value or Literal}`: If you're annotating a single entity, you don't need to specify the entity.
- `{ENTITY_NAME/Entity item or synonym}`: If desired, you can specify the entity name, which is the name of the closed list or RegEx. Providing the entity name makes it easier to read in the YAML, and also matches the syntax used in topics.

The following example illustrates the annotation syntax:

- "book it for `{New York}`"
- "book it for `{City/New York}`"

The screenshot shows the Copilot Studio interface. On the left, the 'Entities' section is selected under 'Settings'. A list of entities is shown, including 'City', 'Class', 'BookingRefNo', 'month', 'Payment', 'phone', 'Age', 'Boolean', 'Color', 'Continent', 'Country or region', 'Date', 'Date and time', 'Date and time without timezone', and 'Duration'. On the right, a detailed view of the 'City' entity is open. The 'Name' field is set to 'City'. The 'Method' dropdown is set to 'List', with a note explaining it tries to match an item on the list based on what the customer says. The 'Modified' field shows '21 days ago'. The 'Smart matching' toggle is turned 'on'. A note explains it helps match misspellings, grammar variations, and words with similar meanings. Below this, there's a link to 'Learn more about entities'. In the top right corner of the entity view, there are tabs for 'List items' and 'Phrases', with 'Phrases' currently selected. The 'Phrases' tab contains a text input field with 'yes (NYC)' and a note about pasting multiple phrases. At the bottom right of the entity view are 'Save' and 'Close' buttons, along with a magnifying glass icon.

## Custom list entities

For NLU+, [list entities](#) are considered to be partially open. This consideration means the model extracts entity literals that aren't explicitly defined in the list, so the model can handle entity data that isn't explicitly defined.

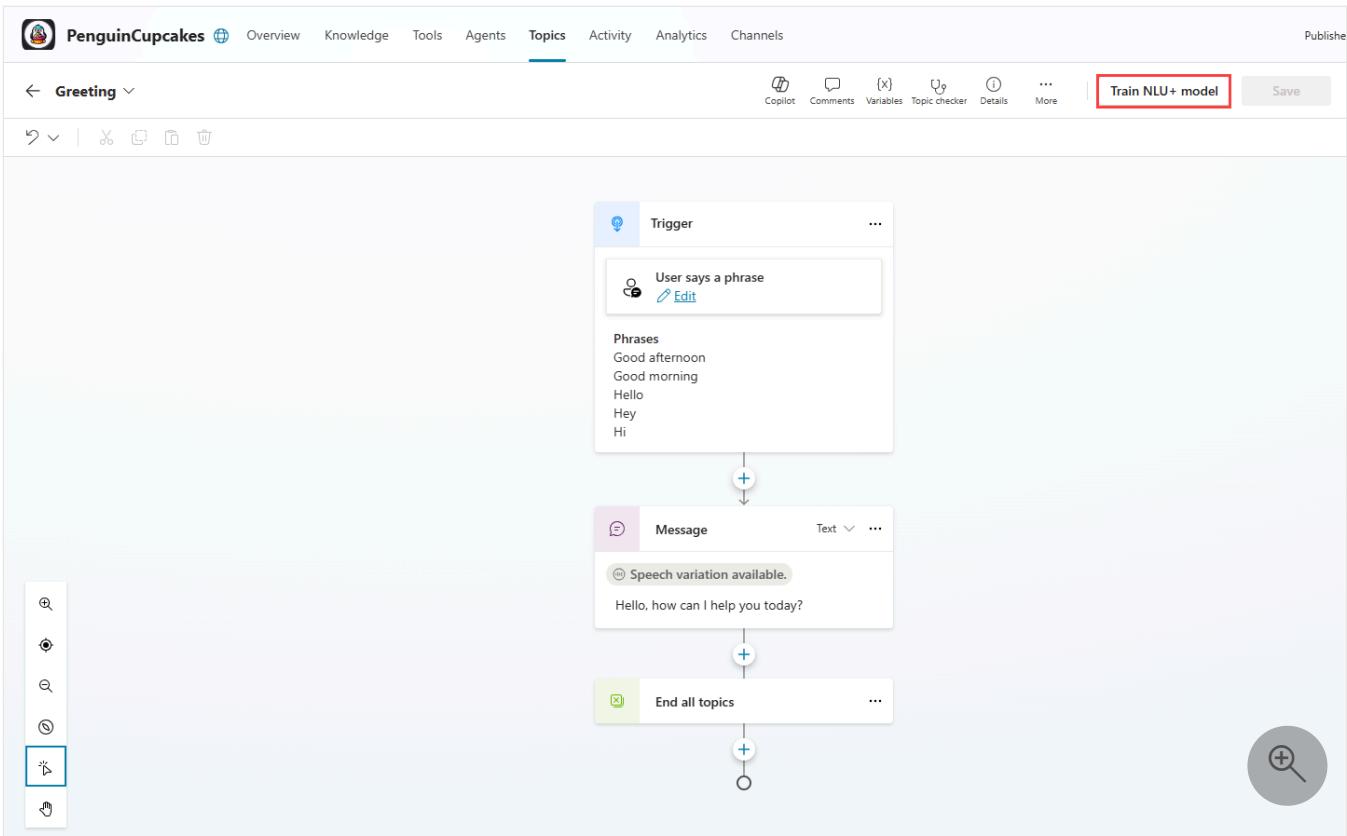
For example, you have a custom list with "Movie titles" that your app handles. If a user requests a title that isn't in your list, the model still marks that title as a "Movie entity." When this happens, the entity value is blank, because the model doesn't know what value to assign the entity.

To influence how *open* an entity is, alter the way you annotate your entity. If you add training data where the entity is annotated with items and synonyms already defined in your entity list, the model considers the entity mostly *closed*. The model still might extract new entity items, but the probability of that happening is low. The more training data you add with the entity annotated with literals not in your entity definition, the more *open* that list becomes. The model is more likely to extract entity literals not in your entity definition.

## Build your NLU+ model

NLU+ requires that the maker explicitly builds their NLU+ model before they can test or publish their agent. This is different from the original NLU option, where changes are automatically incorporated. The NLU+ compiled model has a more predictable latency performance for large models, but requires model training.

After adding your training data and you're satisfied with it, select the **Train NLU+ model** button. The button is available in the **Topics** page or the **Entities** settings page.



### Settings

		<a href="#">+ Add an entity</a>	<a href="#">Train NLU+ model</a>	<input type="text"/> Search entities
<a href="#">Generative AI</a>				
<a href="#">Agent details</a>				
<a href="#">Security</a>				
<a href="#">Connection Settings</a>				
<a href="#">Authoring Canvas</a>				
<a href="#">Entities</a>				
<a href="#">Skills</a>				
<a href="#">Voice</a>				
<a href="#">Languages</a>				
<a href="#">Language understanding</a>				
<a href="#">Component collections</a>				
<a href="#">Advanced</a>				

**Entities**

Name	Description	Method	Errors
eCupcakeFlavour	cupcake flavours	ClosedList	
eSprinklesFlavour	sprinkles flavours	ClosedList	
eTransferCupcakeFrom	from continent	ClosedList	
eTransferCupcakeTo	to continent	ClosedList	
eNLUMainMenu		ClosedList	
Age	Age of a person, place, or thing, extracted as a number	Prebuilt	
Boolean	Positive or negative responses, extracted as a Boolean	Prebuilt	
City	City names, extracted as a string	Prebuilt	
Color	Primary colors and hues on the color spectrum, extracted as a string	Prebuilt	
Continent	Continent names, extracted as a string	Prebuilt	
Country or region	Country and region names, extracted as a string	Prebuilt	
Date	Dates, days of the week, and months relative to a point in time, extracted as a string	Prebuilt	
Date and time	Dates, times, days of the week, and months relative to a point in time, extracted as a string	Prebuilt	
Date and time without timezone	Dates, times, days of the week, and months relative to a point in time, extracted as a string without the timezone	Prebuilt	
Duration	Lengths of time, extracted as a string, in standard Timezone format	Prebuilt	

NLU+ model training times vary, based on the complexity of the model. The **Channels** page displays the model training status. Once training is completed, details about the trained model are displayed, including the user who initiated the training, when the training completed, and the status.

The screenshot shows the 'Channels' tab selected in the navigation bar. Under 'Published agent status', a card for the 'Latest NLU+ model. Trained by' is highlighted with a red border. It shows the training date as '5/26/2025, 2:53 PM - Success'. Below this, it says 'Published by' and the date '5/27/2025, 12:06 AM'. There is a link to 'Demo website'. In the 'Share a preview' section, there are buttons for 'Demo website', 'Teams and Microsoft 365 Copilot', and 'SharePoint'. The 'Other channels' section lists 'Web app', 'Native app', 'Facebook', 'Slack', 'Telegram', 'Twilio', 'Line', 'GroupMe', 'Direct Line Speech', and 'Email'. A magnifying glass icon is also present.

Select the NLU+ model training details in the **Channels** page to open the NLU+ training dialog. This dialog provides details about your model's training, such as information about each of your languages. If you have **Optimize for voice** turned on, you can view details of the ASR training. If the training had any errors or warnings for any region or locale, you can download the individual details file for more information about the specific issues.

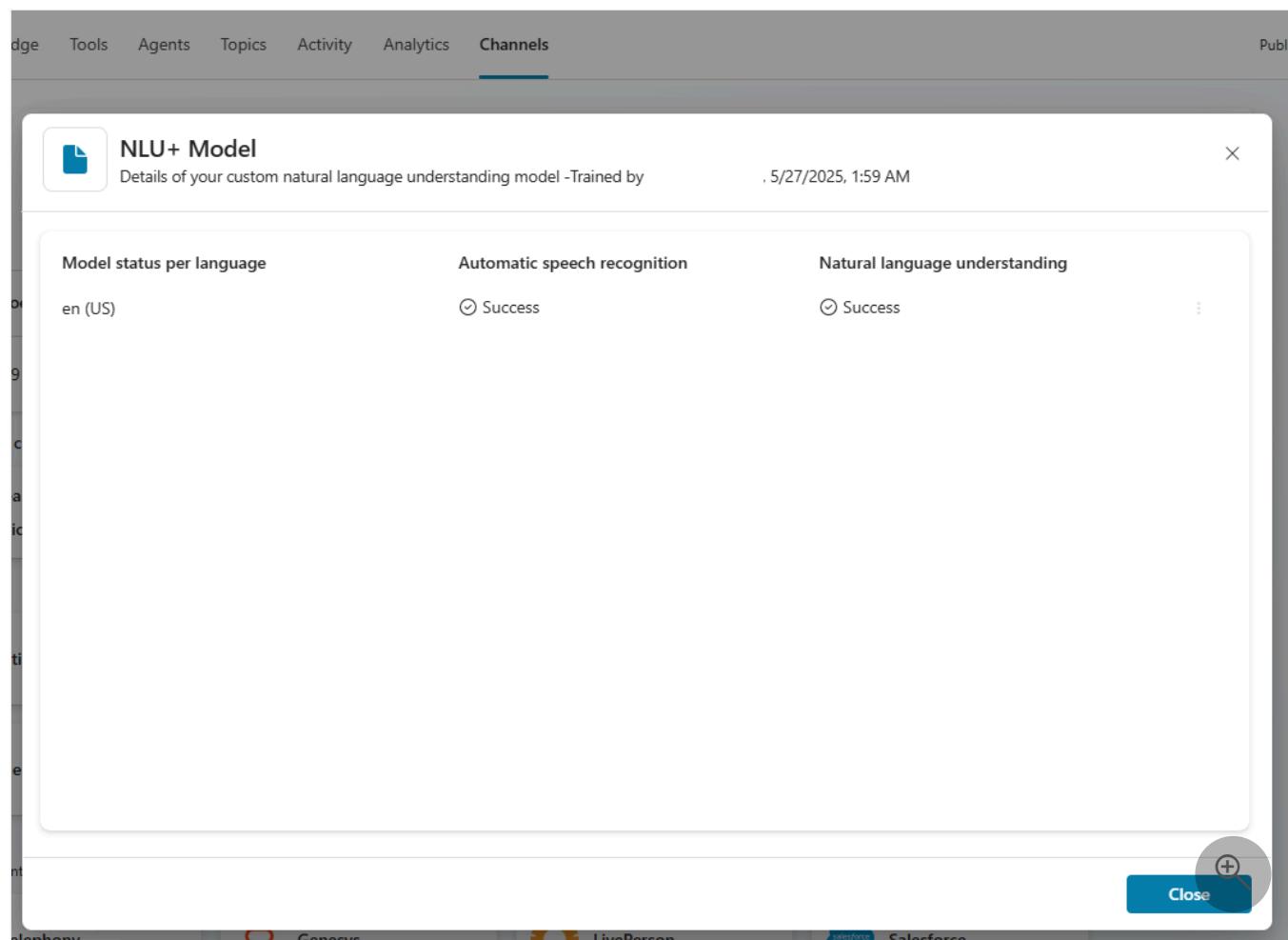
The screenshot shows the 'NLU+ Model' dialog box. At the top, it says 'Details of your custom natural language understanding model - Trained by' and the date '5/26/2025, 2:53 PM'. The dialog is divided into two main sections: 'Model status per language' and 'Natural language understanding'. Under 'Model status per language', it shows 'en (US)' with a status of 'Success'. Under 'Natural language understanding', it shows a status of 'Success'. A 'Close' button is at the bottom right of the dialog.

### (!) Note

- You must wait for training to complete before initiating another model training.
- You can train the models as many times as you want. Copilot Studio only retains the last successfully trained model, and this model is used when testing or publishing your agent.

## Publish your NLU+ agent

When you're ready to publish your agent and its NLU+ model, Copilot Studio uses the last successfully trained model. Select **Publish**, and the **Publish** dialog displays information about the last successfully trained model. This information allows the maker to know which version of the model is being published.



# Conversational language understanding integration overview

10/10/2025

You can integrate a [conversational Language Understanding \(CLU\)](#) model with a Copilot Studio agent. Conversational language understanding is a feature offered by Azure AI Language. It's a cloud-based service that applies machine-learning intelligence. You can use it to build a natural language understanding (NLU) component for a conversational application. Language service APIs are available (but not required for you to integrate with Copilot Studio agents), and [Azure Language Studio](#) provides natural language processing (NLP) features for analyzing conversational text.

For projects created with this capability, you can access intents and entities from the CLU model directly in Copilot Studio. You map CLU intents to trigger topics in the same way as native Copilot Studio [trigger phrases](#). In Copilot Studio, entities imported from a CLU model appear on the [Entities](#) page and you can use them just like the other custom and prebuilt entities. You can add [Question](#) nodes in topics and select entities and intents from the imported CLU model. CLU entities are bound to equivalent Copilot Studio objects. You can specify a name, the data type, or the [JSON structure](#) of a custom data type, if desired, for each entity.

## Note

For an existing agent that you want to take advantage of CLU integration, you must map the CLU model to the Copilot Studio agent, then you can update the agent's trigger phrases to bind each topic to a corresponding CLU intent. You can also manually manage the relationship between the CLU model and your agent.

The concepts discussed in this article help you understand how to integrate CLU models with Copilot Studio agents. For more information, see [Get started with conversational language understanding integration](#).

## CLU connectors

A [connector](#) is a wrapper around an API that allows Azure AI Language to talk to Copilot Studio agents. It provides a way for you to connect your accounts and apply a set of prebuilt actions and triggers to build your apps and workflows. For more information, see [Azure Cognitive Service for Language](#) in the *Copilot Studio, Power Platform, and Azure Logic Apps connectors* documentation.

Power Platform connectors allow Microsoft services to talk to the CLU API. For more information, see the [Copilot Studio, Power Platform, and Azure Logic Apps connectors documentation](#). Although you can import any connector types, Copilot Studio doesn't currently validate connectors.

## Connections in Copilot Studio

Connections are stored authentication credentials for a connector, for example OAuth credentials for the SharePoint connector. A connection reference is a solution component that contains a reference to a connection about a specific connector.

Copilot Studio connections are environment specific. When you import an agent, you must set up a connection for it. Power Platform offers a standard way of achieving this capability by way of a connection reference. For more information, see [Use a connection reference](#), in the Power Apps documentation.

## External recognizers

CLU integration supports specific external recognizers. The `OnRecognize` trigger fires in the following scenarios:

- When a topic is triggered, `LanguageUnderstandingReason.TriggerTopic`
- When a **Question** node doesn't support interruptions and requires an answer, `LanguageUnderstandingReason.AnswerQuestion`
- When a **Question** node supports interruptions and requires an answer, `LanguageUnderstandingReason.AnswerQuestionWithInterruptions`

## External intents

CLU integration supports recognized external intents that utilize the `System.Recognizer.IntentOptions` system variable in the following scenarios:

- Reuse *TopicId* for the external intent ID
- Reuse *TriggerId* for the external intent ID
- Create a new property *IntentId* for the external intent ID
- Use system variables that support the reason for triggering the recognizer

 Expand table

<b>Property name</b>	<b>Type</b>	<b>Description</b>
DisplayName	String	The display name for the recognized intent; localized in the current language and to be shown in the "Did you mean..." prompt (if applicable)
Score	Number	The recognizer score
TopicId	String	The Dataverse schema name of the topic
TriggerId	String	The unique ID of the trigger within the AdaptiveDialog topic

For a complete list of Copilot Studio built-in variables, including system variables, see [Variables overview](#).

## Mix and match entity component types

CLU entities are relevant information extracted from NLU utterances. Entities can be extracted using different methods. They can be learned through context, mixed and matched from a list, or detected from a prebuilt recognized entity. For more information, see [Component types](#), in the Azure AI Language documentation.

 **Note**

You can use Copilot Studio [prebuilt entities](#) along with CLU entities.

## Related content

- [Use system topics](#)
- [Use entities and slot filling](#)
- [Best practices for conversational language understanding](#)
- [Design and control conversation flow](#), in the Bot Framework SDK documentation
- [Create expressions using Power Fx](#)

# Get started with conversational language understanding integration

10/10/2025

This article shows you how to integrate a CLU model to an agent.

## ! Note

Ensure the [environment for your agent](#) is in a [region supported by Copilot Studio](#) with an equivalent [CLU region](#).

Not all regions supported by Copilot Studio have an equivalent CLU region.

## Prerequisites

- A fully [trained CLU model](#), including intents for all system topics and for any custom topics you want to author
- The key and prediction URL for the deployed model
- The CLU project name and deployment name
- An account key for Cognitive Services
- A site URL for Azure AI Language
- A [Copilot Studio account](#)
- A [Copilot Studio agent](#) configured to use classic orchestration
- A Copilot Studio language connector configured to use [maker-provided credentials](#)

## Prepare your CLU project

1. If you don't have an Azure subscription, [create a free account](#).
2. [Create a language resource in Language Studio](#), with the [conversational language understanding](#) feature enabled.

To ensure your Copilot Studio agent functions correctly, the CLU model you create must have intents for all [system topics](#) and for [any custom topics](#) you add to your agent.

3. [Define the entities](#) you want your Copilot Studio agent to use.
4. Train and deploy the CLU project in the same region as your Copilot Studio agent.

# Prepare your agent

1. Go to the **Settings** page for your agent. Settings for the **Generative AI** category appear.
2. If generative orchestration is turned on, switch to classic orchestration: under **Orchestration**, select **No**.
3. Select **Language understanding** in the side pane.
4. Select **Utilize prebuilt Azure NLU**.
5. If your environment *isn't* yet connected to Azure AI Language or if you want to use a new connection, select **Manage connections**, go to Power Apps, and [create a CLU connection](#).
6. Select the desired CLU connection.
7. Select **Save**. A window appears, prompting you to save a snapshot of your agent before deleting all existing trigger phrases for your topics.
8. Select **Save snapshot**, and save the resulting ZIP archive (*botContent.zip*), to the desired location. The ZIP archive contains a single YAML file (*botContent.yml*) with your agent content, including trigger phrases and messages.
9. Select **Yes, delete my trigger phrases**, and then select **Continue**.
10. Enter the name and model deployment information for the appropriate Azure AI Language project, and select **Save**. Once this operation completes, you can see a new system topic named **Analyze Text**. This topic is connected to your CLU model and its purpose is to recognize intents and entities in conversations between customers and your agent.

## ⓘ Note

Changing the language understanding configuration for your agent back to **Microsoft Copilot Studio NLU** removes the **Analyze Text** system topic. It also requires you to manually add example phrases for topics currently mapped to external intents.

# Map CLU intents and entities

In your Copilot Studio agent, start mapping existing topics to CLU intents. You can manually map [intents](#) and [entities](#), or [perform bulk mapping](#).

## Manually map CLU intents to topics

1. Go to the **Topics** page for your agent.
2. Select the desired topic.
3. On the **Trigger** node, select **Edit**. The **Phrases** panel appears.
4. Under **Intent name**, enter the name of the CLU intent that you want to map to this topic.  
The intent name appears on the **Trigger** node under **External intent**.

 **Note**

The CLU intent name must be typed exactly as stored in the CLU model, including matching case.

5. Select **Save**.
6. Repeat these steps for any remaining topic you want to map to an external CLU intent.

## Manually map entities

1. Go to the **Settings** page for your agent.
2. Select **Entities**.
3. Select **Add an entity > Register an external entity**.
4. In the panel that opens, enter the desired name and a description (optional).
5. For **Data Type**, select **From sample data**.
6. Select **Get schema from sample JSON**, enter a JSON code snippet for your CLU entity, and select **Confirm**. Find example JSON snippets at [Entity registration for conversational language understanding integration](#).
7. Select **Save**, and close the panel.
8. Repeat these steps for any remaining entity you want to map to an external CLU entity.

## Perform bulk mapping

1. Go to the **Settings** page for your agent.
2. Select **Language understanding**.

3. Select **Add topics and entities from model data**. The **Add intents and entities** wizard appears.
4. Select **Choose file** to select the file with your CLU model data.

Your model data appears in the **Preview** pane.
5. Select **Next**. The **Map existing topics** screen appears.
6. Select the desired CLU intent for each topic.
7. Review your selections and select **Next**. The **Create new topics** screen appears, showing intents that aren't mapped to an existing topic.
8. If desired, create a new topic for each of these intents: under **Create a new topic**, enter a name for each topic you want to create.

 **Note**

The wizard ignores any fields that you leave blank.

9. Select **Next**. The **Register entities** screen appears, showing the CLU entities from your model data file.
10. Select the appropriate data type for the entities you want to use, then select **Next**. The **Review** screen appears.
11. Review the mappings on the **Existing topics**, **New topics**, and **New entities** tab, and select **Save**.
12. Review the information on the **Success** screen, and select **Done**.

Once you're done with this procedure, you can go to the **Topics** page to review your topics. For more information, see [Manage topics](#).

## Create a CLU connection in Power Apps

1. In Power Apps, if the **Connections** page isn't already in focus, select **Connections** in the left pane.
2. Select **New connection**.
3. Select **Azure Cognitive Service for Language** (use the search field to narrow down the list if needed). A window appears prompting you for authentication information.

4. For the authentication type, select **Api Key**, and enter the required account key.
5. Enter the appropriate root site URL, if any.
6. Select **Create**. The new connection, with the default name "Azure Cognitive Service for Language," appears in the list of connections for your environment.
7. Select the three dots (:) next to this connection and select **Edit**.
8. In the window that appears, replace the default display name with something else, to distinguish this connection from other CLU connections, and select **Update**.

## Related content

- [Conversational language understanding integration overview](#)
- [Entity registration for conversational language understanding integration](#)

# Entity registration for conversational language understanding integration

10/10/2025

This article discusses adding conversational language understanding (CLU) entities to Copilot Studio agents. In most cases, you can use [Copilot Studio prebuilt entities](#) for your projects. To use CLU entities:

- You can map CLU entities of the following data types directly to the corresponding prebuilt entities:
  - BooleanDatatype: `Choice.Boolean`
  - StringDatatype: `Geography.Location`, `Regex`, `List`, `General.Event`, `General.Organization`, `IP Address`, `Person.Name`, `Phone Number`, `URL`
  - NumberDatatype: `Number`

## ⓘ Note

Composite entities (entities with multiple components) map to `StringDatatype`.

- For CLU entities with custom JSON resolutions, you can use sample JSON code to [register these external entities](#) with your agent. These entities resolve to complex data types. You can manually map CLU entities to Copilot Studio data types by copying and pasting JSON code blocks from this article for the relevant entity.

For more information, see [Data types](#) in the Power Fx documentation, and [Supported prebuilt entity components](#) in the Azure AI Language documentation.

## Age

JSON

```
{
 "unit": "Year",
 "value": 10
}
```

## Currency

JSON

```
{
 "unit": "Egyptian pound",
 "ISO4217": "EGP",
 "value": 30
}
```

## Temperature

JSON

```
{
 "unit": "Fahrenheit",
 "value": 88
}
```

## Ordinal

JSON

```
{
 "offset": "3",
 "relativeTo": "Start",
 "value": "3"
}
```

## Dimensions

JSON

```
{
 "unit": "KilometersPerHour",
 "value": 24
}
```

## Datetime entity types

`Datetime` is a special entity type that changes the returned resolution based on user input.

The following examples show how to define entities for different types of date and time utterances. Depending on the type of input you expect from users of your agent, you can create your own mappings, based on these examples.

## Date

Example input: *Jan 1st, 1995*

JSON

```
{
 "dateTimeSubKind": "Date",
 "timex": "1995-01-01",
 "value": "1995-01-01"
}
```

## Datetime (year)

Example input: *I'll be back on April 12th*

JSON

```
{
 "dateTimeSubKind": "Date",
 "timex": "XXXX-04-12",
 "value": "2022-04-12"
}
```

## DatetimeRange (duration)

Example input: *I'm out between 3 and 12 of Sept.*

JSON

```
{
 "resolutionKind": "TemporalSpan",
 "timex": "(XXXX-09-03,XXXX-09-12,P9D)",
 "duration": "P9D",
 "begin": "2022-09-03",
 "end": "2022-09-12"
}
```

## DatetimeRange (set)

Example input: *Every Tuesday*

JSON

```
{
 "resolutionKind": "DateTime",
 "dateTimeSubKind": "Set",
 "timex": "XXXX-WXX-2",
 "value": "not resolved"
}
```

## Datetime (since)

Example input: *I've been out since August*

JSON

```
{
 "resolutionKind": "TemporalSpan",
 "timex": "XXXX-08",
 "begin": "2022-08-01",
 "modifier": "Since"
}
```

## Time

Example input: *It's half past seven o'clock*

JSON

```
{
 "resolutionKind": "DateTime",
 "dateTimeSubKind": "Time",
 "timex": "T07:30",
 "value": "07:30:00"
}
```

# Edit agents in Visual Studio Code (preview)

Article • 05/28/2025

The Copilot Studio extension for Visual Studio Code is designed to enhance the development experience of Copilot Studio agents. It provides language support, IntelliSense code completion and suggestions, and authoring capabilities for Copilot Studio agent components.

After installation, the extension prompts you to sign in to Copilot Studio. It can then show you a list of the agents associated with your environment. Clone an agent to see its editable components, including knowledge sources, actions, topics, and triggers.

## Important

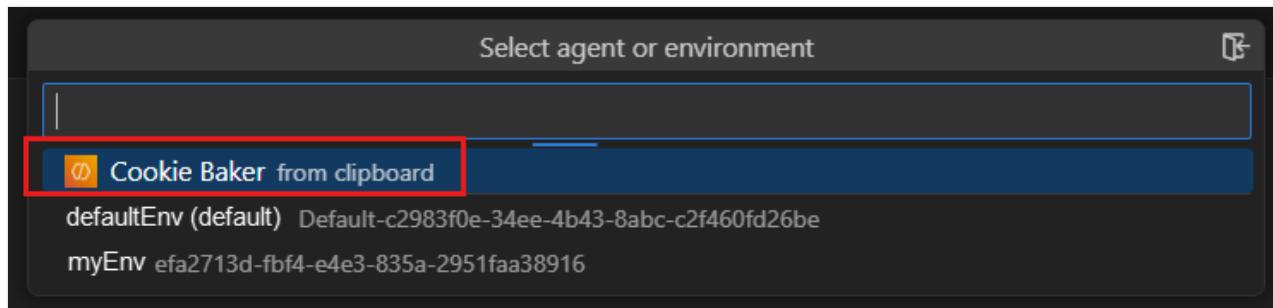
The Copilot Studio extension for Visual Studio Code is a technology preview release, only available for Windows/x64 versions of Visual Studio Code.

## Install the Copilot Studio extension for Visual Studio Code

1. Go to the **Extensions** panel in Visual Studio Code. Alternatively, go to the Visual Studio Marketplace.
2. Search for *Copilot Studio*.
3. Select **Install**.
4. Select the Copilot Studio  icon in the primary side bar of Visual Studio Code. The extension asks for your permission to sign in.
5. Select **Allow**, and sign in with the appropriate credentials for your Copilot Studio environment.

## Clone an agent

1. (Optional) Open the desired agent in Copilot Studio and copy its URL from your browser's address bar.
2. In the **Copilot Studio** panel of Visual Studio Code, select **Clone agent**.
3. Select your agent (marked with *from clipboard* if you already copied the URL); otherwise, select the desired environment and then select the desired agent.



The extension prompts you to select a folder to hold your agent's files (similar to a local repository).

4. Select the desired folder.

## Edit your agent

To edit any component, open the corresponding file and make the desired changes. Since Visual Studio Code natively supports YAML files, the Copilot Studio extension supports IntelliSense code completion and guided tips.

A screenshot of Visual Studio Code showing the "ResetConversation.mcs.yml" file open in the editor. The file content is as follows:

```
1 #· Name: · Reset · Conversation
2 kind: AdaptiveDialog
3 startBehavior: UseLatestPublishedContentAndCa
4 beginDialog:
5 kind: OnSystemRedirect
6 id: main
7 actions:
8 - kind: SendActivity
9 id: sendMessage_OPsT10
10 activity: What can I help you with?
11
12 - kind: ClearAllVariables
13 id: clearAllVariables_73bTFR
14 variables: ConversationScopedVariables
15
16 - kind: CancelAllDialogs
17 id: cancelAllDialogs_12Gt21
```

## Sync your changes

The Copilot Studio extension uses the same source control features as Visual Studio Code. **Fetch changes**, **Pull changes**, and **Push changes** icons are available in both the **Explorer** panel and the **Source Control** panel of Visual Studio Code.

- To preview any remote changes from Copilot Studio, use **Fetch changes**.
- To get all remote changes from Copilot Studio, use **Pull changes**.
- To push your local changes from Visual Studio Code to Copilot Studio, use **Push changes**.

When you push changes, they're saved directly to Copilot Studio. This behavior is different than having a local instance of the agent, which you would then deploy to Copilot Studio. The extension provides a *live editing* experience of a cloud resource.

## Connect to another Copilot Studio account

1. In the **Copilot Studio** panel of Visual Studio Code, select **Clone agent**. The environment selector shows all the environments for the account you're already signed in to.
2. Select the **Switch account**  icon. The extension asks for your permission to sign in.
3. Select **Allow**. The extension shows the accounts you already used.
  - If the desired account appears in the list, select it.
  - Otherwise, select **Sign in to another account**, and sign in with the appropriate credentials.

Now you can [clone the desired agent](#).

# Agent flows overview

Article • 05/19/2025

Agent flows are a powerful way to automate repetitive tasks and integrate your apps and services. Agent flows can be triggered manually, by other automated events or agents, or based on a schedule.

Agent flows consume [Copilot Studio capacity](#) for each agent flow action they execute.

## Benefits of agent flows

Agent flows offer several benefits.

- **Consistent execution:** Agent flows are deterministic—they execute actions or tasks following a rule-based path. The same input always produces the same output, making them reliable and predictable.
- **Simple workflow creation:** You can design, edit, and automate workflows directly in Copilot Studio, using AI-driven suggestions for triggers, actions, and agent flows.
- **End-to-end process visibility:** Design workflows, monitor their performance, and get actionable insights to improve your automation projects, all in the Flows panel in Copilot Studio's unified interface. You can also view details of an agent flow, such as the flow name, description, and status.

## Create agent flows

You can create an agent flow in Copilot Studio in two ways:

- **Natural language:** Create a flow by describing what you need using everyday language in one or more conversations. Copilot Studio interprets your intent and creates a flow based on your description. You can edit the flow in the designer to add more actions. Learn more in [Build an agent flow with natural language](#).
- **Designer:** Build an agent flow by dragging and dropping components on a canvas in the visual designer. You can add actions, conditions, and loops to create sophisticated automation scenarios. You can also use expressions to manipulate data and perform calculations in your flows. Learn more in [Edit and manage your agent flow in the designer](#).

You can also [convert a Power Automate flow to an agent flow](#).

## Triggers and actions in agent flows

An agent flow consists of a trigger and at least one action. A *trigger* is an event that starts a flow. Triggers can be instant (manually run on demand) or based on a schedule, or they can happen in response to other events. An *action* is a task that an agent flow performs. Let's say you want to get a notification in Microsoft Teams when your manager sends you an email. Receiving an email from your manager is the trigger that starts this flow. Sending a message in Microsoft Teams is the action that happens in response.

Agent flows that have the **Run a flow from Copilot** trigger can be actions in other agents, allowing them to be used as part of larger agent-based automation scenarios. Learn more in [Add an agent flow to an agent](#).

## Work with agent flows in solutions

Agent flows are included in solutions, where you have access to advanced capabilities, such as drafts and versioning, export, import, and customization. You can manage solutions in the Power Platform admin center.

## Manage agent flow capacity usage

Because every action your agent flows execute consumes [Copilot Studio capacity](#), it's a good idea to monitor their capacity usage. In the [Power Platform admin center](#) > **Billing** > **Licenses** > **Copilot Studio**, review the **Agent flow actions** used by each flow.

Testing an agent flow in the designer doesn't consume Copilot Studio capacity. However, even in test runs, actions that call features such as AI Builder that have separate billing rates still incur usage of those features.

Here's how to calculate the capacity that's used when a flow is run as an action in an agent:

- When a flow is run from a topic, one **Classic answer** plus the agent flow actions are consumed.
- When a flow is run using [generative orchestration](#), one **Autonomous action** plus the agent flow actions are consumed.
- When a flow is run from the agent's embedded test chat, either from a topic or as a generative action, only the agent flow actions are consumed. The direct messages in the test chat don't count toward capacity consumption.

## Convert a Power Automate flow to an agent flow

If you already have a Power Automate cloud flow that does what you want, you can convert it to an agent flow. Converting it allows you to manage the flow in Copilot Studio and consume

Copilot Studio capacity instead of Power Automate billing. You can still access the flow from Power Automate.

Converting the flow requires that Copilot Studio capacity, either prepaid or pay-as-you-go, is available in the environment. Conversion is a one-way operation. It can't be reversed due to the change in the flow's billing.

1. Open the cloud flow's detail page in the [Power Automate portal](#).

2. Make sure that the flow is in a solution.

If the **Solutions** tile is present on the right side of the detail page, it is. If the **Solutions** tile isn't present, [add the flow to a solution](#) before continuing.

3. Make sure that the flow is in the Power Platform environment that you want to consume Copilot Studio capacity from.

If Copilot Studio capacity isn't allocated to the environment yet, reach out to your Power Platform administrator for assistance.

4. Select **Edit**.

5. Change the flow's plan to **Copilot Studio**.

6. Select **Save**.

7. When prompted, confirm that you want to convert the flow.

## Related content

- [FAQ for Copilot in cloud flows](#)
- [FAQ for Copilot expression assistant](#)
- [FAQ for generative actions in cloud flows](#)
- [FAQ for Power Automate for desktop natural language to code in scripting actions](#)

# Build an agent flow with natural language

Article • 03/31/2025

You can easily create an agent flow by conversing with Copilot using natural language to describe what you want the flow to do. Based on your description, Copilot generates a flow with a trigger and actions that you can further customize in the [Copilot Studio designer](#). Copilot will:

- Understand your intent and create a flow based on your scenario.
- Set up required connections and apply the necessary parameters.
- Respond to your requests to make changes to your flow.
- Answer questions about your flow and Copilot Studio, such as "What does my flow do?" or "How do I access child flows?".
- Suggest a description for the flow when you edit the flow's details.

## ⓘ Important

- Copilot is new technology that's still being developed. It's optimized for use with the English language and has limited support for other languages. Parts of it might appear in English rather than your preferred language.
- Copilot capability is powered by [Azure OpenAI Service](#).

## Create an agent flow

1. In Copilot Studio, in the navigation pane, select **Flows > New agent flow**.
2. Enter a prompt that describes what you want the flow to do in everyday language. For example, "When an email from my manager arrives, post the email subject in Teams." Read [tips for writing a good prompt](#).  
Alternatively, select an example flow from the list. Select **View all examples** for the complete list of flow templates. [Edit, test, and publish the flow in the designer](#).
3. Select **Submit**. Allow a few seconds for Copilot to generate a flow based on your prompt.
4. If you're satisfied with the flow that Copilot generated, select **Keep it and continue**. If you want to make changes, enter more information in **Add more details for**

Copilot to work with, and then select Add more details.

For example, let's say that Copilot generated the following suggestions. Here are examples of the kinds of additional details you can provide:

 Expand table

Copilot-generated flow suggestion	Add this detail
When an email arrives from mbowen@contoso.com, post in Teams.	I want to send the email subject to a Teams channel.
When an item is created in SharePoint, send me a mobile notification.	I don't want to receive a mobile notification. Send me an email instead.

5. Review the apps and services that Copilot connected to your flow. A green checkmark indicates that the connection is successful.

6. Select **Create**.

7. [Edit your flow, test it, and publish it in the designer](#).

If you need to stop working on your flow and come back to it later, select **Save draft**. A banner at the top of the screen confirms that your flow is saved.

## How to write a good prompt

A good prompt includes specific details about your request, including how you want your results to display. Copilot Studio lets you try out variations of prompts to help you evaluate what works best. If the initial results aren't what you're looking for, revise your prompt and run it again. Here are more tips for writing a good prompt:

- Provide your prompt in *When X happens, do Y* format to get better and more accurate results.
- Be as specific as possible. *When an email arrives, I want to post the subject of the email to 'Contoso' Teams General channel* is a more effective prompt than the generic *I want to process an email*.
- Mention the connector if you can, like Outlook, Teams, or Forms.
- Use the same language you would use when talking to a colleague or friend.
- Avoid using technical terms or jargon.

Learn more in [The art of the prompt: How to get the best out of generative AI](#). More examples are available in the prompt library. Visit the [Sample Solution Gallery](#) and in the Products list, select **Copilot Studio**.

# Next step

Edit and manage your agent flow in the designer

## Related content

- [FAQ for Copilot in cloud flows](#)
- [FAQ for generative actions in cloud flows](#)

---

## Feedback

Was this page helpful?

 Yes

 No

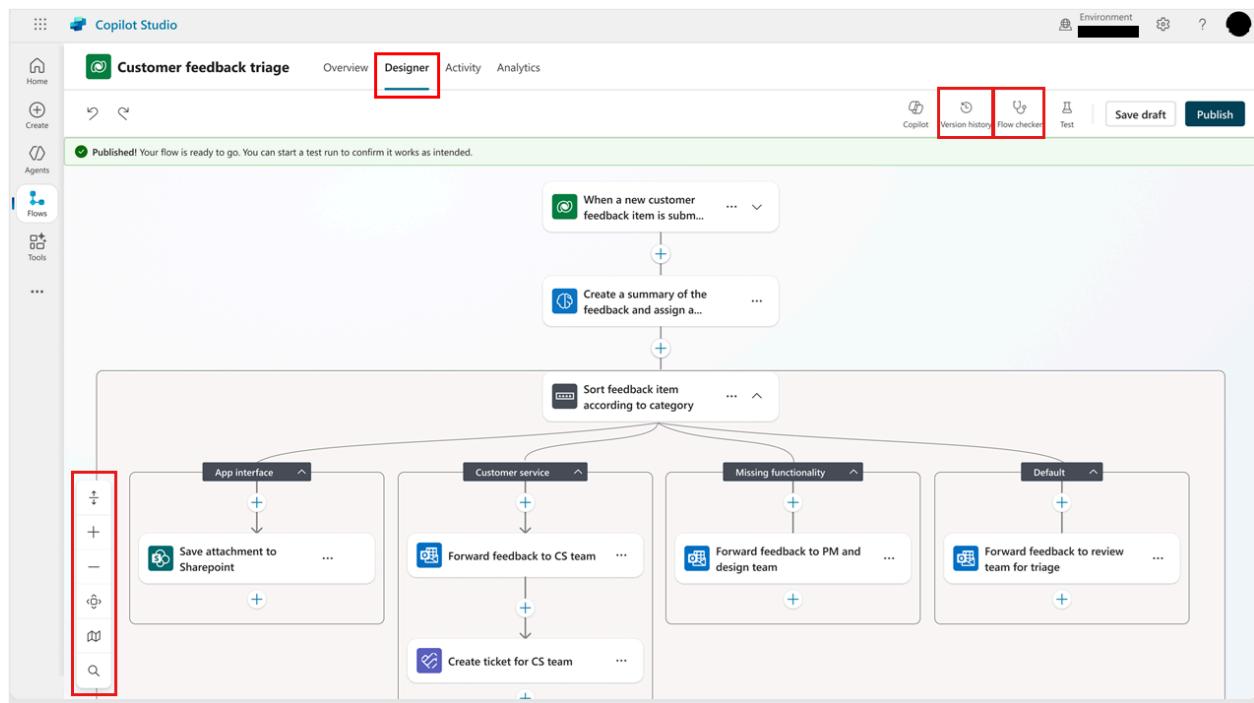
[Provide product feedback ↗](#)

# Edit and manage your agent flow in the designer

Article • 03/31/2025

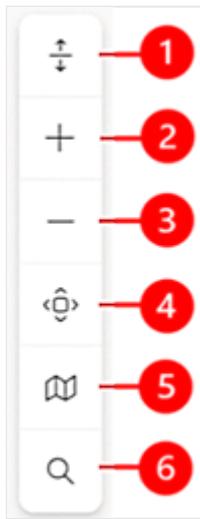
The visual designer in Copilot Studio is a canvas on which you can continue to build or edit an agent flow. Add and remove actions, check for errors, and publish your flow, all without leaving the designer. You can also adjust how your flow is displayed to allow you to focus on specific areas.

To display an agent flow in the designer, open Copilot Studio, select **Flows**, select the flow, and then select the **Designer** tab.



## Change how your flow is displayed

Depending on the size and complexity of your agent flow, you might want to adjust how it's displayed to make it easier to work with. The toolbar at the bottom-left corner of the canvas controls the view.



Legend:

1. **Expand/Collapse**: Expand or collapse all action groups. For example, if an action has multiple conditions, select this icon to show the condition details.
2. **Zoom in**: Increase the size of the flow on the canvas.
3. **Zoom out**: Decrease the size of the flow on the canvas.
4. **Fit view**: Resize the view to fit the entire flow on the canvas.
5. **Minimap**: Navigate to a specific section of a large flow.
6. **Search**: Look for an operation in your flow.

## Add and remove actions

1. Below the card where you want to add the action, select the plus sign (+). The **Add an action** panel opens.
2. Under **Add action**, start typing the action you want your flow to perform, and then select it from the list. Depending on your selection, you might need to provide additional information to complete the action.

If you determine that an action isn't needed, select it in your flow, and then select **More actions (⋮) > Delete**. To confirm the deletion, select **OK**.

## Check flow parameters

To view the parameters for a trigger or action in your agent flow, select its card. The configuration panel opens with the **Parameters** tab shown by default. To enter a value for a parameter, select an option in the dropdown list or enter an expression to set the value dynamically.

# View version history

A version history is recorded in Microsoft Dataverse as you build or modify a flow. To understand how an agent flow evolved, review its version history. Flow versions are grouped by date, with indicators for latest version, published, and past published.

A flow's history is based on saved versions, so it's a good idea to save drafts before the flow is complete.

To use the version history panel, you must save your agent flow at least once.

1. On the menu at the top of the canvas, select **Version history**.
2. To view the recorded versions of the flow, expand an item in the list.
3. To display the version in the designer, select it in the list.

## Check for errors

Errors in agent flow actions are indicated in red. To view details of an error in one action, select the error. To list all errors in the entire flow, on the menu at the top of the canvas, select **Flow checker**. To view the details of individual errors, select the error. Alternatively, select **Open operation** to open the operation in the Flow Checker panel and correct the error.

 **Important**

You can't publish a flow if it contains errors. You must correct *all* the errors before you can publish your flow.

## Publish your agent flow

If your agent flow has no errors, you can publish it. To publish your flow, on the menu at the top of the canvas, select **Publish**.

## Test your agent flow

Just because your flow has no errors and can be published, it doesn't mean it's going to do what you want it to. After you publish it, test your flow to make sure that it works as expected.

You can run a test manually or automatically. After the test runs, you can check the output of each action to confirm that the flow is working.

1. Save and publish your flow.
2. On the menu at the top of the canvas, select **Test**. The **Test Flow** panel opens.
3. Select whether to run the test **Manually** or **Automatically**.
4. In the **Run Flow** panel, select **Test > Run flow**. A green checkmark indicates that the flow ran successfully.
5. Select **Done**.

Testing an agent flow in the designer doesn't consume Copilot Studio capacity for the agent flow actions that are directly executed by the test run. However, actions that call features such as AI Builder that have separate billing rates do incur usage of those features.

## Related content

- [Parameters in the action configuration pane](#)
- [Add multiple actions and advanced options to a cloud flow](#)
- [Drafts and versioning for cloud flows](#)
- [Perform bound actions or unbound actions](#)
- [Find and fix errors with Flow Checker](#)
- [Troubleshoot common issues with triggers](#)
- [FAQ for Copilot in cloud flows](#)
- [FAQ for Copilot expression assistant](#)
- [FAQ for generative actions in cloud flows](#)

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Savings on agent flows (preview)

09/30/2025

[This article is prerelease documentation and is subject to change.]

The flow savings feature lets you calculate how much time and/or money an agent flow is saving your organization. You can add a user-defined saving rule to each flow, which keeps track of savings on successful flow runs.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

The following table describes key concepts related to savings in agent flows.

 Expand table

Concept	Definition
Savings rule	Maker-defined baselines defining how to calculate money and time savings for an agent run, compared to when the task is not automated.
Savings	The amount of time or money saved, generated by the saving rule for the agent flow.

## Note

- Only solution-based agent flows can use savings rules.
- Only successful runs generate savings. Successful *test* runs don't generate savings.
- Environments with Dataverse for Teams can't use the savings feature.

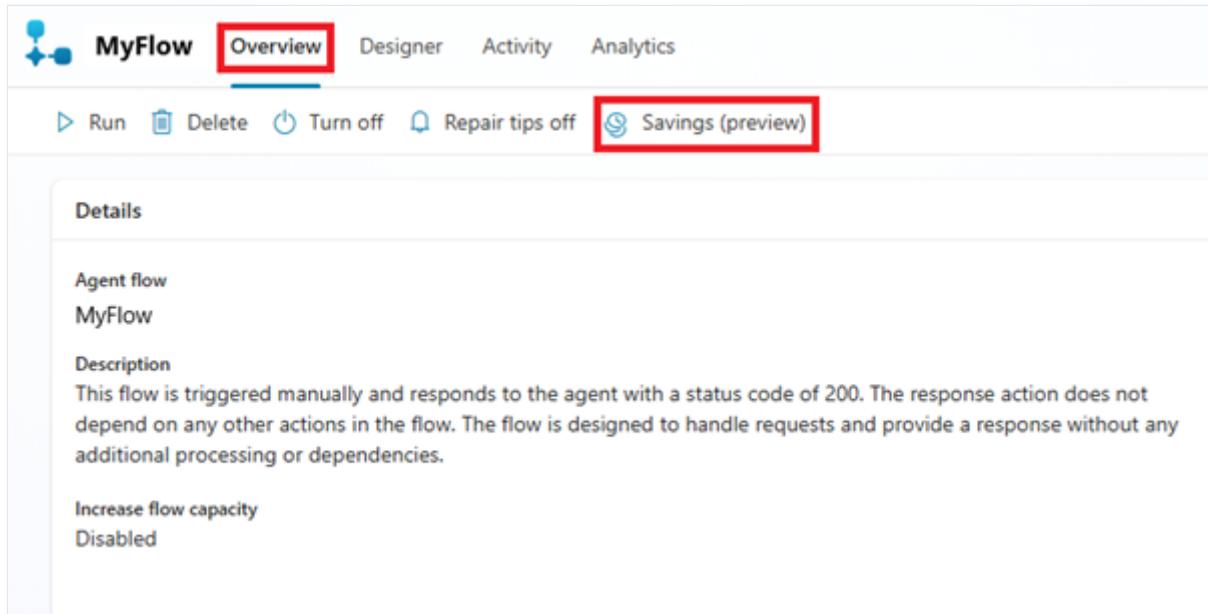
## Set up a savings rule

To set up a savings rule for your agent flow:

1. Go to your agent flow **Overview** page by selecting **Flows** in the left pane, then selecting your flow from the list.

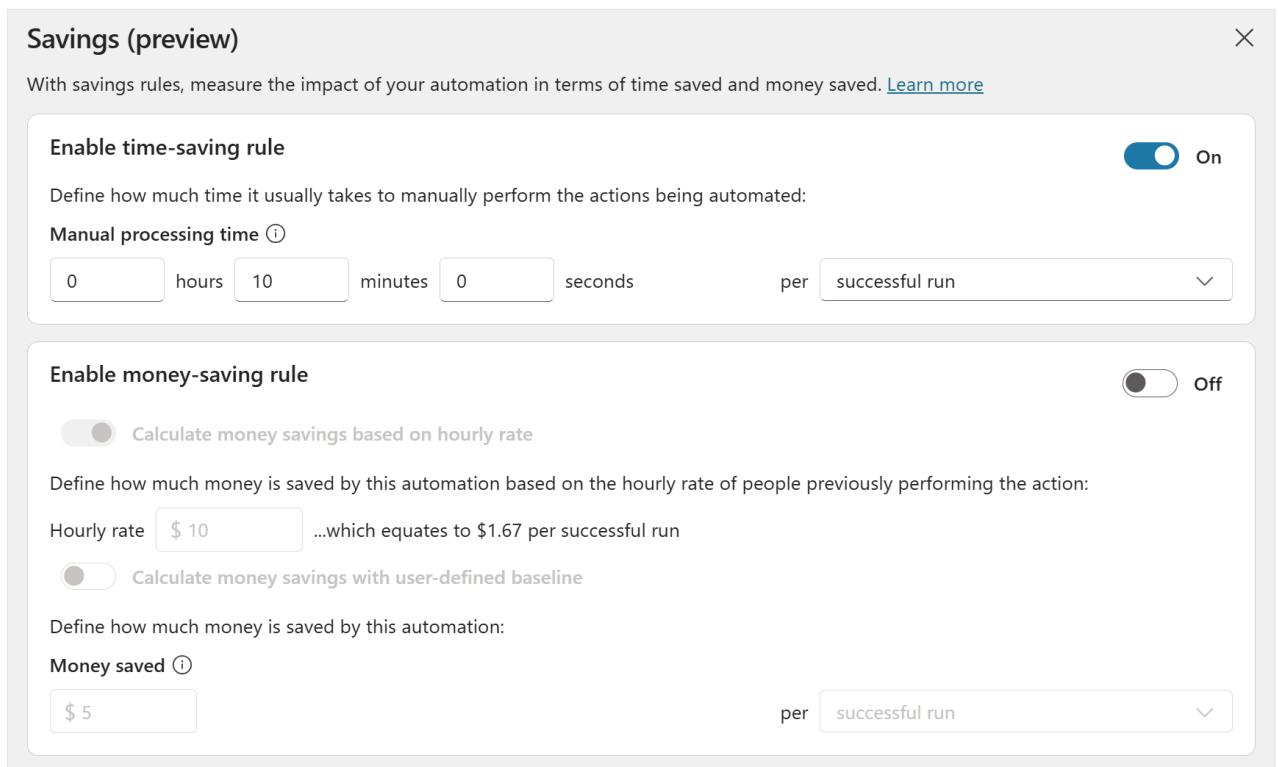
The agent flow **Overview** page is displayed.

2. Select **Savings (preview)** on the **Overview** page action bar.



The screenshot shows the 'MyFlow' agent flow in the 'Overview' tab. The 'Savings (preview)' button in the top navigation bar is highlighted with a red box. Below the navigation bar, there are several buttons: Run, Delete, Turn off, Repair tips off, and Savings (preview), also with a red box around it. The main content area displays details about the agent flow, including its name 'MyFlow', a description of its function, and a section for increasing flow capacity which is currently disabled.

This opens the **Savings** dialog.



The 'Savings (preview)' dialog is open. It starts with a message about measuring automation impact. The 'Enable time-saving rule' section is active, with a toggle switch set to 'On'. It asks for manual processing time, with fields for hours (0), minutes (10), and seconds (0), and a dropdown for 'per successful run'. The 'Enable money-saving rule' section is inactive, with a toggle switch set to 'Off'. It includes options to calculate savings based on hourly rate or user-defined baseline, and a field for 'Money saved' per successful run (\$5).

3. On the **Savings** dialog, edit time-saving and/or money-saving rules for the agent flow. When you are done, select **Save**.

## Savings rules types

The following table describes details about the different types of savings rules.

 Expand table

Saving rule	Definition
Time-saving rule	Define the manual processing time it usually takes to manually perform the actions being automated.
Money-saving rule based on hourly rate	The money-saving rule can be based on an hourly rate converting time-savings into money-savings. This mode is only available if a time-saving rule is enabled.
Money-saving rule with user-defined baseline	Define the amount of money saved by the actions being automated.

## Money-saving rule based on hourly rate

The money savings generated are rounded to the nearest whole number and don't include decimals.

For example:

- You define a time-saving baseline of 1 hour and 15 minutes per successful run.
- You define a money-saving rule with an hourly rate of \$1/hour.
- The money-saving associated with the agent flow run should be \$1.25 for a successful run (1.25 hours x \$1 per hour). The savings are rounded to \$1 however.

## When savings are generated

Once you define a savings rule on an agent flow, every successful run of this flow contributes to generate savings. However, the savings don't immediately generate after the successful run. The savings amount for the agent is updated periodically.

### Note

It can take up to one hour for a run to be accounted for in the savings amount of its agent flow.

## Track the savings generated by your agent flow

You can view the savings generated by an agent flow on the flow details page.

Time saved (last 7 days)

**20 hours**

Last generation: 23/09/2025, 14:29:28 ⓘ

Money saved (last 7 days)

**\$1000**

Runs which happened after the last generation time are not yet accounted for in terms of savings.

## Permissions required to view and edit saving-rules

To view and edit a savings rule, you need a security role with privileges to the **Saving Rule** and the **Flow Aggregation** tables. For example, people with the Environment Maker role can view and edit savings rules on agent flows.

## Frequently asked questions

### Why is the hourly rate mode disabled in the money-rule section?

A money-saving rule can only use the hourly rate mode if a time-saving rule is also enabled.

### Why is the money rule entirely disabled?

Your environment administrator blocked the usage of money rule in the environment. Money savings generated before the administrator decision remain unchanged, but no new money savings are generated.

# Monitor your agent flows

Article • 03/31/2025

The **Designer tab** in Copilot Studio is the canvas on which you build agent flows. Other tabs are available to help you manage and monitor your flows:

- Edit the details of an agent flow on the **Overview** tab.
- Monitor its activity and check for any failures on the **Activity** tab.
- Analyze its performance over time on the **Analytics** tab.

## Overview

In the **Overview** tab, view and edit your flow's name, description, and connections; view the owner, any co-owners, and when it was created and last modified; and view the four most recent runs. You can also delete the flow or turn it off to prevent it from triggering.

The screenshot shows the 'Overview' tab for an agent flow named 'When someone responds to a Microsoft Forms su...'. The 'Details' section includes the agent flow description, status (Off), creation date (Feb 26, 09:45 AM), modification date (Mar 13, 10:51 AM), type (Automated), and plan (Copilot Studio). The 'Connection References' section lists three connections: Microsoft Forms Cr57, Office 365 Users Cr57, and Office 365 Outlook Cr. The 'Co-owners' section shows 'Evan Smith' as a co-owner. The '28-day run history' section displays four recent runs, all of which failed. The first run started on Feb 28, 02:05 PM (2 wk ago) and took 649 ms. The second run started on Feb 28, 02:04 PM (2 wk ago) and took 222 ms. The third run started on Feb 28, 02:04 PM (2 wk ago) and took 361 ms. The fourth run started on Feb 28, 01:59 PM (2 wk ago) and took 432 ms.

Start	Duration	Status
Feb 28, 02:05 PM (2 wk ago)	649 ms	Failed
Feb 28, 02:04 PM (2 wk ago)	222 ms	Failed
Feb 28, 02:04 PM (2 wk ago)	361 ms	Failed
Feb 28, 01:59 PM (2 wk ago)	432 ms	Failed

## Activity

In the **Activity** tab, view the history of an agent flow's runs, their status, and how long each run took. Select a run to step through the activity view and troubleshoot as needed.

You can select multiple runs and either resubmit them or cancel runs that are in progress.

# Analytics

In the **Analytics** tab, view a summary of your flow's activity, including the number of times it ran, failure rate, and performance trends over time.

## Known issues and limitations of the Analytics tab

- Flow runs might take up to 30 minutes to appear in run-related visualizations.
- Metrics related to run analysis are available only for solution flows.
- The number of flow runs might vary from the numbers shown in the **Activity** tab due to differences in data sources, availability, and refresh rates.
- Data shown in the *Actions performed trend* visualization might differ from the *Flow runs trend* graph due to differences in data sources, availability, and refresh rates. The refresh rate for this data is approximately 24 hours.
- Flow runs that are associated with co-owned or shared flows are unsupported. They aren't included in the total *Runs* count, *Run error rate*, and *Flow runs trend* visualizations.

---

## Feedback

Was this page helpful?

 Yes

 No

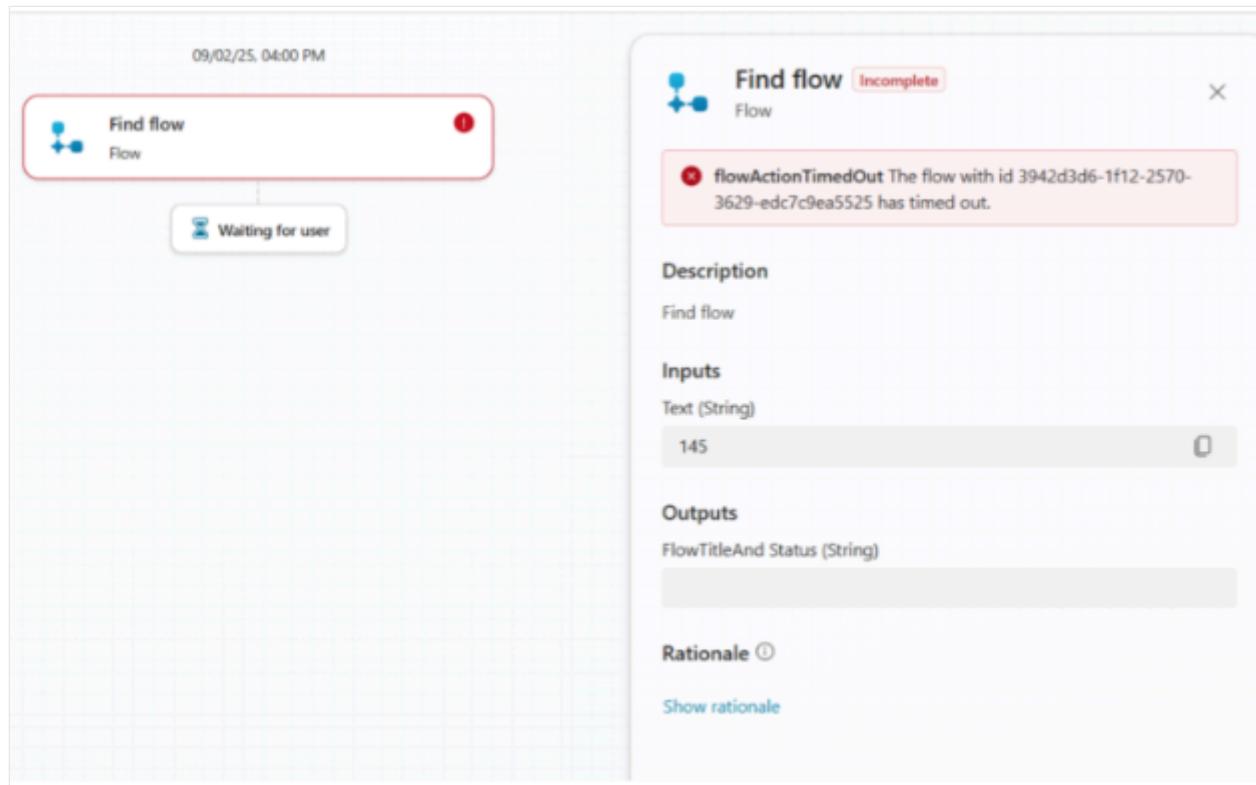
[Provide product feedback ↗](#)

# Speed up agent flow execution with express mode (preview)

10/20/2025

[This article is prerelease documentation and is subject to change.]

By default, an agent flow initiated by an agent or app fails if it takes longer than two minutes to respond to the calling agent or app. This failure affects the functionality and usability of the agent or app that calls the flow.



## ⓘ Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

With express mode, a flow can achieve faster execution times. This feature increases the likelihood that agent flows can complete within the two minute window, and provide a response in time.

The following image shows an example of the speedup for a specific flow with express mode and without it.

28-day run history		
Start	Duration	Status
Sep 2, 03:56 PM (1 min ago)	00:01:23	Succeeded
Sep 2, 03:53 PM (4 min ago)	00:00:12	Succeeded

### !Note

The speedup varies based on the complexity of the flow and the actions used within it. Express mode works best for flows that do not move a lot of data from connectors, but take a lot of processing time. We recommend testing your flows to see if express mode is suitable for your specific scenario and how much it improves flow performance. If you encounter a flow runtime error mentioning a data limitation, we suggest that you turn off express mode, and try it with other flows instead.

## Prerequisites

To use express mode, your flow must meet the following requirements:

- The flow must have either the **When an agent calls a flow** or the **When an app calls a flow** trigger.
- The flow must have a Response action, either **Respond to agent** or **Respond to app**.
- The flow must have a Copilot Studio plan associated with it.
- The Power Automate environment must be upgraded to support express mode.

## Turn on express mode

1. Select **Flows** and then select the flow for which you want to turn on express mode.
2. In the **Details** section of the **Overview** page, select **Edit**.  
The **Details** panel appears.
3. Turn on **Express mode**. The toggle is at the bottom of the panel.

### Description

This flow starts when a manual trigger is activated. It retrieves a list of items from a SharePoint site and then checks each item to see if its ID matches a specific value. If a match is found, it combines the item's title and owner into a single string. After processing all items, the flow sends a

 Use AI to create a description for you (this is a preview feature, so review the suggestion before saving). [See terms](#)



### Primary owner \*



### Plan \*

Copilot Studio 

[Learn more](#) to understand which plan is best for your flow.

### Increase flow capacity



### Express mode



#### Note

You can also turn on express mode from the details page for your flow in the Power Automate portal.

## Check availability for your environment

Express mode is currently only available for some environments that are upgraded to a new infrastructure that supports it. Microsoft is in the process of upgrading environments to support express mode. No action is required on your part to upgrade your environment.

To check if express mode is available in your environment:

1. Make sure you have a simple agent flow in Copilot Studio with either the **When an agent calls a flow** or **When an app calls a flow** trigger, and some action. If you don't have a

flow with these elements, create one.

2. Publish the flow.

3. Check for the express mode toggle on the **Details** section of the **Overview** page for your flow.

If express mode doesn't appear in the **Details** section, your environment doesn't support this feature yet.

## Limitations

Here are some known limitations to be aware of when using express mode.

### Limitations at flow publish

*Delay* and *webhook* actions in the flow aren't supported when express mode is turned on.

Errors appear when saving the flow. For such cases, try turning off express mode and try again.

### Limitations at flow runtime

There are some runtime limitations when using express mode.

#### Execution time

With express mode turned on, flows must complete in two minutes. Flows that take longer time out.

#### Flow size

- You shouldn't include more than 100 actions in a flow run. The actions count includes loops, which count as one action for each loop iteration. The following limits apply for loops:
  - **Apply to each** loop on an array: max 100 items
  - **Do until** loop: max 100 iterations
- Variable content limit: 1,024 characters

100 actions is a safe limit for flow size. You may be able to get in more than 100 actions, depending on how heavy the flow is, and on how much data processing the flow does. See [Guidelines for using express mode](#) for more information on managing flow size.

## Message size

The message size limit for data passed in an individual action (connector response) is 64 KB.

## Error messaging

We show error messaging at runtime detailing why the flow failed if it's due to any of the previously mentioned runtime limitations. If you have a flow that doesn't meet the prerequisites for express mode, we recommend running the flow with express mode **Off**.

## Other limitations when using express mode

- [Testing your flow automatically](#) doesn't work in the flow designer. Testing automatically uses flow resubmit, and flow resubmit can't be used with agent-triggered or app-triggered flows.
- Loop iterations don't appear in the [Run details](#) view when a connector response exceeds the memory limit. In some cases, you might notice that loop iterations don't show up when you monitor a loop ([Apply to each](#) or [Do until](#)). Test your flow in a development environment, and then turn on express mode for debugging before you publish to production.

## Guidelines for using express mode

 Expand table

When to use express mode	When not to use express mode
<p><b>Logic heavy flows:</b> Your agent flow that doesn't return large amounts data from connectors response (Large data amounts exceed the memory limit of express mode).</p>	<p><b>Data heavy flows:</b> Your agent flow which is moving a lot of data (for example, list the rows in a table, or a get items action returns 1500 rows of data with 100 columns).</p>
<p>Your agent flow has a response action, meaning the flow is time sensitive. An agent waits for the flow to execute and send back a response as quickly as possible.</p>	<p>Your agent flow doesn't have a response action. The flow is a fire-and-forget type. It doesn't need to send back a response to a calling agent or app.</p>
<p>Express mode runs successfully and optimally during test.</p>	<p>Express mode fails during test due to a memory limit or actions limit. If the flow fails during a test, you will likely hit these issues when the flow is deployed and run in production.</p>

# Add an agent flow to an agent as a tool

06/10/2025

Although agent flows can be used effectively on their own, they can also be added to agents as tools to create larger agent-based automation scenarios. To add an agent flow to an agent as a tool, it must satisfy the following criteria:

- Have the **When an agent calls the flow** trigger and a **Respond to the agent** action.
- Be configured to respond in real time, not asynchronously. The **Asynchronous response** toggle must be set to **Off** under **Networking** in the **Respond to the agent** action settings.
- Respond to the agent within the 100-second action limit. Optimize the flow logic, queries, and the amount of data returned so that a typical run is below this 100-second limit. Actions in the flow that need to run longer can be placed after the **Respond to Copilot** action to continue to run up to the flow run duration limit of 30 days.

For simplicity, this article assumes you already have an agent flow configured to work with an agent. If you don't, see [Create an agent flow](#) for more details.

You can add an existing agent flow to an agent at either the agent level or the topic level.

## Add an agent flow to an agent as an agent-level tool

When you add an agent flow to an agent, the agent orchestrator can trigger it directly at runtime.

To add an agent flow to an agent as an agent-level tool:

1. In Copilot Studio, select **Agents**, and then select the agent that you want to add an agent flow to.
2. Go to the **Tools** page and select **Add a tool**. The **Add a tool** panel appears.
3. To list the agent flows that are available as tools, select **Flow**.

If the flow you want to add isn't listed, make sure that it has the **Run a flow from Copilot** trigger and a **Respond to Copilot** action. If it doesn't, it can't be added as a tool to an agent.

4. Select the agent flow.
5. Modify the flow configuration as needed. For example, you can update the description to help your agent understand the flow's purpose.

## 6. Select **Add action**.

The agent flow should appear in the agent's list of tools.

# Add an agent flow to a topic as a topic-level tool

When you add an existing agent flow to a topic, it only becomes available to that topic.

To add an agent flow to a topic as a topic-level tool:

1. In Copilot Studio, select **Agents**, and then select the agent that you want to add an agent flow to.
2. Go to the **Topics** page and select the topic that you want to add an agent flow to.
3. Select the **Add node** icon  below any node, and select **Add a tool**.
4. Select the agent flow you want to add. A new **Action** node appears in the topic.

## Related content

- [Use agent flows with your agent](#)

# Multistage and AI approvals in agent flows (preview)

09/04/2025

[This article is prerelease documentation and is subject to change.]

This article describes how to use *multistage and AI approvals* in agent flows.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

## What are multistage approvals?

A multistage approval is a capability that caters to more complex approval processes. It offers a rich set of capabilities that allow organizations to build robust approval workflows for their business processes, now enhanced with AI-powered decision-making.

Multistage approvals combine human and AI reviews to make approval decisions based on given instructions, documents, and images. This combination enables automated processing of routine requests while maintaining human oversight for complex decisions, significantly accelerating approval cycles while ensuring compliance with organizational policies.

Multistage approvals build upon standard approvals found in cloud flows. Before getting into multistage approvals, you should familiarize yourself with *standard approvals* first. Here are some resources to get you started:

- [Get started with approvals](#)
- [Microsoft Power Platform and Azure Logic Apps approvals connector reference](#)

## Access multistage approvals

Multistage approvals are available exclusively in *agent flows*. Agent flows are automations or flows built in Copilot Studio and used standalone or with Copilot Studio agents. Learn more about agent flows in [agent flows overview](#).

## Key functionalities of multistage approvals

Multistage approvals offer the following key functionalities that set them apart from standard approvals:

- 1. Manual approval stages:** Request approval decisions from human stakeholders at various stages of the process.
- 2. AI approval stages:** Implement automated decision-making at key points in your approval workflow using AI models.
- 3. Conditions/conditional approvals:** Place conditions between stages to dynamically control the approval workflow based on specific criteria.

All of these capabilities are in preview.

### Manual approval stages

Manual stages in your multistage approval let you request approval decisions from different *human* stakeholders at various stages of the process. In these stages, you can define the approval type to be either "First to respond" or "Everyone must approve," the title of the approval request, who the approval should be assigned to, and the relevant details with which the human approver needs to make a decision.

### AI approval stages

AI stages in your multistage approval enable you to implement automated decision-making at key points in your approval workflow. In these stages, you provide the AI model with specific instructions and relevant inputs such as documents, images, text, and organizational knowledge, and the AI produces a clear Approve or Reject decision with clear rationale based on your instructions. You can configure approval decisions from AI stages for review and approval by human approvers in subsequent stages. The human review ensures that humans remain in control always while still benefiting from faster processing of routine evaluations.

### Conditions and conditional approvals

Conditions can be placed between stages in your multistage approval to provide dynamic workflow control by evaluating specific criteria to determine the next action in your approval process. With conditions, you can define logical rules based on inputs and configure the workflow to automatically approve, reject, route to another stage, or skip stages entirely based on whether those conditions are met. Conditions enable flexible decision-making that adapts your approval workflow to different scenarios without manual intervention.

## Set up your multistage approval

There are a few steps to set up a multistage approval.

### Add the multistage action

In your agent flow, you can add *Run a multistage approval* as an action via the new *Human in the loop* connector.

1. Open the designer of the agent flow into which you want to add a multistage approval.
2. Select the + button in the location between nodes where you want to add the multistage approval.
3. From the **Add an action** card or pane, you see the *Run a multistage approval* action in the *Human in the loop* section. You can also find the same action in the *Human in the loop* connector from the list. If you don't see the action or connector, search for them.
4. For easy access next time, favorite the action by selecting the star icon on the action card.
5. Select *Run a multistage approval* action. This action starts an approval, sends it to the required assignees, and waits for the approval requests to be completed, before the rest of the flow continues.

Add an action

Search for an action or connector

> Favorites

▽ AI capabilities [See all \(25\)](#)

-  Run a generative action [Preview](#)
-  Let AI use a connector [Preview](#)
-  Run a prompt
-  Respond to the agent

▽ Human in the loop

-  Request for information (previ... [Preview](#)
-  Run a multistage approval (pr... [Preview](#)
-  Create an approval
-  Start and wait for an approval

▽ Connect to systems [Preview](#) [See all \(5\)](#)

-  Microsoft Dataverse >
-  Office 365 Outlook >

6. By default, the name of your multistage approval is the action name, "Run a multistage approval." To rename your approval, you must do so in the left pane. If you're using the new inline designer, select the button to open the left pane.



The screenshot shows the 'Run a multistage approval' action card from the previous step. A red box highlights the three-dot menu button in the top right corner of the card.

7. In the left pane, select the action name and rename as you choose.

Expense Reimbursement Approval

Parameters ● Settings Code view Testing About

Approval Process \*

**Set up your approval process**

Create advanced, multistage approvals to streamline even the most complex process. [Learn More](#)

**Set up the approval**

8. Select **Set up the approval** to open the modal.

## Add and configure a manual stage

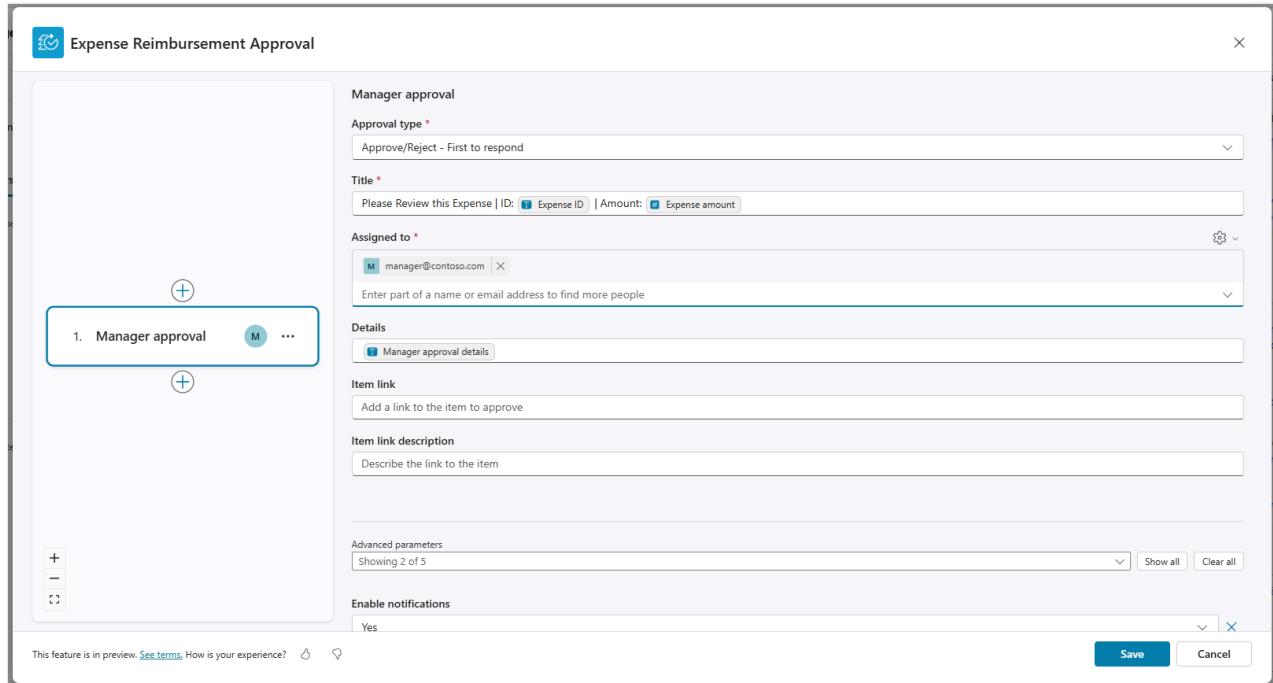
Approvals are configured in *stages*. Stages represent different gate points in the approval process where the flow pauses and waits for a *designated approver* to review and approve before continuing. You can configure each stage independently, customizing to meet the needs of each group of stakeholders.

Let's start with how to configure a manual stage.

1. In the modal, select **Create** and from the dropdown, select **Manual stage**. A single tile appears in the approval viewer.
2. On the right-hand side of the dialog, select the approval type you want. Check out [approval types and their behaviors](#) to learn about the different types of approvals and how they work.
3. Other configuration options appear. Configure these fields as you would configure a standard approval. The three main fields that should be configured are: **Title** (signifies the title of the approval request); **Assigned to** (details that human the approval should go to); **Details** (provides all the relevant details that the human approver need to make an approval decision).

**!** Note

In the "Assigned to" field, don't assign the same approver to multiple stages as this causes the flow to fail.



## Using inputs in a manual stage

If you have tokens or variables created in your agent flow that you wish to use in your manual stage approval, you need to create inputs. You can't directly use tokens or variables inside the approval without creating inputs first. In the previous image, we created three inputs namely, *Expense ID*, *Expense amount*, and *Manager approval details*. To add an input, simply hit slash, / on your keyboard where you want the input to be, choose from one of these types, `String`, `Number`, and `Boolean`, and then give your input a name. If you were to save this approval process, you would see fields for the inputs you created in within the flow designer. You can now map a token or variable from the flow to this input.

The screenshot shows a Microsoft Copilot Studio interface. At the top center is a circular button with a blue plus sign and a downward arrow. To the left is a blue square icon with a white circular arrow and a checkmark. To the right are three small icons: a clipboard, a gear, and three dots. Below these are sections for 'Approval Process \*' (with an 'Edit' button), 'Expense ID \*' (with a placeholder 'Expense ID X'), 'Expense amount \*' (with a placeholder '{x} Expense amount X'), and 'Manager approval details \*' (with a placeholder 'Approval details X').

## Add and configure an AI stage

### Important

Before you add an AI stage to your multistage approval, please ensure that you have Copilot Studio Copilot Credits assigned to the environment you're working in. See [Copilot Studio licensing](#) for more information on licenses and billing. If you don't have sufficient Copilot Credits, your approval stalls and doesn't proceed.

Let's now configure an AI stage.

The screenshot shows the 'Expense Reimbursement Approval' application. On the left, there's a sidebar with a plus sign icon and a manual stage section. In the main area, there's a 'Initial AI review' stage (1) with an 'Add content' button (3). The 'Instructions' pane (2) contains sample text and a model picker (4). The 'Test' pane (5) shows a rejected result (6) with reasons like budget exceedance and overspend. A 'Next step based on decision' section (7) is also visible.

1. In the approval viewer on the left pane, select the plus button "+" above the manual stage that you created.
2. Select **AI stage** from the dropdown. A single tile for the AI stage appears in the approval viewer.
3. Select the text in the tile and rename the stage (1).
4. In the **Instructions** pane (2), start by writing instructions and criteria you want the AI stage to use to make a decision.

### ⓘ Important

See [How to write instructions in your AI stage for better results](#) to learn how to craft your instructions well.

5. If you want the AI stage to use inputs such as documents, images, text, and knowledge to make an approval decision select "/" on your keyboard or select + **Add content** (3) where you want that input in the instructions. A menu appears to select an input type and then to upload a sample input for testing.
6. If you used inputs in your instructions, be sure to upload samples before testing.
7. From the model picker (4), select the AI model you want to use to make the approval decision.

### ! Note

Where possible, favor the more powerful models for approval decisions. For example favor GPT-o3 over GPT-4.1.

8. Select **Test** (5) to test the instructions and wait for some seconds.
9. In the **Approval decision** (6) pane view the test result, and below the rationale for the decision.
10. The **Next step based on decision** (7) section allows you to route your approval based on the decision from the AI stage. Here by default, if the AI stage approves, the approval *continues* to the next stage. If rejected, the approval *ends as rejected*. If the approval fails to give an approve/reject decision, it simply continues to the next stage.

 **Note**

The **Next step based on decision** section ensures that humans are always in control giving them the ability to make final decisions even after AI stages make decisions.

11. At this point if you save and close the multistage approval, you see fields for the inputs you configured in this AI stage in the flow designer.

## Add and configure a condition

Conditions are essential for routing approvals and automatically approving or rejecting based on specific criteria. If no conditions are set between stages, the default behavior is:

- If a stage is approved, the process moves to the next stage.
- If the stage is rejected, the approval process ends.

However, with conditions you can customize the default behavior of your multistage approval.

1. To add a condition, select **+** in the multi-stage viewer between the two stages.
2. Select **Condition** from the dropdown.
3. Add condition statements as you wish.

**Condition type \*** If/else  Switch**Condition \***

OR ▾

{x} Amount x ▾	Is greater than... ▾	50000	
{x} Customer x ▾	Is equal to ▾	High	

[+ Add statement](#) [+ Add group](#)

then

Go to 3. Senior adjuster approval

if not

Go to 2. Adjuster approval

**① Note**

On the left side of the condition statement, you can only use an input. You can select from existing inputs or create new ones.

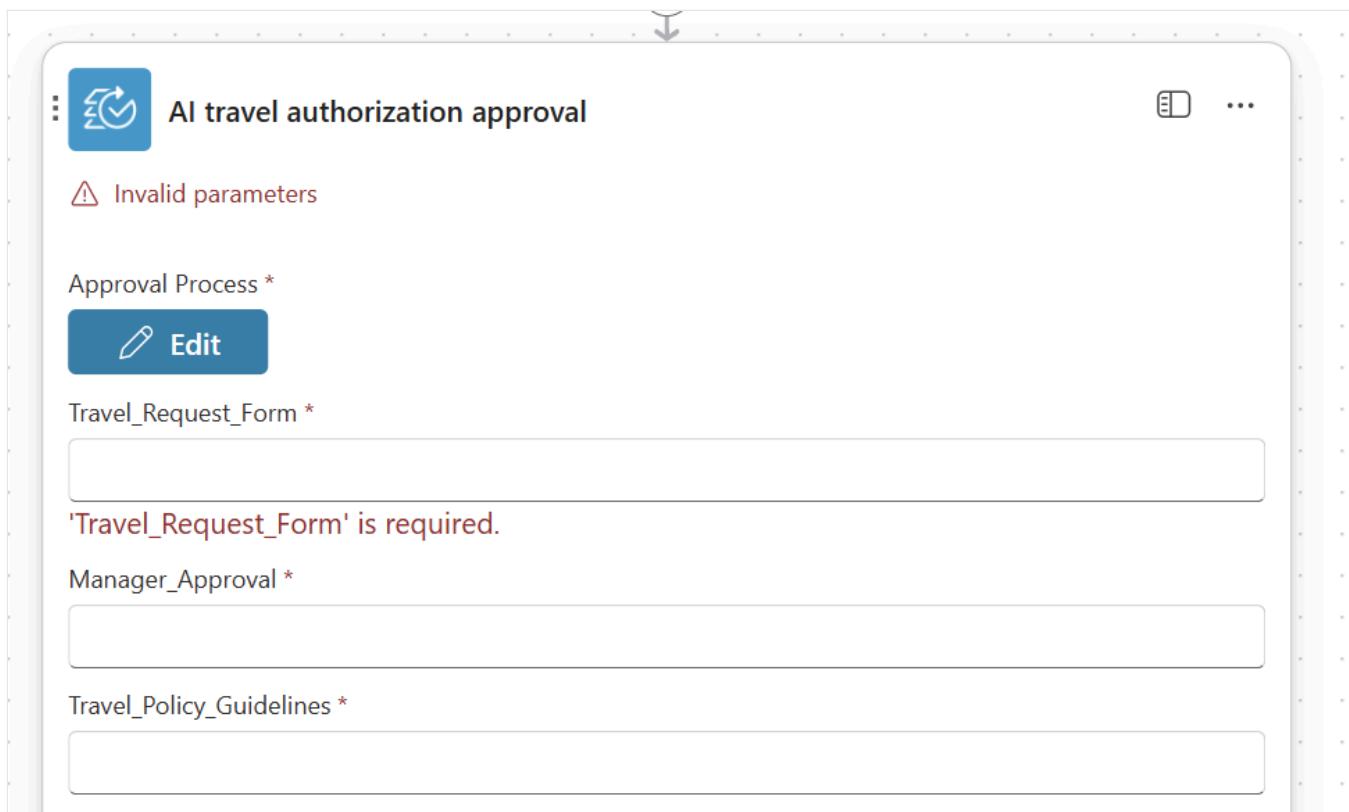
**4. Configure where the approval goes based on whether the condition is met:**

- If the condition is met, you can continue to the next stage, approve/reject, or route to another stage.
- If the condition isn't met, you can continue to the next stage, approve/reject, or route to another stage.

In this simple example, we define that if the expense amount is greater than \$5,000, the request should *continue* to the manager for approval. However, if the amount is under \$5,000 and the AI stage approves it, the request can bypass the manager approval and automatically *end the approval as Approved*.

## Mapping variables from your flow to inputs in the approval

If you created inputs inside of the approval, you need to map variables from your flow to those inputs. When you save and close the approval dialog, you can see fields to perform these mappings in the action card.



Select the fields and then select the items from preceding actions in the flow that you want to be used in the approval at run time.

## Document format requirements for approvals files

Documents used in approvals must be in [base64 format](#). Some connectors return outputs that are already in base 64. But if you're using outputs from those that don't, use the string function from PowerFX on the file as shown in the following image:

The screenshot shows the Microsoft Copilot AI interface. On the left, there's a code editor window with the following code:

```
string(triggerBody()?.['file'])?['contentBytes'])
```

Below the code editor, there's a button labeled "Create an expression with Copilot". Underneath that, there are tabs for "Function" and "Dynamic content", and a search bar.

On the right, there's a title "AI travel authorization approval" with a blue circular icon containing a white arrow. Below the title, there's a section titled "Approval Process \*". It contains three items:

- Travel\_Request\_Form \*
- Manager\_Approval \*
- Travel\_Policy\_Guidelines \*

Each item has a small "Edit" button next to it.

## Bringing it all together

Let's look at an example of how this approval process works with manual and AI stages, and conditions. The following image shows a complete approval process with four stages and a condition check.

The screenshot shows the Microsoft Copilot AI interface for an "Expense Reimbursement Approval" process. The process consists of the following stages:

1. AI expense screening
2. Manager approval
3. AI budget verification
4. Finance review

**AI expense screening**

Instructions: Review the following inputs:

- Budget allocation document
- Expense claim form
- Receipt attachments

And the criteria for rejecting the expense claim are as follows:

- Budget Exceeded:** Reject if the total claimed amount in the Expense Claim Form exceeds the approved budget limit specified in the Budget Allocation Document.
- Category Overspend:** Reject if any individual expense category (meals, travel, accommodation) exceeds its allocated portion by more than 10%.
- Unauthorized Items:** Reject if Receipt Attachments include expenses not covered under the approved budget categories in the Budget Allocation Document.
- Missing Receipts:** Reject if any expense line item in the Expense Claim Form lacks corresponding receipt documentation in Receipt Attachments.
- Invalid Dates:** Reject if any receipt date falls outside the project period specified in the Budget Allocation Document.

Reject if ANY of the criteria is met. In the rationale, ensure you state all the criteria for which you rejected.

**Approval decision**

Based on the prompt, the decision can have one of these two outcomes:

- Approved (radio button selected)
- Rejected

Make sure AI-generated content is accurate and appropriate before use. This feature is in preview. [See terms](#)

**Next step based on decision**

if approved	Continue
if rejected	End approval as 'rejected'
if failed	Continue

Save Cancel

This feature is in preview. [See terms](#). How is your experience? [See terms](#)

1. The approval request reaches the *AI expense screening* stage first when the approval runs.
  - a. If *approved*, the request moves to the condition check. b. If *rejected*, the approval process ends without reaching the condition check.
2. At the condition check: a. If the expense amount is greater than 5000, the approval continues to the Manager approval stage. b. If the expense amount isn't greater than

5000, the approval ends as Approved.

3. For expenses over 5000 that reach the *Manager approval* stage: a. If approved, the request moves to the AI budget verification stage for final compliance checks. b. If rejected, the approval process ends as Rejected.
4. The *AI budget verification* stage performs final budget and compliance validation: a. If the AI approves, the request continues to Finance review for processing. b. If the AI rejects, the approval process ends as Rejected.
5. Finally, the finance professionals in the *Finance review* stage perform their own evaluation based on accounting rules, budget availability, and payment processing requirements: a. If approved, the approval process ends as Approved and payment is initiated inside the agent flow. b. If rejected, the approval process ends as Rejected.

This demonstrates how you can build intelligent approval workflows that combine AI efficiency with human oversight, automatically routing requests based on specific business rules like expense amounts.

 **Note**

In each AI stage, you can configure whether the approval should continue to the next stage even when the AI approves, giving you full control over your workflow routing.

## Responding to approvals

Responding to approvals remains consistent between standard and advanced approvals. Users assigned to approvals can respond through:

- Microsoft Teams approvals app
- Outlook
- Power Automate portal

## More on AI stages

### Using AI approvals for sensitive cases

While AI approvals can streamline many business processes, exercise care when implementing them for highly sensitive approvals such as financial transactions, legal decisions, personnel actions, or compliance-critical processes. For these scenarios, we recommend ensuring that human approval stages are reached in the workflow so that humans remain in ultimate control

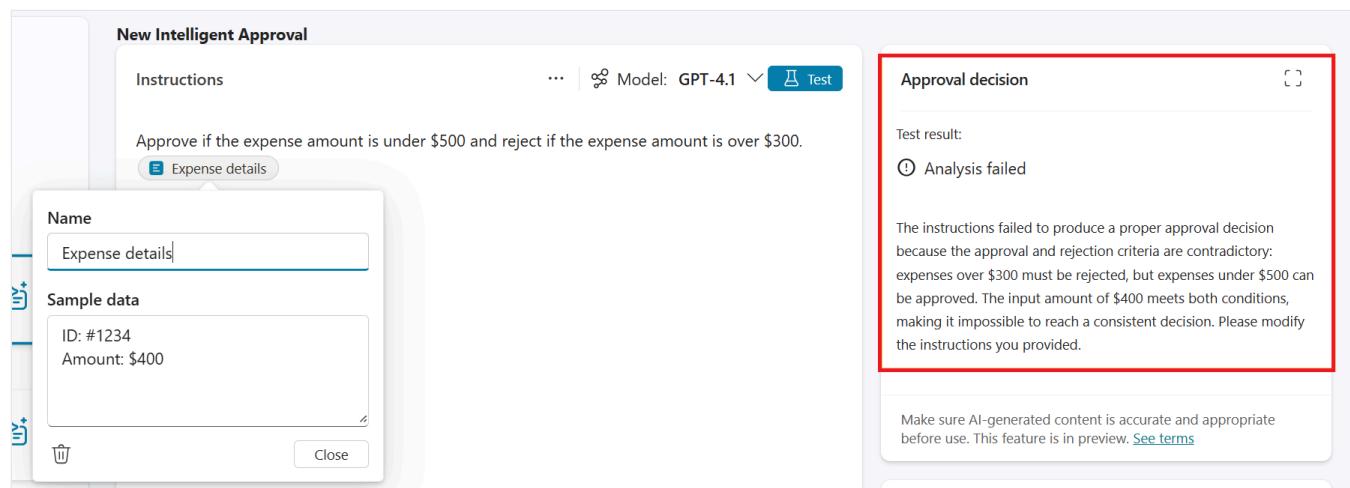
of important decisions. Review these terms <https://www.microsoft.com/en-us/business-applications/legal/legal-landing-page/> for more guidance.

For more information on responsible use of AI for approvals, see [FAQ for AI approvals](#).

## What happens when the AI stage is unsure of the decision to make

Sometimes, the AI stage might return a response labeled "Analysis failed." This typically occurs when the AI model is unable to determine an appropriate approval decision. Common causes include contradictory or off-topic instructions, or insufficient information provided to support a clear decision.

Here's an example where the AI returned an "Analysis failed" response due to conflicting input.



In this example, we instruct the AI stage to approve expenses under \$500 and reject those over \$300. However, the attached expense is \$400, which falls within both criteria. As a result, the AI can't confidently determine whether to approve or reject the request due to the conflicting instructions.

### Important

To avoid the "Analysis failed" response, follow guidelines in the *How to write instructions in your AI stage for better results* section of this article.

## Viewing AI stage decision and rationale in Power Automate

There are two ways to see the decision and rationale of the AI stage after it runs:

1. Approvals center

## 2. AI Builder activity screen

### View in approvals center

If you're an approver in a completed multistage approval, you should see the decision and rationale of the AI stages in Power Automate.

1. Sign in to [make.powerautomate.com](https://make.powerautomate.com).
2. From the environment picker, select the environment that the approval was run from.
3. From the left navigation pane, select **Approvals**.
4. In the **History** tab, select the approval that was completed and that includes the AI stage.
5. View the approval decision and rationale from the preceding AI stages in the right pane that opens.

### View in AI Builder activity screen

1. Sign in to [make.powerautomate.com](https://make.powerautomate.com).
2. From the environment picker, select the environment that the approval was run from.
3. From the left navigation pane, select **AI Builder activity**. If you don't see that option, select ... **More > Discover all**.
4. Under the monitor header in the **Discover** screen, select **AI Builder activity**. You can pin this screen to your navigation for easier access in the future.

# Discover all you can do with Power Automate

Pin your most-used items to the left navigation for easy access

**Approvals**  
View the status of approval requests you've sent, act on approvals assigned to you, and view completed approvals that you've sent or received in the past. [Learn more](#)

**Solutions**  
Author, package, and maintain units of software that extend Microsoft Dataverse. [Learn more](#)

**Automation center**  
Monitor your cloud flows and desktop flows. [Learn more](#)

**Process mining**  
Simplify business processes by identifying workflow bottlenecks and automation opportunities. [Learn more](#)

**AI hub**  
Add intelligence to automate processes and gain insights. [Learn more](#)

**Data**  

- Tables**: Create tables in Dataverse—the smart, secure, and scalable low-code data platform.
- Credentials**: Manage all your credentials for this environment.
- Connections**: View and manage all of your data

**Monitor**  

- Cloud flow activity**: View a summary of daily flow activity.
- AI Builder activity**: Tables and graphs help you monitor model activity and track the credits you've used. (This section is highlighted with a red box.)
- Desktop flow activity**: Monitor and manage your desktop automations and machines.

**AI**  

- AI models**: Leverage AI to create models to gain insights and enhance productivity.
- Prompts**: Create prompts to streamline daily business interactions.

In this screen, you can see all the runs of your AI prompts including the runs from the AI stages.

Processed time	Tool name	Input	Output	Used in	Estimated consumption
07/03/25, 11:26 AM	Unused Prompt	{"Expense_20details": "ID: #123..."}	{"Decision": "Rejected", "Ratio..."}	Quick Test	0 credits
07/03/25, 11:26 AM	Unused Prompt	{"Expense_20details": "ID: #123..."}	{"Decision": "Approved", "Ratio..."}	Quick Test	0 credits
07/03/25, 11:26 AM	Unused Prompt	{"Expense_20details": "ID: #123..."}	{"Decision": "Approved", "Ratio..."}	Quick Test	0 credits
07/03/25, 11:26 AM	Unused Prompt	{"Expense_20details": "ID: #123..."}	{"Decision": "Approved", "Ratio..."}	Quick Test	0 credits
07/03/25, 11:22 AM	Unused Prompt	{"Expense_20details": "ID: #123..."}	{"Decision": "Approved", "Ratio..."}	Quick Test	0 credits

5. Select the **Processed time** part of one of the rows (the blue text in the column) to see the decision and rationale of that run.

← 07/03/25, 11:26 AM

**Tool name**: Unused Prompt

**Input**: {"Expense\_20details": "ID: #123\nAmount: \$300"}

**Output**: {"Decision": "Approved", "Rationale": "The expense request is approved because the amount of \$300 is less than the \$500 threshold specified in the user instruction."}

**Consumption**: 0 credits

**Processed by**: aurora user06

**Model**: -

Here you see the **Input** which is the piece of text, document, or image that was given in the instruction and the **Output** which contains the decision and rationale of the AI stage

represented in the JSON format.

The screenshot shows the Power Automate interface with the 'AI Builder activity' selected. A red box highlights the 'Input' field, which contains the JSON string: `{"Expense_20details": "ID: #1234\nAmount: $300"}`. Another red box highlights the 'Output' field, which contains the JSON string: `[{"Decision": "Approved", "Rationale": "The expense request is approved because the amount of $300 is less than the $500 threshold specified in the user instruction."}]`. The left sidebar lists various categories like Home, Create, Templates, Learn, My flows, Approvals, Solutions, Process mining, Desktop flow activity, Custom connectors, and Power Platform.

## How to write instructions in your AI stage for better results

Approvals are a critical component of business processes, so it's essential that your instructions are clearly written and consistently yield accurate results. This section is especially important. Follow these best practices to ensure your AI approval stages deliver reliable and consistent decisions. While not every guideline might apply in every case, using them where possible helps maximize effectiveness.

- 1. Keep instructions focused on the business process:** Ensure your approval instructions relate to legitimate business process approvals such as expense approvals, purchase order approvals, contract reviews, etc. AI stages work best when evaluating specific business workflows with clear policies.
- 2. Use more powerful models for complex tasks:** Because accurate approval decisions are critical to your business, it's important to use the most capable AI models available whenever possible. For example, prefer GPT-o3 over GPT-4.1 for improved reasoning and consistency. For approval scenarios that involve complex tasks, intensive calculations, multi-step logic, or nuanced analysis, consider using reasoning-optimized models like o3 or your own fine-tuned models from Azure AI Foundry to achieve better accuracy and reliability.
- 3. Focus on either reasons to reject or reasons to approve:** Focus on either approval criteria OR rejection criteria, and avoid both. Mixing approaches can confuse the model. For example, if you provide the AI stage conflicting criteria such as "Approve if the expense is under \$500 but reject if the expense is over \$300 and it's a meal expense," it won't know how to make a correct decision when processing a \$400 meal expense that meets both conditions simultaneously.
  - Good:** "Reject if the expense amount exceeds \$500 or if no receipt is attached"
  - Avoid:** "Approve if under \$500 but reject if over \$500 unless it's travel expenses"

**4. Reference your inputs specifically:** When using uploaded inputs, documents, or data, reference them by name and specify exactly where the AI should look for information in your instructions. This ensures the AI knows which files to check and what data points to extract.

- **Good:** "Check the 'Budget\_Report.pdf' file and approve if the Q3 revenue exceeds \$100,000".
- **Avoid:** Avoid: "Check the budget file for revenue numbers".

**5. Be specific and measurable:** Avoid subjective terms that can be interpreted differently. Use concrete, measurable criteria that leave no room for ambiguity.

- **Good:** "Approve if delivery date is within five business days of request date"
- **Avoid:** "Approve if the timeline is reasonable" (here there's no quantitative rule for what 'reasonable' means)

**6. Ensure file quality:** Verify that inputs such as images and documents are clear, unprotected, and readable. Blurry images, password-protected files, or corrupted documents are more likely to lead to an analysis failure.

**7. Test with realistic data:** To validate your instructions work as expected, use representative examples that mirror your actual approval scenarios. Don't just test with perfect examples —include edge cases. For example, use real documents from your workflow, test with missing information scenarios, include borderline cases near your thresholds, try documents with formatting variations, test both approval and rejection scenarios.

**8. Define edge cases and fallback behavior:** Specify what should happen when the AI encounters unclear, incomplete, or unexpected information. This prevents the AI from making incorrect assumptions. For example, you can state, "If expense category is missing, reject the request."

**9. Set clear thresholds and ranges:** Avoid ambiguous boundaries by defining exact cutoff points. Be explicit about what happens at boundary values.

- **Good:** "Approve if satisfaction score is 4.0 or higher on a 5-point scale"
- **Avoid:** "Approve high satisfaction scores"

**10. Use structured formats for complex criteria:** For multi-step evaluations, break down your logic into numbered steps or decision trees. This ensures the AI processes each condition systematically and makes your approval logic transparent. Here's a good structure:

"Reject the expense report if ANY of the following conditions are true:

- a. Amount in the "Total" field exceeds \$500.00
- b. No receipt image file is uploaded

- c. Expense date is more than 60 days before the date due date found in the expense guidelines
  - d. Manager signature field in the expense form is blank
  - e. Employee ID field is empty or doesn't match format XXX-XXXX (three numbers, dash, four numbers)"
11. **Include relevant organizational context:** Reference specific company policies, compliance requirements, or business rules the AI should enforce via inputs like documents or knowledge. For example, you could state, "Follow Travel Policy v2.3: meals over \$75 require itemized receipts with business justification. Reject if receipt or justification is missing."
12. **Avoid time-sensitive references:** Don't use relative dates or time references that become outdated. Use specific timeframes or calculated periods instead. The AI model might not know what today's date it as an example.
- **Good:** Approve if submission date is within 30 days of expense date
  - **Avoid:** "Approve if submitted this month"
13. **Use consistent terminology:** To avoid confusion, stick to the same terms throughout your instructions. Avoid using different terms for the same thing. For example, use "expense amount" consistently throughout your instructions and avoid switching between "expense amount," "cost," "total," "sum," and "charge"

## Known limitations

The following are known limitations:

- Only base64 encoded file contents in file inputs for AI stages are supported.
- If a user in a different environment from which an approval was sent responds to the approvals, the approval might fail. To prevent the failure, identify the user you want to send the approval to, go to the Power Platform Admin Center, and add the user to that environment.
- You can't assign the same approver to different stages. For example, if you assign John Doe to stage 1, you can't assign him to stage 2 as well. This causes your flow to fail.
- File attachments aren't currently supported.
- Application lifecycle management (ALM) is currently not supported for advanced approvals, meaning that if you import a flow with an advanced approval, you need to recreate the multi-stage approval in the flow.
- Sharing is currently not supported, meaning that if you share a flow that has the human in the loop connector, the recipient needs to recreate the multi-stage approval in the flow.



# Request information from humans in the loop

09/04/2025

Human input, review, and action are essential to ensure that automations and AI workflows run effectively and maintain high quality. The *request for information* (RFI) action in agent flows provides a streamlined way to incorporate human oversight into automated processes.

## What is the request for information action?

Request for information is an action in agent flows that lets workflows:

1. Pause execution
2. Collect input from designated human reviewers before proceeding with subsequent steps
3. Use input from human reviewers in subsequent steps

## Access request for information

The request for information action is available exclusively in *agent flows*. Agent flows are automations built in Copilot Studio that can be used standalone or integrated with Copilot Studio agents. Learn more about agent flows in the [agent flows overview](#).

## Common use cases for request for information

The request for information action addresses critical business needs across various industries. There are many scenarios where this action can be used. Here are a few examples:

- **Insurance claims processing:** When automated claim review identifies missing documentation or requires damage assessment details, the RFI action can request specific information from field adjusters, such as repair estimates, photo documentation, or on-site inspection results.
- **Financial services compliance:** During loan applications or account reviews, workflows can pause to request more verification documents from compliance officers, such as income verification, credit assessments, or regulatory approvals.
- **Supply chain quality control:** Manufacturing processes can request inspection reports from quality assurance teams when automated systems detect potential defects or when supplier certifications need human verification.

- **Legal document review:** Contract approval workflows can pause to request legal counsel input on specific clauses, risk assessments, or compliance requirements that require professional judgment.
- **IT security incident response:** Security workflows can automatically request more investigation details from cybersecurity analysts when threat detection systems identify suspicious activities requiring expert analysis.

## Set up request for information in your agent flow

There are a few steps to set up and test the request for information action in your agent flow.

### Add request for information in your agent flow

If you don't have an existing agent flow for this action, start by creating a new agent flow:

1. Log into Copilot Studio at [copilotstudio.microsoft.com](https://copilotstudio.microsoft.com).
2. Select **Flows** from the left navigation to access the agent flows menu.
3. Select **New agent flow**.
4. Add a trigger to your flow.

If you already have an agent flow and want to add this action, open your existing agent flow in the flow designer and continue with the following steps:

1. In the flow designer, select the add node (+) at the location where you want to add the RFI action.
2. From the *Human in the loop* section of the action picker, select the **Request for information** action. You can also find the action via the *Human in the loop* connector.

Add an action

Search for an action or connector

> Favorites

AI capabilities [See all \(25\)](#)

- Run a generative action [Preview](#)
- Let AI use a connector [Preview](#)
- Run a prompt
- Respond to the agent

Human in the loop

- Request for information (previ... [Preview](#)) **(This item is highlighted with a red box)**
- Run a multistage approval (pr... [Preview](#))
- Create an approval
- Start and wait for an approval

Connect to systems [Preview](#) [See all \(5\)](#)

Microsoft Dataverse > Office 365 Outlook >

3. Select the star to favorite it for easier access next time.

4. Configure the following required fields:

- **Title:** The subject line of the email that is sent with the request. Example: "Missing Information on Claim with ID: ABC123"
- **Message:** The message or note that accompanies the request for information. Example: "This request is being sent to you because... Provide this information at your earliest convenience."
- **Assigned to:** The email addresses of individuals who receive the request. You can enter multiple emails, however, the response from the first person to respond is used.

 Request for information

Title \*

Specify the title of the request.

Message \*

Specify the message for the request.

Assigned To \*

Enter part of a name or email address to find people

Input \*

+ Add an input

### Important

- The response from the first person to respond is used in the agent flow.  
Subsequent responses aren't processed.
- All requests are currently sent via Outlook only. More platforms might be added in the future.
- Requests CANNOT be sent to users outside of your tenant.

5. Select + Add an input.

6. Add your first input by selecting one of the available input types.

7. Five input types are currently supported:

- **Text** (string)
- **Yes/No** (Boolean)
- **Email** (email address)
- **Number** (integer)
- **Date** (date)

## Choose the type of user input

X



Text



Yes/No



Email



Number



Date

Inputs offer flexibility with options for multi-select, single-select, and required fields. For more details on details for inputs, see [More on inputs](#).

8. Give the input a descriptive name.
9. Add more inputs as needed by selecting + **Add an input**.

Here's an example of a fully configured request for information action:

 Request for information from assessor

Title \*

Assessment Needed for Claim with ID:  Claim ID X

Message \*

Please conduct an assessment for claim with:  Claim ID X .

Details:  Claim details X

Assigned To \*

A aurorauser06@capintegration01.onmicrosoft.com X

Enter part of a name or email address to find more people

Input \*

	Date	Please enter the date on which you conducted the assess	...
	Summary	Please enter the summary of the assessment	...
	Damage amt	Please enter the amount of the damage	...
	Assessor	Please enter your first and last name	...
	Assessor ema	Please enter your email address	...
	High priority	Please select yes or no	...

## Use parameters from request for information in other actions

Every input that you specify in your request for information action produces a parameter that can be used in subsequent actions.

For example, in the previously shown configuration, all the parameters such as `Date`, `Summary`, `Damage amt.`, and others become available as dynamic content that can be passed as input into subsequent actions.

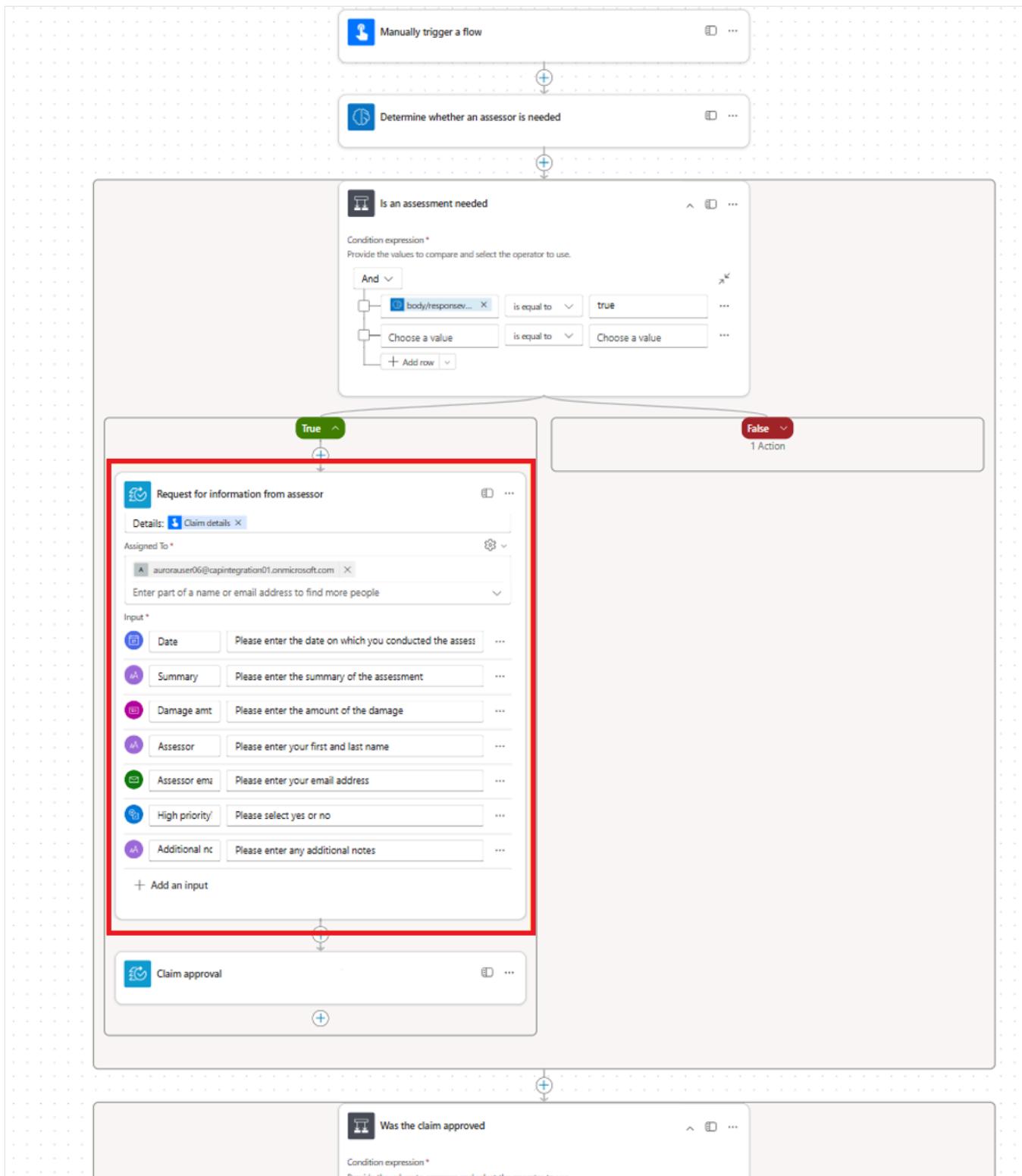
In the example that follows, we use the parameters from the request for information action in a subsequent AI approval action:

The screenshot shows two side-by-side interface snippets. On the left is a 'Request for information from assessor' form with fields for Date, Summary, Damage amt., Assessor, Assessor email, High priority?, and Additional notes. A red box highlights the 'Determine whether an assessor is needed' action below these fields. On the right is an 'AI approval in multistage' process step, showing an 'Edit' button and a field labeled 'AssessorSummary \*' which also has a red box around it.

At runtime, when the human responds to the request, the parameters are automatically populated with the values from their submission.

## Run and test request for information in your agent flow

You now have a request for information action added to your agent flow. Now it's time to test it. Here's a snippet of a claim approval flow with the request for information action highlighted. We use this flow for demonstration purposes.



To run your agent flow:

1. Save and publish the agent flow.
2. Select **Test** to test run your flow.

The flow runs. When the flow reaches the request for information action, it pauses execution and sends the request to the assigned humans via email through Outlook.

The following images show the relationship between the request configurations in the agent flow designer and the email received by the reviewer. Notice that the inputs

specified in the agent flow designer in the first image correspond to the inputs being requested in the email as seen in the second image.

Input \*

 Date	Please enter the date on which you conducted the assessment	...
 Summary	Please enter the summary of the assessment	...
 Damage amt	Please enter the amount of the damage	...
 Assessor	Please enter your first and last name	...
 Assessor email	Please enter your email address	...
 High priority	Please select yes or no	...
 Additional notes	Please enter any additional notes	...

+ Add an input

Assessment Needed for Claim with ID: HGDGR343

**Request for information | Microsoft Copilot Studio**

Please conduct an assessment for claim with: HGDGR343. Details: Flooding

Requested by aurorauser06 <[aurorauser06@capintegration01.onmicrosoft.com](mailto:aurorauser06@capintegration01.onmicrosoft.com)>

Date Created Tuesday, July 22, 2025 3:55 PM

**Date**

**Summary**

**Damage amt.**

**Assessor**

**Assessor email**

**High priority?**

**Additional notes**

This message was created by an agent flow in Microsoft Copilot Studio. Do not reply directly to this address.

**Submit**

 Reply  Forward

3. Within Outlook, the reviewer enters the responses to each of these fields and then selects **Submit**.

After the reviewer submits the responses, the flow resumes execution and continues with the remaining actions. As explained previously, the response data can be used in subsequent actions throughout the flow.

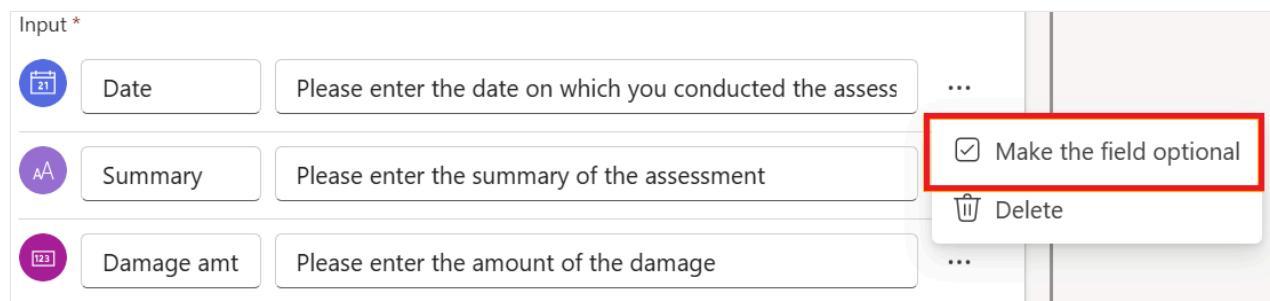
 **Note**

Since requests are sent via Outlook, requests could be sent to users who are outside of the environment in which the agent flow was built.

## More on inputs

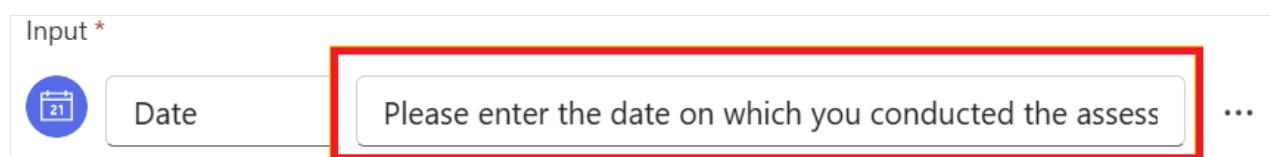
You have some flexibility in configuring your inputs. Here are some ways you can make inputs more flexible:

- **Make some inputs optional (all input types):** When you send requests for information, you might want to make certain fields optional. In the designer, select the three dots beside the input you added and select **Make the field optional**. When this request is sent, that input is optional for the assignee.



- **Placeholder text (all input types):** You can customize the placeholder text that the assignee sees when they receive a request. Use this to provide additional information or hints on how the field should be completed.

To do so in the designer, select the field beside the name field of the input and customize the placeholder text. When the assignee receives the request, they see the corresponding placeholder text in each field. For example, in the image that follows, they would see: "Please enter the date on which you conducted the assessment."

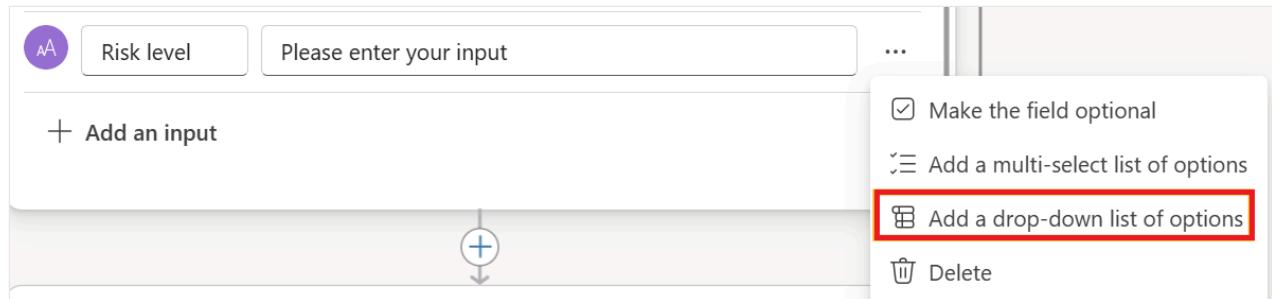


## Date

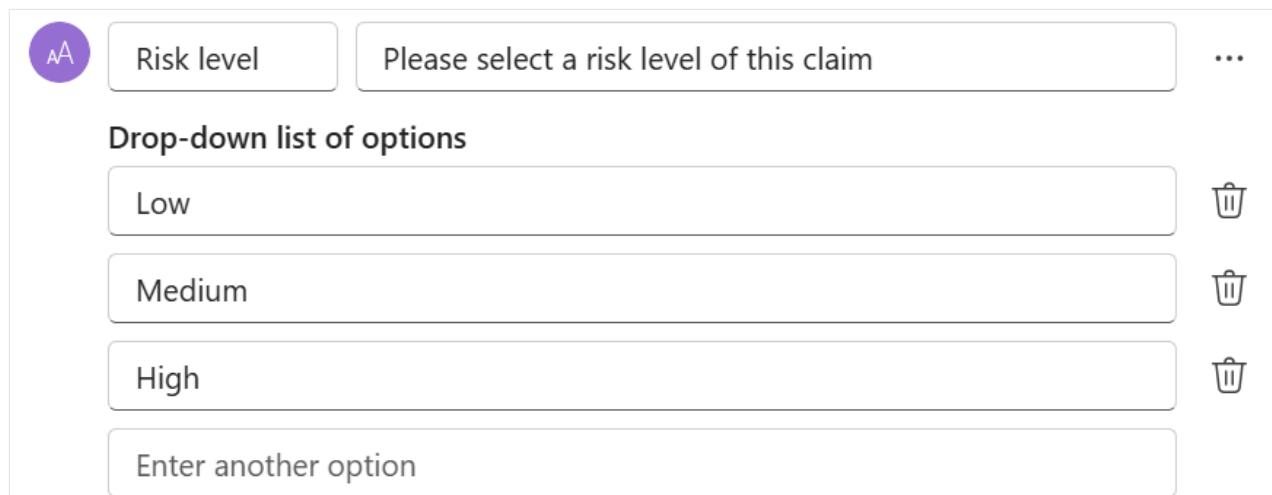
Please enter the date on which you conducted the assessment \*

- **Single-select options (text inputs):** For text inputs, you can provide predefined options to assignees. These options can be configured as single-select, requiring the assignee to choose one option from the list.

To do so in the designer, select the three dots beside the text input of your choice and select **Add a drop-down list of options**.



Add the options you want the assignee to choose from.



When the request is sent, the assignee can select one of the options from the list.

- **Multi-select options (text inputs):** For text inputs, you can also provide predefined options that allow assignees to select multiple choices. Multi-select lets the assignee choose more than one option from the available list.

To do so in the designer, select the three dots beside the text input of your choice and select **Add a multi-select list of options**.

AA

Issues

Please enter all identified issues (select all that apply)

...

**Multi-select list of options**

Cosmetic damage



Structural damage



Mechanical issues



Electrical failures



Enter another option

# Agent flows in Microsoft Copilot Studio

## FAQ

Article • 04/15/2025

This article answers common questions about agent flows in Microsoft Copilot Studio.

## What are agent flows in Microsoft Copilot Studio?

Agent flows are automated sequences of actions that are triggered by specific events or conditions. Agent flows help streamline repetitive tasks, improve efficiency, and ensure consistency in processes. They can range from simple tasks, like sending notifications, to complex workflows that involve multiple systems and integrations.

## How are agent flows different from topics in Copilot Studio?

Both topics and agent flows are deterministic pathways for automation—that is, given the same inputs, they'll always produce the same output, making them reliable and predictable.

However, topics are optimized for managing conversational flows. They can be enhanced with actions for automation behind the scenes of the conversation. Agent flows are optimized for managing business processes. They have more comprehensive automation capabilities for nonconversational processes.

## How are agent flows in Copilot Studio different from cloud flows in Power Automate?

Agent flows in Copilot Studio and cloud flows in Power Automate serve distinct purposes and offer different functionalities, although they can complement each other in automation tasks.

Agent flows in Copilot Studio are designed to provide a native experience for building and integrating agents with business models. They focus on intelligent, AI-driven automation and interaction with systems. Agent flows are optimized for use in Copilot Studio, offering a seamless maker experience that simplifies and accelerates agent development. Unlike cloud flows, which require a Power Automate license, agent flows are billed through Copilot Studio based on consumption, without requiring individual licensing. This model helps customers scale process transformation enterprise-wide. Also unlike cloud flows, agent flows can't be copied, shared, have co-owners, or give run-only permissions in Copilot Studio.

Cloud flows in Power Automate provide a flexible, low-code approach to automating workflows across various applications and services. They offer a wide range of connectors and actions to integrate with different systems and services. Cloud flows are designed for general automation scenarios and can be used independently or with agent flows to create end-to-end automation solutions.

## Why can't I see my flows in Copilot Studio?

Your flows were probably created in Power Automate. The Copilot Studio Flows page only displays agent flows created in Copilot Studio. Flows that were created in Power Automate must be managed in Power Automate, unless you convert them to agent flows.

## I want to add a flow to an agent. Are there any requirements?

To add a flow to an agent, it must be a solution flow and contain the **Run a flow from Copilot** trigger and the **Respond to Copilot** action.

## Can an agent flow be used in multiple agents?

Yes. You can add an agent flow as an action to any of your agents as needed.

## Do I need a Power Automate license to create flows in Copilot Studio?

No. Agent flows are billed in Copilot Studio based on usage. They aren't included entitlements in Power Automate.

## Can I use premium connectors in an Agent Flow?

Yes. Agent Flows are premium flows that can use [premium connectors](#).

# Extend the capabilities of your agent

06/16/2025

Copilot Studio lets you extend the capabilities of agents using various mechanisms. This includes:

- Topics
- Tools
- Knowledge sources
- Other agents (preview)

Tools and knowledge sources that require accessing another service make use of *connectors*. Connectors allow you to integrate data from various enterprise systems and APIs, enabling your agents to access and utilize this data effectively. You can use prebuilt connectors or create custom connectors to connect to your enterprise data sources.

Connectors can include:

- Data sources within the productivity cloud in Microsoft 365
- Business data in Dynamics 365
- Analytical data in Microsoft Fabric
- Non-Microsoft enterprise data sources

Through connectors, you can perform actions and ground your agent responses with multiple enterprise data sources for more relevant and reliable responses from custom agents and Copilot agents.

- For more information on supported knowledge sources using connectors, see [Knowledge sources overview](#).
- For more information on connectors that have enterprise data added using real-time connectors, see [Add real-time knowledge with connectors](#).

## How does it work?

When you sign in to Copilot Studio, you're allowed to access specific agents based on the license associated with your credentials. The list of agents available to you depends on these permissions. If you don't see the agent you wish to extend, verify your credentials. Also try connecting to the agent itself, or try to trigger any jobs that must occur to prepare the agent for use. See the sections detailing the authoring experience for examples.

A maker starts by selecting the agent whose functionality they wish to extend:

- Custom agents

- Other copilots that they have maker access to in their organization
- Microsoft 365 Copilot

For custom agents, you can add tools and knowledge directly to the agent.

For Microsoft 365 Copilot, you don't extend directly with tools and knowledge. Instead, you extend Microsoft 365 Copilot with an *agent* built in Copilot Studio. You can bring tools and knowledge into Microsoft 365 Copilot in a three step process:

1. Create an agent.
2. Add tools and knowledge to the agent.
3. Publish the agent and make it available to Microsoft 365 Copilot.

To select an agent to extend, start by selecting **Agents** in the main menu. For a custom agent, select from the list of custom agents on the **Agents** page. For Copilot agents, select **Copilot for Microsoft** and then select the Copilot agent from the Copilot for Microsoft 365 agents list.

## Requirements

- Maker access to the agent to allow you to extend it
- Copilot Studio license for the maker
- Data sources to use to integrate other data
- Information you need to connect to the content sources (connectors, APIs, and so on)
- Credentials for any systems you're connecting to
- Assistance from the administrator of your agent to enable the extension once you publish the agent

## Related content

[ ] [Expand table](#)

Article	Description
<a href="#">Create and edit topics</a>	Create and edit topics to extend the capabilities of your agent.
<a href="#">Use tools with custom agents (preview)</a>	Extend a <i>custom agent</i> with tools.
<a href="#">Add other agents</a>	Add other agents to your agent to extend its capabilities.
<a href="#">Knowledge sources overview</a>	Learn about the knowledge sources available in Copilot Studio.
<a href="#">Extend Microsoft 365 Copilot with agents</a>	Extend Microsoft 365 Copilot with an agent built in Copilot Studio.

# Extend Microsoft 365 Copilot with agents

06/13/2025

Microsoft 365 Copilot provides default experiences for engaging with content and resources from across an organization. While the default functionality provides many useful capabilities, sometimes customers need to interact with a more focused conversational assistant, accessible directly within Microsoft 365 Copilot. An example of where a more focused assistant is useful is when you need support for a specific subject area with specialized vocabulary, grounded on a subject-specific knowledge base.

You can help Microsoft 365 Copilot handle these situations by extending it with a specialized agent built in Copilot Studio.

There are two different ways agents can be added to Microsoft 365 Copilot:

- Custom agents
- Copilot agents

## (!) Note

Copilot agents are called *declarative agents* in Microsoft 365 Copilot. Learn more about [Microsoft 365 Copilot extensibility](#).

This article covers:

- Authoring a custom agent
- Authoring a Copilot agent
- Adding tools and knowledge to an agent
- Publishing a custom agent to Microsoft 365 Copilot and Microsoft Teams

## What are Copilot agents, tools, knowledge, and suggested prompts?

The following table gives more detailed information about the distinctions between Copilot agents, tools, copilot knowledge, and suggested prompts.

 Expand table

Name	Definition
custom agent	A custom agent is an agent a maker builds from scratch. A custom agent can include a wide range of capabilities, including knowledge, actions, and suggested prompts. Custom agents are authored in a conversational interface that generates prompts for the Copilot Studio model. Custom agents can be published to the Microsoft 365 Copilot and Teams channel, but can also be used with other channels.
Copilot agent	A Copilot agent is created in Copilot Studio by authoring a prompt for Microsoft 365 Copilot. This prompt is used to define a set of runtime behaviors, personalities, and rules that govern the agent's interactions. A Copilot agent is equivalent to a declarative agent created in Microsoft 365 Copilot agent builder. A Copilot agent is used in the context of extending Microsoft 365 Copilot, and can be equipped with actions and knowledge. A Copilot agent can also be used directly as an agent in Microsoft Teams.
Tool	The mechanism that lets an agent call out to other systems and retrieve data or perform other actions. Tools are single-turn interactions. Tools can use authentication, and can include parameters and other functionalities. See <a href="#">Use tools with custom agents</a> .
Knowledge	Knowledge refers to information that lives within your Microsoft 365 tenant, including SharePoint, and Graph connectors.
Suggested prompt	Suggested prompts are maker-authored prompts in an agent that make it easy for customers to do common or sophisticated actions with the agent.

## Keeping agents secure

Microsoft 365 customers and partners can build Copilot agents that extend Microsoft 365 Copilot with custom instructions, grounding knowledge, and tools invoked via REST API descriptions configured by the agent. Tools grab data that influences the behavior of your Microsoft 365 Copilot, including from untrusted sources like emails and support tickets. Attackers can create payloads in these untrusted sources to manipulate the agent's behavior, such as incorrectly answering questions or invoking custom actions.

To protect your agent from attacks, use the controls in Copilot Studio to [configure secure connectors used for knowledge and custom actions](#).

## Prerequisites

- Users and makers must have a license to use Microsoft 365 Copilot.
- Makers must have permissions to create in and have access to a Copilot Studio environment.

## Create and configure a custom agent

Check out the following articles for details on creation and configuration for custom agents:

- [Create a custom agent](#)
- [Add knowledge to a custom agent](#)
- [Add tools to a custom agent](#)

## Create a Copilot agent

Creation of Copilot agents in Copilot Studio is similar to that of custom agents, in that it uses a conversational authoring experience.

However, the starting point is different for Copilot agents.

Custom agents use the main agent creation flow via either **Create > New agent** or **Agents > New agent**.

Copilot agents on the other hand are created from the Copilot for Microsoft 365 agent page under **Agents**.

To create a Copilot agent for Microsoft 365 Copilot:

1. Sign in to Copilot Studio at <https://copilotstudio.microsoft.com>
2. Navigate to **Agents** in the left side navigation panel.
3. Select **Copilot for Microsoft 365** from the list of agents.
4. From here, select **Add** on the **Agents** card. You're sent to the conversational authoring experience.
5. In full sentences, describe what the agent should be able to do and know.
6. When the interaction is completed, select **I'm done** in the chat.

 **Note**

Skip over the conversational interface at any time, by choosing **Skip to configure**.

7. Select **Create** to get to the overview page for the agent.

The Copilot agent is now created. You can interact with the agent in the test canvas, or add more capabilities and knowledge to it. So far, the agent has no added knowledge sources. The only information the agent has, is the data provided to it in the instructions. The agent doesn't

have access to enterprise knowledge or any other actions at this time, so any responses are provided from the generative AI's own knowledge.

You can see the newly created Copilot agent under the **Agents** tab on the **Copilot for Microsoft 365** agent page. Because the agent isn't a custom agent, it doesn't appear in the main agents list.

## Add knowledge to a Copilot agent

When you first create an agent, it starts out with only the base generative AI knowledge. The agent is unaware of any other data sources. To give it more knowledge, add knowledge on the **Overview** page for your agent, or on the **Knowledge** page.

There are three primary types of knowledge that can be added to an agent:

[ ] Expand table

Name	Definition
SharePoint	Paths to SharePoint resources in the organization. Microsoft 365 Copilot uses the credentials of the user to ensure they only are given answers from information they have access to.
Copilot (Graph) connectors	Connectors are data sources configured by the tenant admin to add knowledge to Microsoft Copilot and agents. Learn more about <a href="#">Copilot connectors</a>
Web search	The agent can also use public web search (via Bing) to retrieve web results

### ! Note

**Web Search** is configured via the capabilities card on the overview page, not on the knowledge page.

To start adding knowledge to the agent:

1. Return to the agent overview page.
2. From the **Knowledge** card, select **Add Knowledge**.
3. Choose a data source.
  - If you're using SharePoint, provide a path to root of the library or site. A single page or file being specified isn't supported at this time.
  - If you're using a Copilot connector and the admin configured a Copilot connector in Microsoft 365, you can choose a Copilot connector from the configured data sources.

- Web search can be enabled/disabled from the agent overview page in the **Additional Settings** card.

Multiple knowledge sources can be added with a mix of types to an agent. In addition, for SharePoint, up to four paths can be included.

With the knowledge sources configured, the agent with knowledge is immediately available in the test chat. Close the knowledge panel, choose the refresh gesture at the top of the test chat to get a new chat, and then try asking questions that should be present in the data source.

 **Note**

If results aren't returned, submit feedback to Microsoft for analysis and product improvement.

## Add a tool to a Copilot agent

A tool is used to enable an agent to retrieve information or to perform a task such as updating a record or completing a transaction. Tools can be created from Copilot connectors (also known as Power Platform connectors). Action types include AI Builder prompts, prebuilt connectors, and custom connectors. For more information about Copilot connectors, see [Add actions to custom agents](#).

 **Note**

Currently, Copilot agents don't support agent flows.

Adding a tool requires extra planning on the part of the maker vs adding knowledge. Several considerations come into play:

- Should the tool use the user's authentication or the maker's authentication?
- Can the maker provide descriptions to inform the generative AI when and how to invoke the tool?
- Should the parameters to the function always be automatically filled, or should the maker fill them at authoring time?

Each of these considerations comes in to play when adding tools to an agent.

To add a tool:

1. Return to the Copilot agent's overview page.

2. From the **Tools** card, choose **Add tool**.

3. Search for the desired tool or choose **New tool**.

If you select **New tool**, you're presented with an **Add tool** pane where you can select the type of tool to add. The available options are:

- Prompt
- Computer use
- Custom connector
- REST API
- Model context protocol

Agent flows aren't currently available as an option to extend Copilot agents.

4. For each of the new tool options, once you select the type of new tool to add, you're taken through a guided experience to create the tool. The experience is similar to the one used for adding a new tool to a custom agent.

## View and update tool configurations

Once you create a tool, it shows up in the list of tools in the agent overview page, under **Tools**. You can select the tool entry to view and edit the tool's configuration. There are three tabs in the tool configuration experience:

- Details
- Inputs
- Outputs

### Details section

The **Details** section is used to configure the tool name, display name, and description.

- **Tool name:** The name for the action as seen in Microsoft 365 Copilot in citations. This information also can be influential in helping Microsoft 365 Copilot choose the action to execute.
- **Display name:** Friendly name for the action as seen in the Microsoft 365 Copilot user experience.
- **Description for the agent to know when to use this tool:** A full conversational description of when Copilot should use this action, and the sort of questions that trigger it.

### Inputs and outputs sections

Copilot Studio automatically populates a set of inputs and outputs by inspecting the connector object. The maker can let Copilot fill in those fields or fill them in manually during authoring.

By default, no customization is needed on this experience unless the maker chooses to manually fill a field, or chooses to override a description.

## Test your tool

Once you're done making adjustments, you can test out the tool in the Copilot Studio test chat. Select **Refresh** at the top of the test chat, and then ask a question that should invoke the action.

The first time the user triggers the action, the agent presents a connection card in the chat. This card is used to prompt the user to sign in to the action with their own credentials or use their single sign-on (SSO) credentials, as applicable.

## Suggested prompts

Often when users first interact with any sort of agent, they don't know what they can ask, or don't understand the richness of the agent. As a result they tend to ask simple questions. Microsoft 365 Copilot addresses this issue by providing a suggested prompts experience. Suggested prompts present the user with easy prompts to get started.

The screenshot shows the Microsoft Copilot Studio interface. On the left, there's a vertical sidebar with icons for Home, Create, My Content, Feed, Apps, Copilot (which is selected), Admin, Outlook, Teams, Word, and Excel. At the top, it says "Microsoft 365" and has a search bar. The main area features a large blue speech bubble icon and the text "Relecloud Drones". Below this are six suggested prompts arranged in two rows of three:

- Project timeline**: What is the current timeline for the drone delivery project?
- Budget allocation**: How is the budget allocated for the drone delivery project?
- Team roles**: What are the specific roles of team members in the drone delivery project?

- Risk assessment**: How is risk assessed and managed in the drone delivery project?
- Progress update**: Can you provide an update on the progress of the drone delivery project?
- Resource allocation**: How are resources allocated among different aspects of the drone delivery project?

At the bottom of the main area, there's a text input field with "OK, what can I help with? Try one of the examples or enter your own prompt." and a character count of "0 / 2000". To the right of the input field are "View prompts" and "Need help?" buttons.

Copilot Studio automatically generates suggested prompts when you use the conversational experience to create your agent. You can also edit or create your own at any time from the **Suggested prompts** section on the [Overview](#) page.

## Understanding suggested prompts

One way to use the suggested prompts is as a shortcut for simple questions you expect users to ask. However, they're also an opportunity to present more sophisticated prompts that the user might not otherwise understand are possible.

Within suggested prompts, agent authors have full access to the capabilities of the Microsoft 365 Copilot large language model, including the actions and knowledge you added. Consider creating sophisticated prompts that help users achieve common tasks easily.

## Starter prompts

Suggest ways of starting conversations

 Edit

### Project timeline

What is the current timeline for the drone delivery project?

### Budget allocation

How is the budget allocated for the drone delivery project?

### Team responsibilities

What are the specific responsibilities of each team member in the drone delivery project?

### Risk assessment

What risk assessment has been conducted for the drone delivery project?

### Milestone achievements

What milestones have been achieved so far in the drone delivery project?

### Resource allocation

How are resources allocated among different tasks in the drone delivery project?

## Author suggested prompts

1. Go to the agent's [Overview](#) page.
2. Scroll down to **Suggested prompts** and select **Edit**.
3. Add up to six suggested prompts and select **Save**.

## Publish a custom agent to the Teams + Microsoft 365 channel

You can also extend Microsoft 365 Copilot with a custom agent by publishing the agent to the Teams + Microsoft 365 channel. Refer to [Connect and configure an agent for Teams and Microsoft 365](#) for details on how to publish your custom agent to the Teams + Microsoft 365 channel.

## Publishing a Copilot agent

Once the agent is working how you want in the Copilot Studio test experience, you're ready to publish. Publishing converts the agent from a draft state to a published state, ready to upload to the organization's catalog. Unlike a standalone agent, a Copilot agent isn't automatically deployed to Microsoft 365 Copilot and Teams when published.

## Publish your Copilot agent

1. Navigate back to the agent's [Overview](#) page.
2. Choose **Publish** on the top right.

The information on the subsequent form is used to populate the catalog entry in the organization's Office and Teams Catalogs and the Microsoft Admin Center Integrated Apps list. It isn't used by the Microsoft 365 Copilot language model.

3. Fill in the fields as needed and select **Publish**.

When the operation is complete, you're taken to the **Availability options** page.

 **Note**

When you select **Publish**, a bot resource is provisioned in your tenant's Microsoft Entra ID environment, corresponding to your Copilot agent. The resource allows users to interact with the agent in Microsoft Teams.

## Set availability options

The **Availability options** page is used to distribute the agent to the agent's author's account to use or to small groups of peers governed by distribution lists, or to be published to the organization's catalog by an admin.

 **Note**

If an administrator disabled side-loading or publishing for the tenant, one or more of these features might be blocked. Work with the tenant admin to determine the appropriate distribution mechanism for the tenant.

 [Expand table](#)

Availability Option	Description
Share Link	You can copy a <i>deep link</i> into Microsoft Teams to invoke the published agent. Opening the link in the tenant opens a chat experience with the agent.
Show to my teammates and shared users	Lets you grant access to others to participate in authoring the agent, or to security groups to grant them access to use the agent in Microsoft 365 Chat or Teams.
Show to everyone in my org	Submit to the tenant admin to add to the organizational catalog for all tenant users to install as desired.
Download as a .zip	Download as a zip folder. This availability option lets you manually upload to Teams/Microsoft 365 Copilot to submit to the admin to review, or upload directly to the organizational catalog.

More information about managing apps in the Teams catalog can be found here: [Manage Power Platform Apps](#)

## Use the Copilot agent in Microsoft 365 Copilot or Microsoft Teams

The agent should now be available to try out in Microsoft 365 Copilot or Teams. Users can interact with the agent by either @mentioning the agent in the Microsoft 365 Copilot chat interface or by selecting the agent from the sidebar. After a user selects an agent, the user interacts only with the agent until they end that conversation and return to Copilot or Teams.

In the screenshot, the user requests information that's in the knowledge included in the agent. The generative AI, evaluating the agent's instructions, knowledge, and actions, determines that the best way to resolve the request is by using its knowledge to generate an answer. The generative AI also includes references for the sources it uses to generate the answer.

The screenshot shows the 'New action' dialog box. At the top, there is a header 'New action' and a close button 'X'. Below the header, a sub-header says 'Choose the type of item you'd like to add. Once it's been created, it'll be available in the library.' There are four items listed in a grid:

- Connector**: Provide access to external data sources. (Icon: blue gear with a blue line)
- Conversational**: Create focused topics and responses. (Icon: blue head with a gear)
- Prompt**: Describe a custom action to create it. (Icon: blue speech bubble with horizontal lines)
- Flow**: Enable automated task completion. (Icon: blue flowchart with arrows)

At the bottom left of the dialog is a 'Back' button.

## Tips for agent authors

- Microsoft 365 Copilot caches answers from previously asked questions in a given session, even if it called an action to retrieve data. To break this cache, use the **New Chat** button.
- To see more information about why the generative AI does or doesn't select an action, enter *-developer on* into the Microsoft 365 Copilot input box and ask the question.
- The results you see at runtime in Microsoft 365 Copilot might be different from what you see in Copilot Studio's test canvas and Microsoft Teams.
- Unlike custom agents, Copilot agents don't collect data in the Copilot Studio Analytics tab. This feature is being evaluated for future releases.
- When using Microsoft 365 Copilot to test your agent, use the thumbs down feedback experience to provide feedback to the Microsoft Copilot team to review and address issues. Feedback doesn't directly affect the triggering quality of Microsoft 365 Copilot.



# Use Copilot Tuning to fine-tune models for use in Microsoft 365 Copilot (preview)

06/18/2025

This article describes how to use Copilot Tuning to create fine-tuned models in Copilot Studio you can use with declarative agents for Microsoft 365 Copilot. Fine-tuning is a process that lets you customize a pretrained model for a specific task on your own tenant data. You can use these fine-tuned models to build agents that are expert at performing domain-specific tasks and serve them in Microsoft 365 Copilot.

Fine-tuning helps your model perform better on tasks relevant to your organization. A fine-tuned model is especially useful for organizations with unique data or specialized requirements.

This article provides a *basic overview* of the Copilot Tuning process in Copilot Studio. For more detailed task-specific guidance to help you get best results from fine-tuning for your organization and tasks, see [Copilot Tuning overview](#).

## Benefits of Copilot Tuning

Model fine-tuning is a powerful technique used to tailor large language models to your specific needs. Fine-tuning complements other generative AI optimization techniques, such as Retrieval Augmented Generation (also known as RAG) and prompt optimization. Fine-tuning is well-suited when you want to tightly direct the behavior of your model.

Fine-tuning usually requires a team of expert data scientists to curate datasets and build task-specific data preparation and training pipelines.

Copilot Tuning in Copilot Studio greatly simplifies this process, turning it into a tool that just about any subject matter expert can use.

Copilot Studio abstracts away much of the complexity of the process. The Copilot Studio Copilot Tuning process is low-code, transforming fine-tuning from a complex, resource-heavy project into a streamlined, self-service experience.

Automated data preparation powered by AI turns noisy enterprise content into high-quality training sets with minimal effort. This automation minimizes the need to manually label by requesting human input only where model confidence is low. The automation lets you cut down on data labeling effort.

Finally, This feature saves you the effort of creating specialized data processing and training pipelines.

## Security

Copilot Tuning offers enhanced security compared with conventional fine-tuning techniques by ensuring that only users with the right access controls—defined by your existing [Microsoft Entra Security Groups](#)—can use the model when building Microsoft 365 Copilot agents. Admins can also quickly remove models from production, further enhancing security.

Nobody sees your data, not even during training. All training and inference happen in tenant-isolated environments.

## What kind of tasks can Copilot Tuning perform?

Currently, you can use Copilot Tuning for the following tasks:

1. Q&A: Expert question and answer can accurately answer questions in complex knowledge domains such as HR and professional services scenarios where RAG alone would be insufficient.
2. Document generation: Document generation excels in creating complex, structured documents that must follow specific formats, such as agreements, contracts, and technical documentation.
3. Document summarization: Document summarization precisely distills complex information—such as regulatory or legislative analyses—into tailored summaries.

## Eligibility

Copilot Tuning is an Early Access Program (EAP). See [Introducing Microsoft 365 Copilot Tuning](#) for more details on EAP eligibility.

In an organization where Copilot Tuning is available, a Microsoft 365 admin controls access. The admin can activate Copilot Tuning for the organization or tenant level. The admin can also limit access to this feature for specific users in the organization.

## Access Copilot Tuning in Copilot Studio

Once your Microsoft 365 admin makes Copilot Tuning available in your tenant and grants you model-making permissions, you receive an email inviting you to start building your first model with Microsoft Copilot Studio.

To access Copilot Tuning, do the following:

1. Sign in to [Copilot Studio](#) using a user account with the **Model Maker** role.
2. In the left navigation, select the three dots (...) and then select **Copilot Tuning**.

The **Copilot Tuning** page opens.

If you don't see this option, Copilot Tuning isn't available for your tenant or you don't have permissions to create fine-tuned models.

## Create a fine-tuned model

Copilot Tuning is a multi-step training process. As with any machine learning training process, the quality and quantity of training data are critical to the success of the model.

 **Note**

Copilot Tuning currently only supports Sharepoint files and is limited to Word documents, PDFs, and text files.

## Configure basic model parameters

First, configure high level parameters for what you want your model to do, how it should behave, and the appropriate data sources to use.

1. Go to the **Copilot Tuning** page and select **Create a new model**. You're taken to a **Customize your model to your task** page.
2. Enter a meaningful name and a description for your model.

Describe the model in a way that users in your organization can quickly understand how it can help them in their work.

3. Under **Choose knowledge sources**, select **Add knowledge**.

The **Add knowledge to your model** page appears.

- a. Select a knowledge type. Currently, SharePoint is available.
- b. Select a knowledge source. Browse on your computer for a SharePoint file or enter a URL for the source, and then select **Add**.
- c. Repeat the previous step as needed to add more knowledge sources.

- d. When you're done adding knowledge sources, select **Add** to proceed.
4. Under **Permissions**, specify the Microsoft Entra security groups that should have access to the model when it deploys.
- Copilot Tuning automatically excludes from training any files that your selected security groups can't access. Copilot Studio also automatically suggests other security groups to maximize the breadth of knowledge you can securely incorporate in your model.
5. Under **Task type**, select the desired task type.
6. In the **Model Instructions** section that appears, answer the questions as directed. Enter instruction information as directed. For full details, consult the detailed task-specific guidance in the Microsoft 365 Copilot Tuning documentation.

- [Configure Copilot Tuning for expert Q&A](#)
- [Configure Copilot Tuning for document generation](#)
- [Get started with Copilot Tuning summarization](#)

The model instructions help Copilot Studio identify and prepare the most relevant data from your knowledge sources. Good model instructions provide the model with cues for how to interpret data during the training process.

7. Select **Save draft** to save your progress, or, if you're ready to proceed with the fine-tuning process, select **Prepare labeling data**.

Copilot Studio starts preparing the data for labeling.

Copilot Studio informs you if some of your chosen knowledge sources aren't available for the chosen security groups. Copilot Studio automatically suggests other security groups to maximize the breadth of knowledge you can securely incorporate in your model.

8. Make adjustments to the security groups to expand coverage as desired, and then select **Proceed with selection**.

Copilot Studio prepares the data for labeling.

### **Important**

Depending on the size of your data, the preparation can take up to 24 hours to complete. While the preparation is happening, you can continue to work in Copilot Studio or close the browser tab and return later. You receive an email notification once this step is complete. You can check status at any time by returning to Copilot Studio and refreshing the model list.

## Label the training samples

Once your data is processed, Copilot Studio sends an email notification indicating that your data is ready for labeling.

Copilot Studio presents you with generated training examples relevant to the task and the data you provided. You must review the examples and provide feedback on sample quality.

Labeling is a crucial step as it would essentially teaching the model how to identify ideal training examples. Make sure individuals with domain expertise perform this task. If you aren't a domain expert, you can delegate labeling tasks to subject matter experts via a built-in labeling management workflow.

The labeling process generally goes through multiple batches. Training a model can require up to four to five batches of labels.

Once the labeling is complete, you're ready to train your model. Select **Start Training** to continue.

## Train the model

Copilot Studio trains the model using the labeled data. Training is a fully automated process that requires no further input from you.

 **Important**

Depending on the size of your data, the training process can take up to 24 hours.

You receive an email notification once the training is complete. You can also check status at any time by returning to Copilot Studio and refreshing the model list.

## Evaluate the model

In the final phase, you get a set of side-by-side comparisons between what results from the fine-tuned model output versus results from the baseline, non-fine-tuned model. If you want to continue to improve the quality of the model's responses, you can begin a new model training run.

To improve model outputs in your next training run, ensure your dataset is well-aligned to your model's specific task and that your data are labeled by domain experts.

# Publish the model to Microsoft 365 Copilot

Once you're satisfied with the model's output, publish the model to your Microsoft 365 tenant catalog.

Your model is now available for use by your tenant's agents for Copilot.

## (!) Note

Only members of the security groups you selected at the start of the fine-tuning process can use the model in agents.

For more information about how to use the model in agents for Copilot, see the Microsoft 365 Copilot documentation.

## Limitations and restrictions

There are some limitations and restrictions to be aware of when creating fine-tuned models:

- If you add knowledge sources after training the model, you must restart the fine-tuning process from scratch.
- Copilot Studio doesn't yet support model versioning.
- If a user whose data has been used in training a model submits a valid deletion request under GDPR (or similar regulations), you must retrain the model.
- When you fine-tune a model, the model weights are adjusted based on the training data. You can delete the fine-tuned model at any time.
- You're responsible for how data is collected, stored, and used within your tenant environment.
- You must ensure that your data practices meet legal requirements for transparency, consent, access, and deletion.
- You're responsible for verifying the accuracy, appropriateness, and compliance of any outputs generated from this system before using them. Verification might require reviewing with the subject matter experts.

## Related content

- [Copilot Tuning overview](#)
- [Configure Copilot Tuning for expert Q&A](#)
- [Configure Copilot Tuning for document generation](#)
- [Get started with Copilot Tuning summarization](#)

# Use shared tools from the Tools page

09/24/2025

The **Tools** page provides a centralized place for makers to create and manage functionality that can be used across agents within an environment. You can create, manage, and use tools across various agents within an environment, making tools into versatile and reusable components.

## (!) Note

The **Tools** page replaces the previous **Library** page.

You can access the **Tools** page in Microsoft Copilot Studio by selecting **Tools** in the left hand navigation.

When you go to the **Tools** page, you see a list of any tools that are already added. If no tools are added yet, you can add a new tool by selecting **New tool**. Each entry in the list of tools contains basic information about the tool, including name, type, creator, status, and the one or more agents on which the tool is added.

## Tool details page

Select the name entry for one of the tools in the list to see a details page for the specific tool. From the details page, you can:

- Edit the tool
- Add the tool to one or more agents
- Export the tool in a solution

You can choose one of the selectors above the list to filter by type of item (agent, connector, custom connector, prompt). You can also search for items by name.

## Create a new tool

To create a new tool and add it to the **Tools** page:

1. Select **New tool**. You're presented with several options:

- Prompt: Create a new prompt tool using the embedded AI Builder prompt editor.
- Agent flows: Streamline your business processes and automate repetitive tasks. Preconfigured to start with an agent trigger and end with a response back to your

agent.

- Custom Connectors: Bridge external systems that lack prebuilt connectors. Integrate REST or SOAP APIs with platforms like Logic Apps, Power Automate, Power Apps, or Copilot Studio.
- REST APIs: Provide a guided experience for uploading Swagger files to create REST API tools. These are utilized to connect agents with external data sources or services not covered by prebuilt connectors.
- Computer Use: Enable agents to automate tasks in apps and websites by clicking, typing, and navigating the interface. Computer use is ideal for scenarios like data entry, updating records in a web portal, or retrieving information from systems without APIs.
- Model Context Protocol: Connect agents with existing knowledge servers and data sources to improve the use of structured and unstructured data within Copilot Studio.

2. Select one of the options. You proceed to the creation flow for the tool. For some tools, creation occurs within the context of the Tool page, while for others, you are redirected to another entry point for creation.

Once a tool is created, it is listed on the **Tools** page. All tools within the selected environment are listed here if you are the maker of the tool or the maker shared the tool with you.

## Add a tool to an agent

You can add a tool to an agent from the **Tools** page.

### ! Note

When you add a tool to multiple agents, the underlying tool component (prompt, REST API, custom connector, agent flow, and so on) remains the same. Copilot Studio uses a wrapper around the tool. The wrapper lets you specify specific inputs/outputs and configurations specific to the implementation of that tool with the agent.

## Select tool

1. Select the ellipsis (...) next to the name of tool you want to add.

2. Select **Add to agent**.

The next steps depend on the type of tool.

## Add prompt as a tool to an agent

For prompts, you are taken to the tool details page, where you can add the prompt to an agent from the same page.

1. Select **Add**.
2. Select an agent from the list of agents.
3. Select **Add**.

## Add other tool types to an agent

1. Select the agent to which you want to add the tool, and then select **Add**.

You are redirected to the **Add tool** pane for the selected agent to add the tool.

2. Select the specific tool.

Depending on the tool, you may be asked to select or create a connection for the tool.

3. Select **Add to agent**.

## Tools vs Library comparison

The **Tools** page does not cover exactly the same functionality as the previous **Library** page.

Here is some information about where to find information previously available under the **Library** page.

- Copilot agents: To create Copilot agents (also known as declarative agents) for Microsoft 365 Copilot, select **Agents** page > **Microsoft 365 Copilot** > **Agents**. There, you can see all your existing Copilot agents, and also create new ones. You can also create declarative agents directly from Copilot for Microsoft 365 using Copilot Studio lite.
- Prebuilt Power Platform connectors: The **Tools** page is a place where you can create new tools from scratch. Prebuilt connectors don't fall into this category. You can view and add prebuilt connectors from within the context of an agent.
- Component collections: You can view and create collections from within the **Settings** page for an agent. You can also manage existing collections from the **Solutions** page.
- Custom connectors: You can create custom connectors from the **Tools** page. For prebuilt connectors, you can view and add from within the context of an agent.
- Flow: Available within the **Tools** page.
- Prompt: Available within the **Tools** page.
- REST API: Available within the **Tools** page.

# Known limitations

There are some known issues to be aware of on the **Tools** page. Some of these have mitigations you can use.

 Expand table

Issue	Mitigation
Can't delete a tool from the Tools page	If a tool is created and added to an agent, that tool can be removed from the agent by navigating to the agent.
Computer use tools are not listed on the Tools homepage	If a computer use tool is created and added to an agent, it is listed in the <b>Tools</b> tab of that agent.
Prompts created in Power Apps, Power Automate, or within a topic are not visible in Copilot Studio.	Create a new prompt in <b>Tools</b> . This issue is because the prompt artifact is wrapped in a plugin before it is added to an agent. Prompts created in Power Apps, Power Automate, and within the context of a topic are not wrapped in plugins. As a result, these prompts don't show up as an available tool to add on the * <b>Tools</b> page.
When you edit the title of a prompt tool, the new name doesn't show up under <b>Tools</b> .	Edit the tool name under the <b>Tools</b> tab of the agent where it is added.
When you create a REST API tool, it also creates a custom connector.	This is currently as designed. There are plans to combine the rest API and custom connector tool types.

## Related content

 Expand table

Article	Description
<a href="#">Extend Microsoft 365 Copilot with agents</a>	Extend Microsoft 365 Copilot with an agent built in Copilot Studio.
<a href="#">Use prompts to make your agent perform specific tasks</a>	Add prompts in Microsoft Copilot Studio using the embedded AI Builder prompt editor

# Configure a skill for use in Copilot Studio agents

08/08/2025

Copilot Studio lets you extend your agent by using [skills](#). If you already have agents built and deployed in your organization (using the Microsoft 365 Agents SDK and pro-code tools) for specific scenarios, you can convert those bots into a skill and register that skill in a Copilot Studio agent.

## ⓘ Note

To add a skill from within the Copilot Studio app in Microsoft Teams, you must have a [Microsoft Copilot Studio standalone subscription](#).

This article is intended for system administrators or IT professionals who are familiar with [skills](#). After you register a skill with a Copilot Studio agent, you can seamlessly [trigger skill actions in the conversation](#).

## Prerequisites

- [Review restrictions and validation for skills used in Copilot Studio](#)

## ⓘ Important

Copilot Studio agents only support skills built using [Legacy Bot Framework SDK version 4.12.0+ and Microsoft 365 Agents SDK v1.0.0+](#). New skills should be built using Microsoft 365 Agents SDK.

To configure a skill, you should set the associated app registration in Microsoft Entra ID for your skill to single-tenant configuration.

## Flows actions and skills actions

The following table helps determine when to use skills for a conversation.

 [Expand table](#)

	Flow actions	Skill actions
<b>Persona</b>	Agent authors can build reusable Flows to embed into any agent conversation.	Developers can create, deploy, and host custom skills in their own environment.
<b>Conversation</b>	Use flows for simple, single-turn operations. For example, place an order, or get order status.	Use skills for complex, multi-turn operations. For example, schedule a meeting or book a flight.
<b>Response</b>	Use flows to emit an agent response. For example, show a personalized message or inline images.	Use skills to emit any supported agent response. For example, show an adaptive card or send random responses.
<b>Actions</b>	Use flows to trigger server-side single-turn actions. For example, call an HTTP API or trigger a custom connector.	Use skills to trigger server-side and client-side events and actions. For example, navigate to a page upon agent response.

## Configure a skill

First, use Copilot Studio to [create an agent](#) and create and deploy a skill using pro-code tools into your organization. Refer to these samples:

.Net

[CopilotStudio Skill ↗](#)

Before registering the skill, provide the agent's ID to your skills developer to authorize the agent to call actions in the skill. [Learn more about skill allowlist](#). You can get your agent's ID from the **Add a skill** window.

To add a skill to your agent:

1. In the navigation pane, expand **Settings** and select **Skills**.
2. At the top of the **Skills** page, select **Add skill**.
3. Copy your agent ID and provide that to your skills developer.
4. Enter the URL to the skill manifest. A skill's manifest contains the information that your agent needs to trigger actions within a skill.
5. Select **Next** to begin the [validation process](#). Once the validation is successful, your skill is added to your agent. You can now [use this skill in your topics](#).

# Compliance considerations

To protect user privacy, we require skills to be registered as an app in the signed-in user's Microsoft Entra ID tenant.

## Troubleshoot errors during skill registration

A series of validation checks are made against the URL. These checks ensure compliance, governance, and usability of the skill being added to your agent. You must fix these errors before registering a skill.

 Expand table

Error message	Troubleshoot / Mitigation
We ran into problems getting the skill manifest. <code>(MANIFEST_FETCH_FAILED)</code>	Try opening your manifest URL in a web browser. If the URL renders the page within 10 seconds, re-register your skill.
The manifest is incompatible. <code>(MANIFEST_MALFORMED)</code>	<p>(a) Check if the manifest is a valid JSON file. (b) Check if the manifest contains required properties For example, (<code>name</code>, <code>msaAppId</code>, single <code>endpoint</code>, <code>activities/id</code>, <code>activities/description</code>, <code>activities/type</code> (only <code>event</code> or <code>message</code> supported)).</p>
There is a mismatch in your endpoints <code>(MANIFEST_ENDPOINT_ORIGIN_MISMATCH)</code>	Check if your skill endpoint matches your Microsoft Entra ID application registration's <code>Publisher domain</code> (preferred) or <code>Home page URL</code> field. <a href="#">Learn more about setting the home page for endpoints</a> .
To add a skill, it must first be registered <code>(APPID_NOT_IN_TENANT)</code>	Check if your skill's application ID is registered in your organization's Microsoft Entra ID tenant.
The link isn't valid; The link must begin with https:// <code>(URL_MALFORMED, URL_NOT_HTTPS)</code>	Re-enter the link as a secure URL.
The manifest is too large; <code>(MANIFEST_TOO_LARGE)</code>	Check size of the manifest. It must be less than or equal to 500KB.
This skill has already been added to your agent. <code>(MANIFEST_ALREADY_IMPORTED)</code>	Delete the skill and try registering again.
The skill is limited to 100 actions. <code>(LIMITS_TOO_MANY_ACTIONS)</code>	<p>There are too many skill actions defined in skill manifest. Remove actions and try again.</p>

Error message	Troubleshoot / Mitigation
Actions are limited to 25 inputs. ( <code>LIMITS_TOO_MANY_INPUTS</code> )	There are too many skill action input parameters. Remove parameters and try again.
Actions are limited to 25 outputs. ( <code>LIMITS_TOO_MANY_OUTPUTS</code> )	There are too many skill action output parameters. Remove parameters and try again.
Your agent can have a maximum of 100 skills. ( <code>LIMITS_TOO_MANY_SKILLS</code> )	There are too many skills added into an agent. Remove an existing skill and try again.
It looks like something went wrong. ( <code>AADERROR_OTHER</code> )	There was a transient error while validating your skill. Retry.
Something went wrong while checking your skill. ( <code>ENDPOINT_HEALTHCHECK_FAILED</code> , <code>HEALTH_PING_FAILED</code> )	Check if your skill endpoint is online and responding to messages. Also check that the single tenant skill's app registration was deployed to the same tenant.
This skill has not allow-listed your agent ( <code>ENDPOINT_HEALTHCHECK_UNAUTHORIZED</code> )	Check if your agent has been added to the skills allowlist.

# Use skills in Copilot Studio

06/20/2025

Microsoft Copilot Studio lets you extend your agent using skills. If you have already built and deployed bots in your organization (using pro-code tools, including the Microsoft 365 Agents SDK) for specific scenarios, you can convert such a bot to a skill and embed the skill within a Microsoft Copilot Studio agent.

## Prerequisites

- [Review restrictions and validation for skills used in Microsoft Copilot Studio](#).
- [A standalone Microsoft Copilot Studio subscription](#). If you have a [Teams plan](#), you can't use Agents SDK skills.

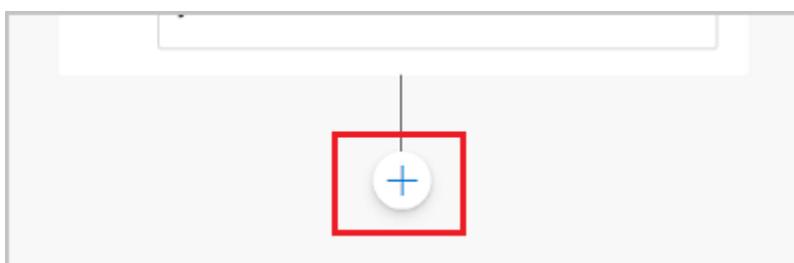
## Add a skill to agent conversations

First, [create a Copilot Studio agent](#) and [create and deploy the skill using pro-code tools](#) into your organization.

Next, [register a skill in Copilot Studio](#).

Add a skill into a conversation:

1. Go to the [Topics page](#) for the agent you want to edit.
2. Open the authoring canvas for the topic you want to call a skill action from.
3. Select **Add node (+)** underneath an existing node to add a new node.



4. In the node selection window, select **Call an action**, and then select the skill you want to add.
5. If your skill has inputs, assign variables to those inputs. Also, if your skill emits outputs to variables, you can use those variables in any nodes that follow.

**Note**

Skills can accept simple data types (integer, Boolean, string) and complex data types (as JSON strings).

If you need to extract variables from complex data types, you must [use Power Automate](#) to further parse and break down complex JSON strings into key value pairs for use in your agent.

6. Select **Save** to persist your changes to the topic.

7. You should [test your copilot](#) to make sure it's using the skill properly.

## Convert an existing skill from multitenant to single-tenant

You can convert existing skills from multitenant support to single-tenant support. To convert a multitenant skill to a single-tenant skill, you need to perform the following changes:

- Create a new single-tenant Entra ID app registration
- Update the skill configuration to use single-tenant
- Deploy the skill
- (Optional) update the source code

For more information, see [Implement a skill for Copilot Studio](#).

## Single-tenant skills and Conditional Access policies

The admins for the Entra ID tenant can configure Conditional Access policies to control access to resources in the tenant. This can potentially affect requests to resources, including skills, based on factors such as geography. Make sure to review the [Conditional Access policies](#) in your tenant.

# Use a classic chatbot as a skill in a Bot Service bot

Article • 05/23/2025

## (!) Note

This article applies to classic chatbots only. This feature isn't available in the new Copilot Studio experience.

When you use a classic chatbot as a [skill](#) with a Bot Service bot, the Bot Service bot determines if anything the user says matches the classic chatbot's [trigger phrases](#). If there's a match, the Bot Service bot can pass the conversation to the classic chatbot. The classic chatbot extracts any [entities](#) and triggers the matching topic.

## (i) Important

You must have a [trial](#) or full Copilot Studio license to use classic chatbots as skills. This capability isn't available with the [Teams Copilot Studio license](#).

To set up your classic chatbot as a skill:

1. [Add the Bot Service bot to the allowlist for the classic chatbot](#).
2. [Download the skill manifest for the classic chatbot](#).
3. [Connect to a Copilot Studio skill from your bot](#).
4. [Confirm the bot works as a skill](#).

## Prerequisites

- [Understand how Bot service skills work](#).
- [Get your Bot Service bot's app ID](#).

## Add your Bot Service bot to the allowlist for your classic chatbot

Add the Bot Service bot's app ID to the allowlist for the classic chatbot that you want to use as a skill. The bots must be in the same tenant.

1. Get the Microsoft app ID for your Bot Services bot. You can find this information in the Azure portal or in the bot configurations.
2. In Copilot Studio, open the classic chatbot you want to use as a skill.
3. In the navigation menu, go to **Settings** and select **Security**. Then select **Allowlist**.
4. Select **Add allowed caller**.
5. Paste your Bot Service bot's [app ID](#) and select **Next**.

 **Caution**

A classic chatbot can't act as a skill for other classic chatbots. If you try to add an app ID that belongs to a Power Virtual Agent bot to the allowlist, you get an error. You can only add an app ID for a Bot bot.

Copilot Studio validates the Bot Service bot's app ID and confirms that it belongs to a bot in the same tenant.

6. (Optional) Add a **Display name** for the bot you added to the allowlist.

7. Select **Save**.

The Bot Service bot appears by its display name if you entered one, or by its app ID if you didn't enter a display name. To delete or edit it at any time, select the icon to the right of the bot's display name or app ID.

 **Note**

Bot Service bots added to the allowlist aren't [exported as part of the bot content](#).

## Download the classic chatbot skill manifest for your classic chatbot

Bot Service bots can use a Copilot Studio skill manifest to configure a connection to the classic chatbot that produced the manifest.

All classic chatbots have skill manifests. A skill manifest is a JSON file that includes the skill's name, interface, and trigger phrases.

A Bot Service bot can use a skill manifest to know when to trigger the classic chatbot (for example, in response to something a user says to it).

Copilot Studio skill manifests follow [version 2.2 of the Bot Service skill manifest schema](#) and consist of *intents.lu* and *manifest.json* files.

A Bot Service bot might decide that a classic chatbot should handle the user's request, based on the manifest data. It passes everything the user said to the classic chatbot. Then, the classic chatbot matches what the user said [to a Copilot Studio topic](#), extracts any [entities that are needed for slot filling](#), and triggers the Copilot Studio topic.

Copilot Studio skill manifests are automatically generated and updated. A classic chatbot has two skill manifests:

- **Test manifest:** Allows the Bot Service bot to connect to the test version of your classic chatbot. Use the test manifest to validate changes to your skill before you publish it.
  - The test manifest is immediately available for every newly created classic chatbot.
  - It's automatically updated to reflect changes every time you [save](#) your classic chatbot.
- **Published manifest:** Allows the Bot Service bot to connect to the published version of your classic chatbot.
  - The published manifest is only available for classic chatbots that were published at least once.
  - It's automatically updated to reflect changes every time you [publish your agent](#).

 **Note**

The **Published manifest** isn't available for classic chatbots that were never published.

To generate your bot's **Published manifest**, [publish your classic chatbot](#).

Both skill manifests are shown on the **Manage allowlist** panel.

The skill manifests are also shown on the copilot's Details page. In the navigation menu, select **Settings**, and then select **Details**. The Details page shows other metadata, including **Environment ID**, **Tenant ID**, and **Bot app ID**.

To download a manifest, select it. It downloads as a .zip file labeled \_<bot name>*manifest* for the **Published manifest**, or <bot name>*test\_manifest* for the **Test manifest**.

 **Note**

The skill manifest doesn't include system topics. Only topics that were created by copilot authors and intent triggers that authors add to the bot are included.

## Connect to a Copilot Studio skill from your bot

You can use the information from your Copilot Studio skill manifest to create a skill connection in your Bot Service bot.

To learn more about the structure of the skill manifest, see [Skills manifest](#).

The activities section of the skill manifest contains the topics that are available in your classic chatbot.

You can [implement a skill consumer](#) in your Bot Service bot to call a classic chatbot as a skill.

## Test your classic chatbot as a skill with a Bot Service bot

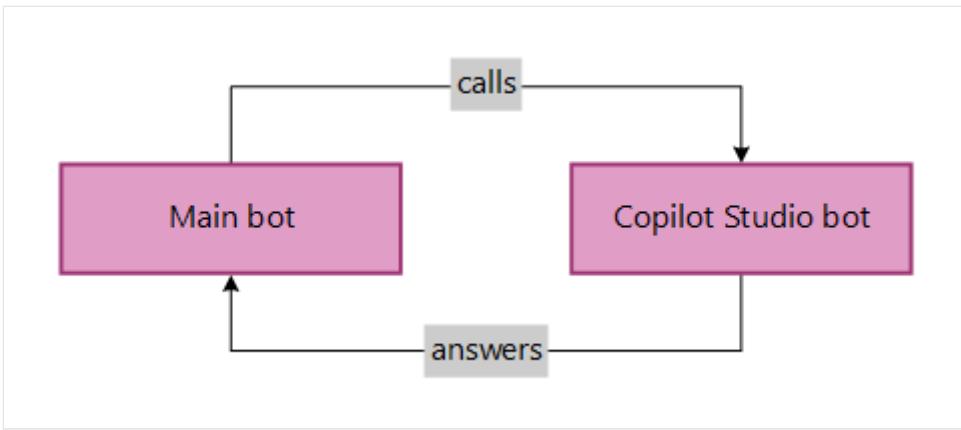
You can the [Bot Service Emulator](#) to test that your Bot Service bot is properly calling a classic chatbot as a skill.

### Important

Your Bot Service bot must be added to the classic chatbot's allowlist for the skill connection to work.

## Update a multitenant Bot Service bot to a single-tenant bot

The following illustration shows the scenario where a Bot Service bot calls a classic chatbot, to use it as a skill.



Both the Bot Service bot and the classic chatbot have a Microsoft Entra ID app registration, which is referenced by the Azure Bot instances. These Microsoft Entra ID app registrations were created as multitenant. The Bot Service bot requests a token from the [Bot Service](#) tenant, and uses this token to authenticate to Copilot Studio using Azure Bot Service. Copilot Studio uses the same mechanism to contact the Bot Service bot.

The Copilot Studio chatbot's application registration is now created as a single-tenant, so this communication no longer works. Microsoft Entra ID refuses to create tokens for single-tenant application registrations that aren't in the same tenant.

Additionally, Copilot Studio validates the incoming token and rejects requests from tenants it doesn't expect, such as the tenant ID in which the classic chatbot's application registration was created.

To resolve this issue, there are two possible fixes.

## Classic chatbots in the same tenant as the Bot Service bot

In scenarios where the Copilot Studio chatbot is already deployed in the same tenant as the Bot Service bot and is only used by this bot, update the bot to a single-tenant configuration, then redeploy your Bot Service bot. For more information on how to configure a Bot Service bot for single-tenant, see [Provision and publish a bot](#).

## Copilot Studio chatbot that can't be updated to single-tenant

In scenarios where the Copilot Studio chatbot is already deployed in the same tenant as the Bot Service bot, but the bot can't be updated to single-tenant, you need to update the code in the Bot Service bot to create the token to communicate with the Copilot Studio chatbot. The token needs to be created for the tenant in which the Copilot Studio chatbot was also created. In addition, the response must be accepted, which probably requires that you follow the steps in [Convert an existing skill from multitenant to single-tenant](#).

## Learn More

- [Skills overview](#)
- [Skills manifest](#)
- [Test a skill in Emulator](#)

# Key concepts - Publish and deploy your agent

09/06/2025

With Copilot Studio, you can publish agents to engage with your customers on multiple platforms or channels, such as live websites, mobile apps, Microsoft 365 Copilot or messaging platforms like Teams and Facebook.

Each time you update your agent, you can publish it again from within Copilot Studio. Publishing your agent applies to all the channels associated with your agent.

## Web app

You need to publish your agent before your customers can engage with it. You can publish your agent on multiple platforms, or *channels*.

After you publish your agent to at least one channel, you can connect it to more channels. Remember to publish your agent again after you make any changes to it.

When you publish your agent, this agent updates on all connected channels. If you make changes to your agent but don't publish after doing so, your customers won't be engaging with the latest content.

The agent comes with the **Authenticate with Microsoft** option turned on. The agent automatically uses Microsoft Entra ID authentication for Teams, Power Apps, and Microsoft 365 Copilot without requiring any manual setup.

If you want to allow anyone to chat with your agent, select **No authentication**.

### ✖ Caution

Selecting the **No authentication** option allows anyone who has the link to chat and interact with your bot or agent.

We recommend you apply authentication, especially if you are using your bot or agent within your organization or for specific users, along with [other security and governance controls](#).

If you want to use other channels and still have authentication for your agent, select **Authenticate manually**.

### **Important**

If you select **No authentication**, your agent can't use [Agent actions](#) with [user credentials](#).

## Publish the latest content

1. With your agent open for editing, in the navigation menu, select **Publish**.
2. Select **Publish**, and then confirm. Publishing can take a few minutes.

## Test your agent

Test your agent after you publish. You can [make the agent available to users in Teams and Microsoft 365 Copilot](#) with the installation link or from various places in the Microsoft Teams app store.

You can share your agent later by selecting **Make the agent available to others** from the **Publish** page, in Teams.

You can also install the agent for your own use in Microsoft Teams by selecting **Open the agent in Teams**.

If you selected **No authentication** or **Authenticate manually**, select the **Demo website** link to open a prebuilt website in a new browser tab, where you and your teammates can interact with the agent.

The demo website is also useful to gather feedback from stakeholders before you roll your agent out to customers. Learn how to [configure the demo website and add the agent to your live website](#).

### **Tip**

#### **What's the difference between the test chat and the demo website?**

Use the test chat (the **Test agent** pane) while you're building your agent to make sure conversation flows as you expect and to spot errors.

Share the demo website URL with members of your team or other stakeholders to try out the agent. The demo website isn't intended for production use. You shouldn't share the URL with customers.

# Configure channels

After publishing your agent at least once, you can add channels to make it reachable by your customers.

To configure channels for your agent:

1. On the top menu bar, select **Channels**.
2. Select the desired channel from the list of available channels.

The connection steps are different for each channel. For more information, see the article for the desired channels, in the following list:

- [Teams and Microsoft 365 Copilot](#)
- [SharePoint](#)
- [WhatsApp](#)
- [Demo Website](#)
- [Custom Website](#)
- [Mobile App](#)
- [Facebook](#)
- [Azure Bot Service channels](#), including:
  - Cortana
  - Slack
  - Telegram
  - Twilio
  - Line
  - Kik
  - GroupMe
  - Direct Line Speech
  - Email

## Channel experience reference table

Different channels have different user experiences. The following table shows a high-level overview of the experiences for each channel. Take the channel experiences into account when optimizing your agent content for specific channels.

 [Expand table](#)

<b>Experience</b>	<b>Website</b>	<b>Teams and Microsoft 365 Copilot</b>	<b>Facebook</b>	<b>Dynamics Omnichannel for Customer Service</b>
Customer satisfaction survey	Adaptive card	Text-only	Text-only	Text-only
Multiple-choice options	Supported	Supported up to six (as hero card)	Supported up to 13 ↗	Partially Supported
Markdown ↗	Supported	Partially Supported	Partially supported ↗	Partially Supported
Welcome message	Supported	Supported	Not supported	Supported for <a href="#">Chat</a> . Not supported for other channels.
Did-You-Mean	Supported	Supported	Supported	Supported for <a href="#">Microsoft Teams</a> , <a href="#">Chat</a> , Facebook, and text-only channels (short message service (SMS) via <a href="#">TeleSign</a> and <a href="#">Twilio</a> , <a href="#">WhatsApp</a> , <a href="#">WeChat</a> , and <a href="#">Twitter</a> ). Suggested actions are presented as a text-only list; users must retype an option to respond.

## Next steps (Web app)

[+] [Expand table](#)

<b>Article</b>	<b>Description</b>
<a href="#">Publish an agent to a live or demo website</a>	Publish your agent on your live website, or use a demo website to share internally.
<a href="#">Connect and configure an agent for Teams and Microsoft 365 Copilot</a>	Use Teams and Microsoft 365 Copilot to distribute your agent.
<a href="#">Publish an agent to Facebook</a>	Add your agent to Facebook Messenger.
<a href="#">Publish an agent to mobile or custom apps</a>	Add your agent to mobile or custom native apps (developer coding required).
<a href="#">Publish an agent to Azure Bot Service channels</a>	Add your agent to Azure Bot Service channels (developer coding required).

# Publish an agent to a live or demo website

08/16/2025

When publishing the agent to the web, you can publish to a prebuilt demo website (which you can use to share the agent with your teammates and stakeholders). You can also publish to your own live website.

## 💡 Tip

### When should I use the demo website and when should I use my own website?

You should use the demo website only to try out the agent and share it with your teammates or other stakeholders who want to try out the agent. It's not intended for production uses, for example, you shouldn't use it directly with customers.

You should publish and use the agent on your live website for production scenarios, such as a help agent on your help webpage for customers to interact with.

Preview

## ⓘ Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

A prebuilt demo website is automatically created for you when you publish your agent. Your teammates and stakeholders can use the demo website to try out the agent while you're building it. You might want to customize the demo site to guide your teammates' testing efforts. Publish the agent to your live website when it's ready for customers to use.

## When should you use the demo website and when should you use your own website?

You should use the demo website only to try out your agent, or to share it with teammates and stakeholders who want to try it out. It's not intended for production use. You shouldn't

give the URL to customers. Publish the agent on your live website for customers to use.

## Customize the demo website

Edit the demo website's welcome message and trigger phrases as often as you like. The welcome message can explain the purpose of the testing and prompt your teammates for what to ask the agent.

1. Open your agent and on the top menu bar, select **Settings**.
2. On the side navigation pane, select **Security**.
3. Select **Authentication** and then **No authentication**, and then select **Save**.
4. Exit **Settings** and on the top menu bar, select **Channels**.
5. Under **Channels**, select **Demo website**.
6. On the **Demo Website** pane, under **Welcome message**, enter the message you want your teammates to see.
7. Under **Conversation starters**, enter a list of trigger phrases.  
Trigger phrases are what initiate specific topics, so you could enter trigger phrases for topics that you want your teammates to test.
8. Select **Save**.
9. Copy the demo website URL and share it with your teammates.

## Add your agent to your website

When your agent is ready for customers, add it to your website as an iframe code snippet. Your live website can be external or internal, like a SharePoint site. You can also add the agent to your Power Platform admin center.

1. Copy the following HTML code and save it to a file named `index.html`. Alternatively, copy and paste the code into the [w3schools.com HTML try it editor](#).

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
<meta name="description" content="Contoso Web Chat Assistant" />
<title>Contoso Sample Web Chat Test</title>
<script src="https://cdn.botframework.com/botframework-
webchat/latest/webchat.js"></script>
<script>
 let webChatInstance = null;
 let directLineUrl = null;

 // Replace with your token endpoint
 const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";
 const styleOptions =
 {"accent": "#0078D4", "autoScrollSnapOnPage": true, "autoScrollSnapOnPageOffset": 0, "avatarBorderRadius": "7%", "avatarSize": 31, "backgroundColor": "#e8e9eb", "botAvatarBackgroundColor": "#ffffff00", "botAvatarImage": "https://powercatexternal.blob.core.windows.net/creatorkit/Assets/ChatbotLogoBlue.png", "botAvatarInitials": "B", "bubbleAttachmentMaxWidth": 480, "bubbleAttachmentMinWidth": 250, "bubbleBackground": "#f0eded", "bubbleBorderColor": "#f5f5f5", "bubbleBorderRadius": 41, "bubbleBorderStyle": "solid", "bubbleBorderWidth": 1, "bubbleFromUserBackground": "#ebffff", "bubbleFromUserBorderColor": "#f5f5f5", "bubbleFromUserBorderRadius": 41, "bubbleFromUserBorderStyle": "solid", "bubbleFromUserBorderWidth": 1, "bubbleFromUserNubOffset": 0, "bubbleFromUserNubSize": 0, "bubbleFromUserTextColor": "#242424", "bubbleImageHeight": 10, "bubbleImageMaxHeight": 240, "bubbleImageMinHeight": 240, "bubbleMessageMaxWidth": 480, "bubbleMessageMinWidth": 120, "bubbleMinHeight": 50, "bubbleNubOffset": 0, "bubbleTextColor": "#242424", "emojiSet": true, "fontSizeSmall": "70%", "hideUploadButton": false, "messageActivityWordBreak": "break-word", "monospaceFont": "Consolas", "paddingRegular": 10, "paddingWide": 10, "primaryFont": null, "sendBoxBackground": "#e8e9eb", "sendBoxBorderTop": "solid 1px
#808080", "sendBoxButtonColor": "#0078d4", "sendBoxButtonColorOnHover": "#006cbe", "sendBoxButtonShadeBorderRadius": 40, "sendBoxButtonShadeColorOnHover": "", "sendBoxHeight": 60, "sendBoxPlaceholderColor": "#171616", "sendBoxTextColor": "#2e2d2d", "showAvatarInGroup": "status", "spinnerAnimationHeight": 16, "spinnerAnimationPadding": 12, "spinnerAnimationWidth": 16, "subtleColor": "#00000FF", "suggestedActionBackgroundColor": "#006FC4FF", "suggestedActionBackgroundColorOnHover": "#0078D4", "suggestedActionBorderColor": "", "suggestedActionBorderRadius": 10, "suggestedActionBorderWidth": 1, "suggestedActionLayout": "flow", "suggestedActionTextColor": "#FFFFFF", "typingAnimationBackgroundImage": "url('https://wpamelia.com/wp-content/uploads/2018/11/ezgif-2-6d0b072c3d3f.gif')", "typingAnimationDuration": 5000, "typingAnimationHeight": 20, "typingAnimationWidth": 64, "userAvatarBackgroundColor": "#222222", "userAvatarImage": "https://avatars.githubusercontent.com/u/8174072?v=4&size=64", "userAvatarInitials": "U"};
 const backgroundImage = "";
 document.addEventListener('DOMContentLoaded', () => {
 const root = document.documentElement;
 root.style.setProperty('--primary-color', createGradient(styleOptions.accent));
 root.style.setProperty('--header-textColor', styleOptions.suggestedActionTextColor);
 if (backgroundImage) {
 const webchatElement = document.getElementById('webchat');
 webchatElement.style.backgroundImage = `url(${backgroundImage})`;
 }
 });
</script>
```

```

 webchatElement.style.backgroundSize = 'cover';
 webchatElement.style.backgroundPosition = 'center';
 webchatElement.style.backgroundRepeat = 'no-repeat';
 const overlay = document.createElement('div');
 overlay.className = 'webchat-overlay';
 webchatElement.appendChild(overlay);
 }
});

function createGradient(baseColor) {
 const r = parseInt(baseColor.slice(1,3), 16);
 const g = parseInt(baseColor.slice(3,5), 16);
 const b = parseInt(baseColor.slice(5,7), 16);
 const lighterColor = `#${Math.min(255,
r+30).toString(16).padStart(2, '0')}#${Math.min(255,
g+30).toString(16).padStart(2, '0')}#${Math.min(255,
b+30).toString(16).padStart(2, '0')}`;
 const darkerColor = `#${Math.max(0, r-
30).toString(16).padStart(2, '0')}#${Math.max(0, g-
30).toString(16).padStart(2, '0')}#${Math.max(0, b-
30).toString(16).padStart(2, '0')}`;
 return `linear-gradient(135deg, ${lighterColor}, ${baseColor},
${darkerColor})`;
}

const environmentEndPoint = tokenEndpoint.slice(
 0,
 tokenEndpoint.indexOf("/powervirtualagents")
);
const apiVersion = tokenEndpoint
 .slice(tokenEndpoint.indexOf("api-version"))
 .split("=")[1];
const regionalChannelSettingsURL =
`${environmentEndPoint}/powervirtualagents/regionalchannelsettings?api-
version=${apiVersion}`;
function showChat() {
 const popup = document.getElementById("chatbot-popup");
 const openButton = document.getElementById("open-chat");
 popup.classList.add("visible");
 openButton.classList.add("hidden");
}
function hideChat() {
 const popup = document.getElementById("chatbot-popup");
 const openButton = document.getElementById("open-chat");
 popup.classList.remove("visible");
 openButton.classList.remove("hidden");
}
function createCustomStore() {
 return window.WebChat.createStore(
 {},
 ({ dispatch }) =>
 (next) =>
 (action) => {
 if (action.type === "DIRECT_LINE/CONNECT_FULFILLED") {
 dispatch({
 type: "DIRECT_LINE/POST_ACTIVITY",
 meta: { method: "keyboard" },
 });
 }
 }
)
}

```

```
 payload: {
 activity: {
 channelData: { postBack: true },
 name: "startConversation",
 type: "event",
 },
 },
 });
}
return next(action);
}
);
}
async function restartConversation() {
try {
if (!directLineUrl) {
console.error("DirectLine URL not initialized");
return;
}
const response = await fetch(tokenEndpoint);
const conversationInfo = await response.json();
if (!conversationInfo.token) {
throw new Error("Failed to get conversation token");
}
const newDirectLine = window.WebChat.createDirectLine({
domain: `${directLineUrl}v3/directline`,
token: conversationInfo.token,
});
const webchatElement = document.getElementById("webchat");
webChatInstance = window.WebChat.renderWebChat(
{
 directLine: newDirectLine,
 styleOptions,
 store: createCustomStore(),
},
webchatElement
);
} catch (err) {
console.error("Failed to restart conversation:", err);
}
}
async function initializeChat() {
try {
const response = await fetch(regionalChannelSettingsURL);
const data = await response.json();
directLineUrl = data.channelUrlsById.directline;
if (!directLineUrl) {
throw new Error("Failed to get DirectLine URL");
}
const conversationResponse = await fetch(tokenEndpoint);
const conversationInfo = await conversationResponse.json();
if (!conversationInfo.token) {
throw new Error("Failed to get conversation token");
}
const directLine = window.WebChat.createDirectLine({
```

```
 domain: `${directLineUrl}v3/directline`,
 token: conversationInfo.token,
 });
 webChatInstance = window.WebChat.renderWebChat(
 {
 directLine,
 styleOptions,
 store: createCustomStore(),
 },
 document.getElementById("webchat")
);
} catch (err) {
 console.error("Failed to initialize chat:", err);
}
}

initializeChat();
</script>
<style>
:root {
 --primary-gradient: var(--primary-color);
 --chat-width: 450px;
 --chat-height: 520px;
 --header-height: 56px;
 --border-radius: 16px;
 --transition-speed: 0.3s;
}
* {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
 Roboto, sans-serif;
}
body {
 min-height: 100vh;
 background-color: #f3f4f6;
}
#chatbot-popup {
 display: none;
 position: fixed;
 bottom: 32px;
 right: 32px;
 width: var(--chat-width);
 height: var(--chat-height);
 background: white;
 border-radius: var(--border-radius);
 box-shadow: 0 18px 40px -5px rgba(0, 0, 0, 0.2),
 0 15px 20px -5px rgba(0, 0, 0, 0.1);
 overflow: hidden;
 opacity: 0;
 transform-origin: bottom right;
 transform: scale(0.95);
 transition: all var(--transition-speed) ease-in-out;
 z-index: 999;
```

```
}

#chatbot-popup.visible {
 display: block;
 opacity: 1;
 transform: scale(1);
}

#chatbot-header {
 background: var(--primary-color);
 padding: 16px 20px;
 height: var(--header-height);
 display: flex;
 justify-content: space-between;
 align-items: center;
 color: var(--header-textColor);
}

.header-title {
 display: flex;
 align-items: center;
 gap: 12px;
 font-size: 16px;
 font-weight: 500;
}

.header-buttons {
 display: flex;
 gap: 12px;
 align-items: center;
}

.icon-button {
 background: none;
 border: none;
 color: var(--header-textColor);
 cursor: pointer;
 padding: 8px;
 border-radius: 8px;
 display: flex;
 align-items: center;
 justify-content: center;
 transition: all 0.2s ease;
}

.icon-button:hover {
 color: var(--header-textColor);
 background: rgba(255, 255, 255, 0.1);
}

.icon-button:focus {
 outline: 2px solid rgba(255, 255, 255, 0.5);
 outline-offset: 2px;
}

#webchat {
 height: calc(100% - var(--header-height));
 background-color: #f9fafb;
 position: relative;
}

.webchat-overlay {
 position: absolute;
 top: 0;
```

```
 left: 0;
 right: 0;
 bottom: 0;
 background: rgba(255, 255, 255, 0.85);
 pointer-events: none;
 z-index: 1;
}
#webchat > div {
 position: relative;
 z-index: 2;
}
#webchat .webchat__basic-transcript__content {
 white-space: pre-wrap !important;
 word-break: break-word !important;
}
#webchat .webchat__bubble__content {
 padding: 8px 12px !important;
}
#webchat .webchat__bubble {
 max-width: 85% !important;
 margin: 8px !important;
}
#webchat .webchat__basic-transcript__content ul,
#webchat .webchat__basic-transcript__content ol,
#webchat .webchat__bubble__content ul,
#webchat .webchat__bubble__content ol {
 padding-left: 24px !important;
 margin: 8px 0 !important;
 list-style-position: outside !important;
}
#webchat .webchat__basic-transcript__content li,
#webchat .webchat__bubble__content li {
 margin: 4px 0 !important;
 padding-left: 4px !important;
}
#open-chat {
 position: fixed;
 bottom: 32px;
 right: 32px;
 width: 64px;
 height: 64px;
 border-radius: 50%;
 background: var(--primary-gradient);
 border: none;
 cursor: pointer;
 display: flex;
 align-items: center;
 justify-content: center;
 box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1);
 transition: all var(--transition-speed) ease-in-out;
 z-index: 998;
}
#open-chat.hidden {
 opacity: 0;
 transform: scale(0.95) translateY(10px);
```

```
 pointer-events: none;
}
#open-chat:hover {
 transform: translateY(-4px);
 box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1);
}
#open-chat:focus {
 outline: 3px solid rgba(79, 70, 229, 0.5);
 outline-offset: 2px;
}
#open-chat svg {
 width: 28px;
 height: 28px;
 color: white;
 transition: transform 0.2s ease;
}
.main-content {
 max-width: 1200px;
 margin: 0 auto;
 padding: 48px 24px;
}
.main-content h1 {
 font-size: 36px;
 color: #111827;
 text-align: center;
}
.main-content p {
 font-size: 18px;
 color: #4b5563;
 line-height: 1.6;
 margin-bottom: 48px;
 text-align: center;
 max-width: 800px;
 margin-left: auto;
 margin-right: auto;
}
.content-grid {
 display: grid;
 grid-template-columns: repeat(2, 1fr);
 gap: 32px;
 margin-bottom: 32px;
}
.content-box {
 background: linear-gradient(135deg, #e6e6e6, #c4c4c4, #9f9f9f);
 padding: 32px;
 border-radius: 12px;
 box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1),
 0 2px 4px -1px rgba(0, 0, 0, 0.06);
 min-height: 300px;
 display: flex;
 flex-direction: column;
 justify-content: center;
 align-items: center;
 text-align: center;
 position: relative;
```

```
 overflow: hidden;
 }
 .content-box::before {
 content: '';
 position: absolute;
 top: 0;
 left: 0;
 right: 0;
 height: 4px;
 }
 .content-box.featured {
 grid-column: span 2;
 min-height: 350px;
 background: linear-gradient(135deg, #e6e6e6, #c4c4c4, #9f9f9f);
 color: #000000;
 }
 .content-box h2 {
 font-size: 24px;
 margin-bottom: 16px;
 position: relative;
 }
 .content-box p {
 font-size: 16px;
 color: #6b7280;
 margin-bottom: 0;
 }
 .content-box.featured p {
 color: #000000;
 }
 @media (max-width: 768px) {
 .content-grid {
 grid-template-columns: 1fr;
 }
 .content-box.featured {
 grid-column: span 1;
 }
 .main-content {
 padding: 24px 16px;
 }
 .main-content h1 {
 font-size: 28px;
 }
 #chatbot-popup {
 width: 100%;
 height: 100%;
 bottom: 0;
 right: 0;
 border-radius: 0;
 }
 }

```

</style>

</head>

<body>

<div class="main-content">

<h1>Header</h1>

```
<p>Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisque
faucibus ex sapien vitae pellentesque sem placerat.</p>
<div class="content-grid">
 <div class="content-box featured">
 <h2>Featured Content</h2>
 <p>Primary content area with custom styling and gradient
background</p>
 </div>
 <div class="content-box">
 <h2>Section One</h2>
 <p>Content box with minimal design</p>
 </div>
 <div class="content-box">
 <h2>Section Two</h2>
 <p>Another content section with consistent styling</p>
 </div>
 </div>
<div id="chatbot-popup" role="complementary" aria-label="Chat
Assistant">
 <div id="chatbot-header">
 <div class="header-title">
 <svg
 class="chat-icon"
 width="24"
 height="24"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M21 15a2 2 0 0 1-2 2H7l-4 4V5a2 2 0 0 1 2-2h14a2 2 0 0 1 2
2z"
 ></path>
 </svg>
 Contoso Assistant
 </div>
 <div class="header-buttons">
 <button
 class="icon-button"
 id="restart-button"
 onclick="restartConversation()"
 aria-label="Restart Conversation"
 >
 <svg
 width="20"
 height="20"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 ></svg>
 </div>
</div>
```

```
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
 >
 <path
 d="M3 12a9 9 0 1 0 9-9 9.75 9.75 0 0 0-6.74 2.74L3 8"
 ></path>
 <path d="M3 3v5h5"></path>
</svg>
</button>
<button
 class="icon-button"
 id="close-button"
 onclick="hideChat()"
 aria-label="Close Chat"
>
<svg
 width="20"
 height="20"
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
>
 <line x1="18" y1="6" x2="6" y2="18"></line>
 <line x1="6" y1="6" x2="18" y2="18"></line>
</svg>
</button>
</div>
</div>
<div id="webchat" role="main"></div>
</div>
<button
 id="open-chat"
 onclick="showChat()"
 aria-label="Open Chat Assistant"
>
<svg
 viewBox="0 0 24 24"
 fill="none"
 stroke="currentColor"
 stroke-width="2"
 stroke-linecap="round"
 stroke-linejoin="round"
 aria-hidden="true"
>
 <path
 d="M21 15a2 2 0 0 1-2 2H71-4 4V5a2 2 0 0 1 2-2h14a2 2 0 0 1 2 2z"
 ></path>
</svg>
</button>
</body>
```

```
</html>
```

2. Retrieve the token endpoint for your agent.
3. In `index.html`, at the line `const tokenEndpoint = "<YOUR TOKEN ENDPOINT>";`, replace the placeholder with the token endpoint for your agent.
4. Open `index.html` using a modern browser (for example, Microsoft Edge) to open the agent in the custom canvas.
5. Test the agent to ensure you're receiving responses from it and that it's working correctly.
6. Provide the snippet (`index.html`) to your web developer to add the agent to your website.

# Publish an agent to Power Pages

Article • 11/19/2024

Power Pages are external-facing websites that allow users outside their organizations to sign in with a wide variety of identities, create and view data in Microsoft Dataverse, or even browse content anonymously. You can add an agent to a website so that visitors can interact with a chatbot directly on your site's page.

## Prerequisites

- [Create a site with Power Pages.](#)
- [Create and delete agents.](#)

## Add an agent to your Power Pages

Power Pages [design studio](#) is a no-code experience used to create and customize Power Pages. It contains various options to add and configure webpages, components, forms, and lists.

To learn how to add an agent with design studio, see [Add an agent to your Power Pages site](#).

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Connect and configure an agent for Teams and Microsoft 365

Article • 05/19/2025

With your agent published, you can make your agent available to users in Teams and Microsoft 365. This way you, your teammates, and your broader organization can interact with it. You must [publish the agent at least once](#) before users can interact with the agent in Teams and Microsoft 365.

You can:

- Customize your agent's appearance in Teams and Microsoft 365 Copilot
- Install the agent for yourself in Teams and Microsoft 365 Copilot
- Share the agent's installation link with other users
- Show the agent in the Teams app store.
  - Show the agent to [shared users](#) by adding it to the Teams app store **Built with Power Platform** section.
  - Share the agent with the organization by submitting the agent for admin approval to be featured in the **Built for your org** section of the Teams app store.
- Add the agent to a team channel.
- Download the app manifest for your agent to distribute it within your Teams tenant.

When you add an agent to Teams, some of your data, such as agent content and user chat content, is shared with Teams. Your data might flow outside of your [organization's compliance and geographic or regional boundaries](#). For more information, see [Understand the permissions of and the information accessed by Teams apps](#).

## Prerequisites

- All agent users in your organization can use Microsoft 365 Copilot.
- Agent makers must have permissions to create and access a Copilot Studio environment.
- [Share your agent with other users](#) so they can install and chat with the agent in Teams and Microsoft 365 Copilot.
- Your organization must allow Microsoft Power Platform apps to be added to Teams before you or other users can directly install an agent in Teams or find it in the **Built with Power Platform** section in the Teams app store. Work with your admin to [Manage Microsoft Power Platform apps in the Microsoft Teams admin center](#) if it's disabled for your organization.
- Agents that appear in the **Built for your org** section (approved by an admin) are managed separately by admins in the Teams admin center's [Manage apps page](#).

- Before installing an agent for yourself or sharing it with others, you should [publish the agent](#) at least once.

## Connect an agent to the Teams and Microsoft 365 Copilot channels

After publishing your agent at least once, you can connect it to the **Teams and Microsoft 365 Copilot** channels to allow users to chat with it in both Teams and Microsoft 365 Copilot.

### Note

We recommend you [turn on security settings](#) for agents built for Teams, Microsoft 365 Copilot, or internal employee use, to prevent individuals outside of your organization from using these agents.

1. [Open the configuration panel for the Teams and Microsoft 365 Copilot channels](#).
2. If you want your agent to be available in both Microsoft 365 Copilot and Teams, under **Turn on Microsoft 365**, make sure **Make agent available in Microsoft 365 Copilot** is selected.

### Note

If you don't select **Make agent available in Microsoft 365 Copilot**, your agent is only available in Teams.

3. Select **Add channel**.

## Teams and Microsoft 365 Copilot

X

Microsoft 365 is your cloud-powered productivity solution and includes Outlook, Word, Excel, PowerPoint, and OneDrive. [Learn more](#)

When you publish your agent to Microsoft 365, we'll publish it to Teams too. You'll get all of your agent's advantages in Teams: meeting summaries and transcripts, pointers to open issues or unresolved questions, and more effective collaboration.

### Turn on Microsoft 365

 You'll need to republish your agent after you turn on/off Microsoft 365 Copilot

Make agent available in Microsoft 365 Copilot Chat 

### Agent preview



Safe Travels

Built using Microsoft Copilot Studio.

 Edit details

[Availability options](#)

 See agent in Microsoft 365

 See agent in Teams

 Add channel

## Customize the appearance of an agent for Teams and Microsoft 365 Copilot

Providing the right description and appearance for an agent before making it available to other users is important as it informs them on the agent's purpose and branding, when applicable.

1. Open the configuration panel for the Teams and Microsoft 365 Copilot channels.
2. Select **Edit details** to change the agent's icon, color, and descriptions. These attributes are visible in the Teams app store and on the app's **About** tab after the user installs the agent. Review the [app icon format guidelines for Teams](#).

 **Important**

Changes to an agent's details aren't available to users who already installed the agent from a shared link or from the [Built with Power Platform](#) section of the Teams app store. Users must reinstall the agent to obtain the updates.

For agents that were [approved by an admin](#), you must resubmit the agent for approval.

3. Select **More** to add the following information:

- Developer name
- Website
- Privacy statement
- Terms of use

You should provide this information to inform the users about your agent. For more information, see [Create a privacy statement and terms of use in Microsoft Teams](#).

4. Select **Save**. Your changes to the color, icon, and short description are now visible on the configuration panel.

 **Note**

Icon customization in this section only applies to the agent avatar for agents that are [approved by an admin](#) and doesn't apply for users who install the agent with a direct [installation link](#) or from the Teams app store [Built with Power Platform](#) section. You can customize the agent avatar at [Customize the look and feel of the agent](#).

## Install an agent in Teams and Microsoft 365 Copilot

With your agent published, you can add the agent to your own Teams profile directly from Copilot Studio.

 **Important**

New content in a freshly published agent doesn't appear in conversations that are currently ongoing.

You can enter **Start over** in an existing conversation to start a new conversation right away and see the latest updates to the agent.

We recommend adding your agent to your own profile in Teams first, before sharing it with others.

1. Open the configuration panel for the Teams and Microsoft 365 Copilot channels.

2. Select **See agent in Teams**.

If you configured your agent to be available in Microsoft 365 Copilot when you first added the channel, selecting **See agent in Teams** installs your agent to both Teams and Microsoft 365 Copilot. If you didn't configure your agent for Microsoft 365 Copilot, your agent installs only to Teams.

3. Select **Add**.

The screenshot shows the Microsoft Copilot Studio app configuration interface. At the top, there's a teal icon with a white robot head and a speech bubble. Next to it is the app name 'Safe Travels' with a copy icon, and below that is the placeholder text 'Your developer name'. There are two buttons: a blue 'Add' button and a grey 'Edit' button. Below these buttons are two tabs: 'Overview' (which is selected) and 'Permissions'. The main content area has a heading 'Built using Microsoft Copilot Studio.' followed by a description: 'Help employees stay informed, productive, and connected. Create agents and add important topics for your organization using an intuitive, graphical interface. No code required. Create your own at <https://aka.ms/microsoftcopilotstudio>'. Below this is a section titled 'App features' with two items: 'Bots' (described as 'Complete tasks, find info, and chat using prompts') and 'Agent' (described as 'Use with Copilot or add to a chat or meeting. Added agents are turned on by default'). Under 'Agent', it says 'Created by: [Your developer name](#)' and 'Version 1.0.3'. At the bottom, there's a 'Permissions' section with a 'Expand All' link, a note about permissions, and a statement: 'By using Safe Travels, you agree to the [privacy policy](#), [terms of use](#), and [permissions](#)'.

The agent appears in the agent list on the left.

4. To ask your new agent a question in Microsoft 365 Copilot, enter @, select your agent from the list that appears, then enter your question and select **Enter**.

# Share a link so others can install an agent

1. Open the configuration panel for the Teams and Microsoft 365 Copilot channels.
2. Select Availability options.
3. Select Copy link.
4. Share the link with users so they can install the agent in Teams.

## Important

Only users who can access the agent can use the link to install the agent. You can [share the agent](#) to give users access to the agent.

It's not possible to use an installation link in the Teams mobile app. To make sure your agent is available to Teams mobile users, configure it to [show in the Teams app store](#).

# Show an agent in the Teams app store

Once you publish your agent, you can show it in the Teams app store so users can find and install it.

## Show to shared users

You can share your agent by adding it to the **Built with Power Platform** section of the Teams app store. Only [shared users](#) can find the agent there.

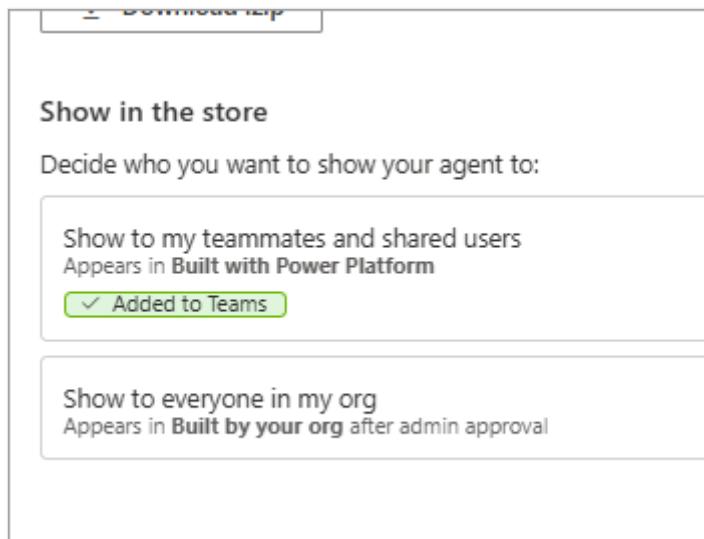
## Important

Only shared users can find and install the agent in the **Built with Power Platform** section of the Teams app store. The agent doesn't show for everyone in the organization even if it's configured to allow everyone to use the agent. To [show the agent to the organization](#), submit the agent for an admin's approval to show it in the **Built for your org** section of the Teams app store.

An agent can only appear in the **Built with Power Platform** section of the Teams app store if the total number of shared users doesn't exceed the limits imposed by the [Teams app store discovery policy](#) for your tenant. To make your agent available to a larger number of users, partner with your admin to [show the agent to the organization](#).

1. Open the configuration panel for the Teams and Microsoft 365 Copilot channels.
2. Select **Availability options**.
3. Select **Show to my teammates and shared users** and review who has access to the agent.
4. Confirm the **Visible in Built with Power Platform** checkbox is selected and select **Share**.

Your agent becomes available in the **Built with Power Platform** section of the Teams app store, and a green **Added to Teams** label appears in the availability options page.



! **Note**

If your app doesn't appear in the **Built with Power Platform** section, your Teams app might be caching information.

Try signing out and back in (if you're using the Teams desktop app) or refreshing your browser (if you're using the Teams web app).

## Remove an agent from the Built with Power Platform section

You can remove an agent from the **Built with Power Platform** section. Removing the agent from there doesn't affect users with the agent installed.

1. Open the configuration panel for the Teams and Microsoft 365 Copilot channels.
2. Select **Availability options**.
3. Select **Show to my teammates and shared users** and clear the **Visible in Built with Power Platform** checkbox.

4. Select **Share**. The agent no longer appears in the **Built with Power Platform** section of the Teams app store.

## Show to the organization

You can share an agent with your organization by submitting it for admin approval to be featured in the **Built for your org** section of the Teams app store. Featuring your app in **Built for your org** is a great way to broadly distribute your agent by partnering with your admin to roll it out to everyone in the organization or to a subset of users.

Admins can also [use app setup policies](#) to automatically install and pin an agent for users in the organization so they can access it directly from their app bar in Teams.

Before submitting an agent to your organization, make sure that:

- Your agent is ready for release and in compliance with company standards, rules, and policies.
- You coordinate with other agent managers. Once you submit an agent, a different user can't resubmit it until an admin approves or rejects the previous submission.
- You [remove the agent](#), if applicable, from the **Built with Power Platform** section of the Teams app store. Otherwise if you submit your agent for admin approval it could end up showing up in two places.

### Note

Once an agent is approved by an admin, the agent availability in the **Built for your org** section is managed by your admin. Only an admin can remove the agent from the **Built for your org** section.

## Submit an agent for the Built for your org section

1. [Open the configuration panel for the Teams and Microsoft 365 Copilot channels](#).
2. Select **Availability options**.
3. Confirm the agent isn't shown to teammates or shared users. [Remove it](#) if it says **Added to Teams**.
4. Select **Show to everyone in my org**.
5. Review the requirements for submission, and select **Submit for admin approval**.

 **Note**

Make sure you review the content and it's ready for an admin to review. You must submit a new approval request to [update the icon and descriptions](#) for agents that were previously approved.

Once an agent is approved, all new and existing users of the agent in Teams receive the update.

A new approval is only required if you modify the agent details. You can make content updates to your agent and publish it as often as desired. You don't need to submit your agent for approval again after publishing such content changes.

A final confirmation prompt appears.

6. Select **Yes** to continue. The agent is sent to your admin [for approval](#), and you can see a note indicating the status of your submission.

 **Note**

After submitting the agent for admin approval, you shouldn't change the agent's access setting to be less than everyone in your organization.

Setting to less than everyone results in users in the organization being unable to chat with the agent after installing it from the **Built for your org** section of the Teams app store.

Work with your admin on the [Microsoft Teams app policy](#) to determine who can find and install the agent.

7. Wait for your [admin's approval](#). You can check the latest status by selecting **Refresh**, or opening the **Make the agent available to others** window from the **Publish** page.
8. After your admin approves the agent, the status is updated to **Approved**. You can directly open the agent and share the approved version's link with other users.
9. Optionally, ask your admin to [use app setup policies](#) to automatically install and pin an agent for users in the organization so they can access it directly from their app bar in Teams.

 **Note**

If your app doesn't appear in the **Built for your org** section even though it's marked as **Approved**, your Teams app might be caching information.

Try signing out and back in again (if you're using the desktop app) or refreshing your browser (if you're using the Teams web app).

## Allow users to add an agent to a team in Teams

In Teams, users can [add an agent to a specific team](#). Once an agent is available for a team, team members can "@mention" it in any team channels, and all teammates see the responses from the agent. This feature is useful for situations such as the following examples:

- Your agent supports collaboration, like sharing the latest sales report insight with the rest of the Sales team.
- You want to allow all team members to engage with the agent in a team channel or via private chat without needing to manually install the agent themselves.

Two options determine whether users can add an agent to any team that they're part of, or to group and meeting chats in Teams. With these options selected, the agent has access to the conversation history from the team channel, group chat, or meeting chat, which helps the agent provide answers better suited to the context of the conversation.

When users install an agent from the [Teams app store](#) or by using the [installation link](#) they can see one of the following options, depending on the collaborative scope you set for the agent:

- Add the agent to a team channel
- Add the agent to group and meeting chats in Teams

Team members can also [remove an agent from a team](#).

### **Important**

The agent should be configured to allow everyone in the organization to chat with it. This setting ensures all team members have permission to chat with the agent. We recommend to not change the security setting of the agent after enabling this capability. Otherwise users might see permission errors when they try to interact with the agent in Teams channels or group chats.

To allow users to add an agent to a team:

1. Open the configuration panel for the Teams and Microsoft 365 Copilot channels.

2. Select **Edit details**.
3. Select **Allow your users to add this agent to a team**.
4. Select **Save**.

Teams caches information, so it can take a while before users see the option to add the agent to a team or group chat. Users can refresh the browser (or sign out of the Teams desktop client and sign back in) to get the latest changes.

If you previously [submitted the agent for admin approval](#), you must submit it again after enabling this option. The admin must approve your changes before users can add the agent to their teams from the **Built for your org** section of the Teams app store.

## Prevent users from adding an agent to a team, or group and meeting chats

Disabling this option prevents users from adding an agent to a new team, group, or meeting chat. If you added an agent to a conversation already, disabling this option doesn't remove it. You must manually [remove](#) the agent.

To prevent users from adding an agent to a team:

1. [Open the configuration panel for the Teams and Microsoft 365 Copilot channels](#).
2. Select **Edit details**.
3. Clear the **Users can add this agent to a team** checkbox.
4. Select **Save**.

## Configure private greeting behavior

Private greetings from an agent are useful for when you want team members to have one-on-one conversations with the agent without needing to manually install the agent themselves.

### Note

Private greeting configuration is available for classic chatbots only.

When enabled, your agent automatically starts a conversation with team members. To customize the agent's greeting message, [modify the Greeting system topic](#).

Consider how relevant the agent is to team members. Make sure the agent's greeting helps them understand what to do with it, describes its limitations and, most importantly, gets them comfortable interacting with it.

To configure when an agent sends a private greeting:

1. [Open the configuration panel for the Teams and Microsoft 365 Copilot channels](#).
2. Select **Settings**.
3. Configure the private greeting options.
4. Select **Save**.

## Download the Teams app manifest for an agent (advanced)

You can download the agent's Teams app manifest to make detailed changes to the manifest directly and share with your admin or users in the tenant. You can also directly upload an app manifest into Teams if your organization's policy allows you to [upload custom applications](#).

1. [Open the configuration panel for the Teams and Microsoft 365 Copilot channels](#).
2. Select **Availability options**.
3. Select **Download .zip**.

The manifest is generated as a .zip file, and is saved according to your browser's save settings.

4. In Teams, select **Apps > Manage your apps > Upload an app**.
5. Select **Upload a custom app**.
6. Browse for the downloaded .zip file.

## Disconnect an agent from Teams

You can take your agent offline so users can no longer chat with the agent in Teams. When your agent is offline, it doesn't respond to anyone who tries to chat with it. You should thoroughly consider the need to disconnect your agent before you do so.

Disconnected agents don't appear in the **Built with Power Platform** section of the Teams app store and can't be installed from a shared link. If a user already has such an agent installed in Teams, it remains available until the user uninstalls it.

After you disconnect an agent that was approved by an admin, it continues to appear in the **Built for your org** section of the Teams app store. You must ask your admin to remove it.

1. [Open the configuration panel for the Teams and Microsoft 365 Copilot channels](#).

2. Select **Remove channel**.

3. Review the information about disconnecting your agent, and select **Disconnect** to continue.

You can always reconnect your agent by selecting **Add channel** in the configuration panel for the Teams and Microsoft 365 Copilot channels, republishing, and then reinstalling the agent in Teams.

 **Note**

If you readd the Teams channel without first uninstalling the agent from Teams, the reinstallation of the agent displays an **Update now** button instead of an **Add** button. In this case, select **Update now** to reinstall the agent in Teams.

## Open the configuration panel for the Teams and Microsoft 365 Copilot channels

1. Open your agent in Copilot Studio.

2. On the top menu bar, select **Channels**.

3. Select the **Teams and Microsoft 365 Copilot** tile. The configuration panel appears.

## Known limitations

The known limitations differ depending on whether you're using Teams or Microsoft 365 Copilot.

### Known limitations in Teams

- [Private greeting configuration](#) isn't supported for agents using the [upgraded unified authoring canvas](#).
- [Setting the agent avatar icon](#) on the details page doesn't show up for agents approved by an admin. Agents approved by an admin use the app icon as the agent avatar.

- Teams has [rate limiting](#) on agents. Depending on your business needs, you might want to keep your agent content concise for agents that are added to Teams.
- Currently, if your agent supports [user authentication](#), the user isn't able to explicitly sign out. This behavior fails the Teams AppSource certification if you're publishing your agent in the Seller Dashboard. This limitation doesn't apply to personal or tenant usage of the agent. Learn more about how to [publish your app to the Teams Store](#) and about the [AppSource certification policy](#).
- After you disconnect your agent from the Teams channel, it might take a few minutes before the agent becomes unreachable by users in Teams.
- Group chats and meeting chats don't support agents configured for [manual authentication](#) with [single sign-on \(SSO\) for Teams](#). Depending on your business needs, you could either [authenticate with Microsoft](#) or remove SSO for Teams from your manual authentication.

## Known limitations in Microsoft 365 Copilot

- **Conversation Start topic not supported:** The Microsoft 365 Copilot channel doesn't support the **Conversation Start** topic, so general greetings aren't possible. Starter prompts are used instead. Variables can be set via Activity Event-triggered topics, but these aren't ideal for greetings due to side effects.
- **Unsupported media types:** Media like GIFs aren't rendered
- **Embedded URLs:** Microsoft 365 Copilot might remove embedded URLs for security. Citations from generative answers should include URLs in the citation section.
- **Unsupported node types:**
  - Speech operations
  - Hand-off to customer service representative
  - Adaptive Cards with `Action.Execute`
- **Unsupported message types:**

 **Note**

You can use Adaptive Cards to show images.

- Basic cards (`ContentFiltered` error)
- Video (`ContentFiltered` error)
- Image (`ContentFiltered` error)
- File
- Speech (not sent)

- Inactivity trigger fires but doesn't post messages back to Microsoft 365 Copilot.

# Create a privacy statement and terms of use in Microsoft Teams

Article • 11/19/2024

When you want to publish your agent so other Microsoft Teams users can access it, you need to provide the following information about your agent:

- A link to a compliant privacy statement
- A link to the terms of use (TOU) for the agent
- A link to the developer's website for more information
- The developer's name

You should update each of these with your own information. See the Microsoft Teams documentation about [Privacy policy, terms of use, and support URLs](#) for some more ideas on what to include.

The following table lists the default values that are provided in a Copilot Studio agent, and suggestions on what you should enter instead:

[ ] Expand table

Requirement	Default value	Suggestions
Privacy statement	<a href="#">Empty privacy statement and terms of use</a>	You should work with your manager or IT department to determine what you want to say in your privacy statement. This should follow your organization's privacy practices.
Terms of use	<a href="#">Empty privacy statement and terms of use</a>	Work with your manager or IT department to define the terms of use. Typically this would include any limitations you might have on how the information can be used. See <a href="#">Teams privacy policy guidelines</a> .
Developer website	Link to the <a href="#">Copilot Studio overview page</a> ↗	You could include a link to your Office profile, or a SharePoint site for your team.
Developer's name	<i>Powered by Microsoft Copilot Studio</i>	You should include a name of someone who can be contacted in case someone has questions about the agent.

## Empty privacy statement and terms of use

ⓘ Important

This app is not developed by Microsoft. If you are a user of an agent and see this page, it means the agent developer did not provide a terms of use and privacy statement for the agent.

For privacy policy or terms of use, contact the developer of this app.

If you do not enter links to a privacy statement or terms of use, then agent users are redirected here.

---

## Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

# Send proactive Microsoft Teams messages

08/28/2025

After you've [published your agent](#) and [made the agent available to users in Microsoft Teams](#), you can notify users ("recipients") in Microsoft Teams with proactive messages. Proactive messages use Power Automate flows to deliver their content, and are useful in many scenarios, including:

- Letting a recipient know that their earlier request has been completed. For example, the user's time-off request has been approved.
- Providing reminders or news updates. For example, the agent could send a reminder message to complete online training.

## Important

Proactive messages can be configured for agents and classic chatbots created and customized in Copilot Studio.

An agent can't deliver messages if the recipient:

- Doesn't have the agent installed in Microsoft Teams.
- Has uninstalled the agent.
- Has blocked the agent.
- Doesn't have permission to chat with the agent, in which case you need to [share the agent with other users](#).

## Prerequisites

- [Use Power Automate flows](#)
- [Publish your agent](#)
- [Connect and configure an agent for Microsoft Teams](#)
- [Getting started with Power Automate](#)

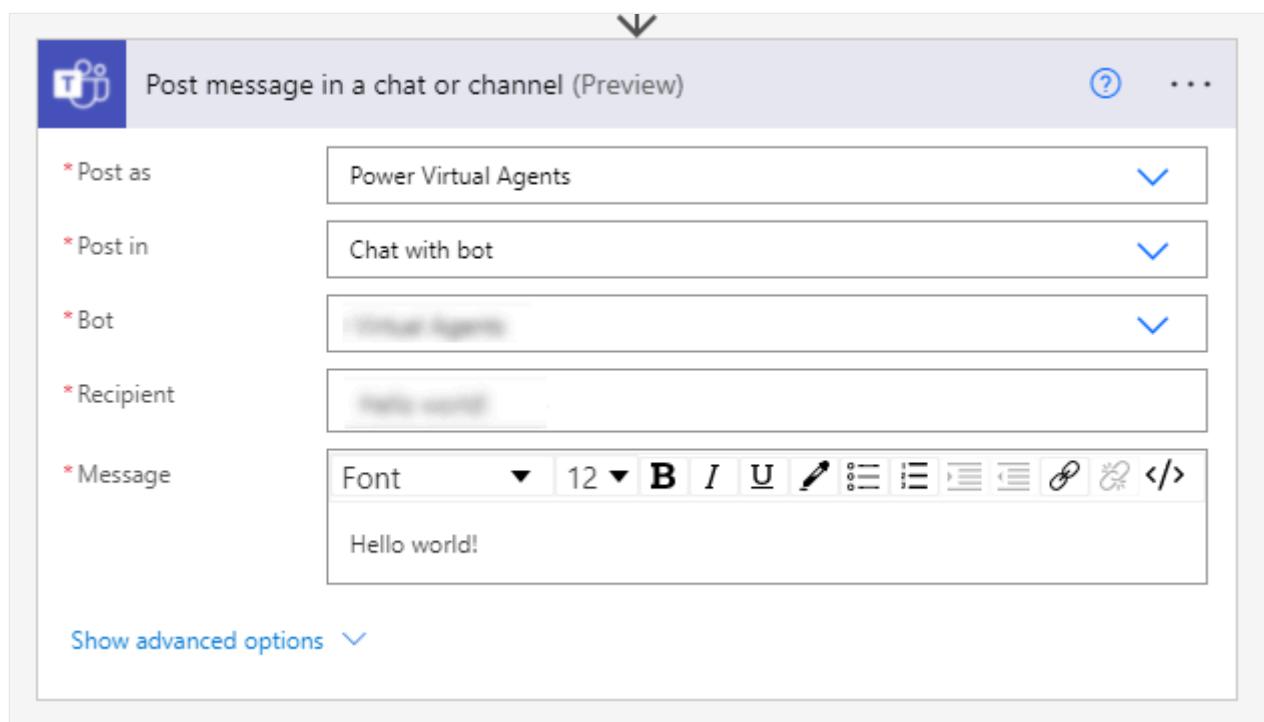
## Known limitations

- If the agent is disconnected and reconnected to Microsoft Teams, users won't receive proactive messages until after they reinstall the agent.
- All proactive messages from Copilot Studio are subject to [limits on Power Automate](#) and [throttling limits of the Microsoft Teams connector](#).

- Proactive messages aren't logged in conversation transcripts or [Analytics sessions](#).
- Proactive messages must be in the same environment as the Power Automate flow.
- Proactive messages can only be posted to a personal chat with the agent.

## Send a proactive message

1. In Power Automate, open the flow that you want to send a proactive message.
2. Add the Microsoft Teams connector action **Post message in a chat or channel**.
3. For **Post as**, choose **Microsoft Copilot Studio (Preview)**.
4. For **Post in**, choose **Chat with bot**.
5. For **Bot**, select the agent that you want the message to be sent from.
6. For **Recipient**, enter the recipient's name or email address. You can also use dynamic content if the recipient info comes from an earlier step in the flow.
7. For **Message**, enter the message you want to send.



When the flow is run, the recipient will receive the proactive message from the agent in Microsoft Teams.

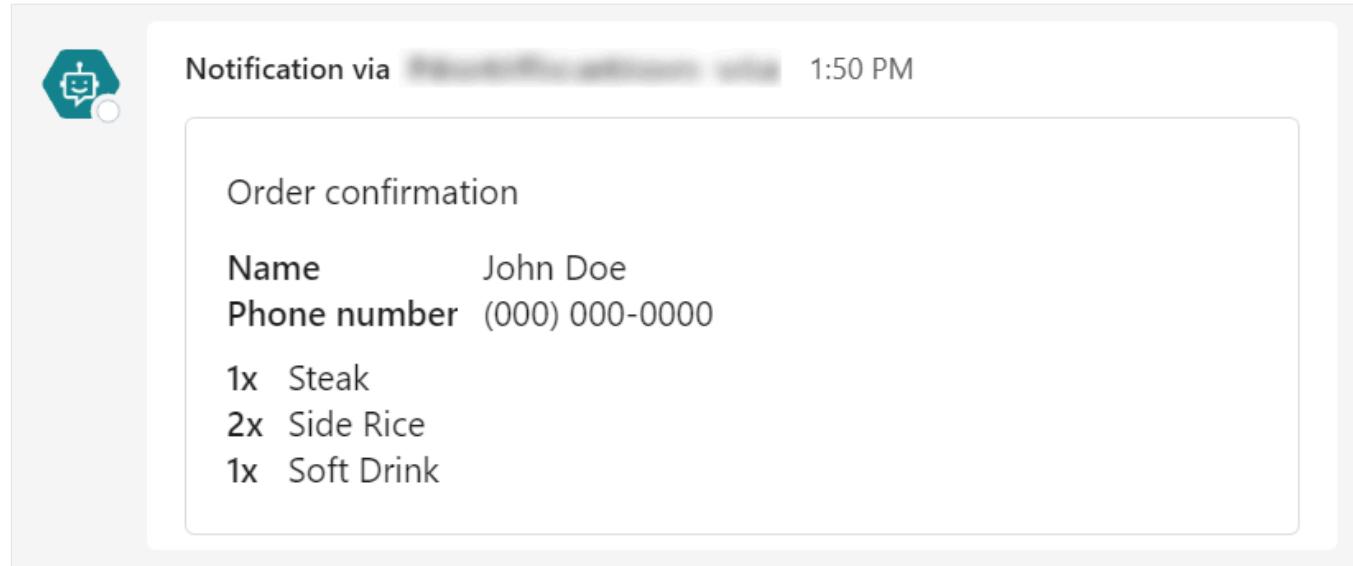
## Send a proactive Adaptive Card

### Important

This section details how to send proactive Adaptive Cards with Power Automate flows.

Proactive Adaptive Cards enables users to receive information from an *inactive* conversation with an agent.

In addition to sending proactive messages, you can also send proactive Adaptive Cards.



Adaptive Cards are an open card exchange format enabling developers to exchange UI content in a common and consistent way. You can author Adaptive Cards by hand in JSON, or if you prefer a drag-and-drop interface, you can use the [Adaptive Cards Designer](#).

 **Note**

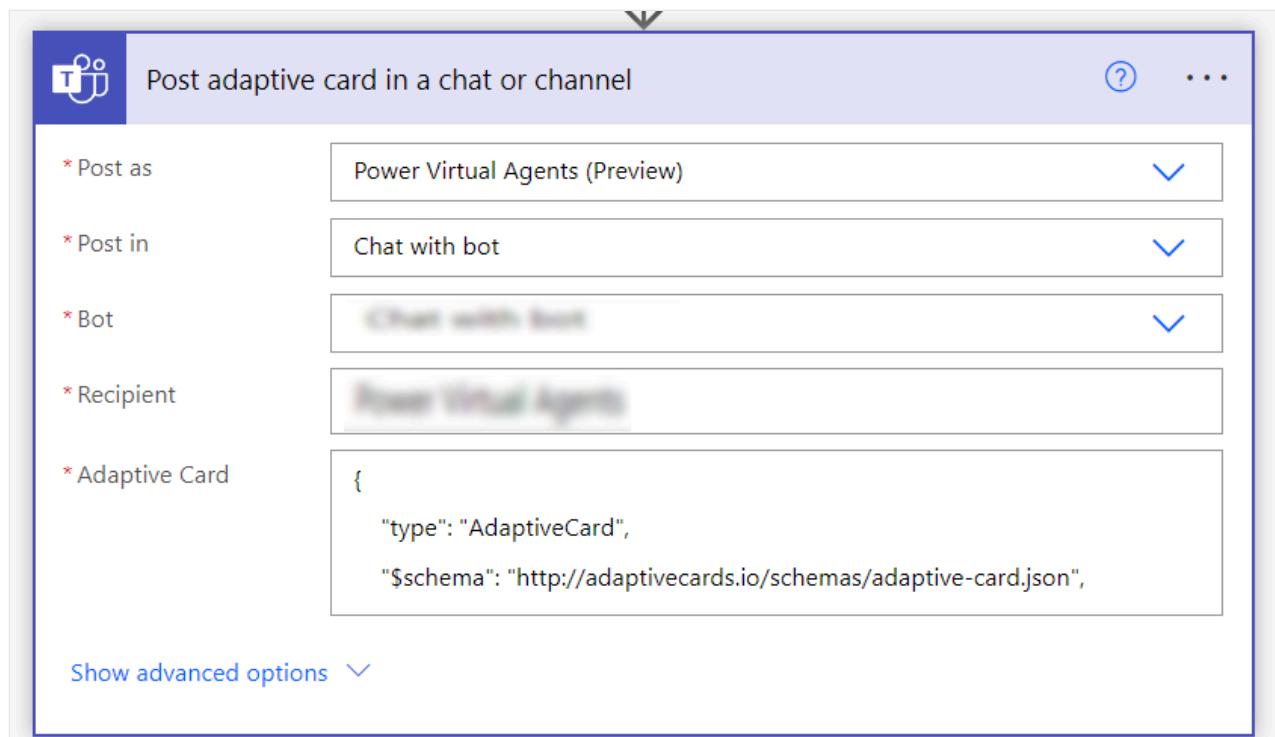
Power Automate does not support Adaptive Cards that use the [templating feature](#).

In this example, you'll send an order summary card for the user to review.

1. In Power Automate, add the Microsoft Teams connector action **Post adaptive card in a chat or channel** at the step where you want to send card in your flow.
2. For **Post as**, choose **Microsoft Copilot Studio (Preview)**.
3. For **Post in**, choose **Chat with bot**.
4. For **Bot**, select the agent that you want the message to be sent from.
5. For **Recipient**, enter the recipient's name or email address. You can also use dynamic content if the recipient info comes from an earlier step in the flow.
6. For **Adaptive Card**, enter the following template JSON:

JSON

```
{
 "type": "AdaptiveCard",
 "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
 "version": "1.5",
 "body": [
 {
 "type": "TextBlock",
 "text": "Order summary",
 "wrap": true,
 "style": "heading"
 },
 {
 "type": "FactSet",
 "facts": [
 {
 "title": "Name",
 "value": "John Doe"
 },
 {
 "title": "Phone number",
 "value": "(000) 000-0000"
 }
]
 },
 {
 "type": "Container",
 "items": [
 {
 "type": "FactSet",
 "facts": [
 {
 "title": "1x",
 "value": "Steak"
 },
 {
 "title": "2x",
 "value": "Side Rice"
 },
 {
 "title": "1x",
 "value": "Soft Drink"
 }
],
 "spacing": "Small"
 }
],
 "spacing": "Small"
 }
]
}
```



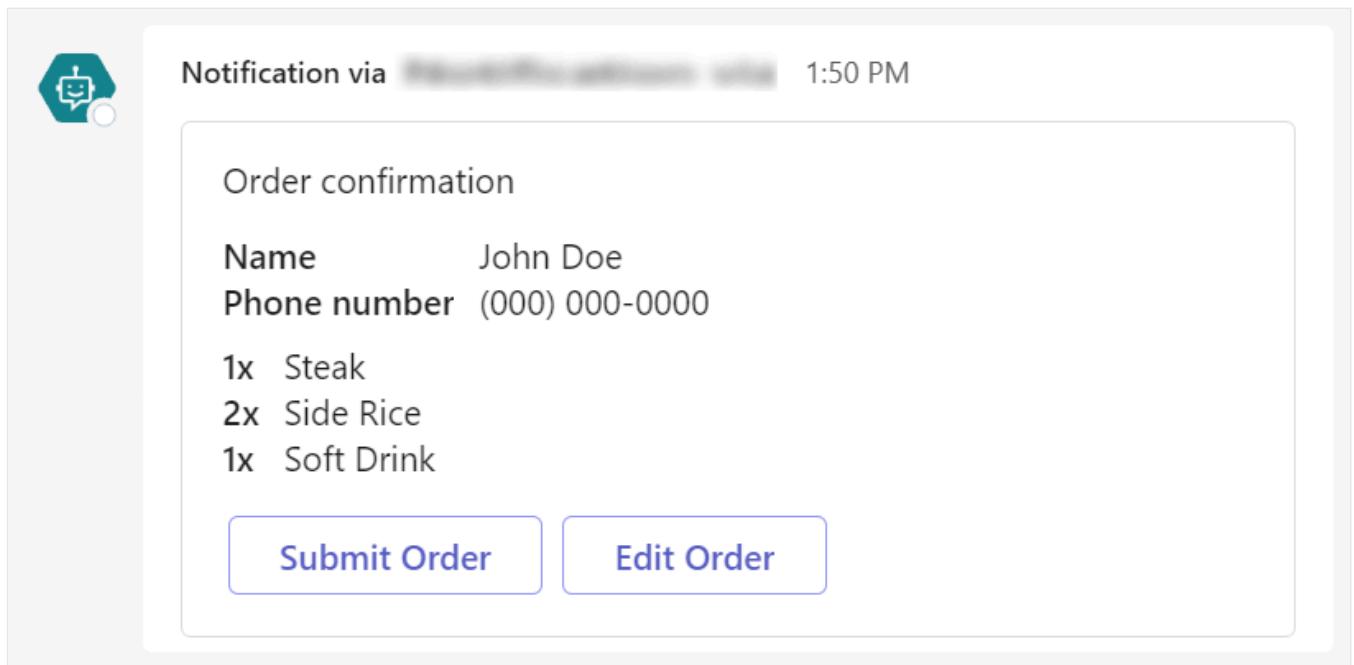
When the flow is run, the recipient will receive the Adaptive Card from the agent in Microsoft Teams.

## Wait for user response

### Important

This section details how to send proactive Adaptive Cards with Power Automate flows. Proactive Adaptive Cards enables users to receive information from an *inactive* conversation with an agent.

Adaptive Cards support collecting user input. In these scenarios, you'll want to wait for the user's response before the flow continues.



### ⓘ Note

Power Automate does not support Adaptive Cards that use the [templating feature](#).

In this example, you'll send an order confirmation card that allows the user to make changes before the order is submitted.

1. In Power Automate, add the Microsoft Teams connector action **Post adaptive card and wait for a response** at the step where you want to send card in your flow.
2. For **Post as**, choose **Microsoft Copilot Studio (Preview)**.
3. For **Post in**, choose **Chat with bot**.
4. For **Message**, enter the following template JSON:

The properties in the `actions` section determine the options that are presented to the user.

```
JSON

{
 "type": "AdaptiveCard",
 "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
 "version": "1.0",
 "body": [
 {
 "type": "TextBlock",
 "text": "Order confirmation",
 "wrap": true,
 "style": "heading"
```

```

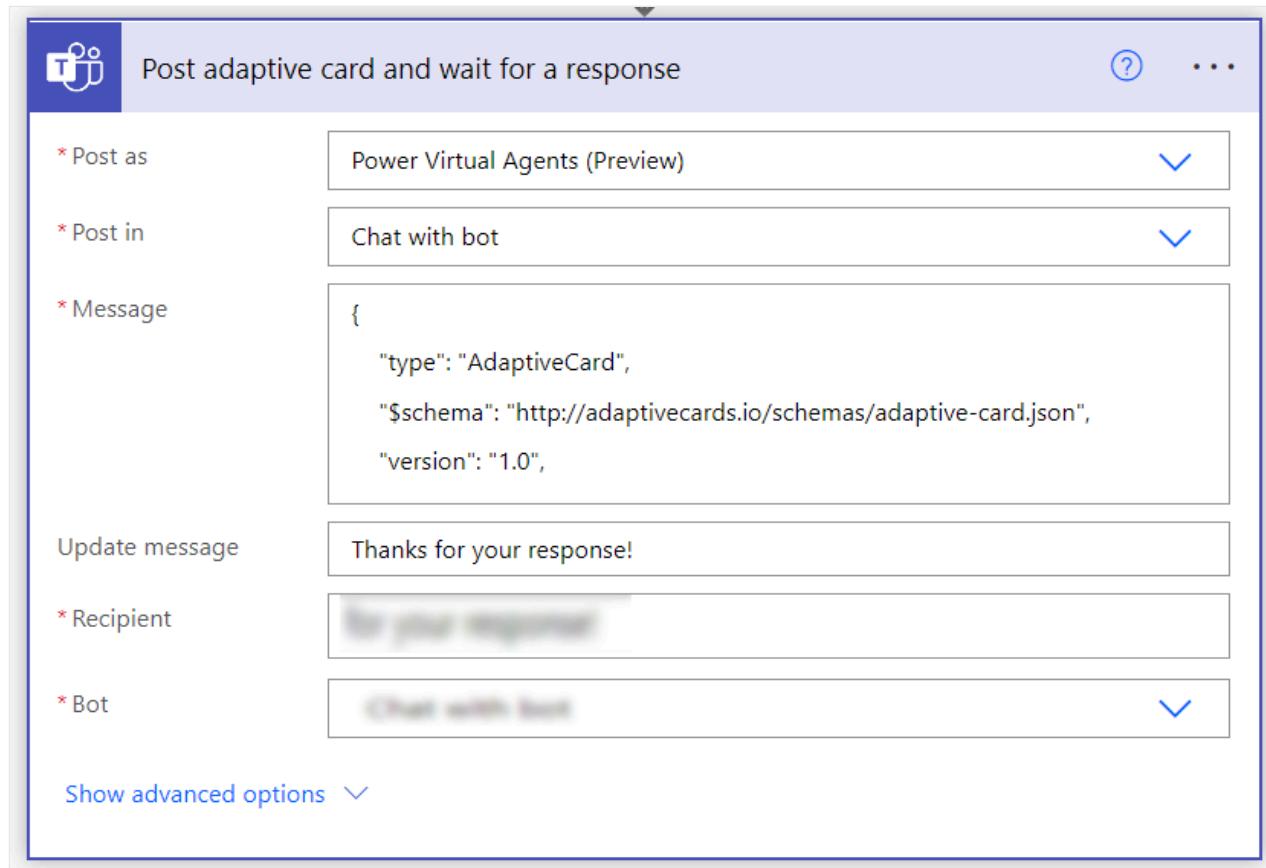
},
{
 "type": "FactSet",
 "facts": [
 {
 "title": "Name",
 "value": "John Doe"
 },
 {
 "title": "Phone number",
 "value": "(000) 000-0000"
 }
]
},
{
 "type": "Container",
 "items": [
 {
 "type": "FactSet",
 "facts": [
 {
 "title": "1x",
 "value": "Steak"
 },
 {
 "title": "2x",
 "value": "Side Rice"
 },
 {
 "title": "1x",
 "value": "Soft Drink"
 }
],
 "spacing": "Small"
 }
],
 "spacing": "Small"
}
],
"actions": [
 {
 "type": "Action.Submit",
 "title": "Submit Order"
 },
 {
 "type": "Action.Submit",
 "title": "Edit Order"
 }
]
}

```

5. For **Update message**, enter the update message that the recipient will see after providing their response.

6. For **Recipient**, enter the recipient's name or email address. You can also use dynamic content if the recipient info comes from an earlier step in the flow.

7. For **Bot**, select the agent that you want the card to be sent from.



When the flow is run, the recipient will receive the adaptive card from the agent in Microsoft Teams that they can then provide a response to.

To use the response from the recipient, select **submitActionId** from the dynamic content flyout menu. The value of this variable will be the **title** of the action the user chose.

## Send proactive messages to multiple recipients

In some scenarios, you might want to send the same proactive message to multiple recipients.

This section contains examples for sending messages to multiple recipients.

### Important

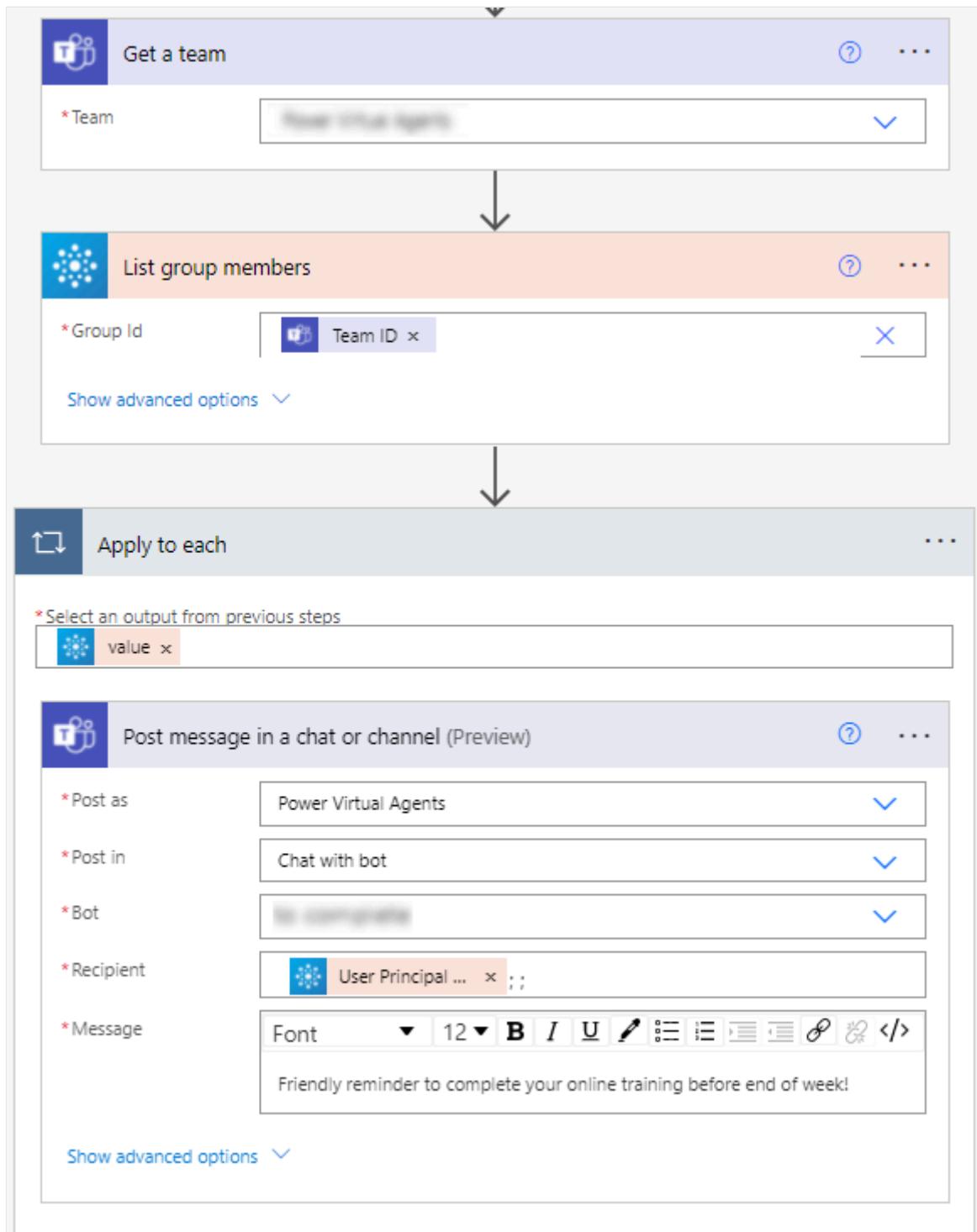
The prerequisites detailed in this topic apply to each recipient.

For example, you can send the same message to 10 recipients in one action. If three of the 10 recipients do not have the agent installed, then those three recipients won't receive the message.

## Send to teammates

In this example, the agent will send a reminder to the members of a team to complete their online training.

1. Add the **Microsoft Teams** connector and select the **Get a team** action in your Power Automate flow.
2. For **Team**, choose the team with the members that you want to send the message to.
3. Add the **Office 365 Groups** connector and select the **List group members** action.
4. For **Group Id**, select **Custom value**.
5. Select **Team ID** from the dynamic content of the **Get a team** action.
6. Add the **Microsoft Teams** connector and select the **Post message in a chat or channel** action.
7. For **Post as**, choose **Microsoft Copilot Studio (Preview)**.
8. For **Post in**, choose **Chat with bot**.
9. For **Bot**, select the agent that you want send the message.
10. For **Recipient**, select the dynamic content **User Principle Name** from the **List group members** action.



11. For **Message**, enter the message you want to send.

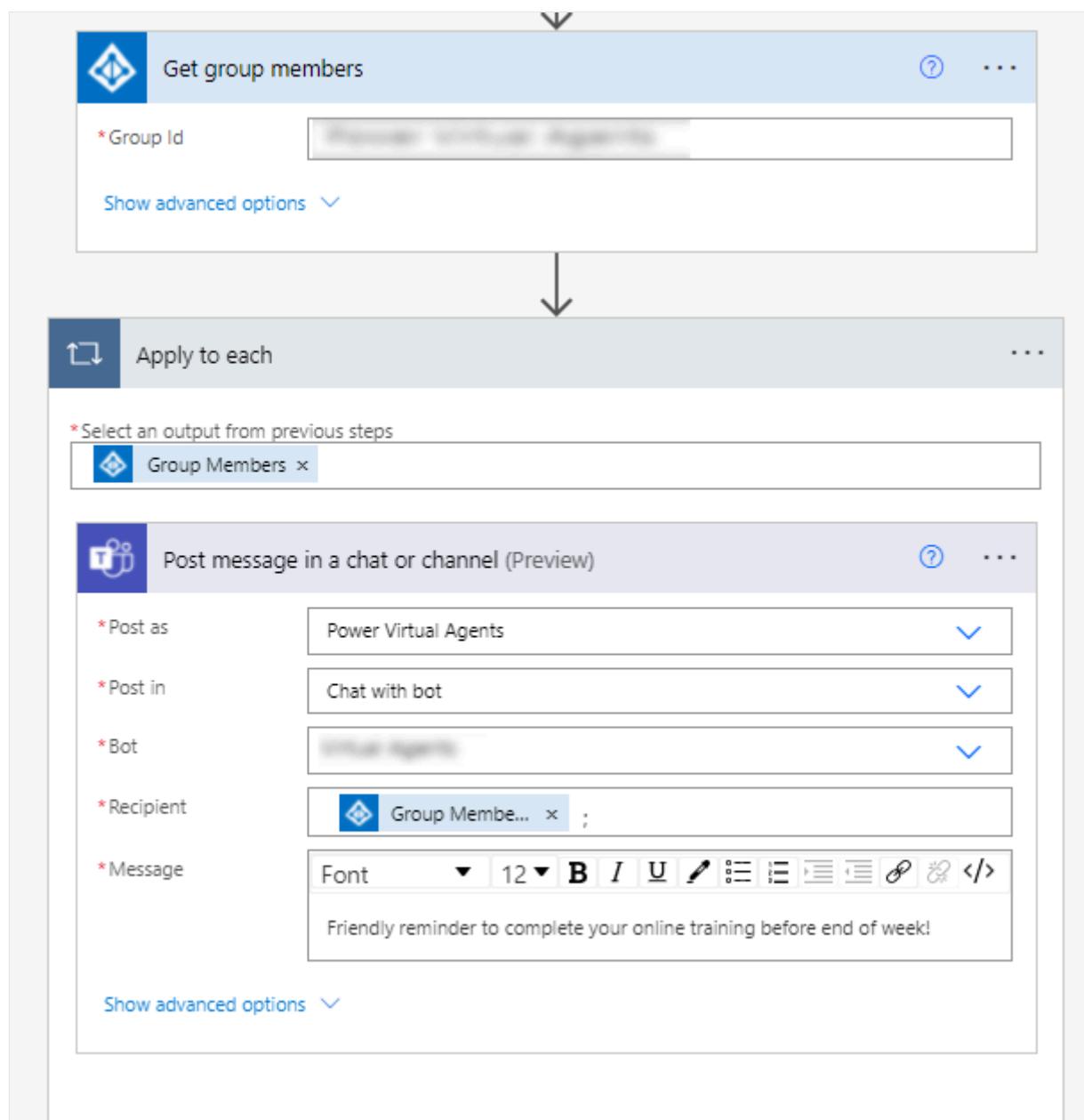
When the flow runs, each user in the team will receive the proactive message in a private chat with the agent.

## Send to a security group

In this example, the agent will send a reminder to a security group to complete their online training.

1. Add the **Microsoft Entra ID** connector and select the **Get group members** action in your Power Automate flow.

2. For **Group Id**, enter the security group's GUID.
3. Add the Microsoft Teams connector and select the **Post message in a chat or channel** action.
4. For **Post as**, choose **Microsoft Copilot Studio (Preview)**.
5. For **Post in**, choose **Chat with bot**.
6. For **Bot**, select the agent that you want send the message.
7. For **Recipient**, select the dynamic content **Group Members User Principle Name** from the **Get group members** action.
8. For **Message**, enter the message you want to send.



When the flow runs, each user in the security group will receive the proactive message in a private chat with the agent.

# Send proactive message to multiple recipients in parallel

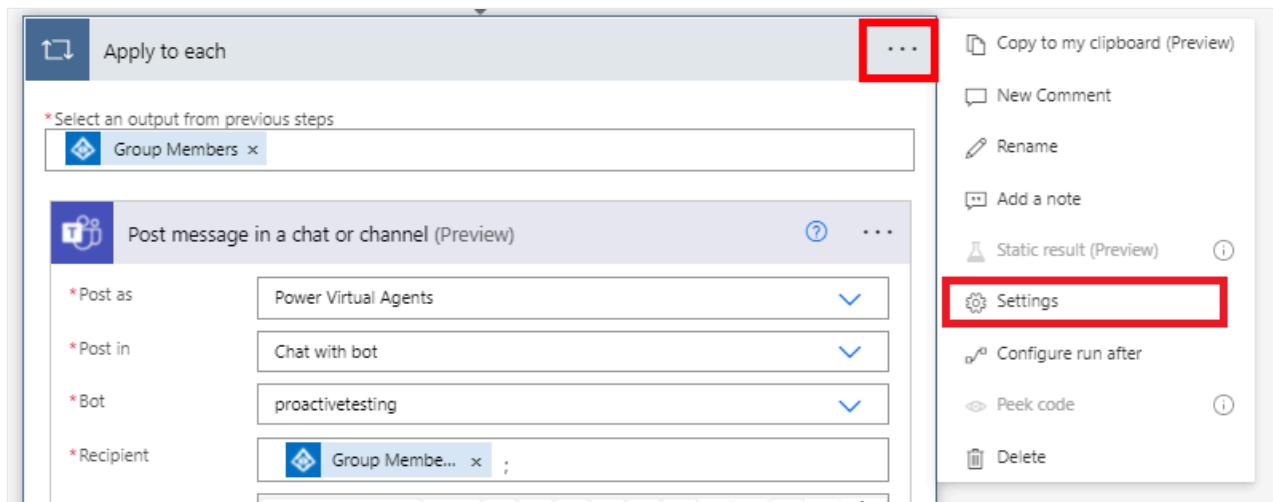
Normally when sending a proactive message to multiple recipients, your agent will send one message after another. However, in some situations, it might be preferable to send the message to multiple recipients at the same time.

## ⚠ Warning

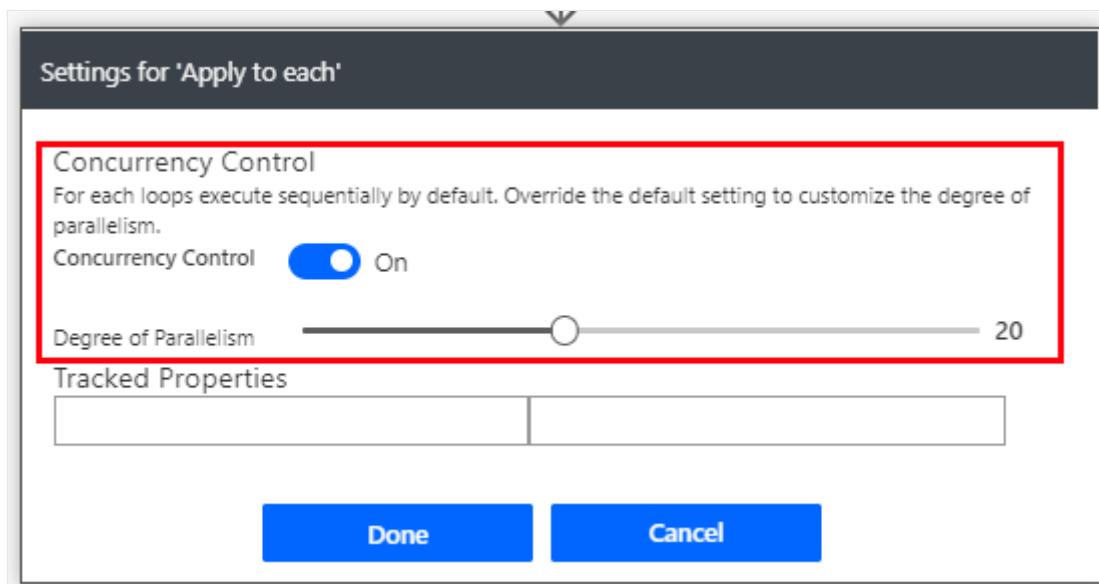
All proactive messages from Copilot Studio are subject to [limits on Power Automate](#) and [throttling limits of the Microsoft Teams connector](#).

If you are sending messages to a large group of recipients, you'll see errors if you hit the throttling limit. You can reduce the degree of parallelism, or reduce the number of recipients in the group.

1. In the **Apply to each** action, select the three horizontal dots (...) and then **Settings**.

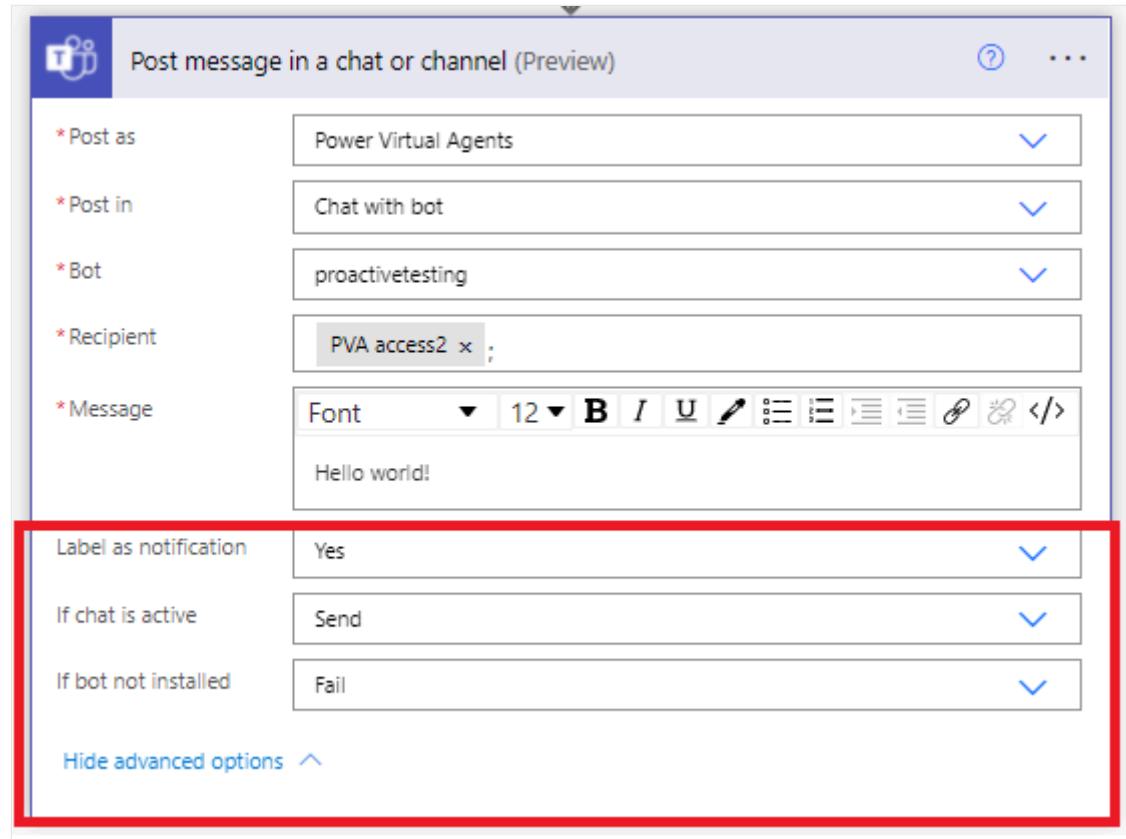


2. Turn on **Concurrency control** and set the degree of parallelism.



# Configure advanced options for proactive messages

Copilot Studio allows you to control detailed behavior on your agent under **Show advanced options** in the Microsoft Teams connector.



The screenshot shows the 'Post message in a chat or channel (Preview)' configuration screen. At the top, there are fields for 'Post as' (Power Virtual Agents), 'Post in' (Chat with bot), 'Bot' (proactivetesting), and 'Recipient' (PVA access2). Below these is a rich text editor with the message 'Hello world!'. A red box highlights the 'Label as notification' section, which contains three dropdowns: 'Label as notification' set to 'Yes', 'If chat is active' set to 'Send', and 'If bot not installed' set to 'Fail'. There is also a 'Hide advanced options' link at the bottom of this section.

## Label sent message as a notification

**Label as notification** controls whether the message has the text "Notification via" in front of the agent's name. Labeling the agent's response allows the recipient to identify the agent's response to their inquiry.



## When the recipient is currently in an active chat with the agent

Sometimes the agent might be sending a proactive message when the recipient is in an active conversation with the agent.

You might want to postpone sending the proactive message until they have finished their conversation to not disrupt the conversation flow.

The **If chat is active** box allows you to control the behavior:

- **Send:** the agent will send the proactive message as normal.
- **Don't send and succeed:** the agent won't send the proactive message when the recipient is in an active conversation. Status code **300** will be returned.
- **Don't send and fail:** the agent won't send the proactive message when the recipient is in an active conversation. The flow run will be marked as a failure.

## When the recipient hasn't installed the agent

The agent can only deliver messages to recipients who have installed the agent in Microsoft Teams. Recipients might not want to install the agent or have uninstalled the agent.

For lower importance messages, you can set the flow run to be marked as succeeded even when the recipient doesn't have the agent installed.

The **If bot not installed** box allows you to control the behavior:

- **Fail:** the flow run will be marked as a failure when the recipient hasn't installed the agent in Microsoft Teams.
- **Succeed with status code:** the flow run will be marked as succeeded even though the recipient can't receive the message because they haven't installed the agent. Status code **100** will be returned.

## Status code definition

You can use the returned status code to define different follow-up behaviors in your flow. For example, you could specify that the flow should try again over a period of time or log a record about the failure.

 Expand table

Status code	Succeeded (Boolean)	Description
200	True	Message is successfully delivered.
100	False	Message couldn't be delivered because the recipient doesn't have the agent installed.

<b>Status code</b>	<b>Succeeded (Boolean)</b>	<b>Description</b>
300	False	Message couldn't be delivered because the recipient is in an active conversation with the agent.

# Publish an agent to Facebook

Article • 03/06/2025

You can add your agent to Facebook Messenger to use the Facebook experience to engage with your customers.

Facebook Messenger and related services are subject to Facebook's own terms and conditions. For support related to Facebook Messenger, contact Facebook directly.

## Important

This article is intended for experienced IT professionals who manage your organization's Facebook page.

## Note

By publishing your agent to a Facebook page, some of your data such as agent content and user chat content will be shared with Facebook (meaning that your data will flow outside of your [organization's compliance and geographic or regional boundaries](#)).

For more information, see [Facebook's Platform Policies](#).

## Add your agent to Facebook Messenger

You need a Facebook app, and an associated developer account, to connect your Copilot Studio agent to Facebook Messenger on your Facebook pages.

To connect your agent to Facebook Messenger:

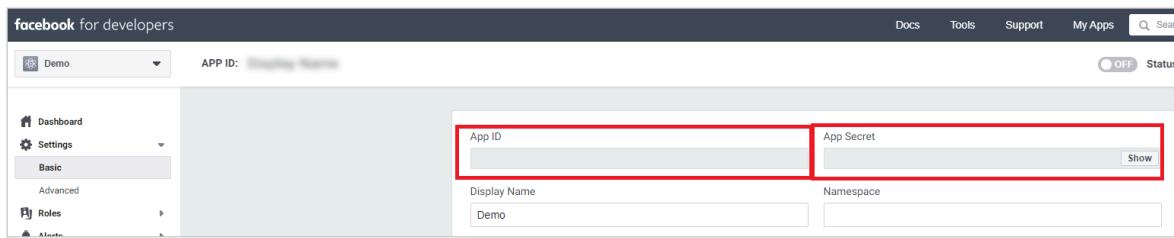
1. Configure Facebook app settings: retrieve your Facebook app information, enable API access, add Facebook Messenger to your app, and configure the Facebook pages your app should appear on.
2. Configure the Facebook publication channel in Copilot Studio.
3. Connect your Facebook app to Copilot Studio using webhooks.

You must submit your app for Facebook review before you can publish your app and make it public.

# Configure Facebook app settings

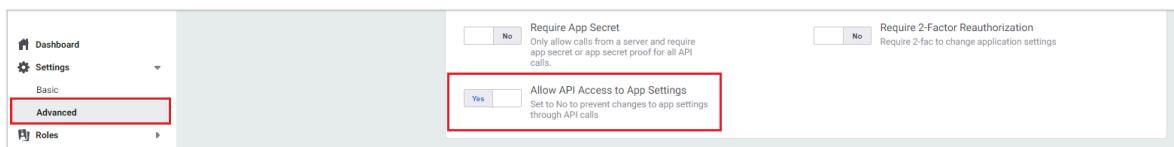
## Retrieve Facebook app information

1. Sign in to the Facebook app that you want to add your agent to at [Facebook for Developers](#).
2. Under **Settings** on the side menu pane, select **Basic**.
3. Copy the *App ID* and *App Secret*. You need these values when you [configure the Facebook channel in Copilot Studio](#).



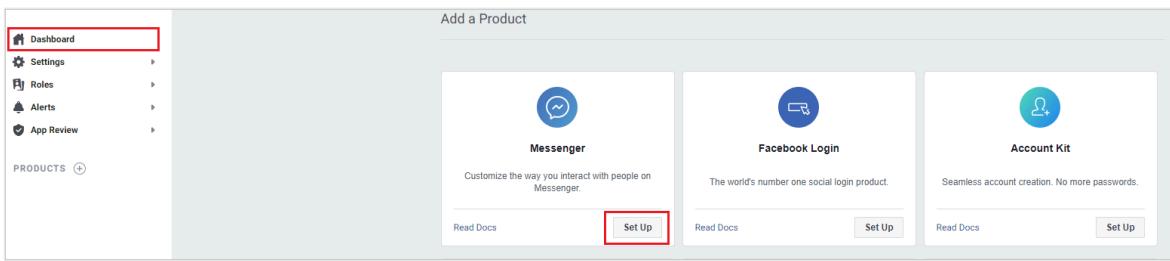
## Enable API access for your Facebook app

1. Sign in to the Facebook app that you want to add your agent to at [Facebook for Developers](#).
2. Under **Settings** on the side menu pane, select **Advanced**.
3. Make sure **Allow API Access to App Settings** is set to **Yes**.
4. Select **Save Changes** to confirm your changes.



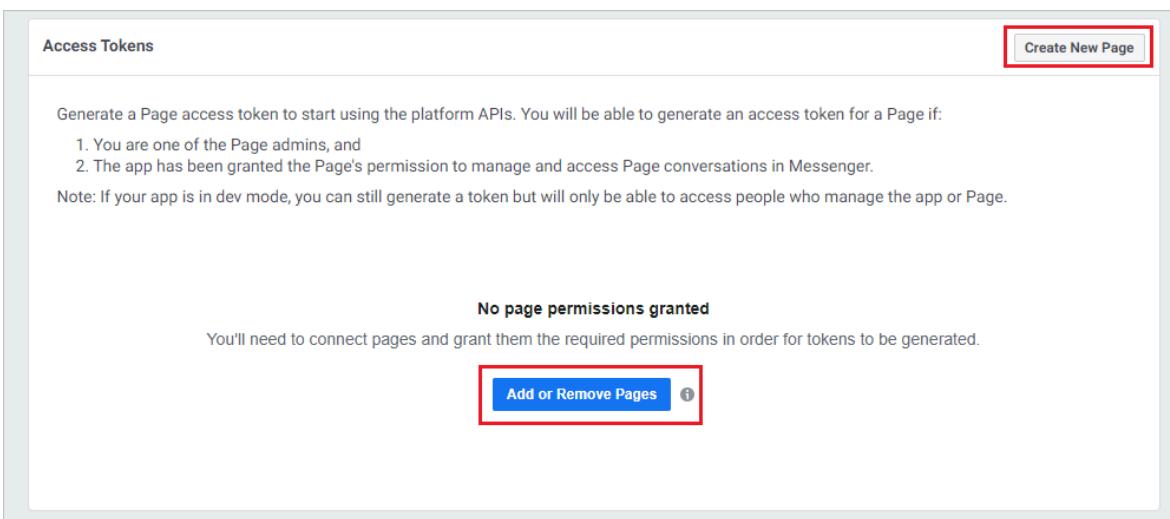
## Add Facebook Messenger to your app

1. Sign in to the Facebook app that you want to add your agent to at [Facebook for Developers](#).
2. Go to the **Dashboard**. Under the **Add a Product** section, select **Set Up** on the **Messenger** tile.



## Configure Facebook pages

1. Sign in to the Facebook App that you want to add your agent to at [Facebook for Developers](#).
2. Select **Settings** under **Products** and **Messenger** on the side menu pane.
3. Add the pages you want to add the agent to by selecting **Add or Remove Pages** under the **Access Tokens** section. You can also create a new page by selecting **Create New Page**.



4. When adding pages, make sure **Manage and access Page conversations in Messenger** is set to **Yes**.

### **⚠ Submit for Login Review**

Some of the permissions below have not been approved for use by Facebook.

[Submit for review now or learn more.](#)

## What is [REDACTED] allowed to do?



[REDACTED] may not work properly if you turn off these options.

Manage and access Page conversations in Messenger

New app



[Cancel](#)

[Back](#)

[Done](#)

[Help Center](#)

5. Copy the *Page ID* and *Token* for each of the pages that you want to add the agent to. Select **Generate Token** for each page. You need these tokens when you configure the Facebook channel in Copilot Studio.

### Access Tokens

[Create New Page](#)

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

Pages ↑	Tokens
D [REDACTED]	—
N [REDACTED]	—

[Add or Remove Pages](#) ⓘ

[Generate Token](#)

[Generate Token](#)

## Configure the Facebook channel in Copilot Studio

1. Open your agent in Copilot Studio.

2. On the top menu bar, select **Channels**.
3. Select the **Facebook** tile to open the configuration window.
4. Paste the Facebook app ID, app secret, page ID, and page access token you retrieved earlier, into the corresponding fields.

## Facebook

X

Set up this channel to let Facebook users chat with your agent. [Learn how to set up Facebook channel](#)

Note that certain agent content may not render the same on Facebook Messenger as it was authored in Microsoft Copilot Studio. For details, refer to our [supported channel content](#) article.

### Credentials

Get this information from Facebook and enter it here.

Facebook app ID \*

Facebook app secret \*

### Pages

Add Facebook page information for each place you want customers to access your agent.

Page ID \*



Page access token \*

+ Add page

### Connection info for Facebook

After adding this channel, copy and paste the values below into Facebook.

#### Callback URL

(Available after channel is added)

Copy

#### Verify token

(Available after channel is added)

Copy

By clicking **Add**, you consent to your data being shared with third party systems and to your data flowing outside your organization's compliance and geo boundaries. This includes Government Cloud environments. Learn more about [where your data is located](#) and the [Microsoft Privacy Statement](#).

Add

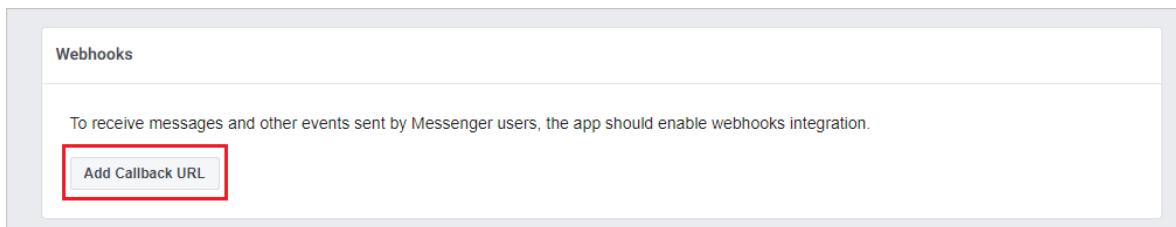
Cancel

5. At least one page is required, but you can select **Add page** to add more pages.

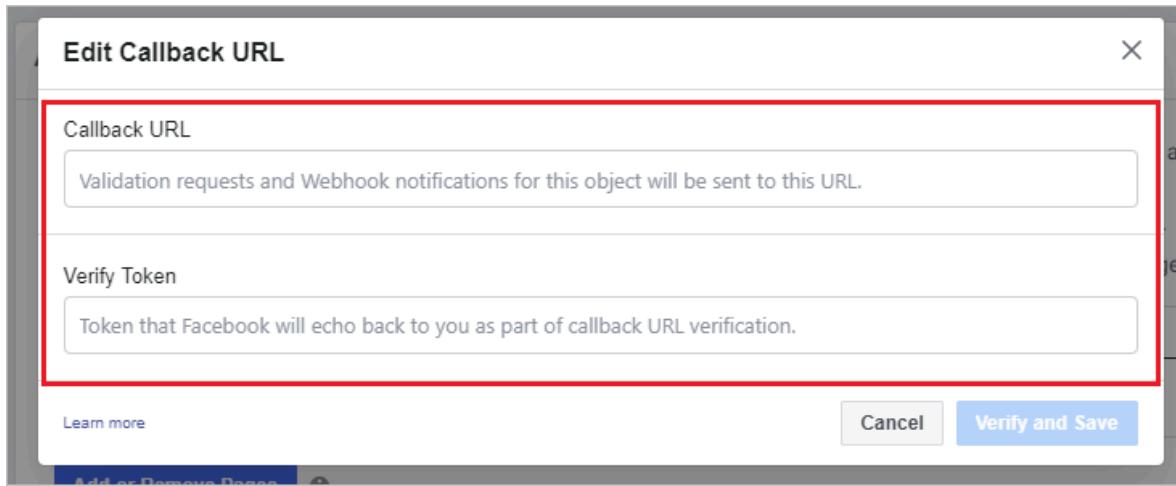
6. Select **Add** and wait for the success confirmation message. Once successful, copy the *Callback URL* and *Verify token*. You need these elements when you [connect your Facebook app to Copilot Studio](#).

## Connect your Facebook app to Copilot Studio

1. Sign in to the Facebook app that you want to add your agent to at [Facebook for Developers](#).
2. Select **Settings** under **Products and Messenger** on the side menu pane.
3. Under the **Webhooks** section, select **Add Callback URL**.



4. Provide the *Callback URL* and *Verify token* from the [Configure the Facebook channel in Copilot Studio](#) section and select **Verify and Save**.



5. Select **Add Subscriptions** for each page that you want to add the agent to.

**Webhooks**

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Callback URL  Verify Token

Validation requests and Webhook notifications for this object will be sent to this URL. Token that Facebook will echo back to you as part of callback URL verification.

[Edit Callback URL](#) [Show Recent Errors](#)

Pages ↑	Webhooks
 [REDACTED]	0 Fields —
 [REDACTED]	0 Fields —

[Add or Remove Pages](#) [?](#)

[Add Subscriptions](#) [Add Subscriptions](#)

6. Select the following fields:

- **messages**
- **messaging\_postbacks**
- **messaging\_options**
- **message\_deliveries**

7. Select Save.

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

**Edit Page Subscriptions** [X](#)

 [REDACTED]

**Subscription Fields**

<input checked="" type="checkbox"/> messages	<input checked="" type="checkbox"/> messaging_postbacks	<input checked="" type="checkbox"/> messaging_optins
<input checked="" type="checkbox"/> message_deliveries	<input type="checkbox"/> message_reads	<input type="checkbox"/> messaging_payments
<input type="checkbox"/> messaging_pre_checkouts	<input type="checkbox"/> messaging_checkout_updates	<input type="checkbox"/> messaging_account_linking
<input type="checkbox"/> messaging_referrals	<input type="checkbox"/> message_echoes	<input type="checkbox"/> messaging_game_plays
<input type="checkbox"/> standby	<input type="checkbox"/> messaging_handovers	<input type="checkbox"/> messaging_policy_enforcement

[Learn more](#) [Cancel](#) [Save](#)

Built-In NLP

## Submit for Facebook review

You must submit your app for Facebook review before you can make your Facebook app public. Facebook requires a Privacy Policy URL and Terms of Service URL. You must

provide those elements on the Facebook basic app settings page (after signing in to your app at [Facebook for Developers](#), select **Basic** under **Settings** on the side menu pane).

The [Code of Conduct](#) page contains non-Microsoft resources to help create a privacy policy. The [Terms of Service](#) page contains sample terms to help create an appropriate Terms of Service document.

Facebook has its own [review process](#) for apps that are published to Messenger. You can learn more about it at [Common rejection reasons](#). Your agent will be tested to ensure it's compliant with [Facebook's Platform Policies](#) before Facebook approves it to become public.

## Make the app public and publish the page

Until the app is published, it's in [Development Mode](#). The agent isn't public and it only works for admins, developers, and testers.

After the review is successful, in the app's **Dashboard** under **App Review**, set the app to **Public**. Ensure that the Facebook Page associated with the agent is published. The status appears in the **Pages** settings.

## Remove your agent from Facebook Messenger

When you don't want the agent to be reachable in Facebook Messenger, you can remove the agent from Facebook.

1. Open your agent in Copilot Studio.
2. On the top menu bar, select **Channels**.
3. Select the **Facebook** tile to open the configuration window.
4. Select **Delete**.

## Updating agent content for existing conversations on Facebook

To prevent disruptions during a chat between a user and the agent, existing conversations aren't updated to the latest agent content immediately after a new publish. Instead, the content is updated after the conversation has been idle for 30 minutes.

New conversations between a user and the agent have the latest published agent content.

## Known limitations

- You need to create a new agent if the Customer Satisfaction (CSAT) survey shows up as a non-interactive card image for an agent created during public preview to access the latest CSAT content.
- It might take a few minutes before the agent becomes reachable by users on Facebook pages after the Facebook channel is added.
- After you remove the Facebook channel, it might take a few minutes before the agent is removed fully and becomes unreachable on Facebook Messenger.
- After you remove a Facebook page, it might take a few minutes before the agent becomes unreachable by visitors to the removed page through Facebook Messenger.

---

## Feedback

Was this page helpful?



[Provide product feedback ↗](#)

# Publish an agent to Azure Bot Service channels

06/17/2025

You can connect your agent to existing [Azure Bot Service channels](#) which can be helpful if you want to connect your agent to customers on Azure Bot Service channels.

Adding your agent to Azure Bot Service channels requires considerable developer expertise. This article is written for IT admins or developers who have experience developing and writing code.

## 💡 Tip

You do not need to follow this document to add your Copilot Studio agent to your [website, Facebook, or Microsoft Teams](#). If your goal is to connect to a custom web-based or native app, your developers can learn more at [Publish an agent to mobile or custom apps](#).

## ⓘ Important

Instructions in this section require software development from you or your developers. It is intended for experienced IT professionals, such as IT admins or developers who have a solid understanding of developer tools, utilities, and IDEs.

## Prerequisites

- An [Azure Bot Service subscription](#).
- An Azure Bot Service bot using v4 SDK.
- [.NET Core SDK](#) version 2.1.
- Nuget package [Microsoft.Bot.Connector.DirectLine](#).
- An agent created in Copilot Studio that you want to connect to an Azure Bot Service channel.
- [Publish an agent to mobile or custom apps](#).

## Code samples

Code snippets used in this document are from [relay bot sample code](#).

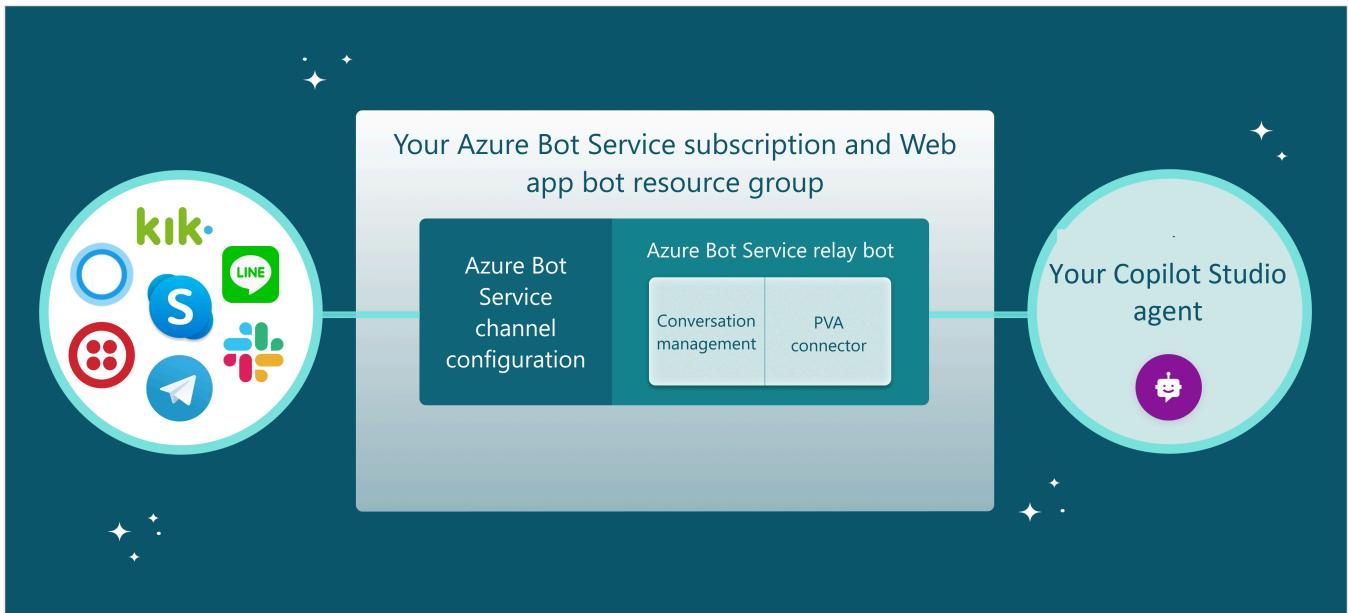
## References

The instructions in this document reference the following documents:

- [Deploy your bot to Azure](#) for instructions on deploying the Azure Bot Service bot.
- [Azure Bot Service Channels](#) to connect to any Azure Bot Service-supported channel.
- [Azure Bot Service debug with the emulator](#) for instructions on debugging the Azure Bot Service bot.

## Create or use an existing Azure Bot Service bot

You need an Azure Bot Service bot that can relay conversations between your Copilot Studio agent and Azure Bot Service channels.



The [relay bot sample code](#) is a good starting point if you don't have an existing Azure Bot Service bot. It's built from Microsoft Bot Framework bot [sample code](#) that can be compiled and deployed to the Azure Bot Service. The sample code is meant to be used as a starting point and not intended to be used in production directly. You need to add code and optimization to match your business needs.

If you already have an Azure Bot Service bot, you need to add a Copilot Studio connector and code to manage conversation sessions. You can then deploy the bot to the Azure Bot Service and connect to channels with the Azure portal.

## Get your Copilot Studio agent parameters

To connect to the agent you built with Copilot Studio, you need to retrieve your agent's name and token endpoint.

1. Copy your agent's name in Copilot Studio.

The screenshot shows the Copilot Studio interface. On the left, a navigation menu includes 'Chatbots', 'Overview', 'Topics', 'Entities', 'Analytics', 'Publish', and 'Settings'. The 'Settings' item is currently selected and expanded, showing 'Details', 'Channels', 'Agent transfers', 'Security', 'AI capabilities', and 'Test your bot'. The main area displays a card for 'Contoso Bot' with a red box around the title. Below the title are 'View Solution (Default Solution)' and a 'Quickstart with topic suggestions' section containing the text: 'Use AI to find question and answer pairs in your organization's documentation or support'.

2. In the navigation menu under **Settings**, select **Channels**.

3. Select the channel you want to connect to. This scenario uses Slack as an example.

The screenshot shows the 'Channels' section of the Copilot Studio settings. The 'Channels' item is highlighted with a red box. To its right are cards for various channels: 'Mobile app', 'Facebook', 'Skype', 'Cortana', 'Slack' (which is highlighted with a red box), 'Telegram', 'Twilio', 'Line', and 'Kik'. Each card has a description below it: 'Add your bot to a native or web-based mobile app.', 'Connect with your customers on Messenger.', 'Expand your bot's reach to customers on Skype.', 'Expand your bot's reach to customers on Cortana.', 'Expand your bot's reach to customers on Slack.', 'Expand your bot's reach to customers on Telegram.', 'Expand your bot's reach to customers on Twilio.', 'Expand your bot's reach to customers on Line.', and 'Expand your bot's reach to customers on Kik.'

4. To copy and save the **Token Endpoint** value, select **Copy**. You need your endpoint to connect your agent to the Azure Bot Service channel.

## Slack

X

Engage your customers on the channel that's right for you. Leverage the resources and sample code we provide to develop a solution that fits your business needs. [Learn more about channel configuration](#)

### Get connected

Copy and provide the information below to your developers so they can use our sample code to integrate with the channel.

#### Token Endpoint

[https://\[REDACTED\].com](https://[REDACTED].com)

[Copy](#)

## Manage conversation sessions with your Copilot Studio agent

There can be multiple conversations between the Azure Bot Service channels and the Direct Line connection with your Copilot Studio agent.

Your Azure Bot Service bot needs to map and relay the conversation from the Azure Bot Service channel to the Direct Line conversation with the Copilot Studio agent and vice versa.

## Sample code example

The following example uses samples from the [relay bot sample code ↗](#).

1. On every new external Azure Bot Service channel conversation start, start a Copilot Studio agent conversation. Refer to [Get Direct Line token](#) and [Use Direct Line to communicate with the agent](#) for instructions on starting a new conversation with the bot.

C#

```
using (var httpRequest = new HttpRequestMessage())
{
 httpRequest.Method = HttpMethod.Get;
 UriBuilder uriBuilder = new UriBuilder(TokenEndPoint);
 httpRequest.RequestUri = uriBuilder.Uri;
 using (var response = await s.httpClient.SendAsync(httpRequest))
 {
 var responseString = await response.Content.ReadAsStringAsync();
 string token = SafeJsonConvert.DeserializeObject<DirectLineToken>(responseString);
 }
}
```

```

 (responseString).Token;
 }
}

/// <summary>
/// class for serialization/deserialization DirectLineToken
/// </summary>
public class DirectLineToken
{
 public string Token { get; set; }
}

```

C#

```

// Use the retrieved token to create a DirectLineClient instance
using (var directLineClient = new DirectLineClient(token))
{
 var conversation = await
directLineClient.Conversations.StartConversationAsync();
 string conversationId = conversation.ConversationId;
}

```

2. To manage multiple sessions, you need to maintain a mapping of external Azure Bot Service channel conversations to corresponding Copilot Studio agent conversations. A Copilot Studio agent conversation can be identified with and connected with two properties: ConversationId and Token.

C#

```

Dictionary<string, PowerVirtualAgentsConversation> ConversationRouter = new
Dictionary<string, PowerVirtualAgentsConversation>();

```

To manage the conversation lifecycle, refresh the Direct Line tokens or clean up idled conversations. Learn more about token refresh at [Refresh Direct Line token](#). A Copilot Studio agent conversation to support refreshing Direct Line tokens is defined as follows:

C#

```

/// <summary>
/// Data model class for Copilot Studio agent conversation
/// </summary>
public class PowerVirtualAgentsConversation
{
 public string ConversationId { get; set; } // The Copilot Studio agent
conversation ID retrieved from step 1

 public string Token { get; set; } // The DirectLine token retrieved from
step 1

```

```

 public string WaterMark { get; set; } // Identify turn in a conversation

 public DateTime LastTokenRefreshTime { get; set; } = DateTime.Now; //
Timestamp of last token refresh

 public DateTime LastConversationUpdateTime { get; set; } = DateTime.Now;
// Timestamp of last active user message sent to agent
}

```

3. When a new Copilot Studio agent conversation starts, add a key value pair

(`external_Azure_Bot_Service_channel_conversationID`, `PowerVirtualAgentsConversation`) to the mapping table.

C#

```

// After new Copilot Studio agent conversation starts
ConversationRouter[external_Azure_Bot_Service_channel_conversationID] = new
PowerVirtualAgentsConversation()
{
 Token = token,
 ConversationId = conversationId,
 WaterMark = null,
 LastConversationUpdateTime = DateTime.Now,
 LastTokenRefreshTime = DateTime.Now,
};

```

4. To continue on an existing conversation, upon a new external Azure Bot Service channel message received, retrieve the existing conversation from the mapping table, relay the external conversation activity to your Copilot Studio agent, and get a response.

The following sample shows relaying conversation by overriding the `ActivityHandler.OnMessageActivityAsync(ITurnContext<IMessageActivity>, CancellationToken)` method

C#

```

// Invoked when a message activity is received from the user
// Send the user message to Copilot Studio agent and get response
protected override async Task
OnMessageActivityAsync(ITurnContext<IMessageActivity> turnContext,
CancellationToken cancellationToken)
{
 // Retrieve agent conversation from mapping table
 // If not exists for the given external conversation ID, start a new
Copilot Studio agent conversation
 ConversationRouter.TryGetValue(externalCID, out
PowerVirtualAgentsConversation currentConversation) ?
 currentConversation : /*await

```

```

StartBotConversationAsync(externalCID)*/;

 // Create DirectLine client with the token associated to current
 conversation
 DirectLineClient client = new
 DirectLineClient(currentConversation.Token);

 // Send user message using directlineClient
 await
client.Conversations.PostActivityAsync(currentConversation.ConversationId,
new DirectLineActivity()
{
 Type = DirectLineActivityTypes.Message,
 From = new ChannelAccount { Id = turnContext.Activity.From.Id, Name =
turnContext.Activity.From.Name },
 Text = turnContext.Activity.Text,
 TextFormat = turnContext.Activity.TextFormat,
 Locale = turnContext.Activity.Locale,
});

 // Update LastConversationUpdateTime for session management
 currentConversation.LastConversationUpdateTime = DateTime.Now;
}

```

5. Refer to [Use Direct Line to communicate with the agent](#) for how to get the Copilot Studio agent's response. When the Copilot Studio agent's response is received, refer to [Parse conversation payload from the agent](#) for how to parse the response to the external Azure Bot Service channel response.

An example of response parsing can be found in the [relay bot sample code ↗](#) ResponseConverter.cs.

## Deploy to Azure Bot Service

After you have your Azure Bot Service relay bot ready, you need to [deploy the bot to your Azure Bot Service](#).

## Set up Azure Bot Service channels

You can set up the channels you want to connect to by signing in to the Azure portal and selecting the Azure Bot Service resource group you deployed to. View the specific instructions for each channel at [Azure Bot Service Channels](#).

# Publish an agent to SharePoint

09/06/2025

With your agent published, you can make your agent available to your SharePoint users. You must [publish the agent](#) before your SharePoint users can interact with the agent.

## Configure the SharePoint channel

After publishing your agent, you can configure the agent for the SharePoint channel.

### Important

- To perform this procedure, you must have *WRITE* access to the SharePoint site.
- To use a Copilot Studio agent in SharePoint, one of the following must be true:
  - The SharePoint user has an M365 Copilot license, **OR**
  - SharePoint *Pay As You Go* (PayG) is enabled in Microsoft Admin Center (MAC) for the user's tenant **AND** billing capability (prepaid messages or PayG) is set up for the agent's environment in Power Platform admin center.
- The agent follows Copilot Studio billing policies and consumes Copilot Studio capacity, despite the agent being used in SharePoint.

1. On the top menu bar, select **Channels**.
2. Select the **SharePoint** channel to open the configuration window.



## Published agent status

Published by [REDACTED] 5/8/2025, 4:56 PM

## Share a preview

## Microsoft channels

Demo website

Teams and Microsoft 365 Copilot

SharePoint

## Other channels

Web app

Native app

Facebook

Slack

Telegram

Twilio

Line

GroupMe

Direct Line Speech

Email

## Customer engagement hub

Connect to a customer engagement app to enable your agent to hand off a chat session to a live agent or other agent.

Dynamics 365 Customer Service

Telephony

Genesys

LivePerson

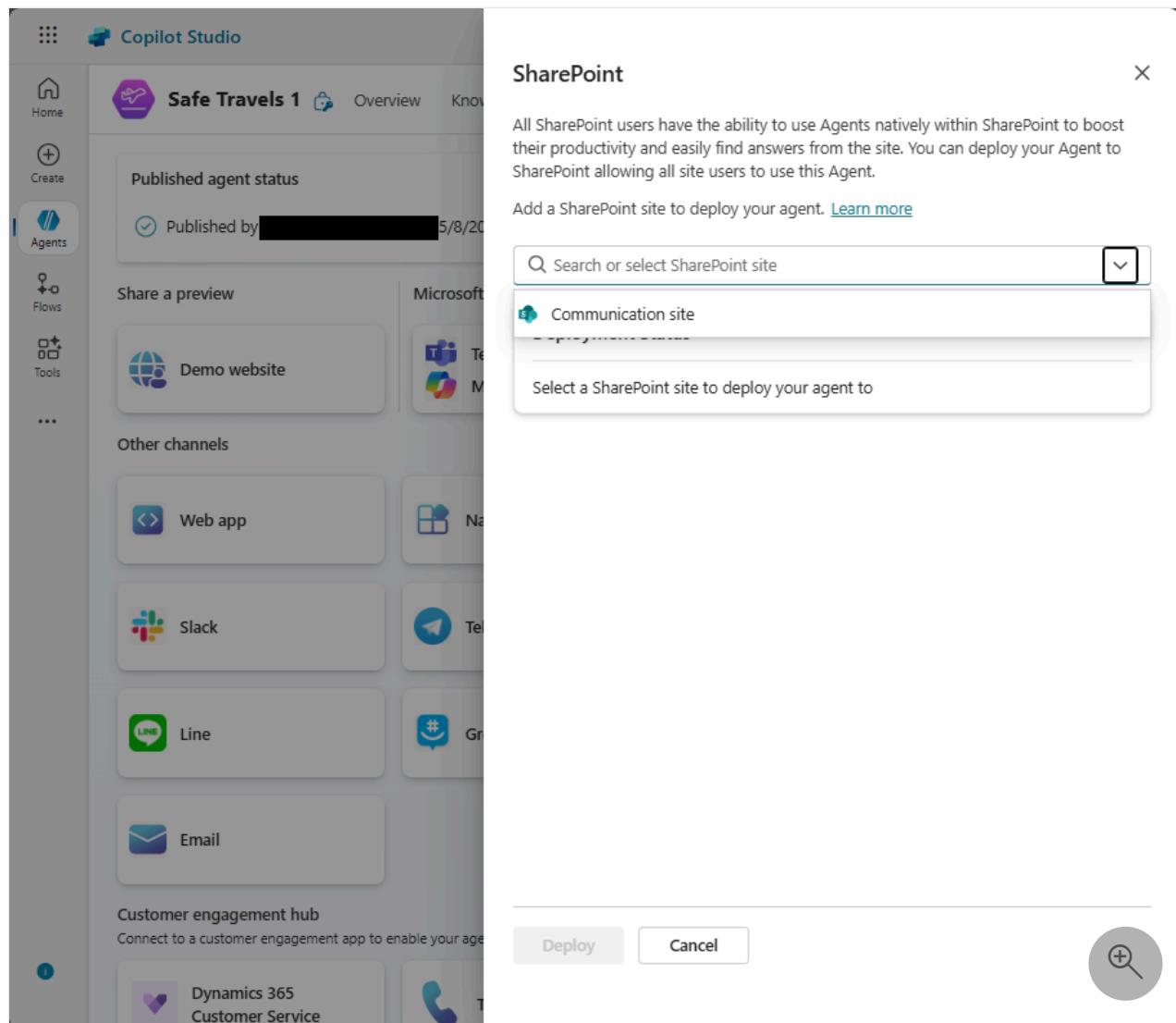
Salesforce

ServiceNow

Custom engagement hub



3. Select your SharePoint site by either using the drop-down menu or entering the name of your SharePoint site.



The drop-down allows the user to select recently used SharePoint sites for easy access. Users can either use the search bar or the drop-down menu.

4. With the SharePoint site selected, select **Deploy**, then **Confirm** to finalize the deployment.
5. Once confirmed, your SharePoint site displays a successful deployment message.

## SharePoint

X

✓ Successfully deployed 1 items

X

All SharePoint users have the ability to use Agents natively within SharePoint to boost their productivity and easily find answers from the site. You can deploy your Agent to SharePoint allowing all site users to use this Agent.

Add a SharePoint site to deploy your agent. [Learn more](#)

Q Search or select SharePoint site

▼

### Deployment Status

Communication site

Deployed ...

If you would like to make this agent easily discoverable to all users of the SharePoint site, please work with the site owner to mark it 'Approved'. [Learn more](#)

Undeploy

Cancel



## Test your deployed agent

Once you deploy your agent to the SharePoint site, you can begin testing it. We recommend you test the agent in both the agent's conversation window and the SharePoint site's Copilot window before promoting the agent to your SharePoint site. To see the agent in Sharepoint, in the Copilot Studio configuration window for SharePoint, select the ellipsis menu in the Deployment Status area, and then select **Open Agent in SharePoint**.

**Deployment Status**

[Communication site](#)      Deployed      [...](#)

**Information:** If you would like to make this agent easily discoverable to all users of the SharePoint site, please work with the site owner to mark it 'Approved'. [Learn more](#)

[Copy Site URL](#)      [Open Agent in SharePoint](#)

## Undeploy the agent from a SharePoint site

Users can choose to remove an agent from a SharePoint site if desired. On the agent's configuration page for SharePoint in Copilot Studio, select **Undeploy**.

**SharePoint** [X](#)

Successfully deployed 1 items [X](#)

All SharePoint users have the ability to use Agents natively within SharePoint to boost their productivity and easily find answers from the site. You can deploy your Agent to SharePoint allowing all site users to use this Agent.

Add a SharePoint site to deploy your agent. [Learn more](#)

Search or select SharePoint site [▼](#)

**Deployment Status**

[Communication site](#)      Deployed      [...](#)

**Information:** If you would like to make this agent easily discoverable to all users of the SharePoint site, please work with the site owner to mark it 'Approved'. [Learn more](#)

[Undeploy](#)      [Cancel](#)

# Enable all SharePoint users to find and use your agent

Once you deploy your agent to SharePoint, it's important to test it directly in SharePoint to verify its usability. If you want your agent to be easily visible to all SharePoint users for the site, marking the agent as *Approved* is an effective way to highlight it. An *Approved* agent appears in a dedicated **Approved** section when using SharePoint and is visible and usable to all SharePoint users in the Agent picker side pane.

For more information, see [Manage agents in SharePoint](#).

# Publish an agent to WhatsApp

07/21/2025

With your agent published, you can make your agent available to your WhatsApp users. You must [publish the agent](#) before your WhatsApp users can interact with the agent.

## Note

When you publish your agent to WhatsApp, some data such as agent content and user chat content is shared with WhatsApp (meaning your data flows outside of your [organization's compliance and geographic or regional boundaries](#)).

For more information, see [WhatsApp Business Platform Policy and Spam Enforcement](#).

## Set up your agent to use WhatsApp

### Important

Make sure [all prerequisites are met](#) before beginning this procedure.

1. Make sure your agent's authentication is set to either [No authentication](#) or [Authenticate manually](#).
2. Make sure your agent is already published. See [Publish the latest content](#).
3. Go to the **Channels** page for your agent.
4. Select **WhatsApp**. The configuration panel appears.
5. On the first page of the configuration panel, select **Continue**.
6. Select your Azure subscription and Azure Communication Services (ACS) resource, then select **Continue**.
7. Select the desired phone number for your agent to connect to, then select **Deploy**.

### Note

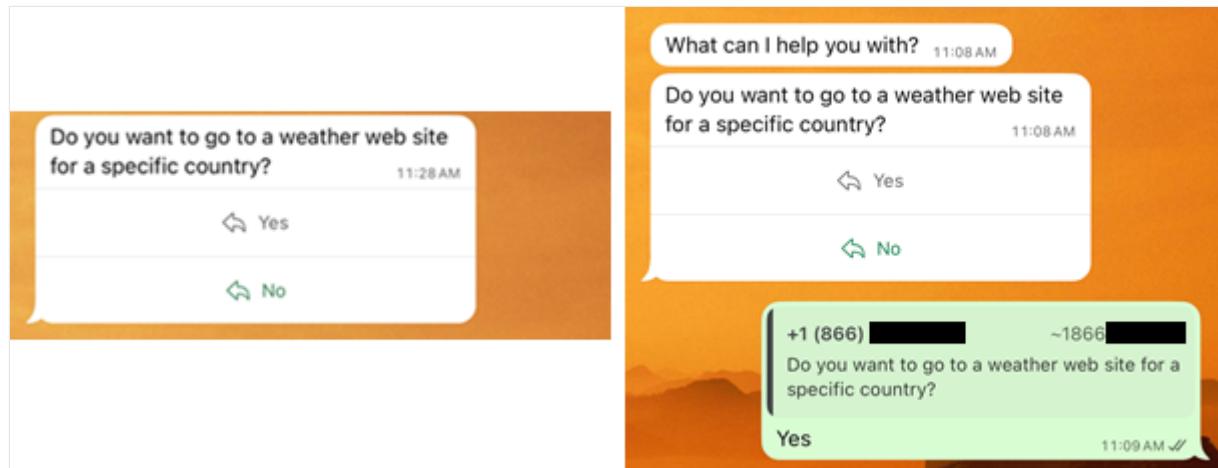
Once you select **Deploy**, Copilot Studio might take a moment before showing a QR code.

- With your WhatsApp device, scan the QR code to begin chatting with the agent over WhatsApp. Alternatively, select **Download** to download the QR code image. Anyone with this QR code who meets all authentication criteria can chat to the agent over WhatsApp.

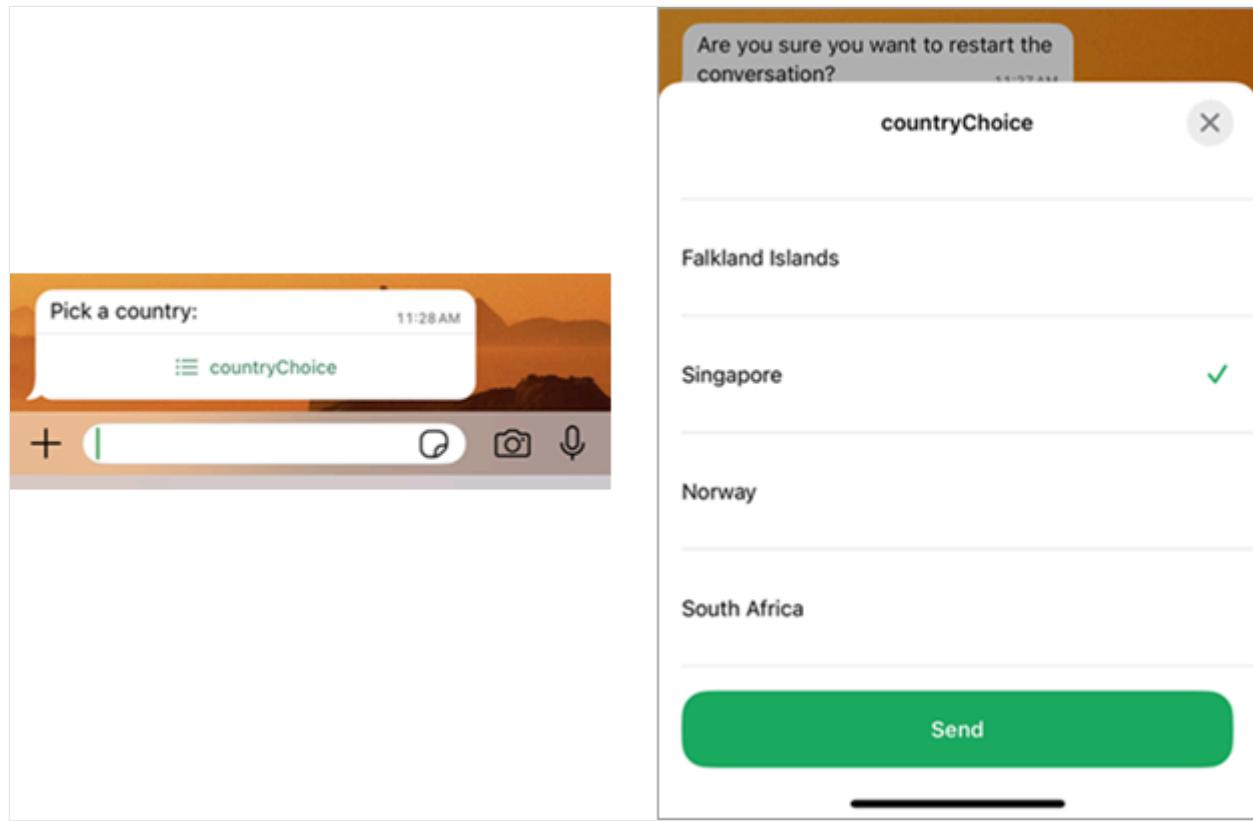
## Use supported Adaptive Cards in your agent's topics

As with all agents in Copilot Studio, agents deployed to WhatsApp make use of topics to define how an agent conversation progresses. Adaptive cards allow you to write platform-agnostic UI snippets in JSON that are interpreted into native UI when delivered to specific apps. However, WhatsApp agents support the following three types of adaptive cards:

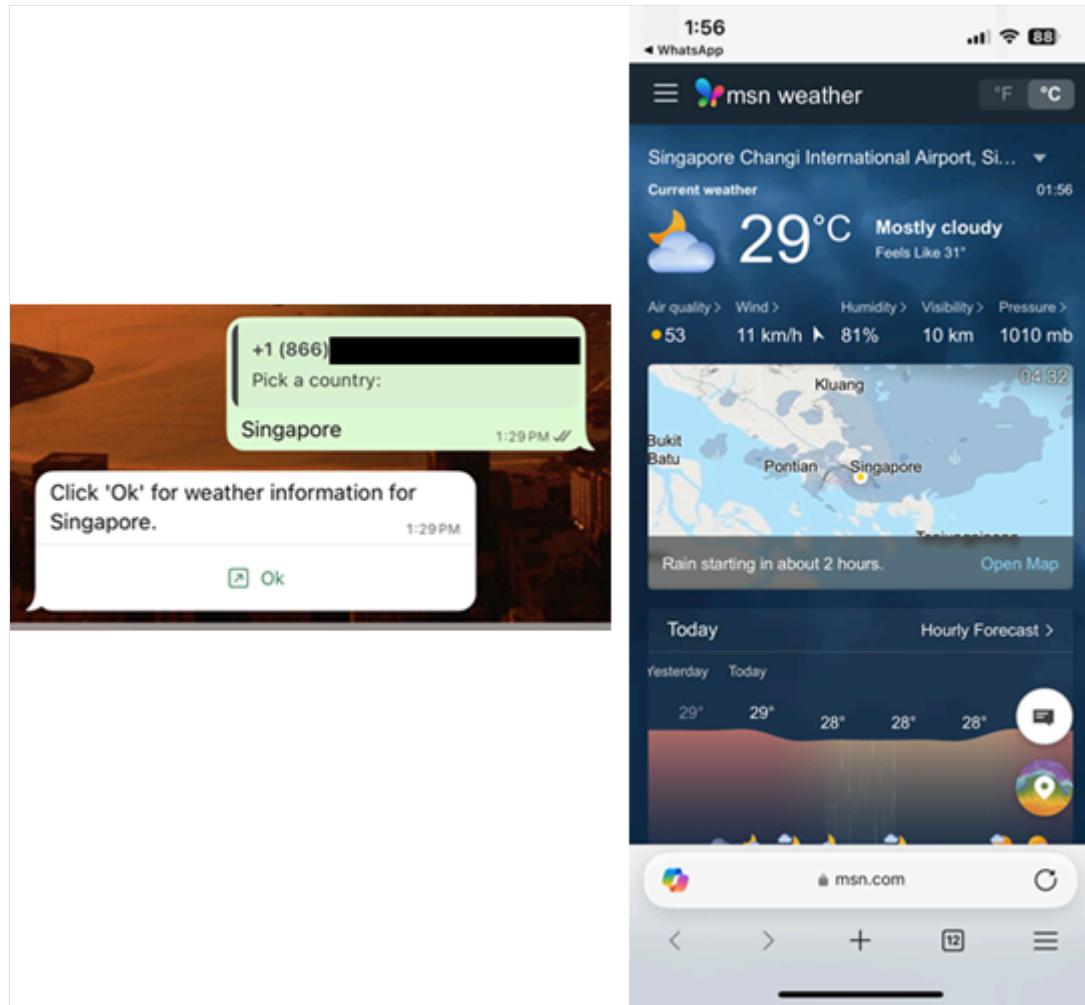
- Interactive actions* adaptive card (`"type": "Action.Submit"`)



- Choice* adaptive card (`"type": "Input.ChoiceSet"`)



- Open URL adaptive card ("type": "Action.OpenUrl")



As a maker, if you want to deploy an agent to a WhatsApp channel, you must therefore make sure you restrict your use of adaptive cards to this subset of card types. For more information

about the use of adaptive cards in Copilot Studio, see [Using Adaptive Cards in Copilot Studio](#).

## Example: Interactive actions adaptive card

Use the *Interactive Actions* adaptive card to allow end-users to select one of up to three options in the form of buttons. This method of selecting from multiple options is quicker for the end-user, with fewer steps, than the alternative method of selecting from a list of options (the *Choice* adaptive card method). However, the Interactive actions adaptive card can't be used for more than three buttons. Makers can invoke this adaptive card with the *Ask with Adaptive Card* functionality.

1. Go to the desired topic and [add an Adaptive Card node](#).
2. In the **Card payload editor** of the Adaptive card designer, copy and paste the following JSON code:

JSON

```
{
 "type": "AdaptiveCard",
 "$schema": "https://adaptivecards.io/schemas/adaptive-card.json",
 "version": "1.5",
 "body": [
 {
 "type": "TextBlock",
 "text": "Do you want to go to a weather web site for a specific
country?",
 "wrap": true
 },
 {
 "type": "ActionSet",
 "actions": [
 {
 "type": "Action.Submit",
 "title": "Yes",
 "id": "yesButtonId"
 },
 {
 "type": "Action.Submit",
 "title": "No",
 "id": "noButtonId"
 }
]
 }
]
}
```

3. In the JSON, in the section with a `type` key of value `ActionSet`, under `"actions":`, make sure you have the right number of actions.

For example, if you need three buttons, under `"actions":`, there should be three sets of actions, with each set comprised of a `type`, `title`, and `id` key.

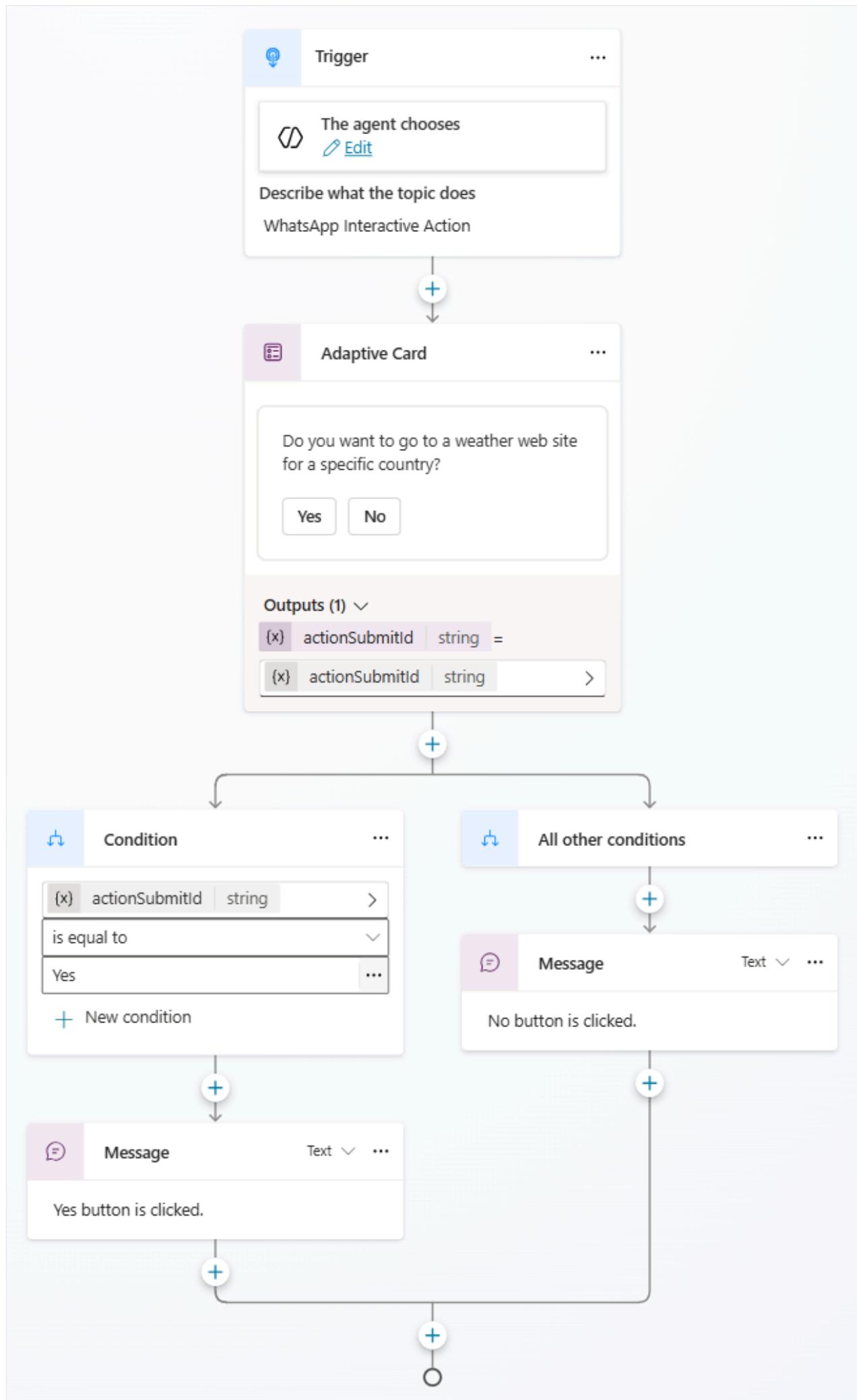
4. Make the following further changes to the JSON:

 [Expand table](#)

Change	To
Value of <code>"text":</code> (under <code>"type": "TextBlock"</code> )	The desired text that prompts end-users to make a choice from among the buttons
Value of <code>"title":</code> (under the first <code>"type": "Action.Submit"</code> key)	The desired text on the first button
Value of <code>"id":</code> (under the first <code>"type": "Action.Submit"</code> key)	The desired ID for the first button
Value of <code>"title":</code> (under the second <code>"type": "Action.Submit"</code> key)	The desired text on the second button
Value of <code>"id":</code> (under the second <code>"type": "Action.Submit"</code> key)	The desired ID for the second button

5. Select **Save** in the **Adaptive card designer**, and then select **Close**.

The authoring canvas for this topic might resemble this:



6. Save your topic.

7. Once you finish all changes to the agent, publish your agent again.

## Example: Choice adaptive card

Use the *Choice* adaptive card to allow end-users to select one option from up to 10 presented as a list. This adaptive card uses a text block the WhatsApp end-user must select which then displays the choice set of options. The end-user must then select one of the options, and then select **Send**. Makers can invoke this adaptive card with the *Ask with Adaptive Card* functionality.

1. In the authoring canvas of the topic you would like to add this adaptive card to, select the **Add node** icon in the desired location, then select **Ask with adaptive card**.
2. Bring the new **Adaptive Card** node into focus by selecting it, and then select the **More** icon for the node and select **Properties**.
3. On the **Adaptive Card Node** properties pane, select **Edit adaptive card**.
4. In the **Card payload editor** of the **Adaptive card designer**, copy and paste the following JSON code:

JSON

```
{
 "type": "AdaptiveCard",
 "$schema": "https://adaptivecards.io/schemas/adaptive-card.json",
 "version": "1.5",
 "body": [
 {
 "type": "TextBlock",
 "text": "Pick a country:",
 "wrap": true
 },
 {
 "type": "Input.ChoiceSet",
 "choices": [
 {
 "title": "Falkland Islands",
 "value": "Falkland Islands"
 },
 {
 "title": "Singapore",
 "value": "Singapore"
 },
 {
 "title": "Norway",
 "value": "Norway"
 }
]
 }
]
}
```

```

 {
 "title": "South Africa",
 "value": "South Africa"
 }
],
 "placeholder": "Select Country",
 "id": "countryChoice"
},
{
 "type": "ActionSet",
 "actions": [
 {
 "type": "Action.Submit",
 "title": "Submit Response",
 "id": "submitResponseId"
 }
]
}
]
}

```

5. In the JSON, in the section with a `type` key of value `Input.ChoiceSet`, under `"choices":`, make sure you have the right number of choices.

For example, if you need seven (7) selectable choices presented in a list, under `"choices":`, there should be seven sets of choices, with each set comprised of a `title` and `value` key.

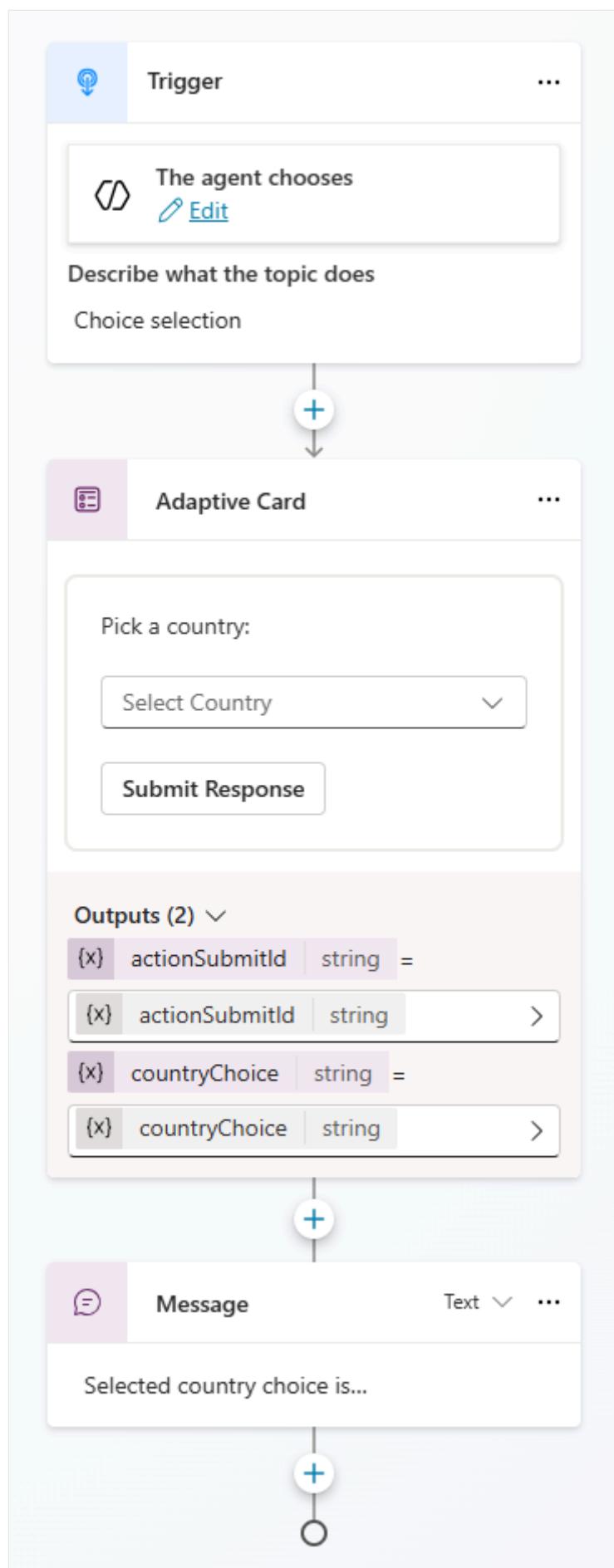
6. Make the following further changes to the JSON:

[Expand table](#)

Change	To
Value of <code>"text":</code> (under <code>"type": "TextBlock"</code> )	The desired text that prompts end-users to display a list of options
Value of <code>"title":</code> (for each <code>choice</code> under <code>"type": "Input.ChoiceSet"</code> key)	The desired text describing or naming the listed option
Value of <code>"value":</code> (for each <code>choice</code> under <code>"type": "Input.ChoiceSet"</code> key)	The value assigned to the <code>Input.ChoiceSet.value</code> , which is then used for further processing
Value of <code>"id":</code> (under <code>"type": "Input.ChoiceSet"</code> )	The ID value for the choice set variable (named <code>countryChoice</code> , in the example). This ID value takes on the recorded <code>value</code> signifying the user's choice from the list. <b>Note:</b> This value should be in <i>camelCase formal</i> format.

7. Select Save in the Adaptive card designer, and then select Close.

The authoring canvas for this topic might resemble the following:



8. Select **Save** in the authoring canvas to commit your adaptive card changes to the topic.
9. Once you finish all changes to the agent, republish the agent.

## Example: Open URL adaptive card

Use the *Open URL* adaptive card to send an end-user to a website. Unlike the other two adaptive cards, the Open URL adaptive card can't be invoked with the *Ask with Adaptive Card* functionality in Copilot Studio. Instead, you must attach this adaptive card to a message created with the *Send a message* functionality.

1. In the authoring canvas of the topic you would like to add this adaptive card to, select the **Add node** icon in the desired location, then select **Send a message**.
2. In the new **Message** node, select **+ Add > Adaptive card**.
3. Select the new **Media** area of the message node, then select the **More** icon for the node and select **Properties**.

 **Note**

Select the new **Media** area of the message node again if the **Edit adaptive card** button doesn't appear on the **Adaptive Card properties** pane.

4. On the **Adaptive Card properties** pane, select **Edit adaptive card**.
5. In the **Card payload editor** of the **Adaptive card designer**, copy and paste the following JSON code:

JSON

```
{
 "type": "AdaptiveCard",
 "$schema": "https://adaptivecards.io/schemas/adaptive-card.json",
 "version": "1.5",
 "body": [
 {
 "type": "TextBlock",
 "text": "Click 'Ok' for weather information for Singapore.",
 "wrap": true
 },
 {
 "type": "ActionSet",
 "actions": [
 {
 "type": "Action.OpenUrl",
 "title": "Ok",
 "url": "https://www.google.com"
 }
]
 }
]
}
```

```
 "url": "https://www.msn.com/en-za/weather/forecast/in-Singapore"
 }
]
}
]
```

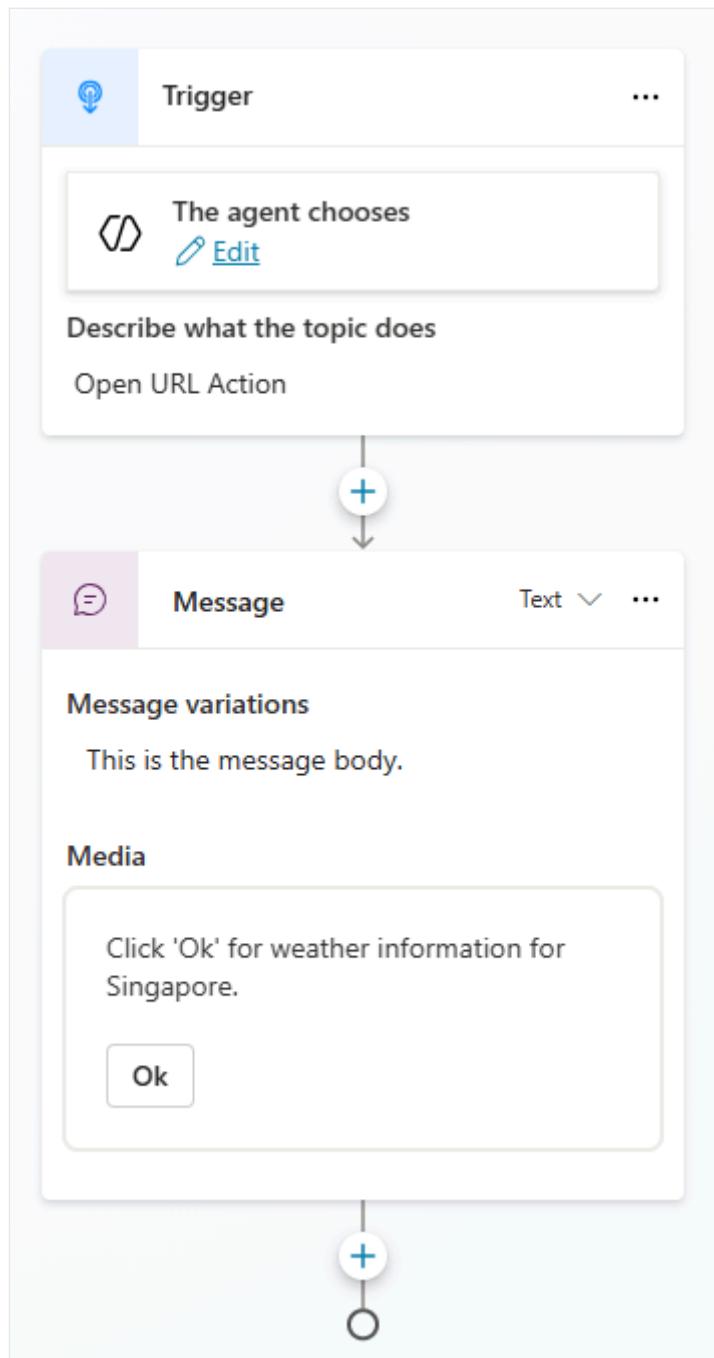
6. Make the following changes to the JSON:

 Expand table

Change	To
Value of <code>"text"</code> : (under <code>"type": "TextBlock"</code> )	The desired text that prompts end-users to open the website
Value of <code>"title"</code> : (under <code>"type": "ActionSet"</code> key)	The desired selectable text that opens the website in a browser when selected
Value of <code>"url"</code> : (under <code>"type": "ActionSet"</code> key)	The desired value of the website's uniform resource locator (URL)

7. Select **Save** in the Adaptive card designer, and then select **Close**.

The authoring canvas for this topic might resemble the following:



8. Select **Save** in the authoring canvas to commit your adaptive card changes to the topic.
9. Once you finish all changes to the agent, republish the agent.

## Configure user authentication using phone numbers in WhatsApp

When you publish an agent to WhatsApp, you can choose to authenticate or not to authenticate user. If you choose not to authenticate at all, you don't need to do anything further.

If you choose to authenticate, your options are:

- Authenticate manually (see [Authenticate manually](#))

- Authenticate using a phone number as a recognized WhatsApp ID (see [Example: Authenticate with a phone number](#))
- Authenticate manually *and* by using phone number recognition. Authenticate with both methods for added security.

 **Important**

Authenticating with Microsoft isn't supported for the WhatsApp channel.

## Example: Authenticate with a phone number

WhatsApp uses phone numbers as identifiers (IDs). When a maker chooses to use phone number authentication, a backend API verifies the number associated with the WhatsApp account of a given user is included in the list of authorized phone numbers in the database. You must configure a trigger in Copilot Studio to send an HTTP request to the API, manage the response schema, and set conditions to manage the conversation flow based on whether the user is registered or not.

 **Note**

Make sure the backend server provides an API to validate if a given phone number is registered in the database.

In our example, the server provides the following API:

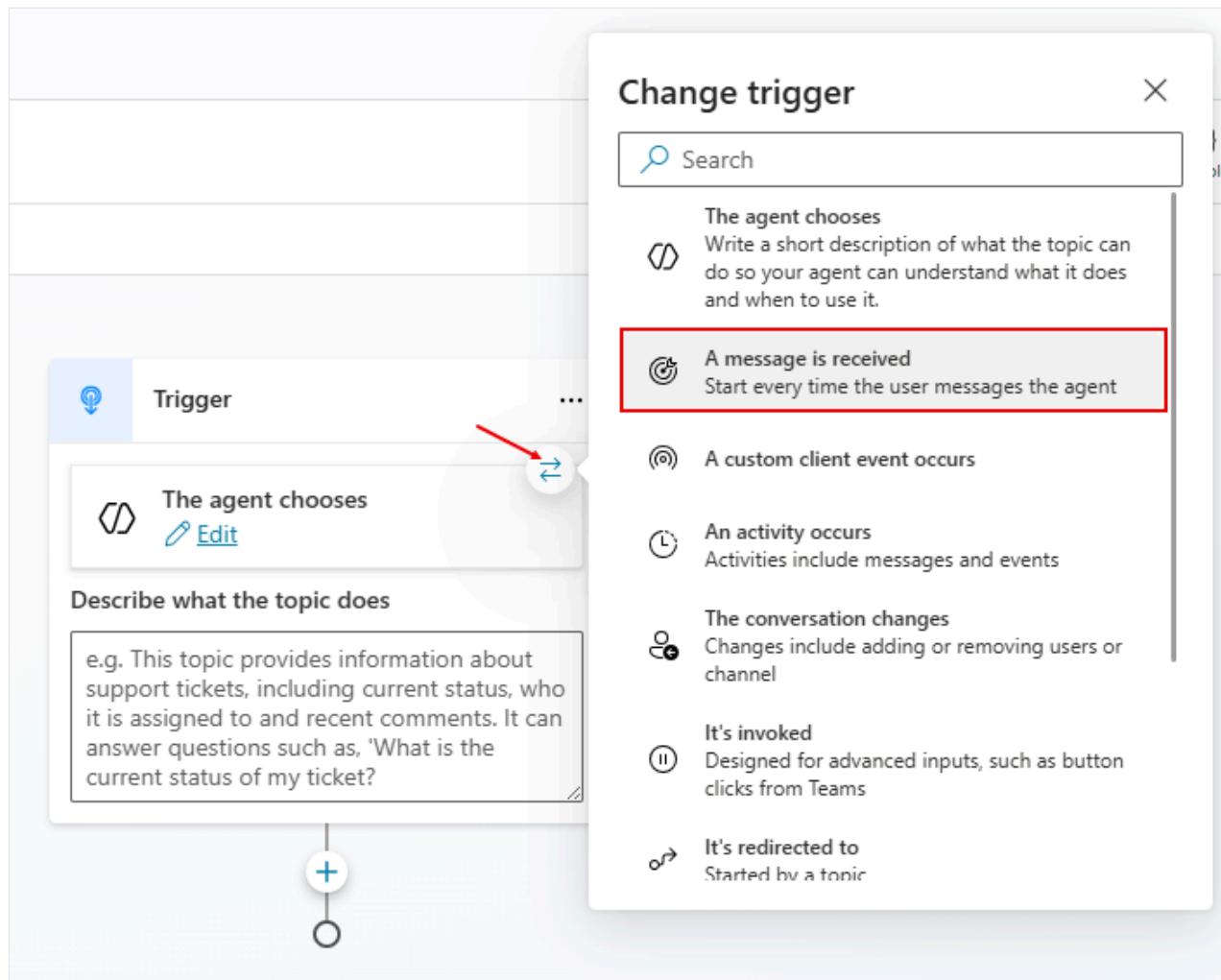
**URI:** /exists/\_{phoneNumber}\_

**Response:**

HTTP

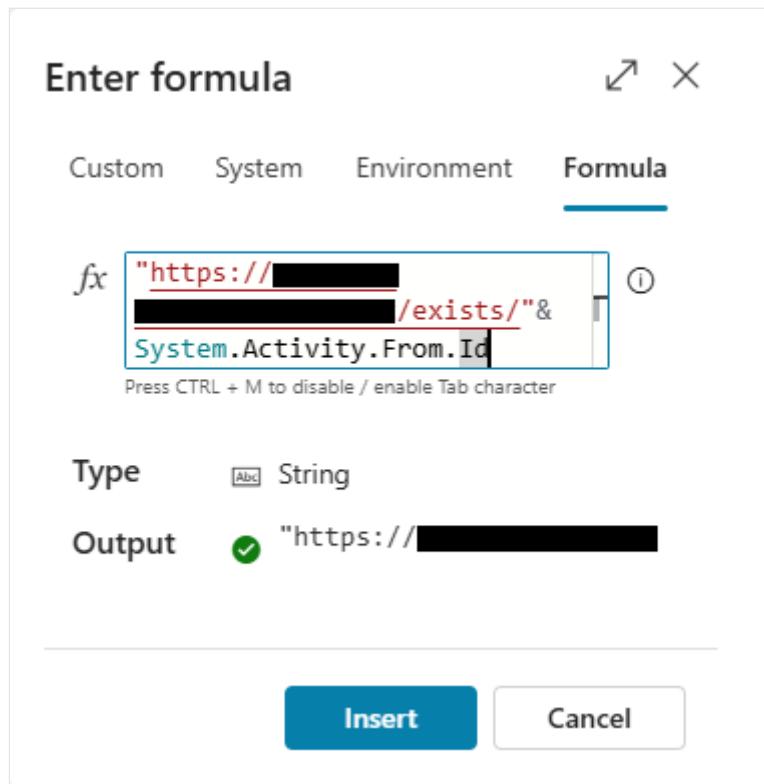
```
{
 phone: string
 exists: boolean
}
```

1. Go to the **Topics** page for your agent.
2. Select **Add a topic > From blank**.
3. Select the **Change trigger** icon  of the **Trigger** node, and then select **A message is received**.



4. Select the **Add node** icon under the trigger node and select **Variable management > Set a variable value**.
5. In the **Set variable value** node, select the **Set variable** field.
6. In the **Select a variable** window, select **Create new**.
7. In the **Set variable value** node, select the variable name in the **Set variable** field and then rename the variable name in the **Variable properties** pane.
8. In the **Set variable value** node, in the **To value** field, select the three dots (...).
9. In the **Enter formula** window, select the **Formula** tab.
10. Copy the following formula to the formula field, substituting `<><BaseUri>>` with the uniform resource identifier (URI) of the API, and then select **Insert**.

```
<><BaseUri>/exists/"&System.Activity.From.Id
```



11. Select the **Add node** icon under the **Set variable value** node and select **Advanced > Send HTTP request**.
12. Select **Record** under **Response data type**.
13. Select **Edit Schema** and provide the response schema.

```
HTTP

kind: Record
properties:
 exists: Boolean
 phone: String
```

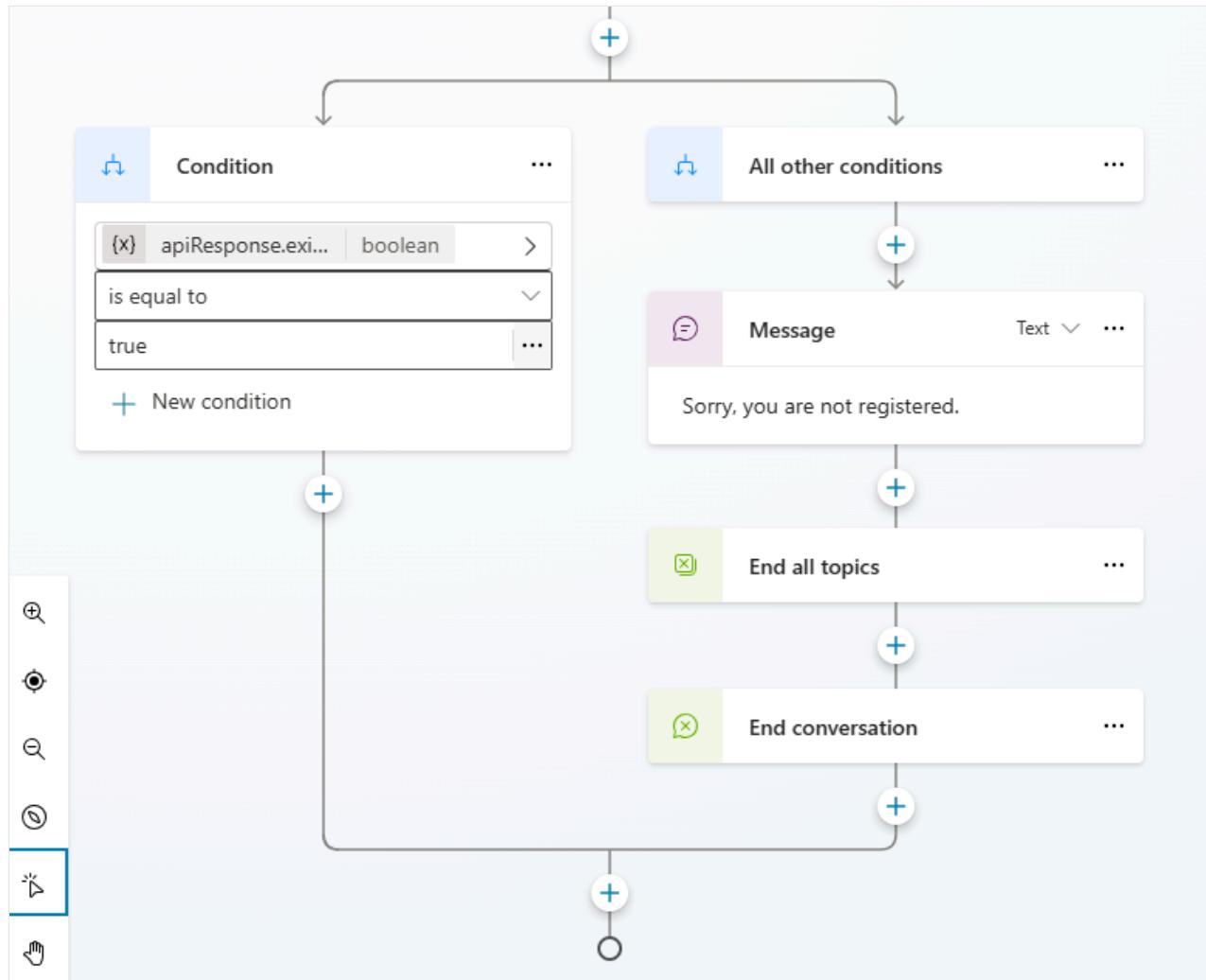
14. For **Save response as**, create a new variable named `apiResponse`.
15. Select the **Add node** icon under the **HTTP Request** node, and select **Add a Condition**.
16. On the **Condition** node, select **Select a variable > apiResponse.exists**.
17. On the second input field, select **Is equal to**.
18. Under **Enter or Select a value**, type `true`.
19. Select the **Add node** icon under the **All other conditions** node, and select **Send a message**.
20. Write a message to display to unregistered users.

**① Note**

You can customize the message sent to unregistered users. In this example, the message is `Sorry, you are not registered.`

21. Select the **Add node** icon under the **Message** node, and select **Topic management > End all topics**.

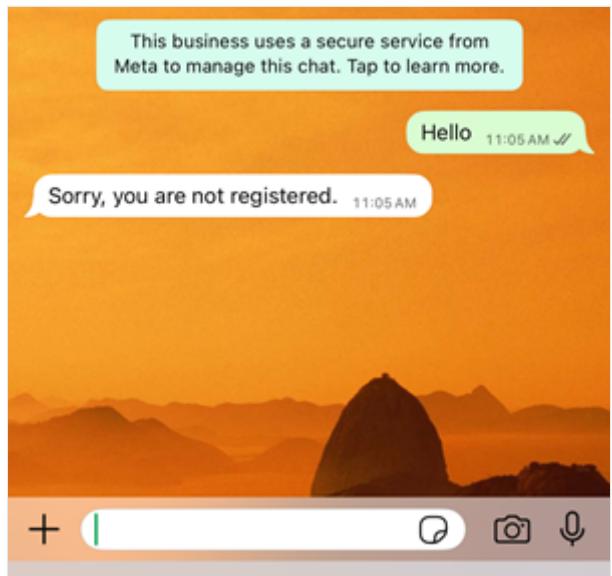
22. Select the **Add node** icon under the **End all topics** node, and select **Topic Management > End conversation**.



23. Select **Save** in the authoring canvas to commit your changes to the topic.

24. Once you finish all changes to the agent, republish the agent.

After this change is published, if the phone number associated with a user's WhatsApp device isn't registered in the database, the message you configured as the *unregistered* message appears.



## Disconnect an agent from a phone number

To disconnect your agent from the phone number configured for the WhatsApp channel, open the configuration panel and select **Disconnect**.

# Integrate with web or native apps using Microsoft 365 Agents SDK

07/16/2025

## Important

This content is intended for experienced IT professionals, such as IT admins or developers, who are familiar with developer tools, utilities, and integrated development environments (IDEs). It requires software development expertise.

After you create and test your agent in Copilot Studio, deploy it to your preferred channel to test how your target users will interact with it.

## Note

If you want to use the Embed code for the web app in Copilot Studio, you must set your security authentication options to **No authentication**. Navigate to **Settings > Security > Authentication** and select **No authentication** to make it publicly available.

This guide covers how to take your Copilot Studio agent and integrate it with your existing web application (typically a website) or native application using the Copilot Studio client with the Microsoft 365 Agents SDK.

## Note

For another option to connect a Copilot Studio or Microsoft 365 Agents SDK agent to a native mobile or Windows app, see [Integrate with native apps using the Agents Client SDK](#).

Different parts of this guide are relevant depending on whether you have an existing UX/UI that you want to integrate the agent into, or if you plan to use a Microsoft-provided UX/UI. Use the following table to find the section that fits your scenario.

 Expand table

Deployment Method	How, where, and why	Quick Link
Use the Default Web Chat Embed Code	With an agent that has <b>No Authentication</b> security setting enabled, it's available on the <b>Web</b> channel publishing pane in Copilot Studio. <b>Note:</b> This option only appears when you have <b>No Authentication</b> selected in Copilot Studio.	<a href="#">Use the default Web Chat Embed code (without development/code)</a>
Connect to Copilot Studio with Agents SDK User sign-in	Use the Agents SDK connection string or configuration settings to directly integrate to your agent using user credentials using the copilot.	<a href="#">Configure your app registration for user interactive sign-in</a>
Connect to Copilot Studio with Agents SDK Service Principal sign-in	To be used where you want the agent to have its own identity and not use on behalf of for the user accessing the agent. Useful in scenarios where you don't require user context but you still need to access privileged APIs or information and requires authentication. <b>Note:</b> To perform this task, you must have authentication for this agent in Copilot Studio set to <b>No Authentication</b> .	<a href="#">Configure your app registration for Service Principal</a>
Use legacy DirectLine to connect to Copilot Studio	You can connect to DirectLine using the DirectLine API and should be used where the Agents SDK doesn't support your scenario.	<a href="#">Publish an agent to mobile or custom apps</a>

## Prerequisites

- .NET Core / JS/ Python
- Packages—Copilot Studio client
- An agent

## Methods to Integrate your Copilot Studio agent

- **Copilot Studio client (using the Agents SDK):** This method is the preferred way to integrate with Copilot Studio.
- **DirectLine:** DirectLine is the legacy way to integrate with Copilot Studio and should be used where there's no support for your use case with Copilot Studio client.
  - Doesn't support service principal token

## Get Started: Basic configuration and agent connection testing using the Microsoft 365 Agents SDK

## 1. Download the Copilot Studio client Sample from the Microsoft 365 Agents SDK.

We simplified integration with the Agents SDK for your web and native apps by providing a client library that allows developers to enter a few details about your agent and to easily integrate it into your applications.

## 2. Get the [Copilot Studio client sample](#) from the Agents SDK GitHub repo in either C#, JavaScript or Python.

## 3. Get the Embed code or connection string for your agent in Copilot Studio.

You need access to your agent in Copilot Studio to get the connection string details (or details for the configuration setting) to be able to connect to it.

In Copilot Studio, open your agent, select **Settings > Security > Authentication**, and then review your agent's settings.

- If either **Authenticate with Microsoft** or **Authenticate manually** is selected, you see only the *connection string* option to integrate with Agents SDK.
- If **No authentication** is selected, you see both the Embed code that you can add to your website and the connection string. The Embed code option doesn't use the Agents SDK and uses the standard out-of-the-box Web Chat component.

The screenshot shows the 'Authentication' settings page. At the top, there is a back arrow and the title 'Authentication'. Below the title, a note says: 'When enabled, your agent will use the selected LLM's stored information and understanding to generate responses.' A section titled 'Choose an option' contains three radio buttons:

- No authentication ⚠️  
Publicly available in any channel
- Authenticate with Microsoft  
Entra ID authentication in Microsoft Teams, Power Apps, or Microsoft 365 Copilot
- Authenticate manually  
Set up authentication for any channel

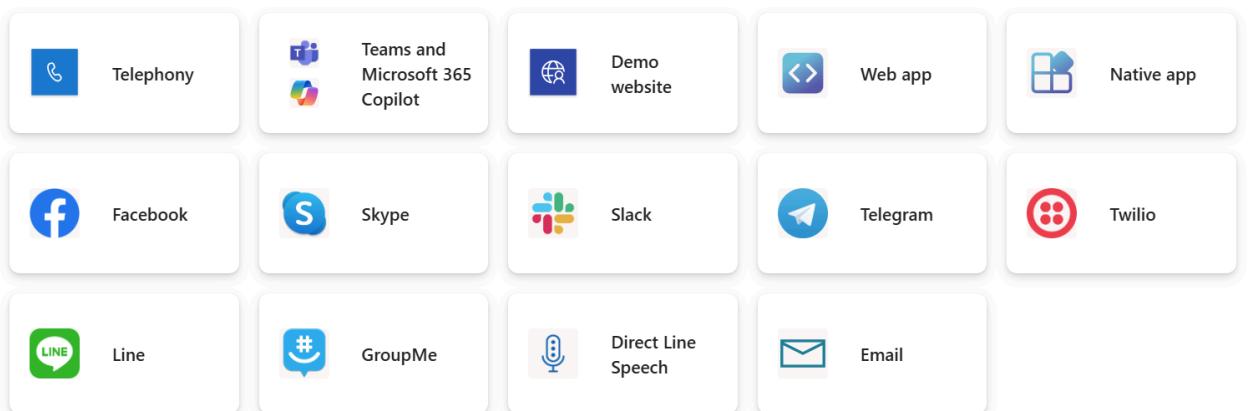
A magnifying glass icon is located in the bottom right corner of the form area.

## 4. Get your connection string.

To get the connection string for your agent in Copilot Studio, select either **Web app** or **Native app** on the **Channels** page. Select **Copy** next to the connection string under **Microsoft 365 Agents SDK**. Use this string to connect to this agent from your web or native app's code.

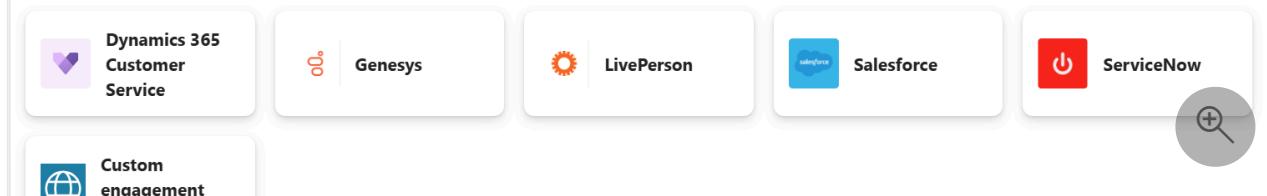
## Channels

Configure your agent channels to meet your customers where they are.



## Customer engagement hub

Connect to a customer engagement app to enable your agent to hand off a chat session to a live agent or other agent.



**Copilot Studio**

Agent Overview Knowledge Topics Actions

Channels

Configure your agent channels to meet your customers where they are.

Telephony Teams and Microsoft 365 Copilot

Facebook Skype

Line GroupMe

Customer engagement hub

Connect to a customer engagement app to enable your agent to hand off a chat session to a live agent or other agent.

Dynamics 365 Customer Service Genesys

Custom engagement

## Web app

There's two ways to add an agent to a web app. The embed code option lets you add an agent to a web-based app with a copied code snippet, while the Microsoft 365 Agents SDK offers a fully integrated experience.

### Embed code

Copy the following code snippet and paste it into your HTML web app. If you don't have access to your code base, share the snippet with the person responsible for your web app.

[REDACTED]

Copy

### Microsoft 365 Agents SDK

To integrate a Copilot Studio agent with anonymous security settings using Python, Javascript, or .NET, copy the connection string and paste it into your app's code. [Learn More](#)

### Connection string

[REDACTED]

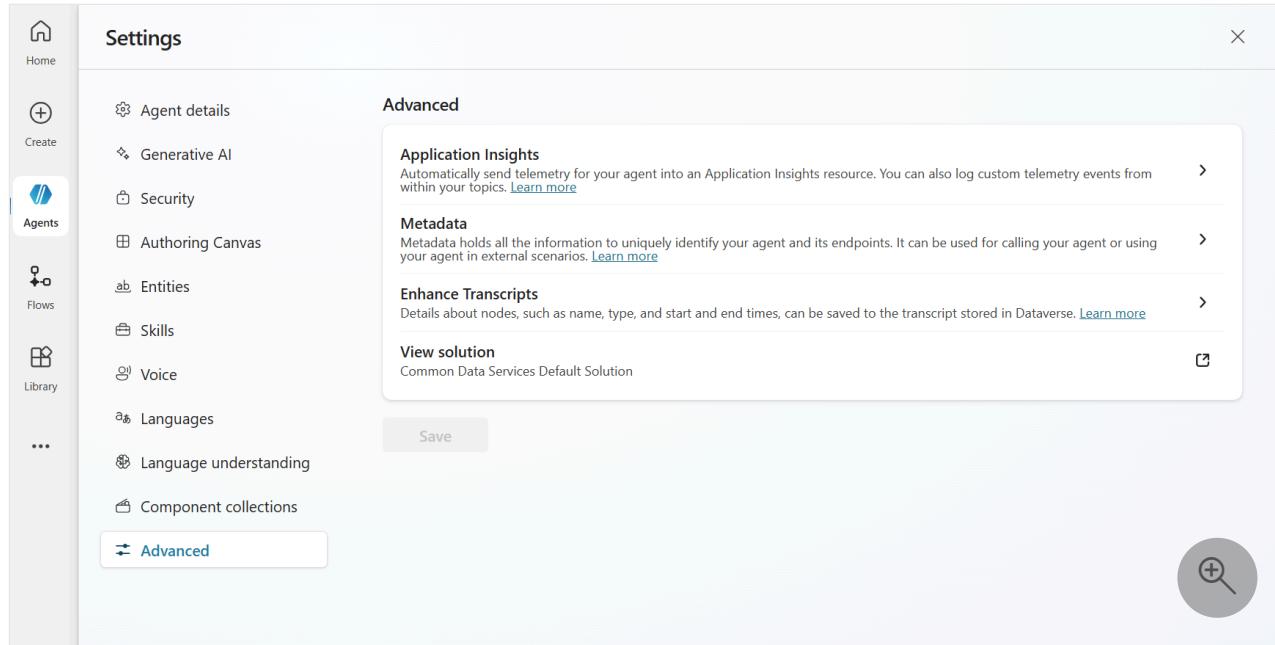
Copy

Using the Microsoft Custom Web Chat UI and other patterns.

## 5. Get information for traditional configuration settings.

If you don't want to use the *connection string* method, and instead use the traditional configuration settings, you need some other metadata from your Copilot Studio agent.

Select **Settings > Advanced**. Under **Metadata**, you need the **Environment ID**, **Tenant ID**, and **Schema name**. Record these metadata values for later.



## 6. Configure your application registration for the correct permissions to access Copilot Studio.

Your web or native app needs to have an app registration in Azure. If you don't have an app registration in Azure, you can follow the full guide on setting one up using the Readme or in the Azure documentation. For more information, see [Register an application in Microsoft Entra ID](#).

### ⓘ Note

Most likely you already have an existing app registration for your registration, and you need to use that instead. You can configure *User* or *Service Principal* authentication methods to access your agent.

## Configure your app registration for user interactive sign-in

1. Navigate to **API permission > Add permissions**, select **APIs my organization uses**, and search for **Power Platform API**.
2. Select **delegated permissions > Copilot Studio > Copilot Studio.Copilots.Invoke** permission. Select **Add Permissions**.

For user sign-in, you can test the sample work with your application registration and your Copilot Studio agent, by:

Adding the connection string into the Copilot Studio client settings in the *appsettings.config* file:

- `DirectConnectUrl`
- `TenantID:`
- `ClientID:`
- `ClientSecret:`

Adding the Copilot Studio client settings in the *appsettings.config* file:

- `EnvironmentID:`
- `SchemaName:`
- `TenantID:`
- `AppClientID:`
- `AppClientSecret:`

You should now be able to run the sample and connect to the agent via your app registration settings using the sample console application.

## Configure your app registration for Service Principal

Alternatively, you might choose to configure your app registration for service authentication rather than for user authentication.

1. On your app registration, go to API permission, add permissions, select **APIs my organization uses**, and search for `Power Platform API`.
2. Select **Application permissions > Copilot Studio**, and check the `Copilot Studio.Copilots.Invoke` permission.
3. Select **Add Permissions**.

For user sign-in, you can test the sample work with your application registration and your Copilot Studio agent, by:

1. Adding the Connection String into the Copilot Studio client settings in the *appsettings.config* file:

- `DirectConnectUrl`
- `TenantID:`
- `UseS2SConnection: true`
- `ClientID:`

- `ClientSecret`:

2. Adding the Copilot Studio client settings in the `appsettings.config` file:

- `EnvironmentID`:
- `SchemaName`:
- `TenantID`:
- `UseS2SConnection: true`
- `AppClientID`:
- `AppClientSecret`:

You should now be able to run the sample and connect to the agent with your app registration settings using the sample console application.

## Integrate the Copilot Studio client into your existing UI/UX

Now you have tested your agent with the Copilot Studio client. Your testing confirms your agent is connected with the sample console app, you're ready to integrate the library into your existing website or app, and connect/surface it with your existing UI.

How you integrate your agent into the application is up to you and your existing code base. Typically, steps to integrate your agent might include:

1. Referencing the library in your application.
2. Implementing objects and methods from the client library in your project.
3. Injecting with DI or managing the client based on the existing design of your application, ensuring you reference the app settings for the configuration details for the agent.

## Use the default Web Chat Embed code (without development/code)

You can add the Copilot Studio agent to your website using an iFrame, embeddable in the HTML of the website/web app.

This code is visible only if the **No authentication** option is selected in the agent's settings in Copilot Studio under **Security**. If **Authenticate with Microsoft** or **Authenticate manually** is selected, the embed code isn't visible.

The screenshot shows the Microsoft Copilot Studio interface. On the left is a sidebar with icons for Home, Create, Agents (selected), Flows, Library, and three dots. The main area has a title 'Settings' and a 'Security' tab selected under 'Authentication'. The 'Authentication' section contains the following text: 'When enabled, your agent will use the selected LLM's stored information and understanding to generate responses.' Below this is a 'Choose an option' section with three radio buttons: 'No authentication' (selected, highlighted in blue), 'Authenticate with Microsoft', and 'Authenticate manually'. A 'Save' button is at the bottom of the panel, and a magnifying glass icon is in the bottom right corner.

Home

Create

Agents

Flows

Library

...

Settings

Agent details

Generative AI

Security

Authoring Canvas

Entities

Skills

Voice

Languages

Language understanding

Component collections

Advanced

Authentication

When enabled, your agent will use the selected LLM's stored information and understanding to generate responses.

Choose an option

No authentication Publicly available in any channel

Authenticate with Microsoft Entra ID authentication in Microsoft Teams, Power Apps, or Microsoft 365 Copilot

Authenticate manually Set up authentication for any channel

Save

## Use DirectLine to connect to Copilot Studio

See [Publish an agent to mobile or custom apps](#) to use DirectLine to integrate with your Copilot Studio agent.

# Integrate with native apps using the Agents Client SDK

07/18/2025

## Important

This content is intended for experienced IT professionals, such as IT admins or developers, who are familiar with developer tools, utilities, and integrated development environments (IDEs). Creating this integration requires software development expertise.

Agents Client SDK is an SDK for Android, iOS, and Windows platforms. This SDK lets you communicate with agents developed using Copilot Studio or the [Microsoft 365 Agents SDK](#), from native Android, iOS, and Windows apps.

With Agents Client SDK, we're introducing multimodality to user-agent interaction. Users can now communicate with agents through text or voice, while sharing context through images or screen capture.

The first version of the SDK supports text-based communication and adaptive cards rendering for unauthenticated or anonymous end users with agents created in Copilot Studio. You can add adaptive cards to topics using the **Ask with adaptive card** option.

## Important

Currently the SDK supports only unauthenticated users and can be used in native apps if you deem such usage appropriate and necessary for your end users.

## Configure your agent for native apps

1. Go to your agent in Copilot Studio by selecting **Agents** and then your agent.
2. Under **Settings > Security > Authentication**, select **No Authentication**.

## < Authentication

When enabled, your agent will use the selected LLM's stored information and understanding to generate responses.

### Choose an option

#### No authentication

Publicly available in any channel

#### Authenticate with Microsoft

Entra ID authentication in Microsoft Teams, SharePoint, Power Apps, or Microsoft 365 Copilot

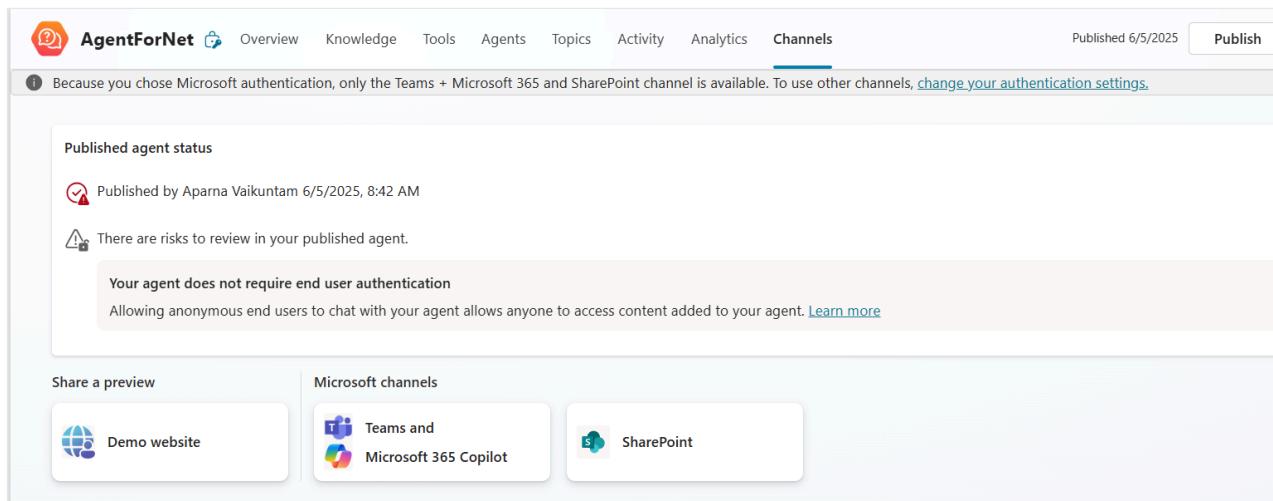
#### Authenticate manually

Set up authentication for any channel

**Save**

### 3. Go to the **Channels** page for your agent.

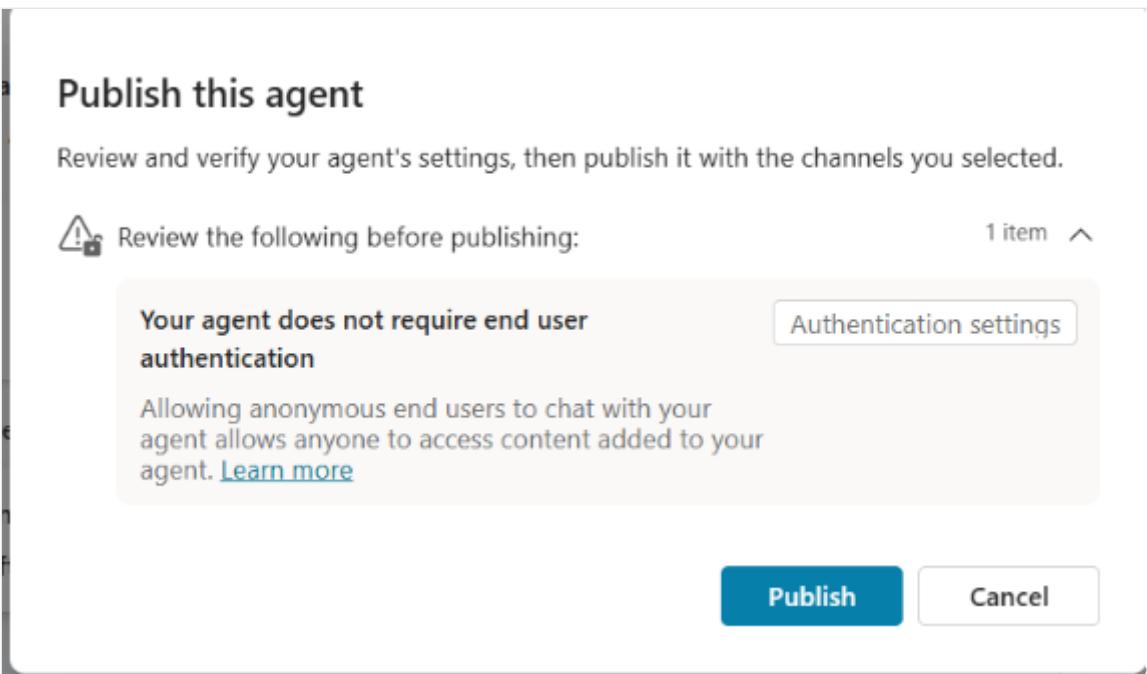
With **No authentication** selected, a review is recommended.



The screenshot shows the 'AgentForNet' application interface. At the top, there are navigation tabs: Overview, Knowledge, Tools, Agents, Topics, Activity, Analytics, and **Channels**. To the right of the tabs, it says 'Published 6/5/2025' and has a 'Publish' button. Below the tabs, a message states: 'Because you chose Microsoft authentication, only the Teams + Microsoft 365 and SharePoint channel is available. To use other channels, [change your authentication settings](#)'. A section titled 'Published agent status' shows a checkmark icon and the text 'Published by Aparna Vaikuntam 6/5/2025, 8:42 AM'. A warning icon indicates 'There are risks to review in your published agent.' A callout box states 'Your agent does not require end user authentication' and explains that allowing anonymous users to chat allows anyone to access content added to the agent. Below this, there are sections for 'Share a preview' and 'Microsoft channels'. Under 'Share a preview', there is a button for 'Demo website'. Under 'Microsoft channels', there are buttons for 'Teams and Microsoft 365 Copilot' and 'SharePoint'.

### 4. Select Publish.

The **Publish this agent** screen appears.



5. Select **Publish**. Your agent is published and available for use with the Client SDK.

## Get the information you need to integrate with the SDK

Before you add integration for your agent into your app code, you need some details from Copilot Studio about your agent.

1. Go to your agent in Copilot Studio by selecting **Agents** and then selecting your agent.
2. Go to **Settings > Advanced settings > Metadata**.

Take note of the **Environment ID** and **Schema name**.

The screenshot shows the 'Advanced' settings page. Under 'Application Insights', it says 'Automatically send telemetry for your agent into an Application Insights resource. You can also log custom telemetry events from within your topics. [Learn more](#)' with a 'more' link. Under 'Metadata', it says 'Metadata holds all the information to uniquely identify your agent and its endpoints. It can be used for calling your agent or using your agent in external scenarios. [Learn more](#)' with a 'more' link. The 'Environment ID' field contains 'Default-c2983f0e-34ee-4b43-8abc-c2f460fd26be'. The 'Tenant ID' field contains 'c2983f0e-34ee-4b43-8abc-c2f460fd26be'. The 'Agent app ID' field contains '6a8689ca-7c18-4d4e-a392-054b25299118'. The 'Schema name' field contains 'cr5f7\_agentForNet'.

3. Take note of the deployment environment for your agent. For `copilotstudio.microsoft.com` and `copilotstudio.preview.microsoft.com`, the deployment

environment is `prod`. For `copilotstudio.preprod.microsoft.com`, the deployment environment is `preprod`.

## Learn more about the Agents Client SDK

Next, using the information you gathered, you can integrate your agent into your native app code using the Agents Client SDK.

For code and more information about how to integrate your agent into your native app, check out the GitHub repo for your app platform of choice:

- [Agents Client SDK for Android ↗](#)
- [Agents Client SDK for iOS ↗](#)
- [Agents Client SDK for Windows ↗](#)

# Agents for customer engagement and handoff

09/06/2025

You can create agents to interact with your customers and integrate with customer service and customer engagement hubs. Such agents provide your customers with self-help based on generative AI. The agent can answer questions and provide information from what's on your company website, within files you upload, or from your knowledge base sources.

When necessary, your agent can transfer the customer to a live agent with integrated handoff to the customer engagement hub that you already use. Customer engagement hubs might be Dynamics 365 Customer Service, ServiceNow, Salesforce, LivePerson, or Genesys.

First, you need to create an agent with generative AI, and tell it where it should look for information when interacting with your customer. The agent uses that information to determine the best answer or response to the user's question and return the information in a conversational tone.

Once you create and test your agent, you can configure integration between the agent and your engagement hub, including handoff to a live agent.

## Create an agent and connect it to your knowledge source

If you're new to creating agents in Copilot Studio, follow the steps in the [Quickstart](#).

If you're familiar with creating agents in Copilot Studio and have an agent now, you can go straight to the instructions for [adding a generative answers node](#) to connect your agent to your data.

### Important

Classic chatbots don't support generative AI features. Classic chatbots are marked as such in Copilot Studio. These are chatbots made with older versions of Copilot Studio and its predecessor, Power Virtual Agents.

## Connect to knowledge sources

Name	Source	Description	Number of inputs supported in generative answers	Authentication
Public website	External	Searches the query input on Bing, only returns results from provided websites	Generative mode: 25 websites Classic mode: Four public URLs (for example, <i>microsoft.com</i> )	None
Documents	Internal	Searches documents uploaded to Dataverse, returns results from the document contents	Generative mode: All documents Classic mode: Limited by the Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	Generative mode: 25 URLs Classic mode: Four URLs per generative answers topic node	Agent user's Microsoft Entra ID authentication
Dataverse	Internal	Connects to the configured Dataverse environment and uses a retrieval-augmented generative technique in Dataverse to return results	Generative mode: Unlimited Classic mode: Two Dataverse knowledge sources (and up to 15 tables per knowledge source)	Agent user's Microsoft Entra ID authentication
Enterprise data using connectors	Internal	Connects to connectors where your organization data is indexed by Microsoft Search	Generative mode: Unlimited Classic mode: Two per custom agent	Agent user's Microsoft Entra ID authentication

### (!) Note

- Agent user authentication for knowledge sources means that when a specific user asks a question of the agent, the agent only surfaces content that the specific user can access.
- Knowledge sources in generative answers nodes currently don't support Bing Custom Search, Azure OpenAI, or Custom Data. Instead, from the generative answers node properties, use the **Classic data** option for [Bing Custom Search](#), [Azure OpenAI](#), or [Custom Data](#) sources.

- For websites, you need to confirm which website(s) your organization owns that Bing will search through Copilot Studio.
- You can perform language-agnostic querying across all supported file types and languages.
- If you're using unstructured data, such as individual SharePoint files and folders, OneDrive files and folders, or connectors, there are different limits and limitations. For more information, go to [Limits and limitations](#).
- Currently, citations returned from a knowledge source can't be used as inputs to other tools or actions.

## Configure handoff to customer engagement hubs

- [Dynamics 365 Customer Service](#)
- [ServiceNow](#)
- [Salesforce](#)
- [LivePerson](#)
- [Generic customer engagement hub](#)

## Configuring agent behavior

The following fields are available under **Agent details** on the side navigation pane and let you customize your agent for tone and behavior.

 Expand table

Field	Description
Greeting message	What the agent says when greeted.
Start over message	What the agent says when the user asks to start the conversation over.
Conversation start message	What the agent says when opened.
Escalate link	A link for the user to talk to an agent.
No match message	What the agent says when it doesn't have topics for the user's input.
Multiple topics matched	What the agent says to ask the user to clarify their meaning.
Reset conversation message	What the agent says after starting the conversation over.



# Configure handoff to any generic engagement hub

09/06/2025

Copilot Studio lets you hand over agent conversations seamlessly and contextually to a human agent through an engagement hub.

With some custom development, you can configure your agent to hand off conversations to any engagement hub. This guide describes how you can do this.

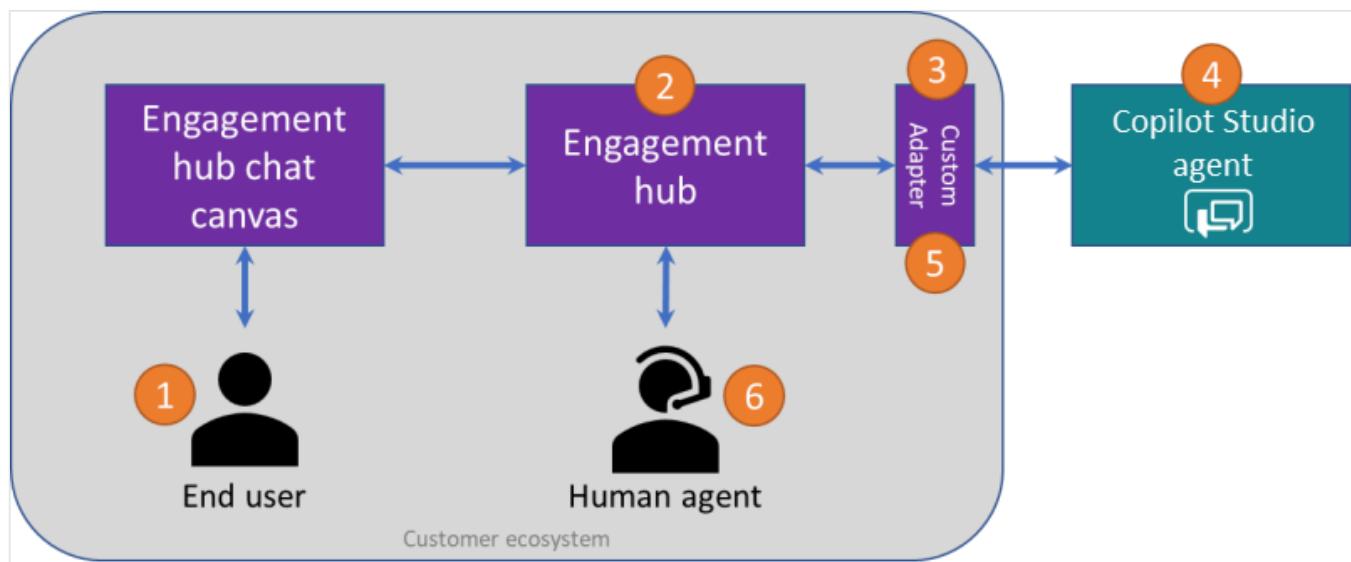
## Prerequisites

- You need to have [built an agent using Copilot Studio](#).
- You need an engagement hub that can interact programmatically using APIs or SDK.

### i Important

Some instructions in this section require software development from you or your developers. It's intended for experienced IT professionals, such as IT admins or developers who have a solid understanding of developer tools, utilities, and IDEs who are looking to integrate third-party engagement hubs with Copilot Studio.

## Overview



A full handoff to an engagement hub follows this pattern:

1. A customer interacts with the engagement hub's chat canvas.

2. The engagement hub routes the incoming chat through built-in chat routing capabilities to an agent.
3. A custom adapter relays the incoming chat messages from the engagement hub to a Copilot Studio agent.
4. Once the customer [triggers handoff](#), Copilot Studio starts handoff with full conversational context.
5. The custom adapter intercepts the handoff message, parses the full conversation context, and seamlessly routes the escalated conversation to a skilled human agent, based on availability.
6. The customer's chat is seamlessly and contextually handed off to a human agent who can resume the conversation.

To hand off the conversation to a human agent, you need to build a custom handoff adapter.

## Build a custom handoff adapter

An adapter bridges conversations to and from your agent engagement hub by relaying and transforming messages between customers, agents, and human agents.

Most popular agent engagement hubs provide software development kits (SDKs) or document their APIs publicly, enabling you to build such adapters.

It's outside the scope of this document to cover what a custom adapter could contain.

However, the following sample handoff message, based on what Copilot Studio generates as part of our [standard handoff to a live agent experience](#), can help get you started.

These code snippets and samples allow you to extract context from the conversation to seamlessly and contextually hand off agent conversations to any generic engagement hub.

## Sample handoff message payload

Handoff is currently only supported over Direct Line. Learn more about [interacting with an agent over Direct Line](#). Upon handoff, an event activity called `handoff.initiate` is raised and sent to the adapter.

You can see a [full sample handoff message activity on our GitHub site](#) ↗.

## Extract context from handoff message

To use [conversational context](#), you must parse the `handoff.initiate` event activity. The following snippet of code parses the `handoff.initiate` event activity and extracts the conversational context. See the [full code sample on GitHub](#) ↗.

C#

```
public void InitiateHandoff(string botresponseJson)
{
 BotResponse response = JsonConvert.DeserializeObject<BotResponse>(botresponseJson);

 // Look for Handoff Initiate Activity. This indicates that conversation needs
 // to be handed off to agent
 Activity handoffInitiateActivity =
 response.Activities.ToList().FirstOrDefault(
 item => string.Equals(item.Type, ActivityTypes.Event,
 System.StringComparison.OrdinalIgnoreCase)
 && string.Equals(item.Name, HandoffInitiateActivityName,
 System.StringComparison.OrdinalIgnoreCase));

 if (handoffInitiateActivity != null)
 {
 // Read transcript from attachment
 if (handoffInitiateActivity.Attachments?.Any() == true)
 {
 Attachment transcriptAttachment =
 handoffInitiateActivity.Attachments.FirstOrDefault(a =>
 string.Equals(a.Name.ToLowerInvariant(), TranscriptAttachmentName,
 System.StringComparison.OrdinalIgnoreCase));
 if (transcriptAttachment != null)
 {
 Transcript transcript = JsonConvert.DeserializeObject<Transcript>(transcriptAttachment.Content.ToString());
 }
 }

 // Read handoff context
 HandoffContext context = JsonConvert.DeserializeObject<HandoffContext>(handoffInitiateActivity.Value.ToString());

 // Connect to Agent Hub
 // <YOUR CUSTOM ADAPTER CODE GOES HERE>
 }
}
```

# Configure handoff to Dynamics 365 Customer Service

10/15/2025

When your customers need to speak with a human agent, your agent can seamlessly hand off the conversation. With the capabilities of the [Chat Add-in for Dynamics 365 Customer Service](#), you can connect both text (messaging) conversations, and voice conversations (in classic chatbots).

## ⓘ Note

Connect to Dynamics 365 Customer Service with a voice-enabled agent. For more information, see [Use interactive voice response in your agents](#).

When your agent hands off a conversation, it shares the full history of the conversation and all variables collected in the interaction. Your Dynamics 365 Customer Service routes incoming escalations to the right queue, and a live agent can seamlessly resume the conversation. For more information about how to use handoff in agent conversations, see [Hand off to a live agent](#).

You can also enable single sign-on (SSO) to allow agents to sign users in if they're logged in to the page where the agent is deployed. For more information, see [Configure single sign-on with Microsoft Entra ID and Pass authentication token to agent during single sign-on in live chat](#).

Web app

## Prerequisites (Web app)

- Sign in with an account that has at least **OC\_Admin** and **Agent Author** roles.
- Have a product license for [Copilot Studio](#) and a [product license for the Chat Add-in for Dynamics 365 Customer service](#).
- Your agent and Dynamics 365 Customer Service must be in the same environment.
- For end-to-end capabilities to work as expected, you must [publish](#) your agent.

## ⚠ Warning

Copilot Studio agents with names that are longer than 30 characters fail to connect when you follow the instructions in this article. Ensure your agent's name contains

fewer than 30 characters before you proceed.

# Connect your agent to Dynamics 365 Customer Service

1. In Copilot Studio, open your agent, and go to the **Channels** page.
2. Under **Customer engagement hub**, select the **Dynamics 365 Customer Service** tile.
3. Select **Connect**.

## Note

- The Dynamics 365 Customer Service package installation must be in the same environment as the agent.
- If you're using application lifecycle management (ALM), you might see a message that we can't determine if Dynamics 365 Customer Service integration is enabled for the environment. For more information, see [Agents with application lifecycle management](#).

4. Select **View in Omnichannel** to continue configuring the agent connection in Dynamics 365 Customer Service.

## Important

To test the agent on your custom website, you must use the **embed** code specified in the chat widget you set up in Dynamics 365 Customer Service. If you use the **embed** code from Copilot Studio, handoff doesn't work. For more information, see [Embed chat widget in your website or portal](#).

## Agents with application lifecycle management (Web app)

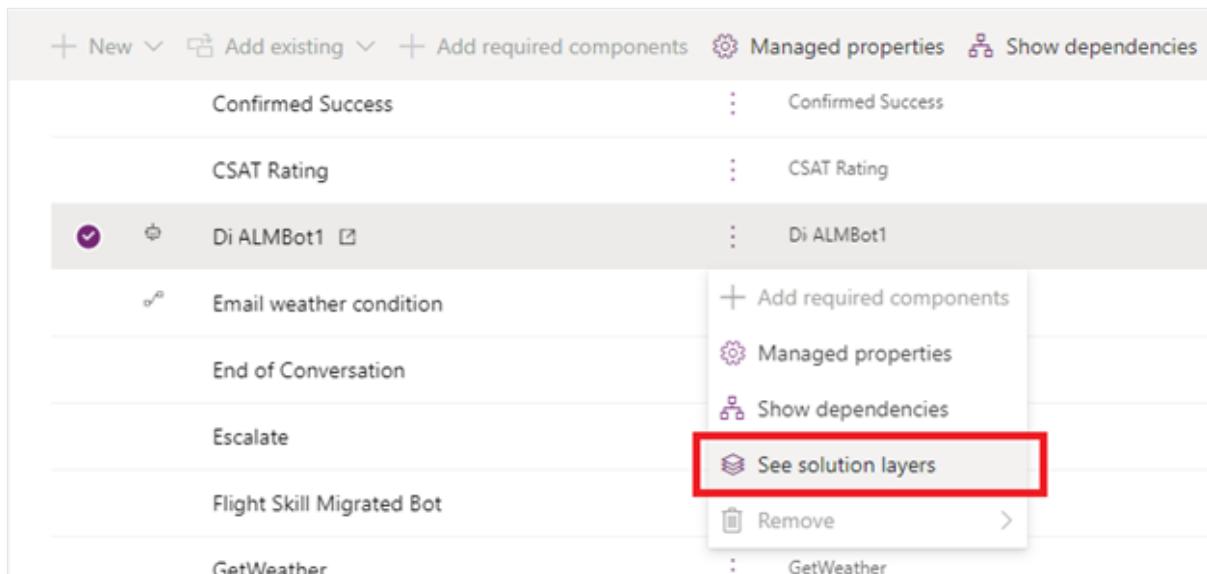
Suppose you [set up application lifecycle management \(ALM\)](#) for your agents, and are exporting and importing agents between [development \(unmanaged\)](#) and [test or production \(managed\) environments](#). In that case, you might see a message that we can't determine if Dynamics 365 Customer Service integration is enabled for the environment.

If the agent you exported has Dynamics 365 capabilities enabled, you can ignore this message. The agent can still work properly. The message will disappear after you export

the latest version of your agent from your development environment and then import it into a targeted test or production environment with managed agents.

If you continue to see the message after you export and import the latest version of your managed agent, make sure to remove any unmanaged layers:

1. Sign in to Power Apps and select the managed agent's environment.
2. In the navigation menu, select **Solutions**, and then select the solution that contains the agent with the unmanaged layer.
3. Next to the agent component in the solution, select **More commands (⋮)**, and then select **See solution layers**.



4. Select the unmanaged layer, and then select **Remove unmanaged layer**.

The screenshot shows the 'Solution Layers' screen. At the top, there are buttons for Compare with previous layer and Remove unmanaged layer (which is highlighted with a red box). Below this, the breadcrumb navigation shows: All > Di ALMBot1 > Solution Layers.

Order	Solution	Publisher
2	Unmanaged layer	Default Publisher for org
1	DiALMSolution	Di publisher

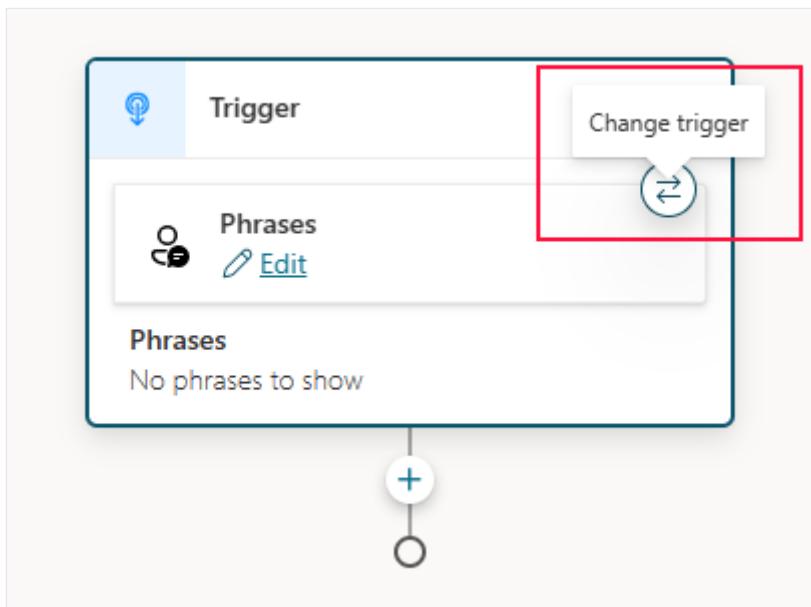
If your agent doesn't have Dynamics 365 capabilities enabled, the message is always displayed.

# Configure inactivity timeout to close conversation

Dynamics 365 Customer Service expects a conversation to end after a period of inactivity. This feature is important to ensure the supervisor view displays the correct ongoing conversations and manages agent loads and metrics correctly. When you create an agent from Dynamics 365 Customer Service, the default agent content includes two topics: **Session Timeout Reminder** and **Session Timeout**. These topics use the [Inactivity trigger](#) to remind the user, and then close the conversation when the configured inactivity timeout expires. But if you create the agent from Copilot Studio, those topics aren't included in the agent. To create them, proceed as follows.

## Create a Session Timeout topic

1. Go to the [Topics](#) page.
2. Select **Add a topic** and **From blank**.
3. Hover the trigger **Phrases** and select the **Change trigger** icon.



4. Select **Inactivity** from the list.
5. Select **Edit** in the **Trigger** node to configure the inactivity duration.
6. Select the desired duration value from the list. Alternatively, use the **Formula** option and enter a value in seconds. This value is the amount of inactive time it takes for the topic to trigger.

7. Select the channel it applies to by using the **Condition** option. For Dynamics 365, under the **Condition** block, select **Select a variable**, select the **System** tab, and **Activity.Channel**.
8. Select **Omnichannel** from the dropdown.
9. Finally, add a message, and at the end, add an **End conversation** node so the conversation ends. Select the **Add node** , point to **Topic management** icon, and then select **End conversation**.
10. Save and publish your agent.

## Manage your agent's Dynamics 365 capabilities (Web app)

1. Go to the **Channels** page.
2. Under **Customer engagement hub**, select the **Dynamics 365 Customer Service** tile.

Here you can disconnect your agent and find the link to go to the Dynamics 365 Customer Service admin center to view the connection details.

## Disconnect your agent from Dynamics 365 Customer Service or disable the connection

If you select **Disconnect**, the application user that represents the agent in your Dynamics 365 Customer Service instance is disabled. Your agent effectively disconnects from the Dynamics 365 Customer Service environment and stops receiving any traffic from your Dynamics 365 Customer Service instance.

To add your agent back, you must [connect it again](#).

## Known limitations (Web app)

See [limitations when using Copilot Studio with the Chat Add-in for Dynamics 365 Customer Service](#).

## Recommended extensions (Web app)

The following extensions aren't required to hand off to Dynamics 365 Customer Service, but they provide agent authors with a better experience by providing [extra variables and](#)

actions.

Install [Dynamics 365 Copilot Studio extension](#).

# Hand off to Genesys

09/06/2025

This article provides supplemental information to the main Genesys documentation for this integration located on [GitHub](#).

## Initial integration

Ensure that the [authentication setting in Copilot Studio is set to No authentication](#).

While the agent connector documentation is for AWS (Amazon Web Services), equivalent steps are possible in Azure as well. Modify the two Python scripts accordingly.

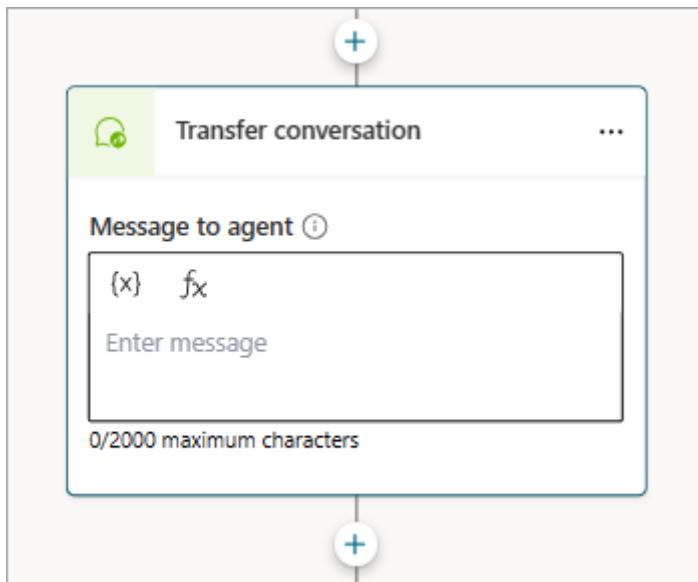
Ensure that the DynamoDB table name created matches the one in the bot\_sessions.py file.

## Agent handoff

Refer to [Configure the transfer to agent node by using the escalate topic](#), in the Dynamics 365 Customer Service documentation.

If you want your agent to pass a message to the agent in the transfer, perform the following steps:

1. Set the message in Copilot Studio:



2. Include as slot for the Escalate intent in the botconnector API call:

JSON

```
{
 "name": "Escalate",
 "slots": {
 "va_AgentMessage": {
 "name": "va_AgentMessage",
 "type": "String"
 }
 }
}
```

The message is visible in Genesys as an output.

The screenshot shows the Genesys interface with the following configuration:

- Add Output** button
- Intents** section:
  - Integration**: There are no slots defined for this intent.
- Escalate** intent:
  - va\_AgentMessage**: A dropdown menu is open, showing the option **A State.sAgentMessage**.
- Failure Outputs** section:
  - errorType**: A dropdown menu is open, showing the option **No output value specified**.
  - errorMessage**: A dropdown menu is open, showing the option **No output value specified**.

3. If there are more variable to pass from Copilot Studio back to Genesys, follow the same process with `va_{variableName}` as the slot name.

# iframe setup

1. See [Add your agent to your website](#) for detailed instructions on how to get a URL to your Copilot Studio agent.

Copy or take note of the iframe src URL. This value is required for configuring Genesys Cloud.

2. Follow the [Genesys documentation on how to set up the widget](#).

Use your iframe src URL as the **Application URL**.

# Hand off to LivePerson

09/12/2025

Integrating LivePerson with Copilot Studio presents a compelling opportunity for businesses seeking to enhance their customer support and engagement capabilities.

## Prerequisites

Before you begin, make sure you have:

- An active LivePerson account.
- A Copilot Studio agent.
- *Admin* or *Developer* access to both LivePerson and Power Platform.

## Steps

Follow these steps to create and connect LivePerson to a Copilot Studio agent:

1. [Retrieve the Direct Line secret](#) for your agent.
2. Follow the instructions in the [LivePerson documentation](#).
3. To set up handoff to LivePerson, see [Transfer to agent](#).

# Hand off to Salesforce Einstein bot

09/06/2025

This document provides step-by-step instructions on how to create and connect an Einstein Bot to a Copilot Studio agent. This integration allows you to use the capabilities of both platforms to enhance your agent experience.

## Prerequisites

Before you begin, ensure that you have the following prerequisites in place:

- An active Salesforce account with Einstein Bots enabled.
- A Copilot Studio account.
- Admin or developer access to both Salesforce and Microsoft Power Platform.

## Step 1: Retrieve Direct Line secret for your agent

1. Retrieve your [Direct LinesSecret](#).

## Step 2: Configure Salesforce remote site settings

1. Sign in to your Salesforce account.
2. In Salesforce, navigate to **Setup**.
3. In the **Quick Find** box, type *Remote Site Settings* and select **Remote Site Settings**.
4. Add a new remote site, then fill in **Remote Site Name** as *Copilot Studio* and **Remote Site URL** as <https://directline.botframework.com>
5. Select **Save**.

 **Note**

For other regions use the appropriate URL for the DirectLine API URL

## Step 3: Create Apex Classes to connect to DirectLine API

1. Sign in to your Salesforce account.
2. In Salesforce, navigate to **Setup**.
3. In the **Quick Find** box, type *Apex Classes*.
4. Create three new classes with the following details:

## DL\_GetConversation

- Inputs: None
- Outputs: `conversationId` (String), `responseCode` (Integer), `errorMessage` (String)
- `InvocableMethod getConversationID`: Use your DirectLine Secret to make an HTTP call to the DirectLine endpoint starting a conversation.

## DL\_PostActivity

- Inputs: `conversationId` (String), `userMessage` (String)
- Outputs: `responseCode` (Integer), `errorMessage` (String), `watermark` (String)
- `InvocableMethod postActivity`: Execute an HTTP call to the post activity DirectLine API using the `conversationId` in the URL and `userMessage`.

## DL\_GetActivity

- Inputs: `conversationId` (String), `watermark` (String)
- Outputs: `message` (String), `watermark` (String), `responseCode` (Integer), `errorMessage` (String)
- `InvocableMethod getActivity`: Execute an HTTP call to the `getActivity` DirectLine API, using the `conversationId` and `watermark` to retrieve and parse the activity object for the message to send back to the user.

## Step 4: Setup Einstein bot

1. Sign in to your Salesforce account.
2. In Salesforce, navigate to **Setup**.
3. In the **Quick Find** box, type *Einstein* and select **Einstein Bots**.
4. Create a new Einstein bot following the steps within the [Salesforce Einstein Documentation](#).

## Step 5: Connect Einstein Bot to Microsoft Copilot Studio with Apex classes

In the Einstein Bot, to have the bot reach out on each utterance outside of Einstein topics you can add references to the previously created Apex classes:

1. In the Welcome topic, add an Apex class action referencing the `DL_GetConversation` class and assign the response variables appropriately.

2. In the Confused topic, add an Apex class action referencing the `DL_PostActivity` class, passing the `conversationId` and last user input. Be sure to set the response variables to other variables.
3. Create a condition to check the Response Code from the `DL_PostActivity` Apex action, if the code is 200 or 204 you can continue, otherwise display the Error Message.
4. Add another Apex class action to `DL_GetActivity`, setting the `conversationId` from `GetConversation` and the `watermark` from the `PostActivity`, and assigning the response variables.

 **Note**

You may want to create `GetActivity` as a separate topic. You might need to loop on `DL_GetActivity` to get the actual response, if it takes longer to fetch.

1. Create a condition to check the response code and that the return message is set, if so, then go to a message node to output the response.
2. The last step in the flow should remain as "Wait for Customer Input."

## Step 6: Handle Agent Hand off

When accessing an agent from Einstein, the agent is only used in a request/response model: based on the user's query, an agent fetches the most appropriate KB, and summarizes the answer.

Because you can continue to configure your Einstein bot, you can implement any other topic triggers or escalation triggers in Einstein. This extra configuration allows your users to escalate to the appropriate Salesforce Live Agent queues.

For more information, see the Salesforce documentation [Route Conversations from an Enhanced Bot](#).

# Hand off to ServiceNow

09/06/2025

This document provides step-by-step instructions on how to create and connect a ServiceNow Virtual Assistant to a Copilot Studio agent and escalate to ServiceNow Live Agent. This integration allows you to use the capabilities of both platforms and generative answers to enhance your agent experience.

## (!) Note

For more detailed information, see [Configure handoff to any generic engagement hub](#). Copilot Studio only provides hooks for you to integrate human agent solutions like ServiceNow.

The following steps are only a suggestion and you need a software development team to perform them. This procedure is intended for experienced IT professionals, developers, or System Integrators who have a solid understanding of developer tools, utilities, and integrated development environments (IDEs).

## Overview

The integration between Copilot Studio and ServiceNow requires the following high-level configuration steps:

1. Create an Azure Function as a relay to DirectLine.
2. Configure Bot Interconnect in ServiceNow.
3. Map topics between ServiceNow Virtual Agent and Copilot Studio.
4. Extend DirectLinePrimaryBotIntegrationInboundTransformer.
5. Configure handoff in Copilot Studio.

## Prerequisites

Before you begin, ensure that you have the following prerequisites in place:

- An active ServiceNow account with ServiceNow Virtual Agent enabled and ServiceNow Bot Interconnect enabled. (ServiceNow Yokohama or later required).

- A Copilot Studio agent, configured with `no authentication` (this pattern uses server-to-server authentication with a Direct Line secret).
- Admin access to ServiceNow.

## Create an Azure Function as a relay to DirectLine

The Azure Function acts as a relay, enabling ServiceNow Bot Interconnect to retrieve activities (such as messages and events) from an ongoing conversation with a Copilot Studio agent.

### Important

The default wait time for the Azure Function is 2,000 ms (2 seconds), which might be too short when using generative orchestration in Copilot Studio. If messages aren't coming through in the virtual agent in ServiceNow, consider increasing the wait time.

## Sample request sent by ServiceNow Bot Interconnect

HTTP

```
GET /api/relayToDirectLine HTTP/1.1
Host: your-function-app.azurewebsites.net
Authorization: Bearer YOUR_DIRECT_LINE_SECRET_HERE
ConversationId: abc123def456-conversation-id
Watermark: 5
WaitTime: 3000
```

## Sample response from the Azure Function

JSON

```
{
 "activities": [
 {
 "type": "message",
 "id": "0000001",
 "timestamp": "2025-07-13T13:46:43.6095506Z",
 "text": "Hello, I'm a Copilot Studio Agent.",
 "from": {
 "id": "bot-id",
 "name": "Copilot Studio Agent",
 "role": "bot"
 },
 "conversation": {
 "id": "wwjmISGzmd3FzMEAgoTiJ-us"
 }
 }
]
}
```

```
 }
 },
{
 "type": "message",
 "id": "0000002",
 "timestamp": "2025-07-13T13:46:45.000000Z",
 "text": "speak to a live agent",
 "from": {
 "id": "user-id",
 "name": "User",
 "role": "user"
 },
 "conversation": {
 "id": "wwjmISGzmd3FzMEAgotiJ-us"
 }
},
{
 "type": "event",
 "id": "0000006",
 "timestamp": "2025-07-13T13:46:51.2854808Z",
 "name": "handoff.initiate",
 "value": {
 "va_LastPhrase": "speak to a live agent",
 "va_Topics": ["Escalate"]
 }
},
],
"watermark": "10"
}
```

## Creating and deploying the Function

You can choose to create and deploy either the v3 Azure Function as described in the ServiceNow documentation or the v4 Azure Function available in the Copilot Studio GitHub repository.

### Option 1: Deploy the v3 Azure Function

To create and deploy the v3 Azure Function, follow the steps outlined in the ServiceNow documentation:

1. [Create a JavaScript function in Microsoft Azure using Visual Studio Code](#).
2. Deploy the function to Azure and retrieve the function URL for use in ServiceNow Bot Interconnect.

### Option 2: Deploy the v4 Azure Function

Alternatively, you can use the v4 Azure Function available in the Copilot Studio GitHub repository:

1. Clone the repository from [Copilot Studio Samples](#). The function is located under `CopilotStudioSamples/IntegrateWithEngagementHub/ServiceNow/DirectLineAzureFunction`.
2. Follow the instructions in [Create a JavaScript function in Azure using Visual Studio Code](#).
3. On the [Overview](#) page of the function app, copy the function URL.

## Configure Bot Interconnect in ServiceNow

1. [Obtain your agent's Direct Line secret](#).
2. [Add the Direct Line secret key to your Virtual Agent Bot Interconnect instance](#).

## Map topics between ServiceNow Virtual Agent and Copilot Studio

1. To call Copilot Studio topics, [create a Virtual Agent Bot Interconnect shell topic](#).

## Extend DirectLinePrimaryBotIntegrationInboundTransformer

The built-in `DirectLinePrimaryBotIntegrationInboundTransformer` doesn't parse the Copilot Studio `handoff.initiate` event when the conversation needs to be handed over to a live agent.

The `shouldConnectToAgent` function in `DirectLinePrimaryBotIntegrationInboundTransformer` always returns false:

```
JavaScript

shouldConnectToAgent: function(response) {
 return false;
}
```

To properly handle handoff events, create a custom transformer that extends the default transformer and detects the `handoff.initiate` event.

```
JavaScript
```

```

shouldConnectToAgent: function() {
 var response = this._response || {};
 var activities = response.activities || [];

 var handoffDetected = activities.some(function(activity) {
 return activity.type === "event" && activity.name ===
"handoff.initiate";
 });

 if (handoffDetected) {
 return true;
 }
 return false;
}

```

## Create the custom transformer

1. In ServiceNow, switch to the **Bot Interconnect** application scope.
  - a. Select the globe icon in the top right corner.
  - b. Select **Bot Interconnect** from the application list.
2. Create a new Script Include.
  - a. Navigate to **System Definition > Script Includes**.
  - b. Select **New**.
  - c. Configure the following settings:
    - **Name:** CustomDirectLineInboundTransformer
    - **API Name:** sn\_va\_bot\_ic.CustomDirectLineInboundTransformer (autogenerated)
    - **Application:** Bot Interconnect (autoselected)
    - **Accessible from:** This application scope only
    - **Active:**  (checked)
3. Replace the default script content with the custom transformer code from the [Copilot Studio Samples repository](#).
4. To save the *Script Include*, select **Submit**.

## Update the Bot Interconnect configuration

1. Navigate to **Workflow Studio**.

- a. In ServiceNow, navigate to **Workflow Studio**.
  - b. Under **Actions**, locate **Direct Line Primary Bot Integration Inbound Transformer**.
2. Modify the transformer action.
- a. Open the **Direct Line Primary Bot Integration Inbound Transformer** action.
  - b. In the script editor, comment out the default transformer and replace it with a call to your custom transformer.

```
JavaScript
```

```
(function execute(inputs, outputs) {
 var clientVariables = JSON.parse(inputs['client_variables']);
 var response = JSON.parse(inputs['response']);
 //set client variables
 response.clientVariables = clientVariables;

 //var transformedResponse = new
 sn_va_bot_ic.DirectLinePrimaryBotIntegrationInboundTransformer(response).tran
 sformResponse();
 var transformedResponse = new
 sn_va_bot_ic.CustomDirectLineInboundTransformer(response).transformResponse()
 ;

 outputs['transformed_response'] = transformedResponse;
})(inputs, outputs);
```

3. Save the changes to the action and publish.

## Configure handoff

To enable handoff to a ServiceNow Live Agent, add a **Transfer conversation** node to your agent's **Escalate** topic.

1. In Copilot Studio, open your agent and navigate to **Topics > System > Escalate**.
2. Select **+** (Add node) > **Topic management > Transfer conversation**.
3. Save your topic and publish your agent.

The **Transfer conversation** node emits a `handoff.initiate` event that ServiceNow's custom transformer detects and processes.

## Sample Escalate topic YAML

You can also import the following YAML configuration for a complete Escalate topic with transfer capabilities:

YAML

```
kind: AdaptiveDialog
startBehavior: CancelOtherTopics
beginDialog:
 kind: OnEscalate
 id: main
 intent:
 displayName: Escalate
 includeInOnSelectIntent: false
 triggerQueries:
 - Talk to agent
 - Talk to a person
 - Talk to someone
 - Call back
 - Call customer service
 - Call me please
 - Call support
 - Call technical support
 - Can an agent call me
 - Can I call
 - Can I get in touch with someone else
 - Can I get real agent support
 - Can I get transferred to a person to call
 - Can I have a call in number Or can I be called
 - Can I have a representative call me
 - Can I schedule a call
 - Can I speak to a representative
 - Can I talk to a human
 - Can I talk to a human assistant
 - Can someone call me
 - Chat with a human
 - Chat with a representative
 - Chat with agent
 - Chat with someone please
 - Connect me to a live agent
 - Connect me to a person
 - Could some one contact me by phone
 - Customer agent
 - Customer representative
 - Customer service
 - I need a manager to contact me
 - I need customer service
 - I need help from a person
 - I need to speak with a live agent
 - I need to talk to a specialist please
 - I want to talk to customer service
 - I want to proceed with live support
 - I want to speak with a consultant
 - I want to speak with a live tech
 - I would like to speak with an associate
```

- I would like to talk to a technician
- Talk with tech support member

```
actions:
 - kind: TransferConversationV2
 id: transferConversationV2_3JXrI7
 transferType:
 kind: TransferToAgent
 messageToAgent:
 context:
 kind: AutomaticTransferContext
```

## Troubleshooting

If you encounter errors in the ServiceNow Virtual Agent chat widget, follow these troubleshooting steps:

### Check Azure Function logs

1. Verify that your Azure Function is working correctly and not returning errors.
2. View real-time logs using [Azure Functions streaming logs](#).
3. Check Application Insights for detailed telemetry and error information.

### Enable ServiceNow debugging

Use ServiceNow's built-in debugging tools to trace the execution flow:

1. Enable [Script Tracer](#) in ServiceNow.
2. Repeat the action that's causing the error.
3. Review the trace logs to identify where the issue occurs.

### Monitor Copilot Studio invocations

Ensure your Copilot Studio agent is being properly invoked:

1. [Connect your Copilot Studio agent to Application Insights](#).
2. Monitor the telemetry to confirm the agent is receiving requests from ServiceNow.
3. Check for any errors or unexpected behavior in the agent's execution.

## Common issues and solutions

- **Handoff not triggering:** Verify the custom transformer is active and the Transfer conversation node is properly configured
- **Authentication errors:** Ensure the Direct Line secret is correctly configured in ServiceNow. Specifically, the credential record in ServiceNow needs to be linked with a connection record.
- **Function timeout:** Check if the Azure Function is timing out and increase the timeout settings if necessary

# Assign licenses and manage access to Copilot Studio

09/17/2025

This article is written for tenant administrators. If you want to sign up for Copilot Studio as an individual, see [Get access to Copilot Studio](#).

To create and manage agents with Copilot Studio, you need:

- A license for each user, also known as a *per user license* (or *Copilot Studio User License* as referred to on the [Microsoft 365 admin center](#)), should be assigned to individual users who need access to create and manage agents.
- A license for your organization, also known as a *tenant license* (or *Copilot Studio* as referred to on the [Microsoft 365 admin center](#)), should be acquired by the tenant administrator. This license can't be assigned to individual users.

## Note

- Users of your agents don't need a special license. After you publish your agent, anyone who can access the published agent can interact with it.
- Guest users of your tenant will not be able to access Copilot Studio.

For more information, see [Copilot Studio licensing](#).

## Important

If you purchase a Copilot Studio license through volume licensing or any channel other than the [Microsoft 365 admin center](#), you need to acquire both a tenant license and a user licenses through that channel, preferably as part of a single transaction.

Web app

## Buy a tenant license

Purchase licenses for your organization by going to the [Microsoft 365 admin center](#). You must sign in with an admin account to buy licenses.

1. Go to the [Microsoft 365 admin center](#) and sign in with your administrator account.

2. On the side pane, expand the **Billing** menu, and then select **Purchase services**.
3. Search for **Copilot Studio**, and complete the checkout process.

## Acquire user licenses

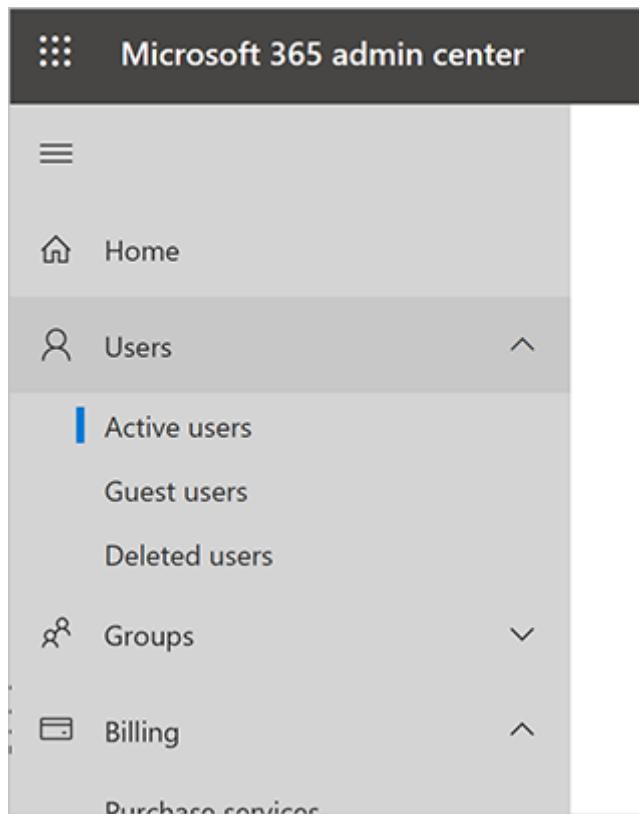
1. After you've purchased a Copilot Studio license from the Microsoft 365 admin center, you need to purchase user licenses to give users access to the product. In the Microsoft 365 admin center, go to **Billing**, and then select **Purchase services**.
2. Scroll down to the **Add-ons** section.
3. Look for **Copilot Studio User License**. Select the number of licenses you need, and complete the checkout process.

 **Important**

If you purchased a Copilot Studio license through volume licensing or any channel other than the [Microsoft 365 admin center](#), you need to acquire both a tenant license and user licenses through that channel, preferably as part of a single transaction.

## Assign licenses to users

1. Go to the [Microsoft 365 admin center](#) and sign in with your administrator account.
2. On the side pane, expand the **Users** menu, and then select **Active users**.



3. Select a name, and then select **Manage product licenses**.
4. On the flyout pane, select the check box next to **Copilot Studio user license**, and then select **Save changes**.

Repeat these steps to add more users, or exit the Microsoft 365 admin center if you're finished.

**(!) Note**

To simplify user license management, you can assign licenses to a Microsoft Entra ID security group. More information: [Assign licenses to users by group membership in Microsoft Entra ID](#)

The users can now sign in to [Copilot Studio](#).

## How to differentiate between tenant and user licenses

If you're unsure whether you have both of the required licenses, on the [Microsoft 365 admin center](#) go to **Billing** and then **Subscriptions** to see the list of active licenses. For Copilot Studio, you should see two licenses:

- Copilot Studio
- Copilot Studio User License

You can only assign the *Copilot Studio User License* to specific users. You don't need to assign the *Copilot Studio* license.

## Trial plans

Users in your organization can try Copilot Studio for a limited time period.

You can [disable or enable the ability for users to sign up for a trial themselves](#) by modifying the `AllowAdHocSubscriptions` flag in your organization settings.

## Subscription capacity

When you purchase a license, you gain capacity for the specified number of billed sessions. Copilot Studio pools this capacity across the entire tenant.

The consumption of the capacity isn't reported at the tenant level, but [can be seen for each individual agent](#).

For more information, see [Quotas and limits](#).

## Using Power Automate with a Copilot Studio license

- [Quotas and limits](#)
- [Use Power Automate flows](#)

# Copilot Studio licensing

09/22/2025

This article covers Copilot Studio licensing details. Review the [Microsoft Copilot Studio Licensing Guide](#) for more information.

If you already have a Copilot Studio user license, see [Assign licenses and manage access to Copilot Studio](#).

Copilot Studio is available in the US Government Community Cloud (GCC) plan.

Copilot Credits are the common currency across Copilot Studio capabilities. They're available through pay-as-you-go meters and Copilot Credit pre-paid pack subscriptions.

## Important

Starting on September 1, 2025, the common currency for agents changed from *messages* to *Copilot Credits*. There's no change in the quantity per prepaid pack or to the pay-as-you-go rate.

## Getting started in Copilot Studio

To use Copilot Studio, you need one of the following configurations:

- Copilot Studio user license (free of charge). The admin assigns this license to the maker in the [Microsoft Admin Center](#). For details, see [Assign licenses and manage access to Copilot Studio](#).
  - You need a Copilot Studio tenant [prepaid Copilot Credit pack subscription](#) before you can get this user license.
- "Copilot Studio authors" role in [Power Platform admin center](#).
  - Create the [security group in Azure portal/Entra](#) and assign users to the group.
  - In [Power Platform admin center](#), assign the security group to the Copilot Studio author setting.
- Microsoft 365 Copilot license
- Copilot Studio [trial license](#). You can access Copilot Studio to create agents, but you can't publish agents.

## Get access to Copilot Studio

The following sections describe the different ways you can use Copilot Studio.

## Copilot Studio prepaid Copilot Credits subscription

The standalone Copilot Studio subscription lets you build agents on any supported channel and connect to any data using premium connectors.

Get a [standalone Copilot Studio subscription](#) from the Microsoft 365 admin center. For more information, see [Assign licenses and manage access to Copilot Studio](#).

- Copilot Credit packs: \$200 per tenant per month for 25,000 Copilot Credits.
- Copilot Studio capabilities are licensed by Copilot Credit packs. One pack equals 25,000 Copilot Credits a month. For extra capacity, sign up for pay-as-you-go to ensure business continuity. The pay-as-you-go meter starts counting after you use your Copilot Credit pack entitlements.

## Copilot Studio pay-as-you-go

Create a [billing policy associated with your Azure subscription](#) in Power Platform admin center.

- Pay-as-you-go: \$0.01 per Copilot Credit.
- With the pay-as-you-go meter, at the end of each month, your organization pays only for the actual number of Copilot Credits its agents use during the month. No up-front license commitment is needed, so your organization can scale usage and ensure business continuity for mission-critical business processes. For more information about setting up the pay-as-you-go meter, see [Set up pay-as-you-go](#).

## Microsoft 365 Copilot

A Microsoft 365 Copilot license lets you use [Copilot Studio to extend Microsoft 365 Copilot with agents](#).

- Copilot Studio in Microsoft 365 Copilot: \$30 per user per month

### Note

With the Copilot Studio in Microsoft 365 Copilot license, agents you build in Copilot Studio for Microsoft Teams, SharePoint, and Microsoft 365 Copilot are included at no extra charge.

## Copilot Studio trial license

The trial license lets you sign up for [Copilot Studio as an individual](#).

Admins can [assign the trial licenses in the Microsoft Admin Center](#) and block the ability to sign up.

 **Note**

The trial license gives you access to Copilot Studio to create agents. You can test your agents using the test chat panel. However, you can't publish the agent.

## Billed Copilot Credit

Copilot Credit measure the time and effort your agent needs to retrieve information, respond to prompts, and use any actions or custom skills. The number of Copilot Credits counted for each response or action depends on the complexity of the task the agent completes. For more information about Copilot Credits, see [Copilot Credits and events scenarios](#).

## Capacity enforcement

Copilot Studio enforces purchased capacity monthly, and unused Copilot Credit don't carry over to the next month. If you exceed your purchased capacity, adjust your quantity per standard Microsoft terms to stay in compliance. Monitoring, reporting, and alerting help you manage Copilot Studio capacity. If your usage exceeds your purchased capacity, technical enforcement applies and can result in service denial. For more information, see [Enforcement policy](#).

## Copilot Studio in Microsoft 365 Copilot

If you have a Microsoft 365 Copilot license, using agents in Copilot Chat, Microsoft Teams, or SharePoint for *classic answers*, *generative answers*, or *Microsoft Graph tenant grounding* doesn't count against the Copilot Studio message pack or meter. For example, when Microsoft 365 Copilot licensed users use agents in Microsoft 365 services or apps, *classic answers*, *generative answers*, or *Microsoft Graph tenant grounding* have zero-rated usage.

A Microsoft 365 Copilot license includes these features and capabilities:

- Microsoft 365 Copilot Chat, including Pages
- Copilot in Microsoft 365 apps, including Teams
- SharePoint agents

- Agent capabilities in Copilot Studio when used in Microsoft 365 Copilot, Teams, or SharePoint
- Copilot Analytics (Microsoft Copilot Dashboard, customizable prebuilt business impact reports, and advanced analytics from Viva Insights)
- SharePoint Advanced Management

## Copilot Studio use rights included with Microsoft 365 Copilot license

Some licenses include limited Copilot Studio use rights and session capacity.

[Expand table](#)

Feature	Copilot Studio pay-as-you-go	Copilot Studio message packs	Copilot Studio use rights with Microsoft 365 Copilot
Included Copilot Credits	Pay as you go	25,000 Copilot Credits	Unlimited
Generative AI	Included	Included	Limited
Create and publish your own agents anywhere	Included	Included	Not included
Create and publish your own agents and plugins to extend Microsoft 365 Copilot	Not included	Not included	Included
Power Automate for Copilot Studio cloud flows	Not included for pay-as-you-go only tenants	Included	Not included
Agent flows	Not included	Not Included	Included
Standard Power Platform connectors	Included	Included	Included
Premium and custom Power Platform connectors	Included	Included	Included
On premises and cloud service data transfer for Power Platform connectors	Included	Included	Included
Dataverse for Copilot Studio	Included	Included	Included
Managed environments	Included	Included	Included

Feature	Copilot Studio pay-as-you-go	Copilot Studio message packs	Copilot Studio use rights with Microsoft 365 Copilot
Available channels to publish your copilot/plugins	External and internal channels	External and internal channels	Microsoft 365 experience

# Billing rates and management

10/03/2025

This article describes the rates for the different features and capabilities used in agents, which are charged to the Copilot Studio pay-as-you-go meter or Copilot Credit packs.

## i Important

For the most up-to-date Copilot Studio licensing and billing information, refer to the the [Microsoft Copilot Studio Licensing Guide](#).

*Copilot Credits* are the unit that measures agent usage. The total cost is calculated based on the sum of the Copilot Credits used by your organization. The number of Copilot Credits consumed by an agent depends on the design of the agent, how often customers interact with it, and the features they use.

The purchase of a Copilot Studio license includes a specific number of billed Copilot Credits. This capacity is pooled across the entire tenant.

## i Important

- Starting on September 1, 2025, the common currency for agents changed from *messages* to *Copilot Credits*. There's no change in the quantity per prepaid pack or to the pay-as-you-go rate.
- Starting on March 25, 2025, deep reasoning is available in AI prompts and agent flows. Charges for deep reasoning in AI prompts use the *Text and generative AI tools (premium)* rate, and charges for agent flows use the *Flow actions* rate. For more information, see the [Copilot Credits and events scenarios](#) table.

## Copilot Credits and events scenarios

The following table illustrates the differences in the subscription models for the cost of Copilot Studio events.

[ ] Expand table

Copilot Studio feature	Billing rate	Use in Microsoft 365 Copilot scenarios <sup>1</sup>	Autonomous triggers <sup>2</sup>
Classic answer	1 Copilot Credit	No charge	N/A
Generative answer	2 Copilot Credits	No charge	2 Copilot Credits
Agent action	5 Copilot Credits	No charge	5 Copilot Credits
Tenant graph grounding for messages	10 Copilot Credits	No charge	10 Copilot Credits
Agent flow actions <i>per 100 actions</i>	13 Copilot Credits	No charge	13 Copilot Credits
AI tools			
- Text and generative AI tools (basic) <i>per 10 response</i>	1 Copilot Credit	No charge	1 Copilot Credit
- Text and generative AI tools (standard) <i>per 10 response</i>	15 Copilot Credits	No charge	15 Copilot Credits
- Text and generative AI tools (premium) <i>per 10 response</i>	100 Copilot Credits	No charge	100 Copilot Credits

<sup>1</sup> Interactive use of classic answers, generative answers, tenant graph grounding and agent actions by authenticated Microsoft 365 Copilot users, in Microsoft 365 apps and services, are included at no extra cost.

<sup>2</sup> Autonomous triggers refer to events or conditions that automatically initiate an agent to take action, without requiring a user to manually invoke it.

- **Classic answers:** These events are predefined responses manually authored by agent makers. They're static and don't change unless manually updated. They're typically used where precise and controlled responses are the only ones we want the agent to generate.
- **Generative answers:** These events are dynamically generated using AI models, such as Generative Pretrained Transformers (GPTs). They can adapt and change based on the context and the knowledge sources they're connected to. They're useful for handling a wide range of topics and providing more flexible and natural interactions.
- **Tenant graph grounding for Copilot Credits:** These events provide higher quality grounding for your agents using retrieval-augmented generation (RAG) over your tenant-wide Microsoft Graph, including external data synced into Microsoft Graph through

connectors. This capability results in more relevant and improved responses and ensures that the grounding information is up-to-date. This capability is optional, and you can turn it on or off for each agent. For more information, see [Tenant graph grounding](#).

- **Agent actions:** Agent Actions refer to steps such as triggers, deep reasoning, and topic transitions that appear on the activity map in Copilot Studio when testing an agent. When the agent invokes either the Knowledge Search/Retrieval tool or the AI Tools prompt, the invocation itself is billed at the Agent Action rate. In addition, usage of the Knowledge Search/Retrieval tool and the AI Tools prompt is metered separately, and they're charged based on their respective consumption rates.
- **Text and generative AI tools:** Prompt tools embedded within an agent enable the creator to direct the underlying model to perform intelligent document and image processing tasks, behave in a task-specific manner, or generate scenario-specific outputs. There are three types of tools, basic, standard, and premium, which are based on the underlying language model of the prompts. The premium text and generative AI tools item are used to charge for advanced reasoning in agents. For more information, see [AI Builder licensing in Microsoft Copilot Studio](#) and [Prompt Tokens](#).
- **Agent flow actions:** Item used to charge for agent flows that enhance AI agents with agent flows, which are predefined sequences of flow actions to execute repetitive tasks quickly, without requiring agent reasoning and orchestration at each step. For more information, see [Agent flows overview](#).

Each interaction with an agent might utilize multiple feature types simultaneously. For example, an agent grounded in a tenant graph could use 12 Copilot Credits (10 Copilot Credits for tenant graph grounding, and 2 Copilot Credits for generative answers) to respond to a single complex prompt from a user.

For example, the following scenarios illustrate the usage of these features:

## Customer support agent

You have a customer support agent on your website that answers questions based on customer return policies, and product manuals that you provided to the agent as a knowledge source.

An average run comprises four classic answers for return-related questions, and two generative answers for troubleshooting questions. The average is 900 customers per day. The estimated cost per day is based on the following calculation: 
$$[(4 \times 1) + (2 \times 2)] \times 900 \text{ customers} = 7200 \text{ Copilot Credits}$$
.

## Sales performance agent

You have a tenant graph grounded agent in Microsoft 365 Copilot Chat. This agent answers employee questions based on sales data connected to Microsoft Graph using Graph data connectors.

An average run comprises four generative answers and four tenant graph grounded Copilot Credits. The average is 50 Microsoft 365 Copilot licensed users and 100 unlicensed users. The estimated cost per day is based on the following calculation:  $[(4 \times 2) + (4 \times 10)] \times 100 \text{ users} = 4,800 \text{ Copilot Credits}$ .

## Order processing agent

An internal-facing agent is autonomously triggered anytime a new order is received by the organization. The agent uses a single knowledge source to get product details about items ordered, and triggers 4 action calls to confirm product availability, view shipping timelines, approve the order, and send an email to the customer with all details. Actions and topics are agent actions in generative orchestration mode. The estimated cost per day is based on the following calculation:  $[(4 \times 5)] = 20 \text{ Copilot Credits}$ .

## Overage enforcement

In an environment, when consumption exceeds available capacity, the environment is in overage. Microsoft allows some level of overage consumption, similar to a grace period, to avoid blocking business processes.

If your environment has no more capacity, you have the following options:

- Reallocate existing capacity from the organization (tenant) or environment level.
- Purchase more capacity and make it available to your environment.
- Set up a consumptive meter or pay-as-you-go meter to handle the overage.

## Enforcement policy

Applies to all tenants operating under the Copilot Studio prepaid capacity model for custom agent usage (conversational and autonomously triggered).

## Usage threshold

Enforcement is triggered when a tenant reaches 125% of their prepaid capacity.

## Action on overage (125%)

Custom agents are disabled. Disabling an agent doesn't interrupt an ongoing conversation. All subsequent attempts to invoke the agent are rejected until capacity is increased or reset.

## Notification mechanism

An email notification is sent to the tenant's designated administrator and the notification is also posted in the Power Platform admin center.

## Agent behavior post-enforcement

After enforcement is triggered and the current conversation concludes, the agent is disabled. When end users attempt to interact with the agent after enforcement, they receive one of the following responses:

- "There is a billing issue."
- "This agent is currently unavailable. It has reached its usage limit."

## Enforcement example

If the customer allocated or reserved capacity in an environment, the system honors the capacity. Consider the following example of a customer having four different environments, and how their Copilot Credit capacity is enforced.

A customer has 25,000 Copilot Credits, and the following allocation structure is being used:

- Environment A has 10,000 Copilot Credits allocated.
- Environment B has no allocation.
- Environment C has no allocation.
- Environment D has an allocation of 500 Copilot Credits and uses pay-as-you-go.

The remaining tenant allocation is 14,500 Copilot Credits. Environment B and Environment C draw and consume against the remaining 14,500 Copilot Credits. If the consumption of Copilot Credits from Environment B and Environment C exceeds 125% of the 14,500 Copilot Credits, the overage enforcement is invoked.

If Environment A draws or consumes Copilot Credits against its allocation of 10,000 Copilot Credits, the following scenario applies. When the 10,000 Copilot Credits are consumed, Environment A can consume from the tenant.

If Environment A consumes from the tenant, it joins Environment B and Environment C in consuming from the tenant capacity. If the tenant reaches 125% Copilot Credit consumption,

enforcement is invoked.

If the tenant is already at 125% of Copilot Credit consumption because of Environment B and Environment C, enforcement isn't placed on the agents in Environment A, so long as Environment A has remaining capacity from its allocation of 10,000 Copilot Credits.

For Environment D, when the tenant is in overage, this environment isn't impacted. Because once Environment D reaches its 500 Copilot Credit limit, the pay-as-you-go meter is invoked.

## Set up pay-as-you-go consumptive meter

[Pay-as-you-go](#) is a way to pay for Copilot Studio using an Azure subscription, which allows you to get started building agents without any license commitment or upfront purchasing.

In the Power Platform admin center, you can link environments to an Azure subscription using a billing policy.

Linking an environment to an Azure subscription enables billing through Azure meters. Any app usage or Dataverse and Power Platform usage that exceeds the included amounts is billed to the Azure subscription.

You can unlink your environment from the Azure subscription at any time and then usage is no longer billed.

 Note

For instructions on how to set up your pay-as-you-go consumptive meter, see [Set up pay-as-you-go](#).

## View Copilot Credit consumption

You can view [Copilot Credit consumption reporting](#) in the Power Platform admin center.

1. In Power Platform admin center, select **Licensing** on the side navigation.
2. From the list of products, select **Copilot Studio**.
3. Select the **Summary** tab to view the capacity summary of your product.
4. Select the **Environments** tab and select the desired environment to view the credit consumption details for the selected environment.

# Manage sessions and capacity

08/22/2025

## !**Note**

This section is for billed sessions in the legacy Power Virtual Agents license, which was available for purchase starting on December 1, 2023. This legacy license is no longer available for purchase starting on January 1, 2024.

A billed session is an interaction between a customer and an agent. It represents one unit of consumption. The purchase of a Copilot Studio license includes a specific number of billed sessions. Copilot Studio pools this capacity across the entire tenant. Your consumed capacity isn't reported at the tenant level. However, you can view consumption for individual agents.

A billed session begins when a user topic or premium functionality is triggered. A billed session ends when one of the following conditions is met:

- The user ends the chat session. When the agent doesn't receive a new message for more than 30 minutes, the session is considered closed.
- The session is longer than 60 minutes. The first message that occurs after 60 minutes starts a new session.
- The session has more than 100 turns. A turn is defined as one exchange between a user and the agent. The 101st turn starts a new session.

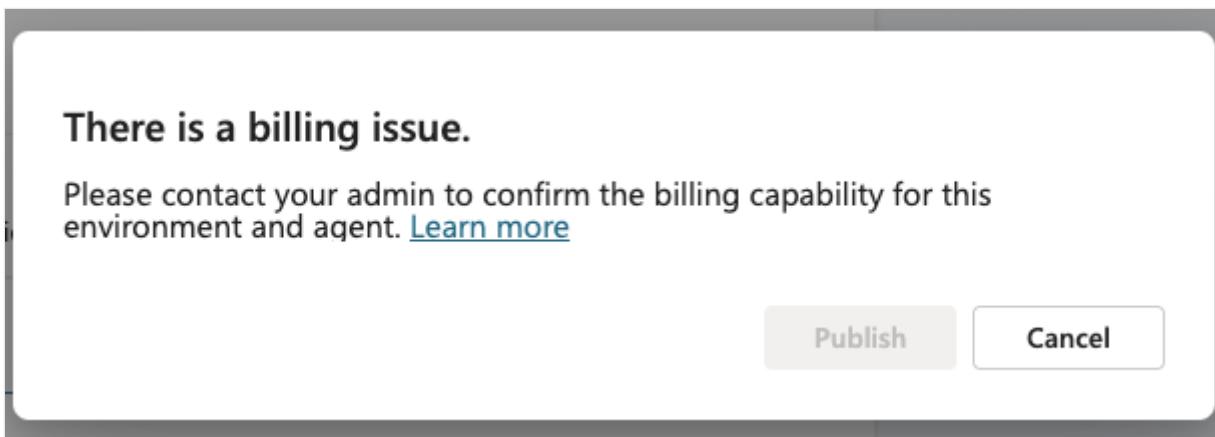
# FAQ for Copilot Studio billing and licensing

09/24/2025

These frequently asked questions (FAQ) describe the billing and licensing for Copilot Studio.

## I'm trying to publish my agent but I'm getting a billing issue error message.

The following error message indicates the environment or tenant doesn't have either enough prepaid or pay-as-you-go messages to support publishing.



## If my agent is configured to proactively greet the end user, will that count as a billed Copilot Credit or an engaged conversation?

A proactive greeting—where the agent initiates a message without being prompted—does count as a billed Copilot Credit, even if the end user doesn't respond. However, for agent analytics, if the end user doesn't reply, the session is categorized as an unengaged conversation.

The following are distinct concepts:

- billing is based on the agent's activity
- engagement metrics reflect the user's response behavior

### ! Note

If the agent is triggered to be ready for a conversation with the user, regardless whether a greeting message is sent, it consumes Copilot Credits as a Classic Answer.

# Do agents built with Copilot Studio For Teams consume Copilot Credits when used in Microsoft Teams?

- Agents created within the Teams environment using Copilot Studio for Teams don't consume Copilot Credits, regardless of who uses them.
- Agents created in the standalone Copilot Studio and deployed to the Teams channel do consume Copilot Credits. However, when they're used by end users with a Microsoft 365 Copilot license, their usage isn't billed. For more information, see the [Copilot Credits and events scenarios](#) table.

## Do preview features count against usage capacity?

Copilot Studio features that are in preview are charged at the rate described in the [Copilot Credits and events scenarios](#) table, and count against your purchased capacity.

## What happens to the remaining capacity at the end of the month? Can this capacity carry over to the following month?

No, unused capacity isn't carried over to the next month. Credit usage is counted on a monthly basis and reset on the first day of the month. Available capacity is based on allocated capacity.

## When is the Copilot Credit consumption count reset? Is resetting based on the license acquisition date?

Credit usage is counted on a monthly basis, which starts on the first day of each month. Credit usage isn't based on the license acquisition date. Copilot Credit usage of an environment shouldn't exceed the Copilot Credit allocation.

## Where can I see Copilot Credit usage for this month?

Overall organization (tenant) level credit consumption is available in the Power Platform admin center in the **Licensing > Products > Summary** tab.

The data is aggregated daily, although some data might be delayed. If delayed, the consumption bar shows all known consumption at the **Last updated** date in the tooltip box.

## **Do chats in the embedded test chat count as billable Copilot Credits?**

Messages in the embedded test chat don't count toward billed sessions.

## **What's the difference between total sessions, engaged sessions, and billed sessions?**

Billed sessions are used to measure usage against the capacity that is allocated when you purchase a license. For more information, see the Microsoft Power Platform Licensing Guide.

Total sessions and engaged sessions are metrics of analytics sessions, used to determine how efficient an agent is in addressing a user's questions.

Billed sessions and analytics sessions are distinct and independent. For more information, see [Analyze agent billed message consumption](#).

## **Where can I see the total number of billed sessions for my organization?**

You can only view billed sessions for individual agents.

## **If I'm on a trial subscription but have some billed sessions, do I have to pay?**

Not necessarily. The trial offer provides you with some billed sessions.

## **If an environment is in overage, do the limitations apply to all environments?**

No. Limitations only apply to the environments in overage. Other environments' allocated Copilot Credits aren't affected.



# Configure customer-managed encryption keys

Article • 05/19/2025

Customers might have data privacy and compliance requirements to secure their data by encrypting their data at-rest. This ensures data is protected from malicious actors, even if the storage is compromised, because the malicious actors can't get access to the data without the encryption key.

All customer data stored in Copilot Studio is encrypted at-rest with strong Microsoft-managed encryption keys by default. Microsoft stores and manages the database encryption key for all your data, so you don't have to. However, Power Platform provides an option to use customer-managed encryption key (CMK) for added data protection control. You can self-manage the database encryption key that is associated with your Microsoft environment. This capability allows you to rotate or swap the encryption key on demand, and prevents Microsoft's access to your customer data when you revoke key access to our services at any time.

Copilot Studio supports CMK, which lets customers control access to their data within Copilot Studio. We support the standard Power Platform implementation, and customers don't need to do anything specific to enable CMK for Copilot Studio. Power Platform only allows Managed Environments to be enabled for CMK.

## Enable CMK for Copilot Studio

Copilot Studio supports the Power Platform implementation of CMK. For more information, see [Manage your customer-managed encryption key](#). When CMK is turned on for the Copilot Studio environment, all Copilot Studio data is encrypted using the customer's key. The customer can cycle keys or turn off CMK as needed.

### Important

- Data within environments that already have CMK turned on before April 7, 2025 continue to use Microsoft managed keys for encryption. In order to use CMK in environments that have CMK turned on prior to that date, remove CMK and then turn it on again.
- Once CMK is turned on, all future changes and data is encrypted using the customer's key. Any previously persisted data continues to use the Microsoft managed keys for encryption.

- Microsoft recommends that you test CMK support for Copilot Studio in a new test environment, and not in a production environment, especially not in an environment with live customer traffic.

## Maker and agent user experience when CMK is applied

Copilot Studio is integrated within Power Platform CMK processes. When CMK is first turned on in Power Platform, it can take up to 48 hours to fully activate, which means Copilot Studio services aren't available until activation is complete.

## Data covered by CMK

The following Copilot Studio data is included in CMK:

- All data in the agent definition
- Published snapshots of the agent definition
- Agent telemetry
- Agent user conversations

### Note

Agent Builder agents aren't covered by CMK, because they're not tied to an environment.

# Configure Virtual Network support for outbound connections from agents

07/04/2025

When you use [Virtual Network support in a Power Platform environment](#), you can securely connect to and integrate Power Platform and Dataverse components with cloud services, or services hosted inside your private enterprise network, without exposing them to the public internet.

Copilot Studio integrates with Power Platform virtual networks over a private endpoint for these scenarios:

- Agents that retrieve keys from Azure Key Vault [over HTTP](#)
- Agents that send telemetry to a private endpoint-enabled instance of [Application Insights](#)
- Agents that use a virtual network-supported connector (like the SQL Server connector) to get data from Azure SQL Server

If you set up a virtual network for a Power Platform environment and enable Copilot Studio to [capture telemetry with Application Insights](#) or [make HTTP requests with your agent](#) over the virtual network, then the calls from Power Platform to Azure resources and Application Insights go through your private network.

## Prerequisites

- Your environment must be [a Managed Environment in Power Platform](#)
- You must have [Virtual Network support enabled for your Power Platform environment](#).  
Also see [Set up Virtual Network support for Power Platform](#) to create virtual networks and delegate subnets that can connect between Azure resources and your Power Platform environment.
- You must be a Power Platform [tenant admin](#) or have the [Environment Admin role](#)

## Enable virtual network support for your environment

To connect to services through a private endpoint, you must have [virtual network support enabled for Power Platform](#).

You can enable virtual network support manually, by following the instructions at [Set up Virtual Network support for Power Platform](#) to create virtual networks and delegate subnets that can

connect between Azure resources and your Power Platform environment.

You can also use a prebuilt Azure Resource Manager (ARM) template to configure and connect your Power Platform environment with Azure and enable virtual network support:

1. Download the [ARM template from the Microsoft Copilot Studio samples repository on GitHub](#).
2. Open PowerShell, connect to your Azure subscription and deploy the template with the [New-AzDeployment command](#) as follows:

```
Connect-AzAccount -Subscription "<Azure subscription>"
New-AzSubscriptionDeployment -Name "<name of deployment>" -TemplateFile "<template.json>" -Location "<Azure geo>"
```

where:

- *<Azure subscription>* is your subscription ID.
- *<name of deployment>* is the name you want to give this deployment.  
The name can be anything you choose, but defaults to the template's filename if you leave it blank.
- *<template.json>* is the path and filename of the template file.
- *<Azure geo>* is the geographic region where you want the deployment management files to go, such as `West US`. The region doesn't control where the template creates the resources.

See [Deploy resources with ARM templates and Azure PowerShell](#) for more information about ARM templates and management.

#### (!) Note

You only need to configure your virtual network using either the ARM template, or manually. You don't need to do both.

Review the overview about [Virtual Network support for Power Platform](#), before following the instructions at [Set up Virtual Network support for Power Platform](#) to create virtual networks and delegate subnets that can connect between Azure resources and your Power Platform environment.

## Retrieve keys from Azure Key Vault over HTTP

When you [set up a virtual network for your Power Platform environment](#), you can configure your Copilot Studio agents to retrieve information from Azure resources with HTTP calls.

First, you set up a private link and endpoint for Azure Key Vault. Then, after validating that the link is working, you add a HTTP Request node from the agent's authoring canvas in Copilot Studio to connect to Key Vault.

## Set up a private link

Follow the instructions at [Integrate Key Vault with Azure Private Link](#) to:

- ✓ [Create a new key vault and establish a private link that scopes the link to your Azure subscription and the resource group where your Key Vault is located, or Establish a private link connection to an existing key vault.](#)
- ✓ [Validate that the private link to Key Vault is working.](#)

### 💡 Tip

If your endpoint isn't correct, review the instructions and related articles for private links and private endpoints in the [Diagnose private links configurations issues on Azure Key Vault](#) article.

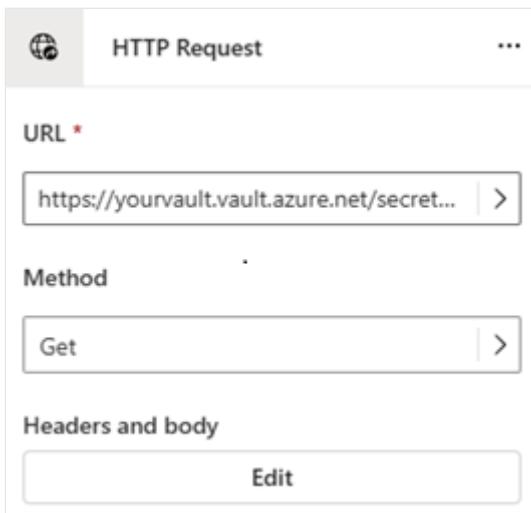
## Use HTTP Request nodes to connect over a private network

After you configure the private link to Key Vault, you add an HTTP Request node to an agent in Copilot Studio to connect over the private network. You specify the connection details to the private endpoint in the node, and when that node is reached in the agent's conversation, the request is made and the information retrieved.

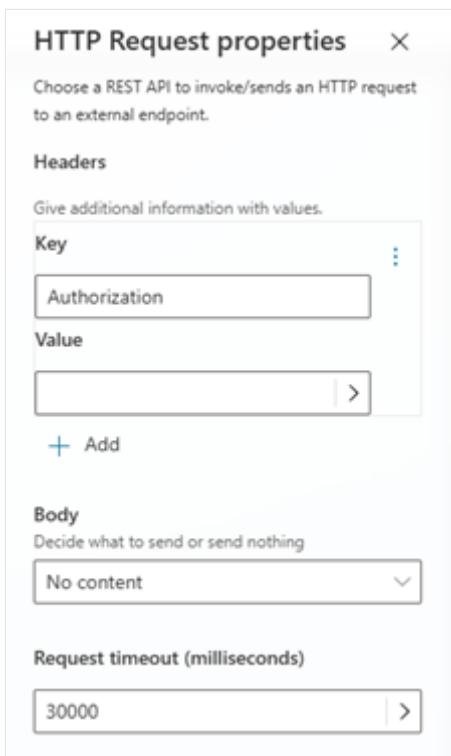
1. In Copilot Studio, on the top menu bar, select an environment where Virtual Network support is enabled.
2. Create or open an existing agent in that environment. If you create a new agent, you can skip the initial configuration steps in the conversational canvas.
3. With the agent open, create or modify a topic in the authoring canvas.
4. Follow the instructions at [Make HTTP requests](#) to add an HTTP request node to the topic.
5. Use the following settings in the HTTP Request node:
  - **URL:** Enter the URL for your Azure Key Vault private endpoint, for example, `https://yourkeyvault.vault.azure.net/secrets?api-version=7.3`. Replace

`yourkeyvault` with the name of your Key Vault.

- **Method:** Select `GET` to retrieve secrets from Key Vault.
- **Headers and body:** Select edit.



- In **HTTP Request properties**, enter `Authorization` as the **Key**, and `Bearer <access-token>` as the **Value**, where `<access-token>` is your Azure access token.



6. Save the topic, and test that the node works by triggering the conversation in the agent's test canvas.

## Send telemetry to a private endpoint-enabled instance of Application Insights

When you [set up a virtual network for your Power Platform environment](#), you can configure your Copilot Studio agents to send telemetry to a private endpoint-enabled instance of Application Insights. Doing so allows you to monitor and analyze the performance and usage of your agents without exposing the data to the public internet.

First, you set up a private link and endpoint for Application Insights. Then, after validating that the link is working, you connect Copilot Studio to Application Insights and it'll send telemetry data over the private link.

## Set up a private link

An [Azure Private Link to Azure Monitor](#) lets Copilot Studio use your virtual network to send agent telemetry to Azure Monitor over a private IP address instead of a public IP address.

Azure Monitor is the backend data platform used to collect and store telemetry data, including Application Insights data.

Follow the instructions at [Configure private link for Azure Monitor](#) and:

- ✓ [Create an Azure Monitor Private Link Scope \(AMPLS\)](#) to scope the link to your Azure subscription and the resource group where your Azure Monitor resources are located.
- ✓ [Connect Application Insights component resources to the AMPLS](#).
- ✓ [Create a private endpoint for the Application Insights resources you added to the scope](#) that Copilot Studio can connect to in your virtual network and over your subnet. This endpoint is used to send telemetry data from the agent to the AMPLS.
- ✓ [Validate that the private link to Azure Monitor is working](#).

You can also [configure which networks can connect to resources in your AMPLS, without using a scope](#), in the [Network Isolation](#) page for your AMPLS. Directly configuring networks is useful if you have multiple virtual networks and want to restrict access to the AMPLS to only certain networks or subnets.

### Tip

If your endpoint isn't correct, review the instructions and related articles for private links and private endpoints in the [Configure private link for Azure Monitor](#) article.

## Connect Copilot Studio to Application Insights

With the private link set up, you can connect Copilot Studio to Application Insights and it'll use your virtual network to send telemetry data.

Follow the instructions at [Capture telemetry with Application Insights](#).

### **Important**

Ensure you get the correct **Connection string** for the private endpoint-enabled Application Insights .

You can validate it's the correct resource by checking the values under **Resource group** and **Subscription** on the **Overview** section for Application Insights in the Azure portal.

Telemetry from Copilot Studio agents appears in the Application Insights resource you configured. You can use the **Live Metrics Stream** to see telemetry data in real time, or use the **Logs** section to query and analyze the data.

## Use virtual network-supported connectors to get data

When you [set up a virtual network for your Power Platform environment](#), you can configure your Copilot Studio agents to use virtual network-supported connectors to connect to data and services over your private network.

You can [use any connector that has native support for virtual networks](#).

Using virtual network-supported connections lets you securely connect to your cloud-hosted data sources, such as [Azure SQL](#) or [SQL Server](#), over private endpoints without exposing them to the internet.

Follow the instructions at [Use Power Platform connectors in Copilot Studio](#) to add and configure the connector you want to use in a topic or tool.

# Use IP firewalls for inbound connections to agents and Copilot Studio (preview)

10/25/2025

[This article is prerelease documentation and is subject to change.]

If you configure [IP firewalls in Power Platform environments](#), connections made into your network to Copilot Studio agents or the Copilot Studio web app from restricted or unallowed IPs or IP ranges are blocked.

An IP firewall helps mitigate threats, helps you apply security principles such as zero trust, and limits access to only those users who need it, when they need it. The IP firewall can help to:

- Mitigate insider threats like data exfiltration. For example, a malicious user who tries to access Copilot Studio from a disallowed IP location is blocked from doing so in real time.
- Prevent token replay attacks. For example, if a user steals an access token and tries to use it to access a conversation from outside an allowed IP range, Copilot Studio denies the access attempt in real time.

When a request is made to Copilot Studio, the requested IP address is evaluated in real time against the IP ranges configured for the Power Platform environment. If the IP address is in the allowed ranges, the request is allowed. If the IP address is outside the IP ranges configured for the environment, the IP firewall denies the request with an error message to the agent user or the agent maker at the time of the block.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

## Prerequisites

- Your environment must be [a Managed Environment in Power Platform](#)

- You must be a Power Platform admin, such as a [tenant admin](#) or have the [Environment Admin role](#)

## Enable IP firewall for Power Platform environments

### Important

You must be using [managed environments in Power Platform](#).

Follow the instructions at [IP firewall in Power Platform environments - Enable the IP firewall](#) to:

- Enable the IP firewall for:
  - Single Power Platform environments, with the Power Platform admin center
  - Single Power Platform environments, with the Dataverse OData API
  - Environment groups, with the Power Platform admin center
- [Validate that the IP firewall is working](#)

You can also configure the firewall to use service tags, operate in audit-only mode, and to allow access for Microsoft trusted services.

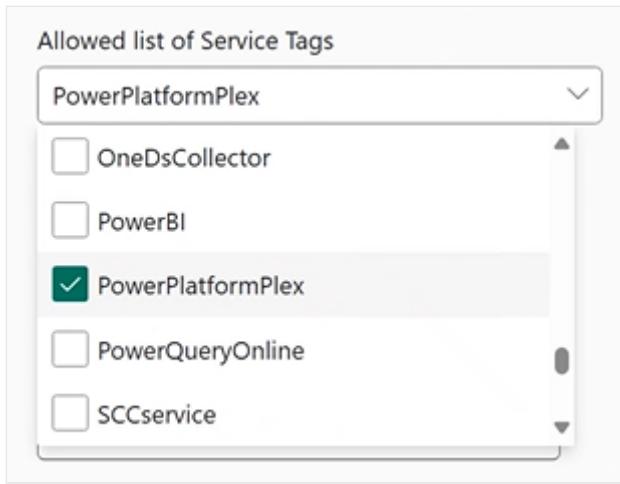
## Service tags

Service tags are predefined sets of outgoing public internet IP ranges and addresses that are used by customer-operated Azure and Power Platform services hosted by Microsoft systems. For example, you might have connections from Azure Web Apps, or you created a Dataverse plugin that needs to access Copilot Studio endpoints.

Because there can be multiple addresses and ranges for these services, and these services may be added to or changed, using service tags removes the manual need to find and enter individual addresses or ranges for the services you're connecting to.

When creating a plugin, tool, or configuring firewalls, you can select the service tags you want to be included in your firewall's configuration.

For example, if you create a Dataverse plugin to connect to Copilot Studio based on an event in your Dataverse instance, you would select the **PowerPlatformPlex** service tag when configuring your firewall:



In this example, the PowerPlatformPlex tag contains the required Power Platform and Dataverse addresses needed to allow the services to interact.

For more details on service tags in Power Platform, see [Power Platform URLs and IP address ranges](#). On that page, PowerPlatformPlex and Power Platform connection requirements are also described under the [IP addresses required](#) section.

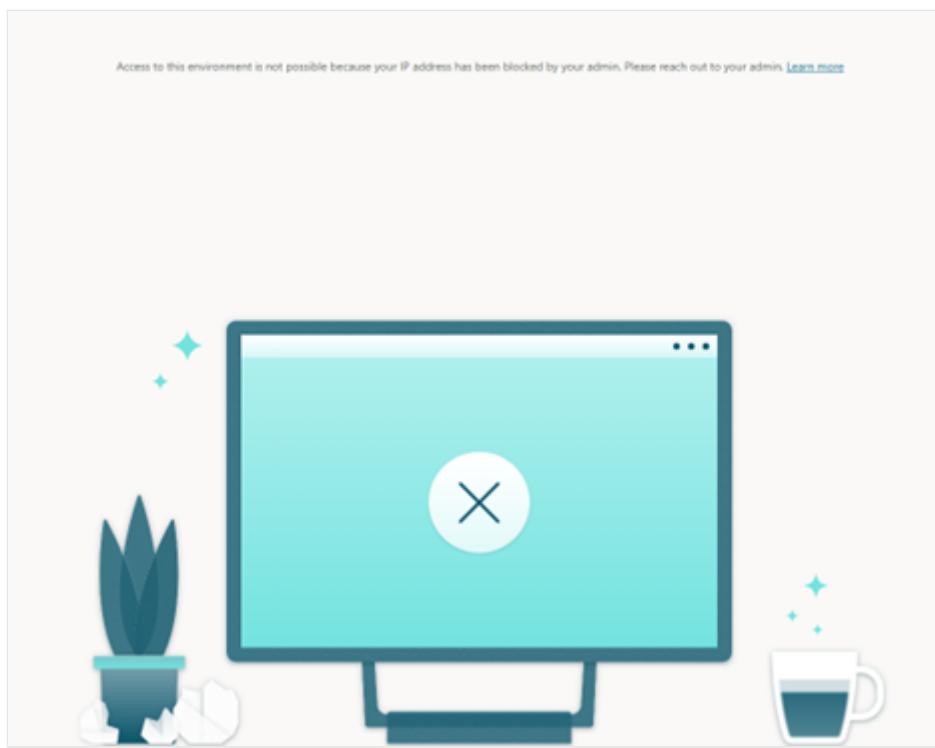
## Validate IP firewall for agent authors and agent users

You should test the IP firewall to verify that it's working:

1. To test that agent author and maker access to Copilot Studio is blocked:
  - a. From an IP address that isn't in the allowed list, open [Copilot Studio](#).
  - b. If the firewall is working, you'll see the following error message, along with an email address for the admin:

*Access to this environment is not possible because your IP address has been blocked by your admin. Please reach out to your admin.*

Access to this environment is not possible because your IP address has been blocked by your admin. Please reach out to your admin. [Learn more](#)



2. To test that agent users are blocked from interacting with an agent:

- From an IP address that isn't in the allowed list, open an agent on [your demo website](#).
- If the firewall is working, you'll see the following error message:

*The agent isn't available right now because your IP address has been blocked. Please contact your agent owner.*

## The agent is not available right now

The agent is not available right now because your IP address has been blocked.  
Please contact your agent owner.  
Session ID: <session identifier>

[Reload](#)

## Known limitations

- Agent IP Firewall restrictions are enforced on web chat and APIs that carry conversational content.
- Agent IP Firewall restrictions will expand to additional endpoints over time.
- Agents published on the following channels don't enforce IP Firewall restrictions:

- Microsoft channels such as [Teams](#) and [Microsoft 365 Copilot](#).
- Other channels such as [Facebook](#).
- [Connected to Dynamics 365 Customer Service](#).
- Existing chat sessions don't disconnect when admins apply IP firewall restrictions. The restrictions apply once the session restarts.

# Block unauthorized self-service agent sign-ups

06/30/2025

## ⓘ Note

Graph PowerShell is required. Azure PowerShell and MSOnline have been deprecated.

[Find Azure AD and MSOnline cmdlets in Microsoft Graph PowerShell](#)

As an administrator, you can prevent users in your organization from signing up for free trials when they don't have administrative permission. To block unauthorized (viral) self-service sign-ups at a company-wide level, perform these steps:

1. Block new users from signing up for viral trial subscriptions (Copilot Studio).
  - a. In Graph PowerShell, in the `Update-MgPolicyAuthorizationPolicy` cmdlet, set the `AllowedToSignUpEmailBasedSubscriptions` parameter to False.

PowerShell

```
Set-Update-MgPolicyAuthorizationPolicy -
 AllowedToSignUpEmailBasedSubscriptions $False
```

For more information, see [Graph PowerShell](#).

2. Remove existing users from viral license assignment, using either the Microsoft 365 admin center or PowerShell.
  - **Microsoft 365 admin center:** Sign in to the Microsoft 365 portal and unassign in bulk.
    - a. Sign in at [admin.cloud.microsoft.com](https://admin.cloud.microsoft.com) and go to the **Billing > Licenses** page.
    - b. Select a product.
    - c. Select the users for whom you want to unassign licenses.
    - d. Select **Unassign licenses**.
    - e. In the **Unassign licenses** box, select **Unassign**.

# Work with Power Platform environments in Copilot Studio

09/24/2025

With Copilot Studio, you can create agents in different environments and easily switch between them.

An environment is a space to store, manage, and share your organization's business data. The agents you create are stored in an environment (apps and flows are also stored in environments). Environments might also have different roles, security requirements and target audiences, and each environment is created in a separate location. For more information, see the following articles:

- [Data locations](#)
- [Power Platform environments overview](#)
- [Power Platform environment routing](#)

## Use environments

There are many strategies for using multiple environments. For example, you can:

- Create separate environments that correspond to specific teams or departments in your company, each containing the relevant data and agents for its audience.
- Create separate environments for different global branches of your company.
- Create separate environments to satisfy data residency requirements.

You can also [enable maker welcome content for your environments](#), and [configure environment routing](#) in the Power Platform admin center.

You can also build all your agents in a single environment if you don't need or want to use different ones.

### Note

We recommend using a non-default production environment for agents that you want to deploy to production.

Learn more about the [types of environments](#).

## Create a new environment for your agents

When you first sign in and create a new agent, a default environment is created for you.

You can, however, create more environments as needed by using the [Power Platform admin center](#).

1. Go to the [Power Platform admin center](#) and sign in using your work account. Use the same tenant that you use for Copilot Studio.
2. On the side navigation, select **Manage**.
3. Select **Environments**.
4. Select **New**. The **New environment** pane opens. Provide the following information:
  - a. **Name**: Specify a unique name for the environment.
  - b. **Region**: Select the [support data region](#) where you want the environment to be created.
  - c. **Get new features early**: Set this option to **Yes** to receive early access to new features.
  - d. **Type**: Select the environment type.

 **Note**

A production environment is intended for production scenarios and isn't subject to the same restrictions as a [trial environment](#). If you are trying out Copilot Studio for free, make sure you set the environment to **Trial**. The standard limitations for [trial environments](#) apply.

- a. **Purpose**: Provide a description of the environment's purpose.
- b. **Add a Dataverse data store?**: Set this option to **Yes** to add a Dataverse data store to the environment.
- c. **Pay-as-you-go-with Azure**: Set this option to **Yes** to enable pay-as-you-go billing for the environment.
- d. Select **Next**.

 **Note**

You must create the environment in a supported region. Learn more about [data locations](#).

5. Specify the details of the Dataverse data store, such as the language, currency type, and security group and then select **Save**. This operation takes you back to the **Environments** tab.
6. Select **Refresh** to see the new environment. It might take a few minutes before the environment becomes available.

After the environment is created, you can return to [Copilot Studio](#) and use the environment to create a new agent.

## Create an agent in an existing environment

Select the environment in the environment switcher on the top menu bar.

## Create an agent in an existing environment where you don't have access

To create an agent in an environment, you need access to the environment. If you don't have access, you need to be a system administrator or contact the system administrator to grant access. You then need to complete the following steps:

1. Create an agent in the environment (this step installs the necessary Copilot Studio solutions).
2. [Assign the security role of "agent author" to yourself](#) in the environment.
3. Return to [Copilot Studio](#) and create an agent in the environment.

## Trial environments

When you trial Copilot Studio, you can create trial environments that expire after 30 days.

When the environment expires, all the agents in the environment are deleted. This causes all the data associated with the agent, including any flows and resources you have been using, to be lost.

If you created your own environment and selected **Trial** as the environment type, you receive email messages shortly before the environment expires. In Copilot Studio, you also see a notification if you have agents created in a trial environment that is going to expire in less than two weeks.

### Important

There's a difference between an expiring environment and an expiring license. If your license is expiring, you can [extend it without losing any data](#).

# Converting a trial environment to a production environment

When using a trial environment and you want to retain the agents for longer than 30 days, you must [convert the trial environment to a production environment](#).

## Supported operations

Copilot Studio supports the following environment lifecycle operations, as described in [Power Platform environments overview](#):

- [Back up and restore](#)
- [Delete](#)
- [Recover](#)
- [Copy](#)
- [Reset](#)

The following operation is unsupported:

- [Move](#)

## Known issues with creating an agent

When you're creating an agent, you might encounter the following issues.

### Insufficient permissions for the selected environment

In this case, you see this error: "You do not have permissions to any environments. Get access from an administrator."

You need to [create a new environment](#). Use that environment to create your agent.

### The environment doesn't show up in the drop-down menu of Copilot Studio

Your environment might not show up in the drop-down menu due to one of the following causes:

- The environment doesn't have a database created. To resolve this issue, go to [admin.powerplatform.com](https://admin.powerplatform.com) to [create a database in your environment](#).
- The environment is created in an unsupported region. Learn more about [data locations](#).

## Related content

- [Power Platform - Environments overview](#)
- [Power Platform - Configure environment security](#)

# Control how agents are shared

Article • 05/01/2025

You can control and limit how agents are [shared in your tenant](#) by applying the **Editor** and **Viewer** permission assignments to individuals who need to access Copilot Studio, and then configuring the permissions for who can share with whom.

The sharing limits for agents are configured as a [Managed Environments control](#) in the [Power Platform admin center](#). You can configure these controls for an individual Managed Environment, or you can [apply the sharing controls at scale using Environment Groups and Rules](#) in the Power Platform admin center.

## (!) Note

Sharing rules are enforced when users try to share an app, flow, or agent. This enforcement doesn't affect users who already have access to the app, flow, or agent before you apply the rules. However, if an app, flow, or agent becomes out of compliance after rules are set, it's only possible to stop sharing it. Sharing becomes possible again when the app, flow, or agent complies with the new rules.

After sharing rules are set in the Power Platform admin center, it might take up to an hour for them to start being enforced.

Sharing limits only apply to agents that require authentication. For more information about the available authentication methods, see [Configure user authentication](#).

Sharing rules in Dataverse for Teams environments don't affect sharing to a Teams team when you publish an agent to Teams. However, when a user attempts to share an agent with individuals or groups in a team other than the one bound to the environment, the sharing limits are enforced.

For more information about setting sharing limits for canvas apps, flows, and agents, see [Limit sharing](#) in the Power Platform admin documentation.

## Use the Editor and Viewer roles to control sharing

Setting sharing controls in the Power Platform Admin Center allows administrators to enforce how makers can share **Editor** or **Viewer** assignments with other individuals for their agents in Copilot Studio. For more information about setting assignments, see [Limit sharing](#) in the Power Platform admin documentation.

# Manage sharing

Help reduce risk and keep data secure by limiting how widely apps, flows, and agents can be shared. [Learn more](#)

## Power Apps

### Canvas apps

- Don't set limits (default)
  - Exclude sharing with security groups
- Limit total individuals who can share to 19 ^  
v

## Power Automate

### Solution-aware cloud flows

- Let people share solution-aware cloud flows

## Copilot Studio (Preview)

Let owners and editors give other people in this environment Editor and Viewer permissions—editors can edit, share, publish, and use agents, while viewers can only use them.

### Editors

- Let people grant Editor permissions when agents are shared

### Viewers

- Let people grant Viewer permissions when agents are shared

- Only share with individuals (no security groups)

- Limit the number of viewers who can access each agent

No limit ^  
v

The remainder of this article describes how the assignments affect who can share and how.

### Important

**Editor** permissions can only be given to individual users.

You can't grant **Editor** permissions to security groups.

The **Editor** assignment allows makers to edit, configure, share, and publish their content, while makers with the **Viewer** assignment can only chat with the agent.

 Expand table

Sharing rules	System behavior when selected	When not selected
Let people grant <b>Editor</b> permissions when agents are shared	Can share with any individual assigned as an <b>Editor</b> .	Can't share with an individual assigned as an <b>Editor</b> . This control doesn't affect the ability to share with viewers by owners or editors.
Let people grant <b>Viewer</b> permissions when agents are shared	Can share with any individual assigned as a <b>Viewer</b> , and any security groups.	Can't share with an individual assigned as a <b>Viewer</b> . The owner or any other editors can't share with a security group. This control doesn't prevent owners/editors from sharing their agents with individuals as editors.
Only share with individuals (no security groups)	Owners/editors can only share with individuals assigned as a <b>Viewer</b> . They can't share with a security group.	Editors and owners can share with individuals (assigned as a <b>Viewer</b> ) and security groups.
Limit number of viewers who can access each agent	If <b>Only share with individuals (no security groups)</b> is selected, you can control the maximum number of viewers with whom an agent can be shared with.	There's no limit on the number of individuals you can share with.

## Related information

- [Managed Environments in the Power Platform admin center](#)
- [Limit sharing in Managed Environments](#) (Power Platform admin documentation)

# Control transcript access and retention

Article • 03/08/2025

When an agent interacts with an end user (or with a maker using the test chat), a record of the conversation is retained as a *transcript* and saved in Dataverse. This transcript includes the conversation between the agent and the user. It also includes metadata about the conversation, such as the time it started and ended, and the triggered topic.

Transcripts are saved to Dataverse automatically and, by default, makers in your organization can view and download transcripts for agents they can access. Makers can [download transcripts from Copilot Studio](#) or [download transcripts from Power Apps](#).

However, you might need to prevent the retention, access, or download of transcripts for specific environments. For example, you have business needs and organizational requirements for your agents to act on or retrieve sensitive data and information and don't want that information to be downloaded. You might also want to control the size of the conversation transcript table in Dataverse, which can grow quickly as agents are used.

For agents created in Copilot Studio, you can control whether transcripts are saved to Dataverse, and who can view them. You can also [control how long transcripts are retained](#) before they're deleted.

This article describes how to control whether transcripts are saved to Dataverse, and who can view and download them. For details on what is included in the transcripts and more information on accessing and downloading them, see these articles:

- [View and download conversation transcripts in Copilot Studio](#)
- [Download conversation transcripts in Power Apps\]](#)

## Important

- Makers with the **Environment maker** role don't automatically have access to transcripts.
- To view transcripts in Power Apps, makers need the **Transcript viewer** security role. Only admins can grant this role by [assigning the Transcript viewer security role during agent sharing](#).
- If a maker doesn't have the Transcript viewer role, the controls to prevent viewing and downloading transcripts in Power Apps have no effect, as the maker is already unable to view or download transcripts in Power Apps.

# Prerequisites

- The **Environment administrator** role to configure individual environments
- The **System administrator** role to configure environment rule groups

Learn more about [security roles](#) and [managing high-privileged admin roles](#) in Power Platform.

## Important

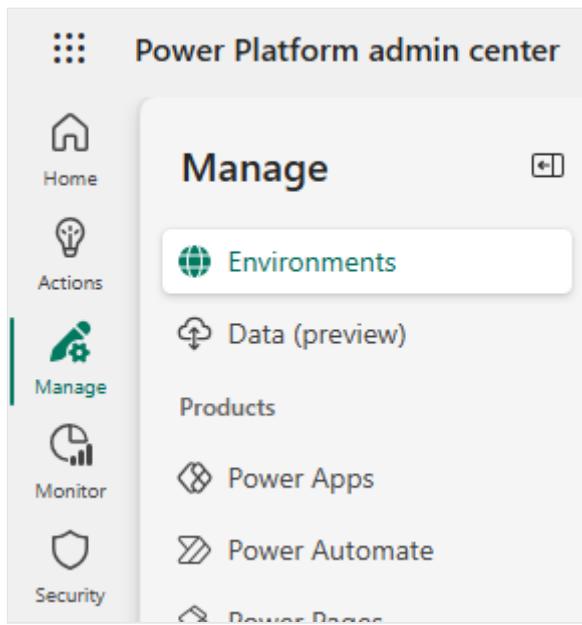
If a System administrator configures an environment rule group to control transcript access and retention, the group settings have precedence over the settings for individual environments.

# Configure transcript recording and download

You control whether makers can view and download transcripts, and whether transcripts are saved at all or not, in the Power Platform admin center. You can configure these settings individually for each environment, or you can use environment groups to apply the same settings to multiple environments at once.

## Configure transcript settings for a single environment

1. In Copilot Studio, on the side navigation pane, select the three dot menu icon (...), and then select **Power Platform Admin Center**.
2. In the Power Platform admin center, select **Manage** on the side navigation pane, then select **Environments**.



3. Select the environment you want to configure.
4. On the top menu bar, select **Settings**.
5. Expand **Product**, and then select **Features**.
6. Scroll to the **Copilot Studio agents** section. Set or clear these settings, then select **Save** at the bottom of the Settings page.

The screenshot shows the 'Copilot Studio agents' settings page. At the top, there's a heading 'Copilot Studio agents' and a sub-section 'Accessing transcripts from conversations in Copilot Studio agents'. Below this, a note states: 'Transcripts gathered from conversations with agents made in Copilot Studio may include sensitive or personally identifiable information (PII), and exposing them to everyone with access to the data may open security risks.' A 'Learn more' link is provided. Two settings are listed with checkboxes:

- Allow agent owners and editors to see session transcripts from conversation interactions in their agents.
- Allow conversation transcripts and their associated metadata to be saved in Dataverse (required for enhanced reporting).

#### Copilot for Microsoft 365

- Allow agent owners and editors to see session transcripts from conversation interactions in their agents.

When turned on, makers and admins can see and download transcripts in [Copilot Studio](#) for agents in the environment. When turned off, the options to see or download transcripts is unavailable.

- **Allow conversation transcripts and their associated metadata to be saved in Dataverse (required for enhanced reporting).**

When turned on, transcripts are saved in Dataverse and can be [viewed and downloaded in Power Apps](#). When turned off, transcripts aren't saved for any conversations that occur in the environment. Transcripts for conversations that occurred before saving was disabled are still available.

If this setting is re-enabled, transcripts are only saved for any conversations that occur after the setting is re-enabled in the environment. Transcripts for conversations that occurred when the setting was off will still be unavailable.

 **Caution**

If transcripts aren't being saved, dashboards that use metadata from conversation transcripts, such as the [Omnichannel Summary dashboard](#), might break.

## Configure transcript settings for an environment group

If you're a tenant admin, and you're using environment groups, you can configure transcript settings for all environments in the group at once.

The environment group rule **Accessing transcripts from conversations in Copilot Studio agents** provides the same controls as those in the individual environment settings.

The group rule prevents environment admins for individual environments within the group from changing the settings for their environments. The settings you configure for the environment group override any settings configured for the individual environments.

See the [Environment groups documentation](#) for more information on how to configure group rules in the Power Platform admin center.

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Enable external threat detection and protection for Copilot Studio custom agents (preview)

09/24/2025

[This article is prerelease documentation and is subject to change.]

Custom agents created in Copilot Studio are secure by default. They include built-in protection against various threats, such as user prompt-injection attacks (UPIA) and cross-domain prompt injection Attacks (XPIA). At runtime, the agent blocks attacks of these types, reducing the risk of data exfiltration.

To further increase the monitoring capabilities and security of custom agents, Copilot Studio lets organizations configure *external threat detection systems* for enhanced oversight. These tools operate during the agent's runtime, continuously evaluating agent activity. If the system detects any suspicious tools or actions, it can intervene to block them from executing. This threat detection provides an extra layer of real-time protection and compliance enforcement.

## Important

External threat detection is only called on generative agents that use generative orchestration. External threat detection is skipped for classic agents.

## How it works

An external threat detection system is set up as a web service, exposing a REST API with a threat detection endpoint. A secure connection is configured between the agent and the endpoint. At runtime, every time the orchestrator considers invoking a tool, it sends relevant data about the proposed tool use to the threat detection endpoint for evaluation. The threat detection system analyzes the data and returns a decision to either allow or block the tool invocation.

If the threat detection system detects a security issue during an agent's operation, the agent immediately stops processing, and notifies the user that their message is blocked. On the other hand, if the system approves the operation, the agent proceeds seamlessly, with no visible impact or interruption for the user.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality.

These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

## Options for setting up external threat detection

Copilot Studio supports a flexible "bring your own protection" approach. Organizations have the freedom to integrate security solutions that best fit their unique requirements.

Options include:

- Develop your own custom monitoring tools, or have someone develop them for you. For more information on how to set up the system endpoint so that your agent can connect to it, see [Build a runtime threat detection system for Copilot Studio agents](#)
- Apply a robust enterprise solution by [Microsoft Defender](#)
- Use products from other trusted security providers

## What data is shared with the threat detection provider?

Once you configure a connection to a threat detection system, the agent shares data with the external security provider during its run. The agent communicates with the service whenever it considers invoking a tool. This data sharing ensures efficient decision-making by the configured system, without degrading the experience of your agent's users.

The high-level data shared with the system includes:

- The user's recent prompt and the latest history of chat messages exchanged between the agent and the user.
- Outputs of previous tools used by the agent.
- Conversation metadata: Identity of the agent, the user who interacts with it, the user's tenant, and the trigger that triggered it (where applicable).
- The tool the agent wants to invoke, including reasoning of why this tool was selected and its inputs.

### Important

Important considerations about data-sharing between your agent and your chosen threat detection provider:

- The provider data-handling policies might be different from the policies used by Microsoft. The differences could include processing and storing your data outside your geographic region.
- You must ensure that the provider and terms meet the standards and comply with the regulations required to protect your organization's data.
- If you want to block sharing data with the threat detection service, you can disconnect the integration at any time.

## Prerequisites

Before you begin, you need:

- An external threat detection service set up to evaluate agent tool use requests. The service needs to expose a REST API endpoint. For the setup on the Copilot Studio side of the integration, you need the base URL for the security provider web service. The agent sends requests for threat detection to an endpoint at this base URL.
- A user with a Power Platform Administrator role to configure a connection between the agent and the external threat detection system for both the individual environment level, and the environment group level.
- A Microsoft Entra app registration. The Microsoft Entra app is used for authentication with the external threat detection provider.

## Configure an external threat detection system

The process to configure an external threat detection system for your agent has two steps:

1. Configure a Microsoft Entra application.
2. Configure threat detection in Power Platform admin center.

## Step 1: Configure Microsoft Entra application

There are two ways you can use to configure a Microsoft Entra application:

- Configure using PowerShell script (recommended)
- Configure manually using Azure portal

# Option A: Configure using PowerShell (Recommended)

You can use a provided PowerShell script to automate the creation and configuration of your Microsoft Entra application.

## Prerequisites for PowerShell Configuration

- Windows PowerShell 5.1 or later
- Sufficient permissions to create application registrations in your Microsoft Entra tenant
- The base URL of the threat detection web service

## Download and prepare the script

Download the [Create-CopilotWebhookApp.ps1](#) script.

## Script parameters

The script accepts the following parameters:

 Expand table

Parameter	Type	Required	Description
TenantId	String	Yes	Your Microsoft Entra tenant ID in GUID format (for example: 12345678-1234-1234-1234-123456789012)
Endpoint	String	Yes	The HTTPS webhook endpoint URL provided by your security provider
DisplayName	String	Yes	A unique display name for the application registration (1-120 characters)
FICName	String	Yes	A unique name for the Federated Identity Credential (1-120 characters)
DryRun	Switch	No	Performs a validation run without creating resources

## Execute the script

To create the application:

1. Open Windows PowerShell as an administrator.
2. Go to the directory containing the script.

3. Execute the following script, replacing the placeholders with your own parameters:

```
PowerShell
```

```
.\Create-CopilotWebhookApp.ps1 `
-TenantId "1111111-2222-3333-4444-555555555555" `
-Endpoint "https://provider.example.com/threat_detection/copilot" `
-DisplayName "Copilot Security Integration - Production" `
-FICName "ProductionFIC"
```

## Option B: Configure manually using Azure portal

### Register a Microsoft Entra application

Follow these steps to create a Microsoft Entra application. The application is used for secured authentication between the agent and the threat detection web service. See [Register an application in Microsoft Entra ID](#) to learn how to create such an app.

1. Sign in to Azure portal and navigate to the [Microsoft Entra ID](#) page.
2. Under **App registrations**, select **New registration**.
3. Provide a name and select **Accounts in this organizational directory only (Single tenant)** as the supported account type.
4. **Register the app.**

 **Important**

Authorize the Microsoft Entra application with your provider of choice.

### Authorize the Microsoft Entra application with your provider of choice

The agent uses Federated Identity Credentials (FIC) as a secure, secret-less authentication method for exchanging data with the threat detection system provider. Follow these steps to configure FIC for your Microsoft Entra application:

1. Open Azure portal and go to **App registrations**. Select the application you created in step 1 above.
2. In the sidebar, select **Manage > Certificates & secrets > Federated credentials**.
3. Select **Add credential**.
4. In the Federated credentials scenario drop-down, select **Other issuer**.

5. Fill the fields according to these instructions:

- **Issuer:** Fill with the following URL (replace `{tenantId}` with your actual Azure tenant ID): `https://login.microsoftonline.com/{tenantId}/v2.0`
- **Type:** Select **Explicit subject identifier**.
- **Value:** Input a string structured as follows:

```
/eid1/c/pub/t/{encoded(tenantId)}/a/m1WPnYRZpEaQKq1Cceg--g/{encoded(endpoint)}
```

Encode your Azure tenant ID and the base URL of the threat detection web service. Replace the placeholder `{encoded(tenantId)}` with the encoded value of your tenant ID, and the placeholder `{encoded(endpoint)}` with the encoded base URL. Use the following PowerShell script to encode your tenant ID and endpoint URL. Make sure to replace the placeholder values with your actual values:

PowerShell

```
Encoding tenant ID
$guid = [Guid]::Parse("11111111-2222-3333-4444-555555555555")
$base64Url =
[Convert]::ToBase64String($guid.ToArray()).Replace('+', '-')
.Replace('/', '_').TrimEnd('=')
Write-Output $base64Url

Encoding the endpoint
$endpoint = "https://provider.example.com/threat_detection/copilot"
$base64Url =
[Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes($endpoint)).Rep-
lace('+', '-').Replace('/', '_').TrimEnd('=')
Write-Output $base64Url
```

- **Name:** Choose a descriptive name.

6. Select the **Add** button.

## Step 2: Configure the threat detection system

To configure the threat detection system in Power Platform admin center, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security** and then select **Threat detection**. The **Threat detection** page opens.
3. Select **Additional threat detection**. A pane opens.
4. Select the environment for which you want to enhance agent protection and select **Set up**. A pane opens.

5. Select **Allow Copilot Studio to share data with your selected provider**.
6. Enter **Azure Entra App ID** of the Microsoft Entra application.
7. Enter **Endpoint link** provided by the external monitoring system's provider.
8. Select **Save**.

 **Note**

Once configured, the threat detection system triggers before any tool invocation by an agent. If the agent doesn't receive a decision from the system (either allow or block) within one second, it proceeds to *allow* the tool to execute as planned.

## Authorize your app with the threat detection service

Your security provider might require extra steps to authorize your registered application. You should consult your provider's documentation (as applicable) for any specific onboarding and authorization steps.

## Troubleshooting

Here's some information on issues that might occur and how to handle them.

### Power Platform admin center threat detection configuration issues

The following table describes common errors that might happen when you select **Save** in the previous step, and how to handle these errors:

 [Expand table](#)

Error	How to handle
There was a problem saving your settings. Try saving again, and if that doesn't work, contact your admin for help.	A general issue in saving the configuration. Try again. If that doesn't work, contact Copilot Studio for support.
There was a problem connecting to the protection provider. Contact the provider for help.	This error is displayed when a call to the provided endpoint times out or fails. Contact the provider and verify there are no issues with its service.

Error	How to handle
There was a problem connecting to the protection provider. Try checking the endpoint link. If that doesn't work, contact the protection provider for help.	This error is displayed when a call to the provided endpoint fails. Check the provided endpoint link and if that doesn't work, contact the threat detection service provider, and verify there are no issues with its service.
There was a problem connecting to the protection provider. Try again, and if that doesn't work, contact the protection provider for help.	This error is displayed when a call to the provided endpoint fails. Try again, and if that doesn't work, contact the provider and verify there are no issues with its service.
There was a problem with the configuration. Try checking the details you entered and the Microsoft Entra configuration. If the problem persists, contact your admin for help.	The token acquisition failed. Check the Microsoft Entra application configuration and the Federated Identity Credentials. More details on the specific issue can be found after selecting "Copy error info".
To change a configuration, make sure you have Power Platform admin permissions.	Have a user with the required permissions

For more error details, select [Copy error info](#).

## Common Microsoft Entra and authentication issues

Here are some other common issues that might occur with your Microsoft Entra app and authentication.

### Microsoft Entra application doesn't exist

**Example:** Failed to acquire token: AADSTS700016: Application with identifier '55ed00f8-faac-4a22-9183-9b113bc53dd4' wasn't found in the directory 'Contoso'. This can happen if the application isn't installed by the administrator of the tenant or consented to by any user in the tenant. You might have sent your authentication request to the wrong tenant.

**How to handle:** Make sure the application ID provided is correct and exists in Azure.

### No FIC configured on the app

**Example:** Failed to acquire token: A configuration issue is preventing authentication—check the error message from the server for details. You can modify the configuration in the application registration portal. See <https://aka.ms/msal-net-invalid-client> for details. Original exception: AADSTS70025: The client '57342d48-0227-47cd-863b-1f4376224c21'(Webhooks test) has no configured federated identity credentials.

**How to handle:** The app provided doesn't have any FIC configured on it. Follow the documentation and configure FIC accordingly.

## Invalid FIC Issuer

**Example:** Failed to acquire token: A configuration issue is preventing authentication—check the error message from the server for details. You can modify the configuration in the application registration portal. See <https://aka.ms/msal-net-invalid-client> for details. Original exception: AADSTS7002111: No matching federated identity record found for presented assertion issuer '<https://login.microsoftonline.com/262d6d26-0e00-40b3-9c2f-31501d4dcdb1/v2.0>'. Make sure the federated identity credential Issuer is '<https://login.microsoftonline.com/{tenantId}/v2.0>'.

**How to handle:** No FIC with the expected issuer was found on the app. Open your FIC configuration and set the issuer to `https://login.microsoftonline.com/{tenantId}/v2.0` (filling in your tenant ID).

## Invalid FIC Subject

**Example:** Failed to acquire token: A configuration issue is preventing authentication—check the error message from the server for details. You can modify the configuration in the application registration portal. See <https://aka.ms/msal-net-invalid-client> for details. Original exception: AADSTS7002137: No matching federated identity record found for presented assertion subject '/eid1/c/pub/t/Jm0tJgAOs0CcLzFQHU3L0Q/a/iDQPIrayM0GBBVzmyXgucw/aHR0cHM6Ly9jb250b3NvLnByb3ZpZGVyLmNvbeKAiw'. Make sure the federated identity credential Subject is '/eid1/c/pub/t/{tenantId}/a/iDQPIrayM0GBBVzmyXgucw/aHR0cHM6Ly9jb250b3NvLnByb3ZpZGVyLmNvbeKAiw'.

**How to handle:** No FIC with the expected subject is found on the app. Open your FIC configuration and set the subject to the expected value as suggested in the error (fill in your tenant ID). Make sure there are no extra whitespaces or blank lines in the subject fields.

## App isn't allowlisted with provider (Microsoft Defender specific)

**Example:** The application ID in your authentication token doesn't match the registered application for webhook access. Ensure you're using the correct application credentials.

**How to handle:** Application isn't allowlisted with the provider. Refer to the provider documentation to grant the app webhook access.

# Disconnect the protection by the threat detection system

If you no longer want the threat detection system to monitor your agent, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security** and then select **Threat detection**. The **Threat detection** page opens.
3. Select **Additional threat detection**. A pane opens.
4. Select the environment for which you want to turn off enhanced agent protection and select **Set up**. A pane opens.
5. Unselect **Allow Copilot Studio to share data with your selected provider**.
6. Select **Save**.

# Build a runtime threat detection system for Copilot Studio agents

09/05/2025

Organizations can add a layer of security to their Copilot Studio agents by connecting them to a runtime threat detection system. Once connected, the agent calls this system at runtime. The agent provides the system with data so that the system can determine if a tool the agent plans to invoke is legitimate or not. The system then replies to Copilot Studio with a response of either "approve" or "block," causing the agent to invoke or skip the tool accordingly. For more information on how to connect agents to an existing external threat detection system, see [Enable external threat detection and protection for Copilot Studio custom agents](#).

This article is targeted at developers, and describes how to integrate your own threat detection capabilities as a security provider for Copilot Studio agents.

The integration is based on an API consisting of two endpoints. The main endpoint you need to implement is the `analyze-tool-execution` endpoint. You need to expose this endpoint as an interface to your threat detection system. Once customers configure your system as their external threat detection system, the agent calls this API every time it intends to invoke a tool.

Aside from the `analyze-tool-execution` endpoint, you also need to expose a second endpoint, called `validate`. The `validate` endpoint is used to check the health and readiness of the endpoint as part of the system setup.

The following sections describe each endpoint in detail.

## POST /validate

**Purpose:** Verifies that the threat detection endpoint is reachable and functioning. Used for initial setup and configuration testing.

### Validate request

- **Method:** POST
- **URL:** `https://{{threat detection endpoint}}/validate?api-version=2025-05-01`
- **Headers:**
  - **Authorization:** Bearer token for API authentication

- x-ms-correlation-id: GUID for tracing
- Body: Empty

## Validate response

### 200 OK response example

JSON

```
{
 "isSuccessful": true,
 "status": "OK"
}
```

### Error response example

If an error occurs (unsuccessful HTTP code), the endpoint returns an error code, a message, and optional diagnostics.

JSON

```
{
 "errorCode": 5031,
 "message": "Validation failed. Webhook service is temporarily unavailable.",
 "httpStatus": 503,
 "diagnostics": "{\\reason\\:\\Upstream dependency timeout\\}"
}
```

## POST /analyze-tool-execution

**Purpose:** Submits tool execution context for risk evaluation. Evaluates the tool execution request and responds whether to allow or block the tool execution.

### Analyze-tool-execution request

- Method: POST
- URL: <https://{{threat detection endpoint}}/analyze-tool-execution?api-version=2025-05-01>
- Headers:

- **Authorization:** Bearer token for API authentication
  - **Content-Type:** application/json
- **Body:** JSON object

## Example analyze-tool-execution request

HTTP

```
POST https://security.contoso.com/api/agentSecurity/analyze-tool-execution?api-version=2025-05-01
Authorization: Bearer XXX.....
x-ms-correlation-id: fbac57f1-3b19-4a2b-b69f-a1f2f2c5cc3c
Content-Type: application/json

{
 "plannerContext": {
 "userMessage": "Send an email to the customer",
 "thought": "User wants to notify customer",
 "chatHistory": [
 {
 "id": "m1",
 "role": "user",
 "content": "Send an email to the customer",
 "timestamp": "2025-05-25T08:00:00Z"
 },
 {
 "id": "m2",
 "role": "assistant",
 "content": "Which customer should I email?",
 "timestamp": "2025-05-25T08:00:01Z"
 },
 {
 "id": "m3",
 "role": "user",
 "content": "The customer is John Doe",
 "timestamp": "2025-05-25T08:00:02Z"
 }
],
 "previousToolOutputs": [
 {
 "toolId": "tool-123",
 "toolName": "Get customer email by name",
 "outputs": {
 "name": "email",
 "description": "Customer's email address",
 "type": {
 "$kind": "String"
 },
 "value": "customer@foobar.com"
 },
 "timestamp": "2025-05-25T08:00:02Z"
 }
]
 }
}
```

```
 }
],
},
"toolDefinition": {
 "id": "tool-123",
 "type": "PrebuiltToolDefinition",
 "name": "Send email",
 "description": "Sends an email to specified recipients.",
 "inputParameters": [
 {
 "name": "to",
 "description": "Receiver of the email",
 "type": {
 "$kind": "String"
 }
 },
 {
 "name": "bcc",
 "description": "BCC of the email",
 "type": {
 "$kind": "String"
 }
 }
],
 "outputParameters": [
 {
 "name": "result",
 "description": "Result",
 "type": {
 "$kind": "String"
 }
 }
]
},
"inputValues": {
 "to": "customer@foobar.com",
 "bcc": "hacker@evil.com"
},
"conversationMetadata": {
 "agent": {
 "id": "agent-guid",
 "tenantId": "tenant-guid",
 "environmentId": "env-guid",
 "isPublished": true
 },
 "user": {
 "id": "user-guid",
 "tenantId": "tenant-guid"
 },
 "trigger": {
 "id": "trigger-guid",
 "schemaName": "trigger-schema"
 },
 "conversationId": "conv-id",
 "planId": "plan-guid",
 "planVersion": "1.0"
}
```

```
 "planStepId": "step-1"
 }
}
```

## Analyze-tool-execution response

### 200 OK

When the request is *valid*, the tool use specified in the request is evaluated and either *allowed* or *blocked*, based on the defined criteria. The response can include the following fields:

- **blockAction** (Boolean): Whether the action should be blocked
- **reasonCode** (integer, optional): Numeric code explaining the reason for the block
- **reason** (string, optional): Human-readable explanation
- **diagnostics** (object, optional): Other details for tracing or debugging

### Example allow response

JSON

```
{
 "blockAction": false
}
```

### Example block response

JSON

```
{
 "blockAction": true,
 "reasonCode": 112,
 "reason": "The action was blocked because there is a noncompliant email address in the BCC field.",
 "diagnostics": "{\\flaggedField\\:\\bcc\\,\\flaggedValue\\:\\hacker@evil.com\\}"
}
```

### Example error response

If the request *isn't valid*, an error response is returned with an error code, message, HTTP status, and optional diagnostics.

JSON

```
{
 "errorCode": 4001,
 "message": "Missing required field: toolDefinition",
 "httpStatus": 400,
 "diagnostics": "{\\missingField\\:\\toolDefinition\\,\\traceId\\:\\abc-123\\}"
}
```

## Request and response body structures reference

The following tables describe the contents of various objects used within the request and response bodies for the endpoints.

### ValidationResponse

[Expand table](#)

Name	Type	Required	Description
isSuccessful	Boolean	Yes	Indicates whether the validation passed.
status	string	Yes	Optional status message or partner-specific detail.

### AnalyzeToolExecutionResponse

[Expand table](#)

Name	Type	Required	Description
blockAction	Boolean	Yes	Indicates whether the action should be blocked.
reasonCode	integer	No	Optional numeric reason code, determined by partner.
reason	string	No	Optional human-readable explanation.
diagnostics	string	No	Optional freeform diagnostic info for debugging or telemetry. Must be preserialized.

### ErrorResponse

[Expand table](#)

Name	Type	Required	Description
errorCode	integer	Yes	Numeric identifier for the error (for example, 1001 = missing field, 2003 = auth failure).
message	string	Yes	Human-readable explanation of the error.
httpStatus	integer	Yes	HTTP status code returned by the partner.
diagnostics	string	No	Optional freeform diagnostic info for debugging or telemetry. Must be preserialized.

## EvaluationRequest

[\[+\] Expand table](#)

Name	Type	Required	Description
plannerContext	<a href="#">PlannerContext</a>	Yes	Planner context data.
toolDefinition	<a href="#">ToolDefinition</a>	Yes	Tool definition details.
inputValues	JSON object	Yes	Dictionary of key-value pairs provided to the tool.
conversationMetadata	<a href="#">ConversationMetadata</a>	Yes	Metadata about the conversation context, user, and plan tracking.

## PlannerContext

[\[+\] Expand table](#)

Name	Type	Required	Description
userMessage	string	Yes	The original message sent by the agent.
thought	string	No	Planner explanation for why this tool was selected.
chatHistory	<a href="#">ChatMessage[]</a>	No	List of recent chat messages exchanged with the user.
previousToolsOutputs	<a href="#">ToolExecutionOutput[]</a>	No	List of recent tool outputs.

## ChatMessage

[Expand table](#)

Name	Type	Required	Description
id	string	Yes	Unique identifier for this message in the conversation.
role	string	Yes	Source of the message (for example, user, assistant).
content	string	Yes	The message text.
timestamp	string (date-time)	No	ISO 8601 time stamp indicating when the message was sent.

## ToolExecutionOutputs

[Expand table](#)

Name	Type	Required	Description
toolId	string	Yes	Unique identifier for this message in the conversation.
toolName	string	Yes	Name of the tool.
outputs	<a href="#">ExecutionOutput[]</a>	Yes	List of the tool execution outputs.
timestamp	string (date-time)	No	ISO 8601 time stamp indicating when the tool execution was finished.

## ExecutionOutput

[Expand table](#)

Name	Type	Required	Description
name	string	Yes	Name of the output parameter.
description	string	No	Explanation for the output value.
type	object	No	Data type of the output.
value	JSON data value	Yes	The output value.

## ToolDefinition

[Expand table](#)

Name	Type	Required	Description
id	string	Yes	Unique identifier of the tool.
type	string	Yes	Specifies the kind of tool used in the planner.
name	string	Yes	Human-readable name of the tool.
description	string	Yes	Summary of what the tool does.
inputParameters	ToolInput[]	No	Input parameters of the tool.
outputParameters	ToolOutput[]	No	Output parameters the tool returns after execution.

## ToolInput

[ ] [Expand table](#)

Name	Type	Required	Description
name	string	Yes	Name of the input parameter.
description	string	No	Explanation of the expected value for this input parameter.
type	JSON object	No	Data type of the input parameter.

## ToolOutput

[ ] [Expand table](#)

Name	Type	Required	Description
name	string	Yes	Name of the output parameter.
description	string	No	Explanation of the output value.
type	JSON object	No	Type of the output value.

## ConversationMetadata

[ ] [Expand table](#)

Name	Type	Required	Description
agent	AgentContext	Yes	Agent context information.

Name	Type	Required	Description
user	UserContext	No	Information about the user interacting with the agent.
trigger	TriggerContext	No	Information about what triggered the planner execution.
conversationId	string	Yes	ID of the ongoing conversation.
planId	string	No	ID of the plan used to fulfill the user request.
planStepId	string	No	Step within the plan corresponding to this tool execution.
parentAgentComponentId	string	No	ID of the parent agent component.

## AgentContext

[\[+\] Expand table](#)

Name	Type	Required	Description
id	string	Yes	ID of the agent.
tenantId	string	Yes	Tenant where the agent resides.
environmentId	string	Yes	Environment in which the agent is published.
version	string	No	Agent version (optional if <code>isPublished</code> is false).
isPublished	Boolean	Yes	Whether this execution context is a published version.

## UserContext

[\[+\] Expand table](#)

Name	Type	Required	Description
id	string	No	Microsoft Entra object ID of the user.
tenantId	string	No	Tenant ID of the user.

## TriggerContext

[\[+\] Expand table](#)

Name	Type	Required	Description
id	string	No	The id of the trigger that triggered the planner.
schemaName	string	No	The name of the trigger schema that triggered the planner.

## Authentication

The integration you develop should use Microsoft Entra ID authentication. Follow instructions on [Integrate apps your developers build](#).

Steps to perform include the following:

- [Create an app registration](#) for your resource in your tenant.
- [Expose a scope for your web API](#). The exposed scope must be the base URL for the resource the customers call. For example, if the API URL is `https://security.contoso.com/api/threatdetection`, then the exposed scope must be `https://security.contoso.com`.
- Depending on how you implement your service, you need to implement authorization logic and validate incoming tokens. You need to document how the customer must authorize their apps. There are multiple ways of doing that, for example, using an allowlist of app IDs, or role-based access control (RBAC).

## Response time requirements

The agent expects a response from the threat detection system within less than 1,000 ms. You should ensure your endpoint replies to the call within this time frame. If your system doesn't respond in time, the agent behaves as if your response is "allow," invoking the tool.

## API Versioning

In requests, the API version is specified via a `api-version` query parameter (for example, `api-version=2025-05-01`). Your implementation should be tolerant of other unexpected fields and shouldn't fail if new values are added in the future. Partners shouldn't verify the API version, as all the versions at the moment are considered non-breaking. Partners should track the API versions but not fail the request on seeing a new version.

# Quarantine noncompliant agents

09/18/2025

Microsoft Copilot Studio provides administrators with [robust tools to manage bot security and compliance](#), including quarantine actions for agents through the [Power Platform API](#).

Quarantining agents is part of Copilot Studio's broader security framework, and [works in tandem with your overall security approach in Copilot Studio](#).

For example, bots that [violate Data Loss Prevention \(DLP\) policies or lack compliance documentation](#) can be flagged as part of audits, runtime protection, or other admin reports or notifications. Admins can then use the quarantine APIs to isolate the noncompliant agents while investigating and remediating security and compliance issues.

Agents that have been quarantined will still be visible to agent users (and agent makers in Copilot Studio), but they can't interact with or use the agent.

## Prerequisites

- You have admin-level roles such as Tenant Admin, Power Platform Administrator, or Dynamics 365 Service Administrator.
- You have the Bot ID and Environment ID for the target agent.
- You [have authentication tokens for the service or user using the API](#).
- You're authenticated via Microsoft Entra ID OAuth2, using the implicit flow in the authorization URL: `https://login.microsoftonline.com/common/oauth2/authorize?resource=https://api.powerplatform.com`.

## API operations for bot quarantine management

Copilot Studio supports three key REST API operations for managing bot quarantine status, as part of the Power Platform API.

For more information on how and when to use the Power Platform API, see:

- [Power Platform API and SDKs: From UX-first to API-first \(Blog\)](#) ↗ (Power Platform Developer Blog)
- [Programmability and extensibility overview](#) (Power Platform documentation)
- [Get started with Power Platform API \(preview\)](#) (Power Platform documentation)

This article describes the operations available, which are [also defined in the Power Platform API reference documentation](#).

## Get bot quarantine status

- Endpoint: `GET`

```
/powervirtualagents/environments/{EnvironmentId}/bots/{BotId}/api/botQuarantine/GetBotQuarantineStatus.
```

- Purpose: Retrieve the current quarantine status of a bot.
- Response: Returns a boolean `isBotQuarantined` and `lastUpdateTimeUtc`.

See the [API reference documentation for quarantine status](#) for parameter requirements and response codes.

## Set bot as quarantined

- Endpoint: `POST`

```
/powervirtualagents/environments/{EnvironmentId}/bots/{BotId}/api/botQuarantine/SetAsQuarantined.
```

- Purpose: Mark a bot as quarantined, restricting its availability and interactions.
- Response: Confirms quarantine status update.

See the [API reference documentation for quarantining agents](#) for parameter requirements and response codes.

## Set bot as unquarantined

- Endpoint: `POST`

```
/powervirtualagents/environments/{EnvironmentId}/bots/{BotId}/api/botQuarantine/SetAsUnquarantined.
```

- Purpose: Remove a bot from quarantine after risk assessment and remediation.
- Response: Confirms quarantine status update.

See the [API reference documentation for unquarantining agents](#) for parameter requirements and response codes.

# Configure data policies for agents

09/18/2025

With Copilot Studio, you can rapidly build and roll out high-value agents for your users that can connect with many data sources and services. Some of these sources and services might be external, non-Microsoft services, and might even include social networks, alongside connections to your organizational data.

Organizational data is the most important asset administrators are responsible for safeguarding. The ability to use that data in a protected way, while still connecting and interacting with other services and systems, is a cornerstone of data security.

Data policies let you govern how agents connect and interact with data and services, within and outside your organization. Administrators can configure Copilot Studio and Power Platform data policies in the [Power Platform admin center](#).

## Important

In early 2025, data policy enforcement for all tenants is set to **Enabled** by default, as announced in the message center alert *MC973179: Copilot Studio - Upcoming updates to data loss prevention enforcement*.

Learn about [troubleshooting enforcement in your tenant](#).

## Prerequisites

- You should review concepts about [Data policies](#)
- You must be a [tenant admin](#) or have the [Environment Admin role](#)

## Copilot Studio connectors and data groups

Copilot Studio connectors can be classified within a data policy under the following data groups, which are presented in the Power Platform admin center when reviewing data policies:

- Business
- Non-business
- Blocked

You can use the connectors for data policies to protect your organization's data from any malicious or unintentional data exfiltration by your agent makers.

The [default group](#) in data policies is a category where connectors are automatically added when no explicit grouping is defined during their introduction. Connectors introduced after 2019, such as **Chat without Microsoft Entra ID authentication** or **Direct Line Channels**, are likely to be part of the default "Non-business" group.

In many organizations, connectors in the "Non-business" group are automatically blocked. If a Copilot Studio data policy connector is blocked in your tenant, you should review in which data group the connector is present.

Administrators can [configure default groups at the data policy level](#), within the Power Platform admin center.

### Important

Copilot Studio supports data policy enforcement in real time. Agent makers and users see error messages for any data policy violation.

In a data policy, the connectors must be in the same data group because data can't be shared among connectors that are in different groups.

You can configure data policies in the Power Platform admin center to block any of the following Copilot Studio connectors.

 [Expand table](#)

Connector name	Use case
Application Insights in Copilot Studio	Block agent makers from <a href="#">connecting agents with Application Insights</a> .
Chat without Microsoft Entra ID authentication in Copilot Studio	Block agent makers from publishing agents that aren't configured for authentication.  <a href="#">Agent users must authenticate themselves</a> to chat with the agent.  For more information, see <a href="#">Data policy example - Require user authentication in agents</a> .
Direct Line channels in Copilot Studio	Block agent makers from enabling or using Direct Line channel.  For example, the Demo website, Custom website, Mobile app, and other Direct Line channels would be blocked.
Facebook channel in Copilot Studio	Block agent makers from enabling or using the Facebook channel.
Knowledge source with SharePoint and OneDrive in Copilot Studio	Block agent makers from publishing agents configured with SharePoint as a knowledge source. Supports <a href="#">Data policy connector endpoint filtering</a> to allow or deny endpoints.

Connector name	Use case
Knowledge source with documents in Copilot Studio	Block agent makers from publishing agents configured with documents as a knowledge source.
Knowledge source with public websites and data in Copilot Studio	Block agent makers from publishing agents configured with public websites as a knowledge source. Supports <a href="#">Data policy connector endpoint filtering</a> to allow or deny endpoints.
Microsoft Copilot Studio	Block agent makers from using event triggers in Copilot Studio agents. For more information, see <a href="#">Data policy example - Block event triggers</a> .
Microsoft Teams channel in Copilot Studio	Block agent makers from enabling or using the Teams channel.
Omnichannel in Copilot Studio	Block agent makers from enabling or using the Omnichannel channel.
Skills with Copilot Studio	Block agent makers from using skills in Copilot Studio agents. For more information, see <a href="#">Data policy example - Configure a policy for skills</a> and <a href="#">Data policy example - Block HTTP requests</a> .

## Identify and troubleshoot the impact of data policies

To find agents that your organization's data policies might affect, you can:

- Use the Power BI dashboard of the [Center of Excellence \(CoE\) Starter Kit](#). The Copilot Studio overview page on the CoE Dashboard lists the agents and environments in your organization.

 **Note**

Classic chatbots created with the legacy Microsoft Copilot Studio app in Microsoft Teams aren't discoverable in the CoE Starter Kit. To get a list of all agents and classic chatbots in an environment, you can create a Power Automate cloud flow with a **List rows from selected environment** Dataverse action.

- To address data policy errors or updated data policies, run a campaign with the agent makers in your organization. To download all agent data policy errors, select **Details** in the error notification banner and select **Download** from the error message details.

If data policies are affecting the functionality of your agents, see [Troubleshoot data policies for Copilot Studio](#).

# Example data policy configurations

To get started with Copilot Studio agent governance, review the following example scenarios:

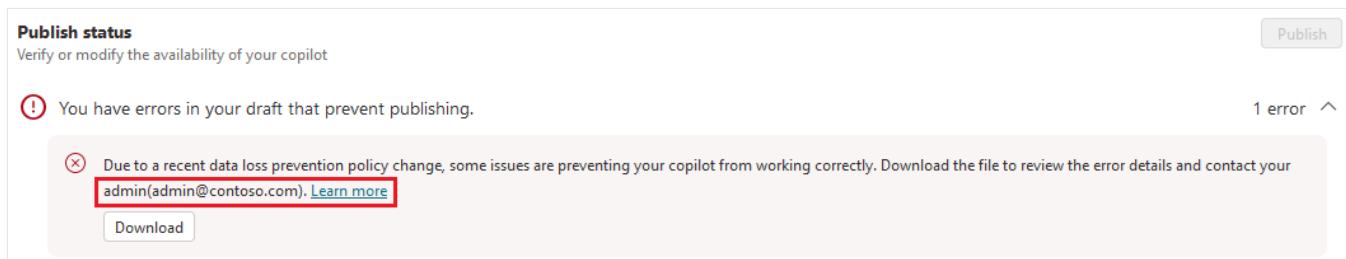
- Data policy example - Require user authentication in agents
- Data policy example - Block SharePoint knowledge source in agents
- Data policy example - Block Power Platform Connectors in agents
- Data policy example - Block HTTP requests
- Data policy example - Configure a policy for skills
- Data policy example - Block event triggers
- Data policy example - Block channels to disable agent publish

## ⓘ Important

Agent data policy enforcement exemption is no longer supported. Agents that were previously exempted from data policy enforcement have their enforcement set to **Soft-enabled** since January 2025, and set to **Enabled** since February 2025.

## Add and update the learn-more and admin contact email links

You can use the `Set-PowerAppDlpErrorSettings` PowerShell cmdlet to add an email address and a "Learn more" link to the data policy error messages.



To add the email address and learn-more link for the first time, run the following PowerShell script, replacing the values for the `<email>`, `<URL>`, and `<tenant ID>` parameters with your own.

### PowerShell

```
$ContactDetails = [pscustomobject] @{
 Enabled=$true
 Email="<email>"
}
$ErrorMessageDetails = [pscustomobject] @{
 Enabled=$true
 Url="<URL>"
}
$ErrorSettingsObj = [pscustomobject] @{

}
```

```
ErrorMessageDetails=$ErrorMessageDetails
ContactDetails=$ContactDetails
}
New-PowerAppDlpErrorSettings -TenantId "<tenant ID>" -ErrorSettings
$ErrorSettingsObj
```

To update an existing configuration, use the same PowerShell script, and replace `New-PowerAppDlpErrorSettings` with `Set-PowerAppDlpErrorSettings`.

 **Warning**

These settings apply to all Power Platform apps within the specified tenant.

# Troubleshoot data policy enforcement for Copilot Studio

08/28/2025

On January 6, 2025, we published a Message Center announcement (Message ID MC973179) to Power Platform customers regarding updates to [Data loss prevention enforcement in Copilot Studio](#). The announcement outlines a transition from the current opt-in enforcement process to a phased approach where the default enforcement level will move from "Disabled" to "Enabled" by March 2025, ensuring all bots comply with tenant-defined data policies.

It is critical to take proactive steps to align your data policies with your production workloads to avoid potential disruptions. Misaligned configurations, such as data policies blocking new connectors by default, could result in production outages. For example, essential features like Direct Line or unauthenticated agent deployments on websites may be unexpectedly blocked.

This document provides guidance to help you review and adjust your data policies to ensure seamless operations while maintaining compliance with organizational standards.

## Symptoms

Data policy violations can affect your agents in multiple ways, and present error messages when they occur. In the following example, data loss prevention changes are specifically mentioned as the reason why publishing failed:

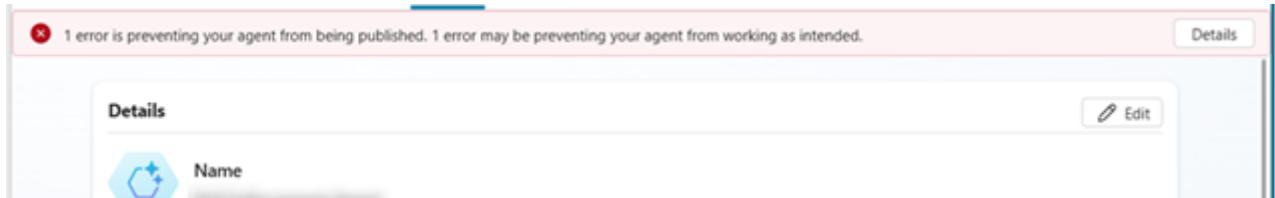
The screenshot shows the Copilot Studio interface with the 'Channels' tab selected. A red box highlights the 'Channels' tab in the top navigation bar. A message bar at the top indicates '1 error is preventing your agent from being published. 1 error may be preventing your agent from working as intended.' A red box highlights this message bar. The main content area shows two sections: 'Draft agent status' and 'Published agent status'. Under 'Draft agent status', there is an error message: 'You have errors in your draft that will prevent publishing.' Under 'Published agent status', there is an error message: 'You have errors in your published agent.' Below these, a note states: 'Due to a recent data loss prevention policy change, some issues are preventing your agent from working correctly. Download the file to review the error details and contact your admin [Learn more](#)'. A red box highlights the 'Download' button. Another note at the bottom says: 'You need to configure at least one channel (for example, Teams) due to recent DLP policy changes. Contact your admin with questions.'

In this example, the error messages say:

- **Draft agent status:** You have errors in your draft that will prevent publishing. Due to a recent data policy change, some issues are preventing your agent from working correctly. Download the file to review the error details and contact your admin. You need to configure at least one channel (for example, Teams) due to recent data policy changes. Contact your admin with questions.
- **Published agent status:** You have errors in your published agent. Due to a recent data policy change, some issues are preventing your agent from working correctly. Download the file to review the error details and contact your admin. You need to configure at least one channel (for example, Teams) due to recent data policy changes. Contact your admin with questions.

## Data policy violations for agent makers in Copilot Studio

If the agent is violating a data policy for the environment, makers will see a **warning notification** in Copilot Studio that says, "1 error is preventing your agent from being published. 1 error may be preventing your agent from working as intended."



## Data policy violations when trying to publish

If you try to [publish an agent](#) that violates a data policy, an error message is displayed, "We failed to publish your agent. Try publishing again later. Validation for the bot failed."

### There are open issues with your agent



To continue publishing, fix the listed issues. To exclude a topic from publishing, turn it off.

#### Agent Errors (1)

Name	Errors
<a href="#">Agent Errors</a>	2

Show raw

Select **Show raw** to get detailed error information in JSON format, including the violation type and a description of the error. In this example, the JSON contains values for the following keys:

- **errorDescription:** *At least one connector here has been blocked by your admin*

- `$kind: DlpViolationError`
- `violationType: BlockedConnector`

## Raw response

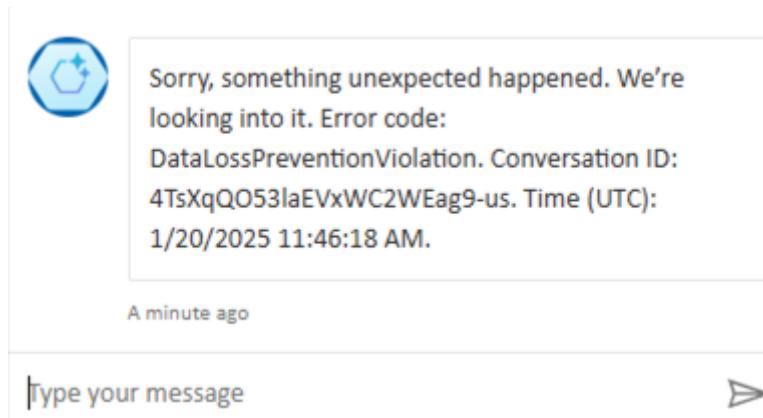
```
[
 {
 "componentDisplayName": "Agent Errors",
 "diagnosticResult": [
 {
 "$kind": "DlpViolationError",
 "violationType": "BlockedConnector"
 },
 {
 "$kind": "DlpViolationError",
 "violationType": "BlockedConnector"
 }
],
 "errorDescription": "At least one connector here has been block
 "diagnosticErrorCount": 2
 }]
```

X

## Data policy violations for end users of the agent

If your published agent is impacted by data policy enforcement and is in violation of your data policies, end users of the agent will see a **DataLossPreventionViolation** error when trying to interact with it.

The message says "Sorry, something unexpected happened. We're looking into it. Error code: DataLossPreventionViolation." and includes the conversation ID and time of the error.



End users should contact their admin to resolve the issue. The admin can check the data policy violations and update the policies or the agent configuration as needed.

# Reason

Microsoft Copilot Studio data policy enablement has been soft enabled for all customers over the course of January and February 2025. With this change, makers see data policy related errors when publishing or managing agents that violate existing data policies without immediately blocking their actions for agents that are already published.

With these changes, data policy exemption is no longer supported, and agents can't be exempted. The ability to exempt agents with a PowerShell command won't work.

Agents that were exempted from data policy enforcement had their enforcement set to **Soft-enabled** in January and February of 2025, and set to **Enabled** in February and March of 2025.

# Mitigation

Makers need to work with admins to check the **publish status of all agents deployed in production** to identify any potential issues caused by data policy violations. Using the insights from the publish errors and downloadable reports in the **Channels** tab, admins can adjust their data policies to align with their production workloads.

## Identify agents in violation of a data policy

From the **Channels** tab in Copilot Studio, you can immediately see warnings if your agent is in violation of data policies.

You can also select the **Details** link in the error notification to get more information about a violation. The **Channels** tab automatically opens and summarizes the data policy violations preventing new publication for an unpublished (or "draft") agent, or that are causing errors for a published agent.

Select **Download** to retrieve an Excel workbook that contains detailed information about the data policy violations. The workbook includes a summary of the errors, including the specific data policy name, ID, and the blocked connector causing the issue.

A	B	C	D	E	F
1	Copilot name				
2	Copilot environment name				
3	Downloaded (UTC)	12:27 March 03, 2025			
4	Documentation	<a href="https://aka.ms/CopilotStudioDLP">https://aka.ms/CopilotStudioDLP</a>			
5					
6	<input checked="" type="checkbox"/> Content	<input checked="" type="checkbox"/> Topic name	<input checked="" type="checkbox"/> Subcomponent	<input checked="" type="checkbox"/> Subcomponent type	<input checked="" type="checkbox"/> DLP policy name
7	Published	<a href="#">MSN Weather - Get forecast for today</a>	Connector	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
8	Published	<a href="#">Conversational boosting</a>	Knowledge source endpoint	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
9	Published	-	Application Insights	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
10	Published	-	Copilot authentication	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
11	Published	-	Knowledge source endpoint	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
12	Latest	<a href="#">MSN Weather - Get forecast for today</a>	Connector	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
13	Latest	<a href="#">Entra ID</a>	Connector	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
14	Latest	<a href="#">Conversational boosting</a>	Knowledge source endpoint	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
15	Latest	<a href="#">Conditional Sources</a>	Knowledge source endpoint	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
16	Latest	-	Application Insights	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012
17	Latest	-	Knowledge source endpoint	Copilot Studio V-Team DLP Policy	bac5e12-69d1-4b62-a936-0d8733170012

There are two worksheets in the Excel file:

- **DLP violations**, containing details for the data policy violations for that agent.
- **Blocked channels**, containing a list of the channels that are currently blocked by data policies for the agent.

The **DLP violations** sheet provides the name of the agent (as **Copilot name**) and its environment, followed by a table with the following columns:

 [Expand table](#)

Column	Description
Content	The publication status of the agent
Topic name	Name of the topic that triggered the violation, if applicable
Subcomponent	Category of the activity
Subcomponent type	Category for the data policy surface area
DLP policy name	The name of the policy (defined by the admin when the policy was created)
Policy id	GUID for the policy
DLP error type	The outcome of the policy (for example, <i>Connector blocked</i> )
Connector (data group)	Name of the connector that triggered the violation

The **Blocked channels** sheet includes the name of the agent (as **Copilot name**), along with the environment name. It's followed by a table with the following columns:

 [Expand table](#)

Column	Description
Channel name	The name of the channel where the agent was blocked by a data policy violation
DLP policy name	The name of the policy (defined by the admin when the policy was created)
Policy id	GUID for the policy

 **Important**

If all channels for the agent are blocked by data policies, you can't publish your agent.

## Identify users with sufficient permissions to update data policies

After identifying data policies that may need to be updated, you'll need an admin to update [Data policies](#) in the [Power Platform admin center](#).

See [Configure data policies for agents](#) for more details and examples of using data policies in Copilot Studio.

When an agent is in violation of a data policy, makers need to determine what policies are impacting them. Data policies can be defined at the tenant level (to impact all environments in a tenant) or for one or more specific environments.

Tenant-wide data policies require tenant-level administrator. Environment-specific data policies can be configurable by users with a less permissive role in the environment.

# Data policy example - Configure a policy to require user authentication in agents

09/24/2025

When you create a new agent, the **Authenticate with Microsoft** authentication option is turned on by default. The agent automatically uses Microsoft Entra ID authentication without requiring any manual setup and only lets you chat with your agent on Teams. However, agent makers in your organization can select the **No authentication** authentication option to allow anyone with the link to chat with your agent.

The screenshot shows the 'Authentication' configuration dialog. It includes a description of verifying user identity, a 'Learn more' link, and a 'Choose an option' section. The 'Authenticate with Microsoft' option is selected (radio button is filled) and highlighted with a red border. This option describes Entra ID authentication for Teams and Power Apps, noting that only these channels are available. A 'Learn more' link is also present here. Other options shown are 'No authentication' (publicly available in any channel) and 'Authenticate manually' (set up for any channel, supporting Microsoft Entra ID or OAuth2 providers). A checked checkbox for 'Require users to sign in' is also visible.

**Authentication** X

Verify a user's identity during a conversation. The copilot receives secure access to the user's data and is able to take actions on their behalf, resulting in a more personalized experience.

[Learn more](#)

Choose an option

No authentication  
Publicly available in any channel

Basic copilot setup with no authentication action or authentication variables.

Authenticate with Microsoft  
Entra ID authentication in Teams and Power Apps

Only Teams and Power Apps channels are available; all other channels will be disabled. [Learn more](#)

Authenticate manually  
Set up authentication for any channel

Supports Microsoft Entra ID or any OAuth2 identity provider. Available in any channel. [Learn more](#)

Require users to sign in

You can use data policies to block your agent makers from configuring and publishing agents that aren't configured for authentication to help prevent data exfiltration.

If an enforced data policy applies to this connector, agent makers must [configure user authentication](#) with **Authenticate with Microsoft** or **Authenticate manually** in Copilot Studio.

For more information about other data policy related connectors, see [Configure data policies for agents](#).

## Configure data policy to require authentication in the Power Platform admin center

To configure the data policy to require user authentication, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security**, and then select **Data and privacy**. The **Data protection and privacy** page opens.
3. Select **Data policy**. The **Data policies** page opens.
4. Create a new data policy, or choose an existing data policy to edit:
  - If you want to create a new data policy, select **New Policy**. For detailed instructions, see [Create a data policy](#).
  - If you want to choose an existing data policy to edit, select the data policy and select **Edit Policy**. For detailed instructions, see [Edit a data policy](#).
5. Select **Next**. The **Add an environment** page opens.
  - If you want to add an environment to your data policy, select the environment in the **Available** tab and select **Add to policy**.
  - If you want to remove an environment from your data policy, select the environment in the **Added to policy** tab and select **Remove from policy**.
6. Select **Next**. The **Assign connectors** page opens.
7. Use the search box to find the **Chat without Microsoft Entra ID authentication in Copilot Studio** connector.
8. Select the connector's **More actions** icon (:), and then select any of the following options:
  - **Move to Business**
  - **Block**
  - **Configure connector**
9. Select **Next**.
10. If you're a tenant admin, or an environment admin for multiple environments, the **Define scope page** opens. Select either of the following options:
  - **Add all environments**: Adds all the environment in your entire tenant. This policy automatically applies to any new environment that is created.
  - **Add multiple environments**: Choose the environments to include in this policy.
  - **Exclude certain environments**: Choose the environments to exclude from this policy.

 **Note**

If the policy has a tenant scope, data policy applies to all agents.

11. Select **Next**.

12. Review your policy, then select **Create policy** if you're creating a new policy or **Update policy** if you're editing an existing policy.

## Confirm data policy enforcement in Copilot Studio

You can confirm that this connector is being used in the data policy from the Copilot Studio web app.

First, open your agent from the environment where the data policy is applied.

If the policy is enforced, you see an error banner with a **Details** button. To see details, on the **Channels** page, expand the error link and select the **Download** button. In the details file, a row appears to describe each violation.

An agent maker can contact their admins with the data loss prevention download spreadsheet details to make appropriate updates to the data policy. Alternatively, the agent maker can update the agent authentication to **Authenticate with Microsoft** or **Authenticate manually (Azure Active Directory or Azure Active Directory v2)** in the **Authentication** configuration page. See [Configure user authentication in Copilot Studio](#).

Authentication options aren't selectable if they don't use Microsoft Entra ID authentication.

## Authentication

X

**⚠** No authentication and service provider Generic OAuth 2 aren't available due to changes in your organization's data loss prevention policy. Contact your admin with questions. [Learn more](#)

Verify a user's identity during a conversation. The copilot receives secure access to the user's data and is able to take actions on their behalf, resulting in a more personalized experience.

[Learn more](#)

Choose an option

No authentication

Publicly available in any channel

Basic copilot setup with no authentication action or authentication variables.

Authenticate with Microsoft

Entra ID authentication in Teams and Power Apps

Only Teams and Power Apps channels are available; all other channels will be disabled. [Learn more](#)

Authenticate manually

Set up authentication for any channel

Supports Microsoft Entra ID or any OAuth2 identity provider. Available in any channel. [Learn more](#)

Require users to sign in

**i** Your organization's data loss prevention policies prevent you turning off this feature. Contact your admin. [Learn more](#)

Redirect URL

<https://token.botframework.com/.auth/web/redirect>

[Copy](#)

Service provider \*

Azure Active Directory v2

▼

Azure Active Directory

Azure Active Directory v2

Generic OAuth 2

# Data policy example - Configure a policy for knowledge sources for agents

09/24/2025

You can use data policies to configure which knowledge sources across environments that agent authors can connect to for data. Doing so can help prevent data exfiltration.

For more information about other data policy configurations, see [Configure data policies for agents](#).

## Configure data policy for knowledge sources for agents in Power Platform admin center

To configure a data policy for knowledge sources, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security**, and then select **Data and privacy**. The **Data protection and privacy** page opens.
3. Select **Data policy**. The **Data policies** page opens.
4. Create a new data policy, or choose an existing data policy to edit:
  - If you want to create a new data policy, select **New Policy**. For detailed instructions, see [Create a data policy](#).
  - If you want to choose an existing data policy to edit, select the data policy and select **Edit Policy**. For detailed instructions, see [Edit a data policy](#).
5. Select **Next**. The **Add an environment** page opens.
  - If you want to add an environment to your data policy, select the environment in the **Available** tab and select **Add to policy**.
  - If you want to remove an environment from your data policy, select the environment in the **Added to policy** tab and select **Remove from policy**.
6. Select **Next**. The **Assign connectors** page opens.
7. Use the search box and type "Knowledge source" to find the connectors pertaining to agent knowledge sources. The following connectors should appear in the results:
  - Knowledge source with SharePoint and OneDrive in Copilot Studio

- Knowledge source with public websites and data in Copilot Studio
- Knowledge source with documents in Copilot Studio

8. Select the connector's **More actions** icon (:), and then select any of the following options:

- **Move to Business**
- **Block**
- **Configure connector**

9. Select **Next**.

10. Review your policy, then select **Create policy** if you're creating a new policy or **Update policy** if you're editing an existing policy.

If admins want to allow or deny SharePoint endpoints their makers can use as knowledge sources in Copilot Studio, they can use [Data policy connector endpoint filtering](#) instead of blocking it.

## Confirm data policy enforcement

You can confirm that this connector is being used in the data policy from Copilot Studio.

1. Open your agent from the environment where the data policy is applied.
2. Go to the **Knowledge** tab, select **Add knowledge**, and add a SharePoint knowledge source.

If the policy is enforced, an error banner with a **Details** button appears. On the **Channels** page, expand the error link and select the **Download** button to see details. The **Published** button is disabled when there's a data policy violation.

In the details file, a row appears for each violation. If a knowledge source has a data policy violation, a row appears for the knowledge page and for each generative answers node that uses that knowledge source.

### Note

Classic chatbots don't support Power Platform connectors.

# Data policy example - Block Power Platform connectors in agents

09/24/2025

You can use data policies to prevent agent authors from configuring connectors. Doing so can help prevent data exfiltration.

For more information about other data policy configurations, see [Configure data policies for agents](#).

## Configure data policy to block Power Platform connectors in the Power Platform admin center

To configure the data policy to block Power Platform connectors, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security**, and then select **Data and privacy**. The **Data protection and privacy** page opens.
3. Select **Data policy**. The **Data policies** page opens.
4. Create a new data policy, or choose an existing data policy to edit:
  - If you want to create a new data policy, select **New Policy**. For detailed instructions, see [Create a data policy](#).
  - If you want to choose an existing data policy to edit, select the data policy and select **Edit Policy**. For detailed instructions, see [Edit a data policy](#).
5. Select **Next**. The **Add an environment** page opens.
  - If you want to add an environment to your data policy, select the environment in the **Available** tab and select **Add to policy**.
  - If you want to remove an environment from your data policy, select the environment in the **Added to policy** tab and select **Remove from policy**.
6. Select **Next**. The **Assign connectors** page opens.
7. Use the search box to find the connector you want to block.
8. Select the connector's **More actions** icon (:), and then select **Block**. You can also see connectors that are already blocked in the **Blocked** tab.
9. Select **Next**.

10. Review your policy, then select **Create policy** if you're creating a new policy or **Update policy** if you're editing an existing policy.

## Confirm data policy enforcement

You can confirm that this connector is being used in the data policy from Copilot Studio:

1. Open your agent from the environment where the data policy is applied and go to the authoring canvas.
2. Create a new topic and add an **Action** node.
3. In the node's properties, select **Connectors** and choose your connection. Save your topic.
4. If the policy is enforced, you'll see an error banner with a **Details** button after the topic is saved. On the **Channels** page, expand the error link and select the **Download** button to see details. The **Published** button is disabled when there's a data policy violation.

In the details file, a row appears for each violation. If a connector has a data policy violation, a row appears for each connector.

 **Note**

Classic chatbots don't support Power Platform connectors.

# Data policy example - Block HTTP requests in agents

09/24/2025

Agent makers in your organization can make HTTP requests with the [Send HTTP request node](#).

You can use data policies to prevent agent makers from configuring HTTP requests to help prevent data exfiltration.

For more information about other data policy related connectors, see [Configure data policies for agents](#).

## Configure data policy to block HTTP requests in the Power Platform admin center

To configure the data policy to block HTTP requests, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security**, and then select **Data and privacy**. The **Data protection and privacy** page opens.
3. Select **Data policy**. The **Data policies** page opens.
4. Create a new data policy, or choose an existing data policy to edit:
  - If you want to create a new data policy, select **New Policy**. For detailed instructions, see [Create a data policy](#).
  - If you want to choose an existing data policy to edit, select the data policy and select **Edit Policy**. For detailed instructions, see [Edit a data policy](#).
5. Select **Next**. The **Add an environment** page opens.
  - If you want to add an environment to your data policy, select the environment in the **Available** tab and select **Add to policy**.
  - If you want to remove an environment from your data policy, select the environment in the **Added to policy** tab and select **Remove from policy**.
6. Select **Next**. The **Assign connectors** page opens.
7. Use the search box to find the **HTTP** connector.
8. Select the connector's **More actions** icon (:), and then select **Block**.

9. If admins want to allow or deny specific HTTP endpoints, they can use [Data policy connector endpoint filtering](#) instead of blocking all HTTP calls.

10. Select **Next**.

11. Review your policy, then select **Create policy** if you're creating a new policy or **Update policy** if you're editing an existing policy.

## Confirm data policy enforcement

### Web App

You can confirm that this connector is being used in the data policy from the Copilot Studio.

First, open your agent from the environment where the data policy is applied. Go to the authoring canvas, create a new topic, add a **Send HTTP request** node (minimally populating the URL property), and save your topic.

If the policy is enforced, you'll see an error banner with a **Details** button. On the **Channels** page, expand the error link and select the **Download** button to see details.

In the details file, a row appears describing each violation. A violation occurs if the HTTP connector is blocked, if the HTTP connector is in a different data group than other connectors in your data policy, or if the HTTP policy isn't blocked but an endpoint is denied.

# Data policy example - Configure a policy for skills

09/24/2025

Agent makers in your organization can [extend their agents with skills](#). Skills can be a useful way of extending the functionality of your agents, however you may want to configure which skills agents can use for security purposes.

You can use the **Skills with Copilot Studio** connector in Power Platform data policies to configure which agents can connect to skills.

See the [Configure data policies for agents](#) topic for information about other data policy related connectors.

## Configure data policy for skills in the Power Platform admin center

To configure a data policy for skills, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security**, and then select **Data and privacy**. The **Data protection and privacy** page opens.
3. Select **Data policy**. The **Data policies** page opens.
4. Create a new data policy, or choose an existing data policy to edit:
  - If you want to create a new data policy, select **New Policy**. For detailed instructions, see [Create a data policy](#).
  - If you want to choose an existing data policy to edit, select the data policy and select **Edit Policy**. For detailed instructions, see [Edit a data policy](#).
5. Select **Next**. The **Add an environment** page opens.
  - If you want to add an environment to your data policy, select the environment in the **Available** tab and select **Add to policy**.
  - If you want to remove an environment from your data policy, select the environment in the **Added to policy** tab and select **Remove from policy**.
6. Select **Next**. The **Assign connectors** page opens.

7. Use the search box to find the **Skills with Copilot Studio** connector.

8. Select the connector's **More actions** icon (:), and then select any of the following options:

- **Move to Business**
- **Block**
- **Configure connector**

9. Select **Next**.

10. Review your policy, then select **Create policy** if you're creating a new policy or **Update policy** if you're editing an existing policy.

## Confirm data policy enforcement in Copilot Studio

You can confirm that this connector is being used in the data policy from Copilot Studio.

Open your agent from the environment where the data policy is applied, and try to [add a skill](#) to the agent.

If the policy is enforced, the **Add a skill** panel reports an error and suggests you contact an admin to add the skill to the allowlist.

# Data policy example - Configure a policy for the channels used to publish agents

09/24/2025

You can use data policies to configure the channels through which agents can be published. Copilot Studio supports connectors for the following channels:

- Teams and Microsoft 365
- Direct Line
- Facebook
- Omnichannel
- SharePoint
- WhatsApp

## Configure data policy for channels used to publish agents in Power Platform admin center

To configure a data policy for the channels used to publish agents, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security**, and then select **Data and privacy**. The **Data protection and privacy** page opens.
3. Select **Data policy**. The **Data policies** page opens.
4. Create a new data policy, or choose an existing data policy to edit:
  - If you want to create a new data policy, select **New Policy**. For detailed instructions, see [Create a data policy](#).
  - If you want to choose an existing data policy to edit, select the data policy and select **Edit Policy**. For detailed instructions, see [Edit a data policy](#).
5. Select **Next**. The **Add an environment** page opens.
  - If you want to add an environment to your data policy, select the environment in the **Available** tab and select **Add to policy**.
  - If you want to remove an environment from your data policy, select the environment in the **Added to policy** tab and select **Remove from policy**.
6. Select **Next**. The **Assign connectors** page opens.

7. Use the search box to find the connectors pertaining to channels through which agents can be published. Search for any of the following:

- Microsoft Teams + M365 Channel in Copilot Studio
- Direct Line channels in Copilot Studio
- Facebook channel in Copilot Studio
- Omnichannel in Copilot Studio
- SharePoint channel in Copilot Studio
- WhatsApp channel in Copilot Studio

8. Select the connector's **More actions** icon (:), and then select any of the following options:

- Move to Business
- Block
- Configure connector

9. Select **Next**.

10. Review your policy, then select **Create policy** if you're creating a new policy or **Update policy** if you're editing an existing policy.

## Limitations of publishing agents based on allowed channels

Makers can only publish agents to the allowed channel. If makers don't configure their agents for the allowed channel except Direct Line (on by default), or if the administrators don't allow any channel, the agents can't be published.

# Data policy example - Block event triggers in agents

09/24/2025

Agent makers in your organization can [add event triggers to agents](#). Event triggers allow your agents to react to external events without human prompting. However, you might want to restrict their use to, for example, prevent data exfiltration or unwanted consumption or quota usage.

You can use the Microsoft Copilot Studio connector in Power Platform admin center data policies to stop agent makers from adding event triggers to their agent.

For more information about data policy related connectors, see [Configure data policies for agents](#).

## Configure data policy to block event triggers in the Power Platform admin center

To configure the data policy to block event triggers, follow these steps:

1. Sign in to the [Power Platform admin center](#).
2. On the side navigation, select **Security**, and then select **Data and privacy**. The **Data protection and privacy** page opens.
3. Select **Data policy**. The **Data policies** page opens.
4. Create a new data policy, or choose an existing data policy to edit:
  - If you want to create a new data policy, select **New Policy**. For detailed instructions, see [Create a data policy](#).
  - If you want to choose an existing data policy to edit, select the data policy and select **Edit Policy**. For detailed instructions, see [Edit a data policy](#).
5. Select **Next**. The **Add an environment** page opens.
  - If you want to add an environment to your data policy, select the environment in the **Available** tab and select **Add to policy**.
  - If you want to remove an environment from your data policy, select the environment in the **Added to policy** tab and select **Remove from policy**.
6. Select **Next**. The **Assign connectors** page opens.

7. Use the search box to find the **Microsoft Copilot Studio** connector.
8. Select the connector's **More actions** icon (:), and then select **Block**.
9. Select **Next**.
10. Review your policy, then select **Create policy** if you're creating a new policy or **Update policy** if you're editing an existing policy.

## Confirm data policy enforcement in Copilot Studio

You can confirm that this connector is being used in the data policy from Copilot Studio.

Open your agent from the environment where the data policy is applied, and try to [add a trigger](#) to the agent.

If the policy is enforced, the **Add a trigger** panel reports an error and suggests you contact an admin to add event triggers to the allowlist.

**Triggers**

Set up your agent to activate when certain events happen. [Learn more](#).

✖ Your organization's data loss prevention policies do not allow the creation of triggers.  
Contact your admin with questions.

cloud When a file is modified → Trigger your agent with certain message upon event: When a file is modified. ✖ ...

# Troubleshoot Conditional Access policy issues

Article • 05/02/2025

Copilot Studio supports [end-user authentication and authorization through Microsoft Entra ID](#), so that users of your agents can use their Microsoft Entra ID credentials to authenticate. Your organization manages these credentials.

However, there may be instances where users encounter issues related to [Conditional Access policies](#) that affect their ability to use Copilot Studio agents effectively.

## Symptoms

Agents may be unresponsive to end users on specific channels, such as Teams, due to Conditional Access policies implemented through Microsoft Entra ID.

Users of your agents might see a blank page in the chat window or receive an error message indicating that the agent isn't available, and the test chat won't respond to queries.

## Reason

**Policy Enforcement:** Due to recent security updates that provide stronger authentication controls, Copilot Studio agent acquires the authentication token specific to a given customer tenant.

With these policy enforcements in place, agents won't initiate a conversation or respond to end users if Conditional Access policies are in place that block the acquisition of the authentication token.

The enforcement applies to existing agents created in tenants with Conditional Access policies that, previously, didn't prevent the agent from responding to end users.

## Mitigation

You can see which Conditional Access policy is blocking the request so that you can investigate and act accordingly. See the additional resources at the end of this article for guidance on how to resolve issues or modify Conditional Access policies.

You might also need to [allow specific IP addresses and IP ranges](#) that are used by Copilot Studio, Power Platform, or other Microsoft services.

You can get to the Conditional Access logs for a specific Copilot Studio agent from the agent's app registration in Entra. You can also see logs for all agents by manually filtering within the Identity section in Entra.

### 💡 Tip

Depending on who made the request, the associated log could be in one of multiple sign-in categories in Entra.

Check each tab on the Conditional Access sign-in logs page.

## Get Conditional Access logs for all Copilot Studio agents

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Reports Reader**.
2. Open the **Identity** section on the side menu. Select **Monitoring & health**, and then **Audit logs**.
3. Select the **Date** range you want to query.
4. Select **Add filters** above the list of sign ins, then select **Application**. After it's added, set the filter to **Application contains: Copilot Studio**.
5. Add the **Conditional Access** filter in the same way, and set it to **Failure**. Select **Apply**.

The screenshot shows the 'Sign-in events' page in the Microsoft Entra admin center. At the top, there are buttons for 'Download', 'Export Data Settings', 'Troubleshoot', 'Refresh', 'Columns', and 'Go'. Below these are two tabs: 'User sign-ins (interactive)' (selected) and 'User sign-ins (non-interactive)'. Underneath are two date filters: 'Date : Last 24 hours' and 'Show dates as : Local'. A red box highlights a modal dialog titled 'Conditional Access : None Selected' with a close button. The dialog has a title 'Conditional Access' and three filter options: 'Not Applied' (unchecked), 'Success' (unchecked), and 'Failure' (checked). At the bottom of the dialog is a blue 'Apply' button. The main table below the filters shows a list of sign-in events with columns for 'Date', 'Request ID', and 'Managed by'. The first few rows show sign-ins from 'Copilot Studio' on 4/16/2025 at various times.

Date	Request ID	Managed by
4/16/2025, 10:28:06 AM	180...	Copilot Studio
4/16/2025, 10:27:35 AM	314...	Copilot Studio
4/16/2025, 10:27:32 AM	5ed...	Copilot Studio
4/16/2025, 10:27:27 AM	d53...	(Microsoft Copi...

## Get Conditional Access logs for a specific Copilot Studio agent

1. Sign in to the [Microsoft Entra admin center](#).

2. Open **App registrations** from the side menu, homepage, or by searching for it in the search bar at the top of the screen.
3. Open the registration for the agent you want to review.
4. On the **Overview** page, under the **Essentials** section, select the link for the **Managed application in local directory**. This takes you to a prefiltered list of Conditional Access logs for that agent.

## Identify and remediate policy failures

By default, the audit logs display all activities. Open the **Activity** filter to narrow down the activities, if necessary. For a list of audit log activities for Conditional Access, see the [Microsoft Entra audit log activities](#) article in the [Entra Conditional Access documentation](#).

1. Review the activities under each tab to locate any that triggered a Conditional Access policy failure for Copilot Studio.
2. Select an entry to open the **Activity Details** blade, then go to the **Conditional Access** tab. The associated policies that triggered the issue is listed, along with the action taken as a result of the policy, such as **Block**.

After you identify the associated policy, you can troubleshoot to determine what you need to do. For example, you can continue allowing the policy to block agent interactions, change the policy's scope, or modify or disable the policy.

The following articles in the Microsoft Entra Conditional Access documentation detail the next steps you can take in Entra to resolve the issue:

- [Troubleshoot policy changes](#)
- [Troubleshoot sign-in problems](#)
- [Troubleshoot using the What If tool](#)

# Audit Copilot Studio activities in Microsoft Purview

Article • 01/31/2025

This article lists and describes the Copilot Studio activities that are logged and available using the [Microsoft Purview compliance portal](#). These logs are also accessible to developers via the [Office 365 Management API](#).

Changes to the content and settings of an agent can affect security and agent behavior. It's important to audit such actions to help mitigate failures, help contain systems of security constraints, adhere to compliance requirements, and act on security threats.

Copilot Studio logs activities related to both [administrative](#) and [maker and user interactions](#) with agents.

The data used to generate the audit logs is stored in accordance with Copilot Studio security and compliance standards. For more information, see [Microsoft Copilot Studio security and compliance](#).

## Important

Administrative activities for Copilot Studio are enabled by default on all tenants. You can't disable activity collection, but you can [disable the audit logs in Purview and use retention policies](#) to prevent the retention of user message text and response text.

## Prerequisites

- Users must have an assigned Microsoft 365 license so that Copilot Studio can record audit events and save transcripts of their conversations with Microsoft 365 Copilot.
- Your tenant isn't a [Federal Risk and Authorization Management Program \(FedRAMP\) tenant](#).

## Access the logs

1. Sign in to the [Microsoft Purview compliance portal](#) as a tenant admin.
2. In the left menu, select **Show all**.

### 3. Under Solutions, select Audit.

Admins can filter for Copilot Studio activities in the Activities list. Activities are mapped to event types and categories, as listed in the following tables in this article.

Search

1 Searches completed | 0 Active searches | 0 Active unfiltered searches

Date and time range (UTC) \*

Start: Jan 13 | 00:00

End: Jan 14 | 00:00

Keyword Search: Enter the keyword to search for

Admin Units: Choose which Admin Units to search for

Activities - friendly names: Interacted with Copilot

Copilot activities:  Interacted with Copilot

M365 Apps Admin Services cloud policy acti

Created policy configuration

Updated policy configuration

Deleted policy configuration

Search | Clear all

Compliance managers can also use the Data Security Posture Management (DSPM) for AI solution to view chat transcripts for CopilotInteraction events. For more information, see [Data Security Posture Management \(DSPM\)](#).

## See audited events (agent authoring)

All logging is done at the SDK layer, so a single action can trigger multiple logged events.

This table lists events typically related to administrative actions in Copilot Studio, such as deleting an agent or updating an agent's name, details, or configuration.

  Expand table

Category	Event label	Description of the event
Agents	BotDeleteCleanup	The cleanup of dependencies after an agent is deleted in Copilot Studio

<b>Category</b>	<b>Event label</b>	<b>Description of the event</b>
Agents	BotNameUpdate	Updating the agent's name in Copilot Studio
Agents	BotCreate	The creation of a new agent in Copilot Studio
Agents	BotDelete	The deletion of an agent in Copilot Studio
Agents	BotAuthUpdate	Updating the authentication settings of an agent in Copilot Studio
Agents	BotIconUpdate	Updating the agent icon in Copilot Studio
Agents	BotPublish	Publishing of an agent in Copilot Studio
Agents	BotShare	Sharing of an agent to other users in Copilot Studio
Agents	BotAppInsightsUpdate	Updating the App Insights logging configuration of an agent in Copilot Studio
Agent Component	BotComponentCreate	The creation of a component (such as a topic or skill) for an agent in Copilot Studio
Agent Component	BotComponentUpdate	The update of a component (such as a topic or skill) for an agent in Copilot Studio
Agent Component	BotComponentDelete	The deletion of a component (such as a topic or skill) for an agent in Copilot Studio
Agent Component Collection	BotComponentCollectionCreate	The creation of a component collection for an agent in Copilot Studio
Agent Component Collection	BotComponentCollectionDelete	The deletion of a component collection for an agent in Copilot Studio
Agent Component Collection	BotComponentCollectionUpdate	The update of a component collection for an agent in Copilot Studio
AI Plugin	AIPluginOperationCreate	Creating an AI Plugin for an agent in Copilot Studio
AI Plugin	AIPluginOperationUpdate	Updating an AI Plugin for an agent in Copilot Studio
AI Plugin	AIPluginOperationDelete	Removing an AI Plugin for an agent in Copilot Studio

Category	Event label	Description of the event
Environment Variable	EnvironmentVariableCreate	Creating an environment variable for an agent in Copilot Studio
Environment Variable	EnvironmentVariableUpdate	Updating an environment variable for an agent in Copilot Studio
Environment Variable	EnvironmentVariableDelete	Deleting an environment variable for an agent in Copilot Studio

## See audited events (agent usage)

### i Note

All logging is done at the SDK layer, so a single action can trigger multiple logged events.

Some channels are excluded from logging. For events listed in the **Audit** solution in Purview, the channel is identified in the audited event's **CopilotEventData** JSON field as the **AppHost** value. In the **DSPM for AI** solution, the channel is identified under the **App accessed in** field.

This table lists events that are logged for user interactions with Copilot agents in Copilot Studio, such as asking questions and viewing responses. The audited event contains metadata for that activity (including the date/time, organization, user and resource IDs, and the transcript thread ID).

Copilot Studio saves the text of the chat (the transcript) separately from the audited event in Purview. The transcript can be seen or accessed directly from the logged event as it appears in the [Data Security Posture Management \(DSPM\) for AI](#) solution.

### i Important

The full text or transcript of the interactions between a user and the agent aren't included in the audit logs in the **Audit** solution, only the transcript thread ID. The [Data Security Posture Management \(DSPM\) for AI](#) solution attempts to retrieve the chat text related to the event, along with links to resources that were accessed as part of the logged event.

[\[+\] Expand table](#)

Category	Event label	Description of the event
Users	CopilotInteraction	User Interactions with a Copilot agent created in Copilot Studio (such as asking questions and viewing responses)

For more information about Copilot interaction events in Microsoft 365, see [Copilot interaction events overview](#).

## Schema audit fields

Schemas define which agent fields are sent to the Microsoft Purview compliance portal. Some fields are common to all applications that send audit data to Microsoft Purview, while others are specific to Copilot Studio. The following are fields common to the Power Platform.

[\[+\] Expand table](#)

Field display name	Logical name	Type	Mandatory	Description
Date	CreationTime	Edm.Date	No	Date and time when the log was generated in UTC.
Id	ID	Edm.Guid	No	Unique GUID for every logged row.
Result Status	ResultStatus	Edm.String	No	Status of the logged row.
Organization Id	OrganizationId	Edm.Guid	Yes	Unique identifier of the organization from which the log was generated.
Operation	Operation	Edm.String	No	Name of the operation, including the event label.
User	UserKey	Edm.String	No	Unique identifier of the user in Microsoft Entra ID.
User type	UserType	Self.UserType	No	The audit type (admin, regular, or system).

## Copilot Studio audit fields

In addition to the fields common to Power Platform administrator activities, Copilot Studio includes the following fields.

[\[+\] Expand table](#)

Name	Type	Mandatory	Description
BotId	Edm.String	No	A unique identifier of the agent
BotSchemaName	Edm.String	No	A unique string identifying the agent
BotUpdateDetails	Edm.ComplexType (Collection)	No	Details of the properties updated on the agent
BotComponentId	Edm.String	No	A unique identifier of the agent components, such as topics or entities
BotComponentSchemaName	Edm.String	No	A unique string identifying the components of an agent, such as topics or entities
BotComponentType	Edm.String	No	The type of the agent component, such as topics or entities
BotComponentUpdateDetails	Edm.ComplexType (Collection)	No	Details of the updated properties of the agent's component
AIPluginOperationId	Edm.String	No	A unique identifier for the operation with the AI plugin
AIPluginOperationName	Edm.String	No	The name of the operation with the AI plugin

Name	Type	Mandatory	Description
EnvironmentVariableDefinitionSchemaName	Edm.String	No	Schema name for the environment variable definition associated with the agent
EnvironmentVariableDefinitionId	Edm.String	No	A unique identifier for the environment variable definition associated with the agent

## Disable audit logging

You can [disable event logging of all audit events across your tenant in Purview](#).

To prevent retention of user and agent text, you can create a Data Lifecycle Management policy in Purview that sets a retention policy of 0 days for Copilot Experiences.

For more information, see [Create and configure retention policies](#).

## Related content

- [Auditing solutions in Microsoft Purview](#)
- [Power Platform connector activity logging](#)
- [Data loss prevention activity logging](#)
- [Manage Dataverse auditing](#)

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# View sensitivity labels in agent responses (preview)

10/17/2025

[This article is prerelease documentation and is subject to change.]

Your organization's makers work with people inside and outside the organization. This means your data can go anywhere, on any device, app, or service. You want to keep your data secure and compliant with your organization's policies.

With Microsoft Purview, you can use sensitivity labels to classify and protect your data, without affecting user productivity and collaboration. Specific to your organization and business needs, your admins can create categories for different levels of sensitive content in your organization.

For example, you can create and apply labels for Personal, Public, General, Confidential, and Highly Confidential content. For more information, see [Learn about sensitivity labels](#) in the Purview documentation.

Copilot Studio supports sensitivity labels for the following [knowledge sources](#) and [sources used in the generative answers classic mode](#):

- SharePoint
- OneDrive for Business
- SQL
- Dataverse
- Cosmos
- Azure Blob Storage
- Word
- Excel
- Outlook
- Office groups
- Office users

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

# Microsoft Purview strengthens information protection for Copilot Studio

An agent uses existing controls to ensure that data stored in your tenant is never returned to the user, or used by generative AI, if the user doesn't normally have access to that data. When the data has sensitivity labels from your organization applied to the content, there's an extra layer of protection.

When an agent is configured with a supported knowledge source, the user sees a shield icon with the highest sensitivity label that has been applied to the content the agent used to generate the response. The user also sees a sensitivity label for each cited file in the agent's response.

The following screenshots show sensitivity labels in agent responses:

- In conversations with agents deployed to the web, the shield is colored to match the sensitivity label, appears below the response, and lists the highest label. The citations after the response also indicate the label applied to that specific file or reference:

Copilot Studio is available as both a standalone web app and as a discrete app within Teams. Most of the functionality between the two versions is the same, but there might be different reasons to choose one version over the other based on how you want to use Copilot Studio. To use Granular Controls, you need to ensure that Copilot Studio follows AI rules and adheres to the "Chain of Protection," such as Sensitivity label. When AI uses existing data specific to an individual user to deliver a new capability, it must maintain a sensitivity label and protection set of the highest labeled and protected source. Around 700+ customers have turned off generative AI copilot publish in their tenants [1] [2].

2 references ▾

	1	2
Copilot Studio.docx	General	Confidential\Any User
Secure Generative Answers Share...		Confidential\Any User

- In conversations with agents on Teams, the shield appears above the response. When a user hovers over the shield icon, the name and summary of the highest label appears. The citations after the response also indicate the label applied to that specific file or reference:



**ⓘ Important**

Microsoft Purview sensitivity labels in Copilot Studio are on by default for agents with supported knowledge sources.

## Related content

- [Learn about sensitivity labels](#)
- [Microsoft Purview data security and compliance protections for generative AI apps](#)
- [Enable sensitivity labels for files in SharePoint and OneDrive](#)

# Configure data movement across geographic locations for generative AI

Article • 11/19/2024

With the [generative AI features](#) of Copilot Studio, you can instantly build useful agents without creating complex conversation flows or manual authoring and configuration. These features draw on Azure OpenAI API Service and Bing Search.

These capabilities are available in environments created for the United States and other [supported geographic locations](#).

In some non-United States geographic locations, global and Power Platform administrators might need to make these features available in your environment.

If these features aren't available for your environment, a message appears in Copilot Studio on the agents overview page.

You can continue to configure some Copilot Studio generative AI capabilities even if the features aren't available. However a message appears when you test your agent.

## Important

Accessing Copilot Studio generative AI features from regions outside the United States results in data movement across regional boundaries. This data movement can be enabled and disabled in Power Platform. Once enabled, any data movement that occurred while this feature was enabled cannot be reversed by removing your consent.

Features powered by Bing are separately governed by the [Microsoft Services Agreement](#) and [Microsoft Privacy Statement](#).

Only global and Power Platform administrators can enable and disable generative AI features for environments outside the United States.

For more information about data movement associated with these generative AI features and how to make these features available, see [Turn on agents and generative AI features](#).

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

# Manage data storage powered by Microsoft 365 services

Agents with generative mode turned on can collect historical activity data that makers can use to troubleshoot and improve their agent. Makers can view activity data as a visual map or as transcripts.

Historical activity is powered by Microsoft 365 services. Makers who want to create an agent with activity data need to have an Exchange license. All activity data is stored in the geographic region of the end user's Exchange mailbox, regardless of the geographic location (geo) selected for the Copilot Studio environment used to create the agent. Power Platform can be configured to use [Customer Managed Keys \(CMK\)](#). *End user* is the person engaging with the agent through conversation or invoking the agent with a [trigger](#). In many cases, the agent maker is also the end user.

When you use activity mapping, data for tracking agent activity is stored using Microsoft 365 services, which is not governed by Azure compliance. Instead, activity data is governed by [Microsoft 365 terms and data residency commitments](#). Global and Power Platform administrators can [turn off data being stored by Microsoft 365 through the Power Platform admin center](#).

Turning this off stops future activity data storage in Microsoft 365. Existing activity data stored in Microsoft 365 will be deleted or purged according to the data retention policy set for Microsoft 365.

If these features aren't available for your environment, a message appears in Copilot Studio on the agent's [Overview](#) page.

---

Last updated on 10/30/2025

# Copilot Studio for US Government customers

10/15/2025

This article is for US government customers who are deploying Copilot Studio as a part of a Copilot Studio Government Community Cloud (GCC) plan. It provides you with an overview of features that are specific to these plans.

Government plans are designed for the unique needs of organizations that must meet US compliance and security standards.

We recommend that you read this article and [Copilot Studio overview](#).

The Copilot Studio US Government Service Description serves as an overlay to the general Copilot Studio Service Description. It defines the unique commitments and differences compared to the general Copilot Studio offerings that are available to our customers since December 2019.

## Copilot Studio US Government plans and environments

Licensing for Copilot Studio US Government plans is the same as for public cloud. They're available through the Volume Licensing and Cloud Solution Provider purchasing channels. For more information, see [Assign user licenses and manage access](#).

The Copilot Studio GCC environment is compliant with the Federal requirements for cloud services, including FedRAMP High.

In addition to the features and capabilities of Copilot Studio, organizations that use Copilot Studio US Government plans benefit from the following unique features:

- Your organization's customer content is physically separated from customer content in non-US-Government plans for Copilot Studio.
- Your organization's customer content is stored within the United States.
- Access to your organization's customer content is restricted to screened Microsoft personnel.
- Copilot Studio US Government complies with all certifications and accreditations that US Public Sector customers require.

## GCC High environment

Beginning February 2022, eligible customers can choose to deploy Copilot Studio US Government to the GCC High environment.

Microsoft designed the platform and our operational procedures to meet the requirements aligning with the DISA SRG IL4 (Defense Information Systems Agency Security Requirements Guide Impact Level 4) compliance framework.

This option enables and requires the customer to use Microsoft Entra ID for Government for customer identities. By contrast, GCC uses the public Microsoft Entra ID.

For the US Department of Defense contractor customer base, Microsoft operates the service in a manner that enables these customers to meet International Traffic in Arms Regulations (ITAR) commitment and Defense Federal Acquisition Regulation Supplement (DFARS) acquisition regulations, as documented and required by their contracts with the US Department of Defense. DISA has granted a Provisional Authority to Operate.

## Customer eligibility

Copilot Studio US Government plans are available to:

- (1) US federal, state, local, tribal, and territorial government entities, and
- (2) other entities, which handle data that is subject to government regulations and requirements and where use of Copilot Studio US Government plans is appropriate to meet these requirements, subject to validation of eligibility.

Microsoft's validation of eligibility includes:

- Confirmation of handling data subject to ITAR
- Law enforcement data subject to the Federal Bureau of Investigation's (FBI) Criminal Justice Information Services (CJIS) Policy
- Other government-regulated or controlled data

Validation may require sponsorship by a government entity with specific requirements for the handling of data.

Entities with questions about eligibility for Copilot Studio US Government should consult their account team. Microsoft revalidates eligibility when it renews customer contracts for Copilot Studio US Government plans.

## Differences between customer data and customer content

Customer data, as defined in the [Online Service Terms](#), means all data provided to Microsoft by, or on behalf of, customers using an online service. This includes all text, sound, video, image files, and software.

Customer content refers to a specific subset of customer data that has been directly created by users. This could include for example content stored in databases through entries in the [Dataverse](#) entities (for example, contact information). Content is generally considered confidential information, and in normal service operations, isn't sent through the Internet without encryption.

For more information on how Copilot Studio protects customer data, see the [Microsoft Online Services Trust Center](#).

## Data segregation for Government Community Cloud

When provisioned as part of Copilot Studio US Government plans, the Copilot Studio service is offered in accordance with the National Institute of Standards and Technology (NIST).

In addition to the logical separation of customer content at the application layer, the Copilot Studio US Government service provides your organization with a secondary layer of physical segregation for customer content. This segregation is achieved by using infrastructure that is separate from the infrastructure used for commercial Copilot Studio customers. This type of usage includes using Azure services in Azure's Government Cloud. To learn more, see [Azure Government](#).

## Customer content located within the United States

The Copilot Studio US Government service runs in datacenters physically located in the United States. It stores customer content at rest in datacenters physically located only in the United States.

## Restricted data access by administrators

Access to Copilot Studio US Government customer content by Microsoft administrators is restricted to personnel who are US citizens. These personnel undergo background investigations in accordance with relevant government standards.

Copilot Studio support and service engineering staff don't have standing access to customer content hosted in the Copilot Studio US Government service. Any staff who requests temporary permission elevation which would grant access to customer content must first have passed the following background checks.

[ ] [Expand table](#)

Microsoft Personnel	Description
<b>Screening and Background Checks</b> <sup>1</sup>	
U.S. citizenship	Verification of U.S. citizenship
Employment history check	Verification of seven (7) year employment history
Education verification	Verification of highest degree attained
Social Security number (SSN) search	Verification that the SSN the employees provides is valid
Criminal history check	A seven (7) year criminal record check for felony and misdemeanor offenses at the state, county, and local level and at the federal level

<b>Microsoft Personnel</b>	<b>Description</b>
<b>Screening and Background Checks</b> <sup>1</sup>	
Office of Foreign Assets Control list (OFAC)	Validation against the Department of Treasury list of groups with whom U.S. persons aren't allowed to engage in trade or financial transactions
Bureau of Industry and Security list (BIS)	Validation against the Department of Commerce list of individuals and entities barred from engaging in export activities
Office of Defense Trade Controls Debarred Persons list (DDTC)	Validation against the Department of State list of individuals and entities barred from engaging in export activities related to the defense industry
Fingerprinting check	Fingerprint background check against FBI databases
CJIS background screening	State-adjudicated review of federal and state criminal history by state CSA appointed authority within each state that has signed up for the Microsoft CJIS IA program
Department of Defense IT-2	Staff who request elevated permissions to customer data or privileged administrative access to DoD SRG L5 service capacities must pass DoD IT-2 adjudication, based on a successful OPM Tier 3 investigation.

- <sup>1</sup>. Applies only to personnel with temporary or standing access to customer content hosted in Copilot Studio US Government (GCC and GCC High)

## Certification and accreditation

Copilot Studio US Government plans are designed to support the Federal Risk and Authorization Management Program (FedRAMP) accreditation at a High Impact level. FedRAMP artifacts are available for review by federal customers who are required to comply with FedRAMP. Federal agencies can peruse these artifacts in support of their review to grant an Authority to Operate (ATO).

 **Note**

Copilot Studio is authorized as a service within the Azure Government FedRAMP ATO.

For more information, including how to access the FedRAMP documents, review the [FedRAMP Marketplace](#).

Copilot Studio US Government plans have features designed to support customers' CJIS Policy requirements for law enforcement agencies.

## Copilot Studio US Government and other Microsoft services

Copilot Studio US Government plans include several features that allow users to connect to, and integrate with, other Microsoft enterprise service offerings such as Power Apps and Power Automate US Government.

Copilot Studio US Government services run within Microsoft datacenters in a manner consistent with a multitenant, public cloud deployment model. However, client applications are limited to the web-user client and aren't available in Microsoft Teams. Government customers are responsible for managing client applications.

Copilot Studio US Government plans use the Office 365 customer administrator UI for customer administration and billing.

The Copilot Studio US Government service maintains the actual resources, information flow, and data management. For purposes of FedRAMP ATO inheritance, Copilot Studio US Government plans use Azure (including Azure for Government) ATOs for infrastructure and platform services, respectively.

If you adopt the use of Active Directory Federation Services (ADFS) 2.0 and set up policies to help ensure your users connect to the services through single sign-on, any temporarily cached customer content will be in the United States.

## Copilot Studio US Government and third-party services

Copilot Studio US Government plans provide the ability to integrate third-party applications into the service through Power Automate Cloud Flow, which uses [Connectors](#) and [Skills](#). These third-party applications and services might involve storing, transmitting, and processing your organization's customer data on third-party systems that are outside of the Copilot Studio US Government infrastructure. As a result, these third-party applications and services aren't covered by the Copilot Studio US Government compliance and data protection commitments.

### Important

Review the privacy and compliance statements provided by the third parties when assessing the appropriate use of these services for your organization.

[Governance considerations](#) can help your organization bring awareness about the capabilities available across several related themes, such as architecture, security, alert and action, and monitoring.

## Copilot Studio US Government and Azure Services

The Copilot Studio US Government services are deployed to Microsoft Azure Government. Microsoft Entra ID isn't part of the Copilot Studio US Government accreditation boundary. However, the services rely on a customer's [Microsoft Entra ID](#) tenant for customer tenant and identity functions. This includes:

- Authentication
- Federated authentication
- Licensing

When a user of an organization employing ADFS attempts to access the Copilot Studio US Government service, the user is redirected to a login page hosted on the organization's ADFS server.

The user provides credentials to their organization's ADFS server. The organization's ADFS server attempts to authenticate the credentials using the organization's Active Directory infrastructure.

If authentication is successful, the organization's ADFS server issues a SAML (Security Assertion Markup Language) ticket that contains information about the user's identity and group membership.

The customer's ADFS server signs this ticket using one half of an asymmetric key pair and then sends the ticket to Microsoft Entra ID via encrypted TLS (Transport Layer Security). Microsoft Entra ID validates the signature using the other half of the asymmetric key pair and then grants access based on the ticket.

The user's identity and group membership information remain encrypted in Microsoft Entra ID. In other words, only limited user-identifiable information is stored in Microsoft Entra ID.

You can find full details of the Microsoft Entra ID security architecture and control implementation in the Azure System Security Plan (SSP).

The Microsoft Entra ID account management services are hosted on physical servers managed by the Microsoft Global Foundation Services (GFS). Network access to these servers is controlled by GFS-managed network devices using rules set by Azure. Users don't interact directly with Microsoft Entra ID.

## Microsoft Copilot Studio US Government service URLs

You use a different set of URLs to access Copilot Studio US Government environments, as shown in the following table. The table also includes the commercial URLs for contextual reference.

[+] Expand table

Commercial	US Government (GCC)	US Government (GCC High)
copilotstudio.microsoft.com	gcc.powerva.microsoft.us	high.powerva.microsoft.us
flow.microsoft.com	gov.flow.microsoft.us	high.flow.microsoft.us
make.powerapps.com	make.gov.powerapps.us	make.high.powerapps.us
flow.microsoft.com/connectors	gov.flow.microsoft.us/connectors	high.flow.microsoft.us/connectors
admin.powerplatform.microsoft.com	gcc.admin.powerplatform.microsoft.us	high.admin.powerplatform.microsoft.us
api.powerva.microsoft.com	gcc.api.powerva.microsoft.us	high.api.powerva.microsoft.us

For those customers that implement network restrictions, ensure access to the following domains is made available to your users' access points:

## GCC customers

- *.azure.net*
- *.azure.us*
- *.azure-apihub.us*
- *.azureedge.net*
- *.crm9.dynamics.com*
- *.microsoft.com*
- *.microsoft.us*
- *.microsoftonline.com*
- *.usgovcloudapi.net*
- *.windows.net*

Refer to the [IP ranges](#) for *AzureCloud.usgovtexas* and *AzureCloud.usgovvirginia* to enable access to Dataverse instances that users and administrators may create within your tenant.

## Connectivity between Copilot Studio US Government and Public Azure Cloud services

Azure is distributed among multiple clouds. By default, tenants are allowed to open firewall rules to a cloud-specific instance, but cross-cloud networking is different and requires opening specific firewall rules to communicate between services. If you're a Copilot Studio customer and you have existing SQL instances in Azure public cloud which you need to access, you must open specific firewall ports in SQL to the Azure Government Cloud IP space for the following datacenters:

- USGov Virginia
- USGov Texas

Refer to the [Azure IP Ranges and Service Tags - US Government Cloud](#) document, focusing attention on *AzureCloud.usgovtexas*, and *AzureCloud.usgovvirginia*. Also note that these are the IP ranges required for your users to have access to the service URLs.

## Copilot Studio US Government feature limitations

Some of the features available in the commercial version of Copilot Studio aren't available to Copilot Studio US Government customers. The Copilot Studio team is actively working on making these features available to US Government customers and will update this article when these features become available.

 Note

Currently, the only way to approve an agent for Teams is to submit the agent to an admin for approval.

[+] Expand table

Feature or capability	Available in GCC	Available in GCC High
Copilot Studio Microsoft Teams app experience	No	No
Teams channel in the Copilot Studio web app	Yes	No
Transfer to agents	Yes	No
Teams & M365 Copilot Channel	No	No
Triggers / Autonomous Agents	No	No
Generative Orchestration	Yes	No
Copilot Agents extend M365	No	No
Generative Answers Enhanced Search	No	No
Prompt Action	Yes	No
Preview Models for answer generation	No	No
Azure AI Search as a knowledge source	No	No

## Requesting support

Having a problem with your service? You can create a support request to get the issue resolved.

More information: [Contact Technical Support](#)

# Manage extensions for Microsoft 365 Copilot for Sales (preview)

10/21/2025

[This article is prerelease documentation and is subject to change.]

Copilot Studio can be used to extend Microsoft 365 Copilot for Sales with actions or plugins. These plugins can be either developed within your enterprise or sourced from Microsoft and other partners.

For more information on connector plugins, see [Agent connectors, actions, and extensions overview \(preview\)](#).

As a Microsoft 365 Copilot for Sales administrator, you can manage plugins for Microsoft 365 Copilot for Sales within your CRM instance by enabling or disabling specific plugins from Copilot Studio.

## Note

By enabling a connector action for Microsoft 365 Copilot for Sales, you may be allowing Microsoft 365 Copilot for Sales users on Microsoft 365 and Teams applications to send and receive data from external sources using Microsoft 365 Copilot for Sales. Data transfer may occur even if the same connector action has been disallowed for use directly with Microsoft 365. You should ensure that this action complies with your organization policies before enabling it.

The data and insights that connector actions bring to Microsoft 365 Copilot for Sales experiences are powered by third parties and may be subject to third party terms and conditions and/or privacy policies. We recommend you validate connector actions for compliance with your organization policies.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

# Who can manage plugins for Microsoft 365 Copilot for Sales?

To manage plugins for Microsoft 365 Copilot for Sales in Copilot Studio, you must be:

- Assigned as a Microsoft 365 Copilot for Sales administrator. For more information, see [Microsoft 365 Copilot for Sales admin settings](#).
- Given either of the following combination of roles in Dataverse:
  - Both `Microsoft Copilot administrator` and `Environment maker`
  - `System customizer`

For Salesforce administrators, these roles can be given for the `msdyn_viva` Power Platform environment. For more information, see [Assign a security role to a user](#).

## Enable a plugin for Microsoft 365 Copilot for Sales

1. Sign in to Copilot Studio.
2. Select the right Power Platform environment for your CRM instance:
  - For Dynamics 365 CRM, refer to [Find your environment and organization ID and name](#).
  - For Salesforce CRM, you can use `msdyn_viva`.
3. Select **Copilot for Sales > Manage**. Here you can see all the plugins that are enabled in this environment.
4. Select **Enable Plugins**.
5. Find and select the plugin you want to enable. You can review the information about the plugin and choose to enable it.
6. Use the **Enable plugin** toggle to turn on the plugin.
7. Select the users you want to enable the plugin for. You can use one of the three options:
  - Only agent administrators (default)
  - Entire organization
  - Specific users/groups

 **Note**

If a plugin is enabled, it is automatically enabled for all agent administrators.

8. Select **Save**. This enables the plugin for the users you selected. You also need to:

- [Create and assign security role to user](#)

The plugin is now enabled for the users.

## Assign security role to user

Once you enable a plugin for a user, you must add appropriate security permissions for the user to be able to use the plugin. Assign the role *Microsoft Copilot User* to the user. For more information, see [Configure user security in an environment](#).

## Disable a plugin for Microsoft 365 Copilot for Sales

1. Sign in to Copilot Studio.

2. Select the right Power Platform environment for your CRM instance:

- For Dynamics 365 CRM, refer to [Find your environment and organization ID and name](#).
- For Salesforce CRM, you can use *msdyn\_viva*.

3. Select **Copilot for Sales > Manage**. Here you can see all the plugins that are enabled in this environment.

4. Select the plugin you want to disable.

5. Use the **Enable plugin** toggle to turn off the plugin.

6. Select **Save**.

This disables the plugin in Microsoft 365 Copilot for Sales. It might take a few minutes for the changes to take effect.

## Related content

- [Agent connectors, actions, and extensions overview \(preview\)](#)
- [Extend Microsoft 365 Copilot for Sales with partner applications](#)

# Configure user authentication in Copilot Studio

06/03/2025

Authentication allows users to sign in, giving your agent access to a restricted resource or information. Users can sign in with [Microsoft Entra ID](#), or with any [OAuth2 identity provider](#) such as Google or Facebook.

## !Note

In Microsoft Teams, you can configure a Copilot Studio agent to provide authentication capabilities, so that users can sign in with a Microsoft Entra ID or any [OAuth2 identity provider](#), such as a Microsoft or Facebook account.

You can [add user authentication to topics](#) when you edit a topic.

## !Important

Changes to the authentication configuration only take effect after you publish your agent. Make sure to plan ahead before you make authentication changes to your agent.

## Choose an authentication option

Copilot Studio supports several authentication options. Choose the one that meets your needs.

1. Go to **Settings** for your agent, and select **Security**.
2. Select **Authentication**.

The following authentication options are available:

- [No authentication](#)
- [Authenticate with Microsoft](#)
- [Authenticate manually](#)

3. Select **Save**.

## No authentication

No authentication means your agent doesn't require your users to sign in when interacting with the agent. An unauthenticated configuration means your agent can only access public information and resources. Classic chatbots are configured by default to *not* require authentication.

 **Caution**

Selecting the **No authentication** option allows anyone who has the link to chat and interact with your bot or agent.

We recommend you apply authentication, especially if you are using your bot or agent within your organization or for specific users, along with [other security and governance controls](#).

## Authenticate with Microsoft

 **Important**

When the **Authenticate with Microsoft** option is selected, all channels except the **Teams + Microsoft 365** channel are disabled.

Additionally, the **Authenticate with Microsoft** option isn't available for agents that are integrated with [Dynamics 365 Customer Service](#).

This configuration automatically sets up Microsoft Entra ID authentication for Teams without the need for any manual configuration. Since Teams authentication itself identifies the user, users aren't prompted to sign in while they're in Teams, unless your agent requires an expanded scope.

Only the Teams channel is available if you select this option. If you need to publish your agent to other channels but still want authentication for your agent, choose **Authenticate manually**.

If you select **Authenticate with Microsoft**, the following variables are available in the authoring canvas:

- `User.ID`
- `User.DisplayName`

For more information about these variables and how to use them, see [Add user authentication to topics](#).

`User.AccessToken` and `User.IsLoggedIn` variables aren't available with this option. If you need an authentication token, use the **Authenticate manually** option.

If you change from **Authenticate manually** to **Authenticate with Microsoft**, and your topics contain the variables `User.AccessToken` or `User.IsLoggedIn`, they're displayed as **Unknown** variables after the change. Make sure to correct any topics with errors before you publish your agent.

## Authenticate manually

Copilot Studio supports the following authentication providers under **Authenticate manually** option:

- Microsoft Entra ID V2 with federated credentials
- Microsoft Entra ID V2 with certificates
- Microsoft Entra ID V2 with client secrets
- Microsoft Entra ID
- Generic OAuth 2 - Any identity provider that complies with the [OAuth2 standard](#)

The following variables are available in the authoring canvas after you configure manual authentication:

- `User.Id`
- `User.DisplayName`
- `User.AccessToken`
- `User.IsLoggedIn`

For more information about these variables and how to use them, see [Add user authentication to topics](#).

Once the configuration is saved, make sure to publish your agent so the changes take effect.

### (!) Note

- Authentication changes only take effect after the agent is published.
- This setting can be controlled by the corresponding admin control in Power Platform. When the control is enabled, it prevents the **Authenticate manually** option from being enabled or disabled within Copilot Studio. The control is always enabled, and the **Authenticate manually** option can't be modified in Copilot Studio.

# Required user sign in and agent sharing

Require users to sign in determines whether a user needs to sign in before talking with the agent. We highly recommend that you turn on this setting for agents that need to access sensitive or restricted information.

This option isn't available for the **No authentication** and **Authenticate with Microsoft** options.

## ⓘ Note

This option is also not configurable when DLP policy in the Power Platform admin center is configured to require authentication. For more information, see [Data loss prevention example - Require user authentication in agents](#).

If you turn off this option, your agent doesn't ask users to sign in until it encounters a topic that requires them to.

When you turn on this option, it creates a system topic called **Require users to sign in**. This topic is only relevant for the **Authenticate manually** setting. Users are always authenticated on Teams.

The **Require users to sign in** topic is automatically triggered for any user who talks to the agent without being authenticated. If the user fails to sign in, the topic redirects to the **Escalate** system topic.

The topic is read-only and can't be customized. To see it, select [Go to the authoring canvas](#).

## Control who can chat with the agent in the organization

Your agent's authentication and **Require user to sign in** setting in combination determines whether you can [share the agent](#) to control who in your organization can chat with it. The authentication setting doesn't affect sharing an agent for collaboration.

- **No authentication:** Any user who has a link to the agent (or can find it; for example, on your website) can chat with it. You can't control which users in your organization can chat with the agent.
- **Authenticate with Microsoft:** The agent works only on [the Teams channel](#). Since the user is always signed in, the **Require users to sign in** setting is turned on and can't be turned off. You can use agent sharing to control who in your organization can chat with the agent.
- **Authenticate manually:**

- If the service provider is either **Azure Active Directory** or **Microsoft Entra ID**, you can turn on **Require users to sign in** to control who in your organization can chat with the agent using agent sharing.
- If the service provider is **Generic OAuth2**, you can turn **Require users to sign in** on or off. When turned on, a user who signs in can chat with the agent. You can't control which specific users in your organization can chat with the agent using agent sharing.

When an agent's authentication setting can't control who can chat with it, if you select **Share** on the agent's overview page a message informs you that anyone can chat with your agent.

## Manual authentication fields

The following are all the fields you can see when you're configuring manual authentication. Which fields you see depends on your choice for service provider.

[Expand table](#)

Field name	Description
Authorization URL template	The URL template for authorization, as defined by your identity provider. For example, <a href="https://login.microsoftonline.com/common/oauth2/v2.0/authorize">https://login.microsoftonline.com/common/oauth2/v2.0/authorize</a>
Authorization URL query string template	The query template for authorization, as provided by your identity provider. Keys in the query string template vary, depending on the identity provider ( <code>?client_id={ClientId}&amp;response_type=code&amp;redirect_uri={RedirectUrl}&amp;scope={Scopes}&amp;state={State}</code> ).
Client ID	Your client ID, obtained from the identity provider.
Client secret	Your client secret, obtained when you created the identity provider app registration.
Refresh body template	The template for the refresh body ( <code>refresh_token={RefreshToken}&amp;redirect_uri={RedirectUrl}&amp;grant_type=refresh_token&amp;client_id={ClientId}&amp;client_secret={ClientSecret}</code> ).
Refresh URL query string template	The refresh URL query string separator for the token URL, usually a question mark (?).
Refresh URL template	The URL template for refresh; for example, <a href="https://login.microsoftonline.com/common/oauth2/v2.0/token">https://login.microsoftonline.com/common/oauth2/v2.0/token</a> .
Scope list delimiter	The separator character for the scope list. Empty spaces aren't supported in this field. <sup>1</sup>
Scopes	The list of <a href="#">scopes</a> that you want users to have after they sign in. Use the <b>Scope list delimiter</b> to separate multiple scopes. <sup>1</sup> Only set necessary scopes and follow the <a href="#">least</a>

Field name	Description
	privilege access control principle.
Service provider	The service provider you want to use for authentication. For more information, see <a href="#">OAuth generic providers</a> .
Tenant ID	Your Microsoft Entra ID tenant ID. Refer to <a href="#">Use an existing Microsoft Entra ID tenant</a> to learn how to find your tenant ID.
Token body template	The template for the token body. (code= <code>{Code}&amp;grant_type=authorization_code&amp;redirect_uri={RedirectUrl}&amp;client_id={ClientId}&amp;client_secret={ClientSecret}</code> )
Token exchange URL (required for SSO)	This optional field is used when you're <a href="#">configuring single sign-on</a> .
Token URL template	The URL template for tokens, as provided by your identity provider; for example, <code>https://login.microsoftonline.com/common/oauth2/v2.0/token</code> .
Token URL query string template	The query string separator for the token URL, usually a question mark (?).

<sup>1</sup> You can use spaces in the **Scopes** field if the identity provider requires it. In that case, enter a comma (,) in **Scope list delimiter**, and enter spaces in the **Scopes** field.

## Turn off authentication

1. With your agent open, select **Settings** on the top menu bar.
2. Select **Security**, then select **Authentication**.
3. Select **No authentication**.

If authentication variables are used in a topic, they become **Unknown** variables. Go to the [Topics](#) page to see which topics have errors and fix them before publishing.

4. Publish the agent.

**ⓘ Important**

If your agent has *tools* configured to [require user credentials](#), don't turn off authentication at the agent level, since this would prevent these tools from working.

# Configure user authentication with Microsoft Entra ID

06/09/2025

Adding authentication to your agent allows users to sign in, giving your agent access to a restricted resource or information.

This article covers how to configure Microsoft Entra ID as your service provider. To learn about other service providers and user authentication in general, see [Configure user authentication in Copilot Studio](#).

If you have tenant administration rights, you can [configure API permissions](#). Otherwise, you need to ask a tenant administrator to do it for you.

## Prerequisites

[Learn how to add user authentication to a topic](#)

You complete the first several steps in the Azure portal, and complete the final two steps in Copilot Studio.

## Create an app registration

1. Sign in to the [Azure portal](#), using an admin account in the same tenant as your agent.
2. Go to **App registrations**.
3. Select **New registration** and enter a name for the registration. **Don't** alter existing app registrations.  
It can be helpful later to use the name of your agent. For example, if your agent is called "Contoso sales help," you might name the app registration "ContosoSalesReg."
4. Under **Supported account types**, select **Accounts in this organizational directory only (Contoso only - Single tenant)**.
5. Leave the **Redirect URI** section blank for now. Enter that information in the next steps.
6. Select **Register**.
7. After the registration is complete, go to **Overview**.
8. Copy the **Application (client) ID** and paste it in a temporary file. You need it in later steps.

## Add the redirect URL

1. Under **Manage**, select **Authentication**.
2. Under **Platform configurations**, select **Add a platform**, and then select **Web**.
3. Under **Redirect URIs**, enter `https://token.botframework.com/.auth/web/redirect` or `https://europe.token.botframework.com/.auth/web/redirect` for Europe. You can also copy the URI from **Redirect URL** textbox under Copilot Studio **Security** setting page under **Authenticate manually**.

This action takes you back to the **Platform configurations** page.

4. Select both **Access tokens (used for implicit flows)** and **ID tokens (used for implicit and hybrid flows)**.
5. Select **Configure**.

## Configure manual authentication

Next, configure manual authentication. You can choose from multiple options for your provider, however we recommend you use Microsoft Entra ID V2 with federated credentials. You can also use client secrets if you're unable to use federated credentials.

Configure federated credentials (recommended)

### Configure manual authentication using federated credentials

1. In Copilot Studio, go to **Settings** for your agent, and select **Security**.
2. Select **Authentication**.
3. Select **Authenticate manually**.
4. Leave **Require users to sign in on**.
5. Enter the following values for the properties:
  - **Service provider:** Select **Microsoft Entra ID V2 with federated credentials**.
  - **Client ID:** Enter the application (client) ID that you copied earlier from the Azure portal.
6. Select **Save** to see the federated credential issuer and value.

7. Copy the **Federated credential issuer** and **Federated credential value** and paste it in a temporary file. You need it in later steps.
8. Go to the Azure portal and the app registration you previously created. Under **Manage**, select **Certificates & secrets** and then **Federated credentials**.
9. Select **Add credential**.
10. Under **Federated credential scenario**, select **Other issuer**.
11. Enter the following values for the properties:
  - **Issuer**: Enter the federated credential issuer value that you copied earlier from Copilot Studio.
  - **Value**: Enter the federated credential value data that you copied earlier from Copilot Studio.
  - **Name**: Provide a name.
12. Select **Add** to finish the configuration.

## Configure API permissions

1. Go to **API permissions**.
2. Select **Grant admin consent for <your tenant name>**, and then select **Yes**. If the button isn't available, you might need to ask a tenant administrator to do enter it for you.

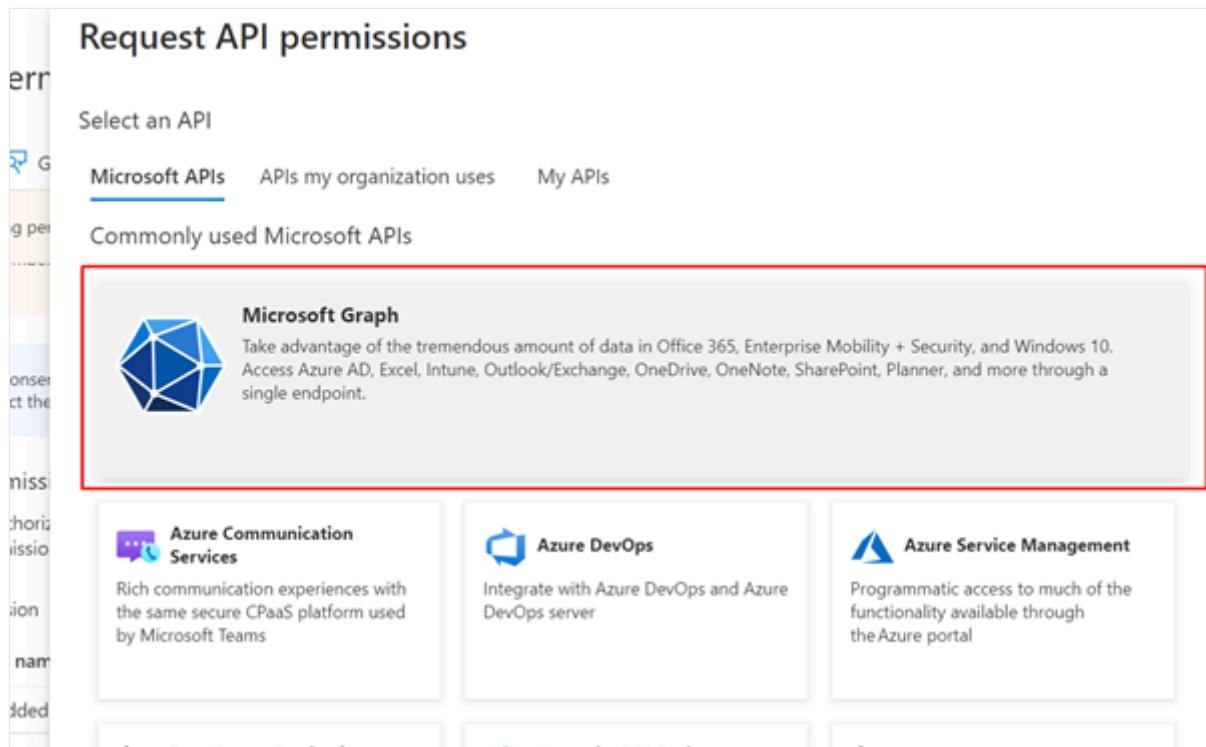
The screenshot shows the 'API permissions' section of the Azure portal. At the top, there's a 'Refresh' button. Below it, a heading says 'Configured permissions' with a note: 'Applications are authorized to call APIs when they are granted permissions by users/admins as part of all the permissions the application needs.' There are two buttons: '+ Add a permission' and 'Grant admin consent for Contoso', with the latter being highlighted by a red box. A table below lists the configured permissions, showing 'Microsoft Graph (1)' with a 'User.Read' entry. The table has columns for 'API / Permissions name', 'Type', and 'Description'.

API / Permissions name	Type	Description
Microsoft Graph (1)		
User.Read	Delegated	Sign in and read user profile

## Important

To avoid users having to consent to each application, someone assigned at least the Application Administrator or Cloud Application Administrator role can [grant tenant-wide consent](#) to your application registrations.

3. Select **Add a permission**, and then select **Microsoft Graph**.



The screenshot shows the 'Request API permissions' page. At the top, there's a search bar labeled 'Select an API'. Below it, three tabs are visible: 'Microsoft APIs' (which is selected), 'APIs my organization uses', and 'My APIs'. A section titled 'Commonly used Microsoft APIs' contains several cards. The 'Microsoft Graph' card is highlighted with a red box; it features a blue geometric icon, the text 'Microsoft Graph', and a description: 'Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.' Other cards shown include 'Azure Communication Services' (with a description about rich communication experiences), 'Azure DevOps' (with a description about integrating with Azure DevOps and DevOps server), and 'Azure Service Management' (with a description about programmatic access through the Azure portal).

4. Select **Delegated permissions**.

# Request API permissions

<https://graph.microsoft.com/> Docs

What type of permissions does your application require?

## Delegated permissions

Your application needs to access the API as the signed-in user.

## Select permissions

Start typing a permission to filter these results

### Permission

#### OpenId permissions

email ⓘ

View users' email address

offline\_access ⓘ

Add permissions

Discard

5. Expand OpenId permissions and turn on openid and profile.

## Request API permissions

### Select permissions

Start typing a permission to filter these results

**i** The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)

Permission	Admin consent required
✓ OpenId permissions (2)	
<input type="checkbox"/> email ⓘ View users' email address	No
<input type="checkbox"/> offline_access ⓘ Maintain access to data you have given it access to	No
<input checked="" type="checkbox"/> openid ⓘ Sign users in	No
<input checked="" type="checkbox"/> profile ⓘ View users' basic profile	No

### 6. Select Add permissions.

## Define a custom scope for your agent

Scopes allow you to determine user and admin roles and access rights. You create a custom scope for the canvas app registration that you create in a later step.

### 1. Go to Expose an API and select Add a scope.

The screenshot shows the Azure portal interface for managing an API. On the left, there's a sidebar with various navigation options: Overview, Quickstart, Integration assistant (preview), Manage, Branding, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API (which is highlighted with a red box), and Owners. The main content area is titled 'Scopes defined by this API'. It contains a sub-section 'Scopes' with the message 'No scopes have been defined'. Below this, there's another section 'Authorized client applications' with the message 'Authorizing a client application indicates that this API ...'. At the top of the main content area, there's a 'Set' button next to 'Application ID URI'. A red box highlights the '+ Add a scope' button in the 'Scopes' section.

### 2. Set the following properties. You can leave the other properties blank.

Property	Value
Scope name	Enter a name that makes sense in your environment, such as Test.Read
Who can consent?	Select <b>Admins and users</b>
Admin consent display name	Enter a name that makes sense in your environment, such as Test.Read
Admin consent description	Enter Allows the app to sign the user in.
State	Select <b>Enabled</b>

3. Select **Add scope**.

## Configure authentication in Copilot Studio

1. In Copilot Studio, under **Settings**, select **Security > Authentication**.
2. Select **Authenticate manually**.
3. Leave **Require users to sign in on**.
4. Select a **Service provider** and provide the required values. See [Configure manual authentication in Copilot Studio](#).
5. Select **Save**.

### 💡 Tip

The token exchange URL is used to exchange the On-Behalf-Of (OBO) token for the requested access token. For more information, see [Configure single sign-on with Microsoft Entra ID](#).

### ❗ Note

The scopes should include `profile openid` and the following, depending on your use case:

- `Sites.Read.All Files.Read.All` for SharePoint
- `ExternalItem.Read.All` for Graph Connection
- `https://[OrgURL]/user_impersonation` for Dataverse structured data

For example, Dataverse Structure Data should have the following scopes: `profile openid`

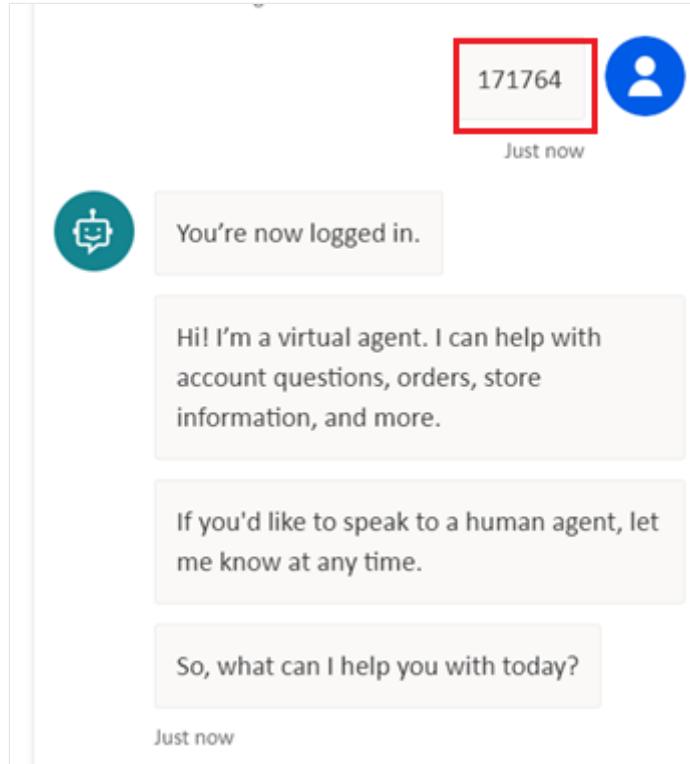
`Sites.Read.All Files.Read.All https://myorg123.com/user_impersonation`

## Test your agent

1. Publish your agent.
2. In the **Test your agent** panel, send a message to your agent.
3. When the agent responds, select **Login**.

A new browser tab opens, asking you to sign in.

4. Sign in, and then copy the displayed validation code.
5. Paste the code in the agent chat to complete the sign-in process.



# Control maker-provided credentials for authentication

09/03/2025

Controlling the use of maker-provided credentials is a governance feature in Microsoft Copilot Studio that allows administrators to determine how makers can authenticate tools they add to agents.

When configuring an agent, [a maker could add a tool that requires authentication](#) to another service (like a connector or Power Automate flow) using their personal credentials. When someone uses the agent, the agent uses the maker's credentials, not the end user's credentials, to authenticate with connected services. Using maker credentials could lead to oversharing of data or capabilities - for example, an end user might retrieve information or perform actions that only the maker's account is permitted to do.

By configuring how makers can authenticate tools, you can help prevent damaging actions because each end user only has access to what their own account allows.

The **Control maker credential options** feature enables this by letting you control how a maker can authenticate tools in an agent. You choose whether makers can select to use their own credentials for authenticating tool connections, to use the end-user's credentials, or to have access to both.

Enforcing the use of end-user credentials prompts the end user to sign in (to the relevant service or connector) when needed. No stored maker credentials are used at run time, aligning the agent's behavior with the end user's actual permissions.

Admins can enable and disable the selection of maker-provided credentials in the [Power Platform admin center](#), as described in this article.

## ✖ Caution

By default, both **End-user** and **Maker-provided** credentials are enabled.

## Getting Started

## ⚠ Warning

When configured as instructed, the feature **immediately** turns off the maker's ability to select their credentials for authenticating tools within the environment or environment

group. All tools for all agents in the impacted environments instantly change to require end-user credentials at runtime.

You should prepare your makers and users for this change, as any new conversations with the impacted agents prompts the agent user for their sign-in details for the connected service.

## Prerequisites

- [Power Platform admin permissions](#) (Environment Admin for a single environment, or Power Platform Administrator/Global Admin for environment groups).

## Prevent the use of maker-provided credentials

Choose whether end-user or maker credentials (or both) can be selected by makers for agents in an environment, or for all agents in all environments in an environment group.

### Tip

If the environment you want to configure is part of an environment group, you must be a tenant admin with access to the group.

You can't configure this feature on the individual details page for environments that are in a group, you must use the group's details page.

1. Go to the [Power Platform admin center](#) and sign in with [an admin account](#).
2. On the side navigation menu, select **Manage**, then **Environments** or **Environment groups**. In the list that appears, select the name of the environment or environment group you want to configure. The details page for the environment or group opens.
3. In the environment's details page, select **Settings** on the top menu bar. Expand the **Product** section, and select **Features**.
4. Scroll to the **Copilot Studio agents** section, and under **Control maker credential options** select **End-user credentials**, **Maker-provided credentials**, or both.

Power Platform admin center

Search for settings, pages, and more

your tenant preferred data location for Viva Insights is different than the location of your Copilot Studio environment

**Control maker credential options**

By default, end users' own credentials are used to sign into agent processes like connectors and actions. For more seamless operation, you can allow makers to provide credentials instead. In this case, makers should ensure the credentials include only the permissions that are OK to share. [Learn more](#)

End-user credentials (default)

Maker-provided credentials

This environment is part of a group and this setting is managed there. If you are a tenant admin, go there to change the setting.

Preview and experimental AI models

### 💡 Tip

If the environment is in a group, the options are unavailable and a message directs you to configure the setting for the group, not the individual environment.

5. Select **Save** at the bottom of the settings page to apply the setting.

The update might take a minute to propagate. Once active, all existing and new agents in that environment adhere to this rule, and the types of authentication options for makers change in Copilot Studio.

## Scope of enforcement and experience

### ✖ Caution

#### Impact on autonomous agents

When maker-provided credentials are prevented from being used, agents require real-time user interaction because each tool call must be authenticated with a live user sign-in. As a result, agents triggered by scheduled or autonomous events, or that attempt to run in the background, fail due to missing credentials. All agent triggers must involve an active user.

## Maker experience

The Copilot Studio authoring UI automatically reflects this policy. Any toggle or dropdown for authentication methods have maker-provided credentials disabled or hidden. The maker sees

that only end-user (or maker) authentication can be chosen. This notification might be labeled simply as "End user credentials" in the UI. If the maker previously had a tool configured with their credentials, they might be prompted to change it before publishing the agent.

## End-user experience

When an end user interacts with an agent (for example, in Teams or on a website) and triggers a tool action, the agent prompts that user to sign in if they haven't already. The prompt could be a login card or link. Once the user signs in with their own account for the required service, the agent proceeds with the action using the user's credentials. If the user is already signed in (for example, their Microsoft 365 or Teams account is also authorized for the needed service), the agent might use that existing authentication session. The action runs under the end-user's identity. If the user lacks permission for something, the agent is unable to do it on their behalf - by design.

## Power Automate flows

The control over the authentication type covers connectors, built-in actions, and embedded Power Automate flows equally. A Power Automate flow as a tool in the agent also requires each user to sign in for any connections that flow uses.

## Scope of enforcement

The policy is applied per environment (or environment group). If an environment is part of a managed group where the policy is enabled, you can't individually disable it for one environment in that group - environment group settings override any environment settings.

## Next steps

As with all security features, this feature should form part of your defense-in-depth strategy. For example, you could:

- Use credential restriction in sensitive or production environments where agents are shared with other end users and data security is paramount - for example, production agents that access your organization's internal systems, like Microsoft 365. Using credential restriction in these environments ensures that only authorized users (by virtue of their own credentials) can execute sensitive operations via the agent.
- Combine credential restriction [with agent sharing controls](#) for even tighter security. By also preventing makers from freely sharing agents (or by limiting who can use certain agents), you reduce the risk that a maker could, for example, share an agent to someone

who shouldn't have access. Control Maker Credential Options ensures credentials aren't inappropriately shared, and also prevents any type of stored credentials, including API keys.

## Related information

- [Configure end-user authentication in Copilot Studio](#)
- [Security and governance settings in Copilot Studio](#)
- [Microsoft 365 Tek Blog Hands-on Tutorial: Block Maker-Provided Authentication ↗](#)

# Configure web and Direct Line channel security

09/12/2025

When you create a Copilot Studio agent, it's immediately available in the **Demo website** and **Custom website** channels to anyone who knows the agent ID. These channels are available by default, and no configuration is needed.

For the Microsoft Teams app, you can configure advanced web channel security options.

## (!) Note

If you have a Teams-only license, you can't generate secrets to enable secure access.

Secure access tokens are created automatically for you and secure access is enabled by default.

Users can find the agent ID directly from within Copilot Studio or by receiving it from someone. But, depending on the agent's capability and sensitivity that might not be desirable.

With Direct Line-based security, you can enable access only to locations that you control by enabling secured access with Direct Line secrets or tokens.

You can also swap and regenerate secrets and refresh tokens, and you can easily disable secured access if you no longer want to use it.

## (!) Note

Copilot Studio uses the Bot Framework Direct Line channel to connect your web page or app to the agent.

## Enable or disable web channel security

You can enforce the use of secrets and tokens for each individual agent.

Once this option is enabled, channels need the client to authenticate their requests either by using a secret or by using a token that is generated by using the secret, obtained at runtime.

Attempts to access the agent that don't provide this security measure don't work.

1. Go to the **Settings** page for your agent, select **Security**, then select **Web channel security**.

Settings X

- ❖ Generative AI
- ❖ Agent details
- Security**
- ❖ Connection Settings
- ❖ Authoring Canvas
- ab Entities
- Skills

**Security**

**Authentication**  
Verify a user's identity during a chat.

**Web channel security**    
Review other enhanced security options.

**Allowlist**  
Let other agents call your agent as a skill.

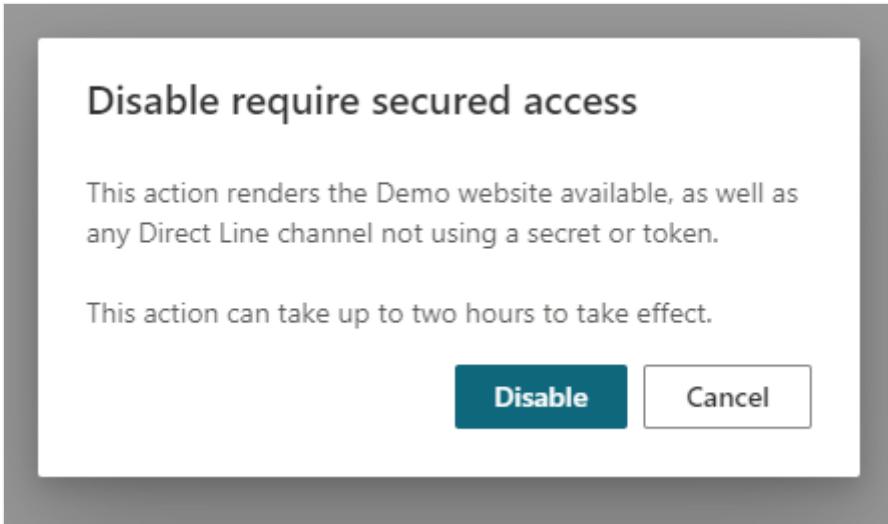
## 1. Enable **Require secured access**.

### ⚠ Warning

Once "Require secured access" is enabled or disabled, the system can take up to two hours to propagate the settings and take effect. Until then, the previous setting is in effect. You don't need to publish the agent for this change to take effect.

You should plan ahead to avoid exposing your agent unintentionally.

If you need to disable the web channel security option, you can do so by clearing the **Require secured access** toggle. Disabling secured access can take up to two hours to propagate.



## Use secrets or tokens

If you're creating a service-to-service app, specifying the secret in the authorization header requests might be the simplest approach.

If you're writing an app where the client runs in a web browser or mobile app, or otherwise the code might be visible to customers, you must exchange your secret for a token. If you don't use a token, your secret can be compromised. When you're making the request to acquire the token in your service, specify the secret in the authorization header.

Tokens only work for a single conversation and expire unless refreshed.

Choose the security model that works best for your situation.

### Warning

We strongly discourage exposing the secret in any code that runs in the browser, either hard-coded or transferred through a network call.

Acquiring the token using the secret in your service code is the most secured way to protect your Copilot Studio agent.

## Obtain the secrets

You need the secret so you can specify it in your app's authorization header requests or similar.

1. In the navigation menu, under **Settings**, select **Security**. Then select the **Web channel security** tile.
2. Select **Copy** for either **Secret 1** or **Secret 2** to copy it to the clipboard. Select the visibility icon  to reveal the secret. A warning prompt appears before you can reveal it.

## Swap secrets

If you need to change the secret being used by your agent, you can do so without any downtime or interruption.

Copilot Studio provides you with two secrets, which work simultaneously. You can swap the secret being used with the other one. Once the secrets are swapped and your users are all connected using the new secret, you're safe to regenerate the secret.

## Regenerate a secret

To regenerate a secret, select **Regenerate** next to the secret.

### Warning

Any user profile connected using the original secret or a token obtained from that secret is disconnected.

## Generate a token

You can generate a token that can be used when starting a single agent conversation. For more information, see the section **Get Direct Line token** in [Publish an agent to mobile or custom apps](#).

1. [Obtain the secret](#).
2. Issue the following request in your service code to exchange the secret for a token.  
Replace <SECRET> with the value of the secret you obtained in Step 1.

HTML

```
POST https://directline.botframework.com/v3/directline/tokens/generate
Authorization: Bearer <SECRET>
```

The following snippets provide examples of the generated token request and its response.

## Sample generate token request

HTML

```
POST https://directline.botframework.com/v3/directline/tokens/generate
Authorization: Bearer
RCurR_XV9ZA.cwA.BKA.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0
```

## Sample generate token response

HTML

```
HTTP/1.1 200 OK
[other headers]
```

JSON

```
{
 "conversationId": "abc123",
 "token":
"RCurR_XV9ZA.cwA.BKA.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfiCpg4Fv
```

```
0y8qqb0F5x8qb0F5xn",
 "expires_in": 1800
}
```

If the request is successful, the response contains a token that's valid for one conversation and an `expires_in` value that indicates the number of seconds until the token expires.

For the token to remain useful, you must [refresh the token](#) before it expires.

## Refresh a token

A token can be refreshed an unlimited number of times, as long as it isn't expired.

An expired token can't be refreshed.

To refresh a token, issue the following request and replace `<TOKEN TO BE REFRESHED>` with the token you want to refresh.

HTML

```
POST https://directline.botframework.com/v3/directline/tokens/refresh
Authorization: Bearer <TOKEN TO BE REFRESHED>
```

The following snippets provide examples of the refresh token request and response.

## Sample refresh request

HTML

```
POST https://directline.botframework.com/v3/directline/tokens/refresh
Authorization: Bearer
CurR_XV9ZA.cwA.BKA.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfiCpg4Fv0y
8qqb0F5x8qb0F5xn
```

## Sample refresh response

If the request is successful, the response contains a new token that's valid for the same conversation as the previous token and an `expires_in` value that indicates the number of seconds until the new token expires.

For the new token to remain useful, you must refresh the token again before it expires.

HTML

```
HTTP/1.1 200 OK
```

```
[other headers]
```

JSON

```
{
 "conversationId": "abc123",
 "token":
 "RCurR_XV9ZA.cwA.BKA.y8qb0F5xPGfiCpg4Fv0y8qqb0F5x8qb0F5xniaJrC8xpy8qb0F5xnR2vtCX7C
 Zj0LdjAPGfiCpg4Fv0",
 "expires_in": 1800
}
```

For more information on refreshing a token, see the section [Refresh a Direct Line token in Direct Line API - Authentication](#).

## Related content

- [Key Concepts - Publish and deploy your agent](#)
- [Publish an agent to mobile or custom apps](#)
- [Publish an agent to Azure Bot Service channels](#)
- [Configure user authentication with Microsoft Entra ID](#)

# Configure single sign-on with Microsoft Entra ID

10/15/2025

Copilot Studio supports single sign-on (SSO). SSO allows agents on your website to sign customers in if they're already signed in to the page or app where the agent is deployed.

For example, the agent is hosted on the corporate intranet or in an app that the user is already signed in to.

There are five main steps to configuring SSO for Copilot Studio:

1. Enable manual authentication for your agent with Microsoft Entra ID
2. Create an app registration in Microsoft Entra ID for your custom canvas.
3. Define a custom scope for your agent in Microsoft Entra ID.
4. Add the custom scope to your agent configuration.
5. Configure your custom canvas client side code to enable SSO.

## Prerequisites

- [Enable user authentication with Microsoft Entra ID](#)
- [Add an authentication topic to your agent](#)
- [Use a custom canvas](#)

### ⓘ Note

To configure SSO using other OAuth 2.0 providers, see [Configure single sign-on with generic OAuth providers](#).

## Supported channels

The following table details the [channels](#) that currently support SSO. You can suggest support for extra channels [at the Copilot Studio ideas forum ↗](#).

 Expand table

Channel	Supported
Azure Bot Service channels	Not supported
Custom Website	Supported
Demo Website	Not supported
Facebook	Not supported
Microsoft Teams <sup>1</sup>	Supported
Mobile App	Not supported
Omnichannel for Customer Service <sup>2</sup>	Supported
SharePoint <sup>1</sup>	Supported

<sup>1</sup> If you also have the Teams channel enabled, you need to follow the configuration instructions on the [Configure single sign-on with Microsoft Entra ID for agents in Microsoft Teams](#) documentation. Failing to configure the Teams SSO settings as instructed on that page causes your users to always fail authentication when using the Teams channel.

<sup>2</sup> Only the live chat channel is supported. For more information, see [Configure handoff to Dynamics 365 Customer Service](#).

### Important

SSO is currently not supported when an agent is published to a [Power Apps portal](#).

Web app

## Create app registrations for your custom website

To enable SSO, you need to create two separate app registrations:

- An *authentication app registration*, which enables Microsoft Entra ID user authentication for your agent
- A *canvas app registration*, which enables SSO for your custom web page

We don't recommend reusing the same app registration for both your agent and your custom website for security reasons.

1. Follow the instructions in [Configure user authentication with Microsoft Entra ID](#) to create an authentication app registration.
2. Create a second app registration to serve as your canvas app registration.

## Add token exchange URL

To update the Microsoft Entra ID authentication settings in Copilot Studio, you need to add the token exchange URL to allow your app and Copilot Studio to share information.

1. In the Azure portal on your authentication app's registration page, go to **Expose an API**.
2. Under **Scopes**, select the **Copy to clipboard** icon.
3. In Copilot Studio, in the navigation menu under **Settings**, select **Security**, and then select the **Authentication** tile.
4. For **Token exchange URL (required for SSO)**, paste the scope you copied earlier.
5. Select **Save**.

## Configure your canvas app client ID to expose an API menu

1. In the Azure portal on your authentication app registration page, go to **Overview**.
2. Copy the **Application (client) ID** value under **Essentials**.
3. On the navigation bar, select **Manage > Expose an API**.
4. Under **Authorized client applications**, select **+ Add a client application**, and then paste the copied client ID.
5. Select **Save**.

## Configure your canvas app registration

1. After you create your canvas app registration, go to **Authentication**, and then select **Add a platform**.
2. Under **Platform configurations**, select **Add a platform**, and then select **SPA**.
3. Under **Redirect URIs**, enter the URL for your web page; for example,  
`http://contoso.com/index.html`.

## Configure Web

X

◀ All platforms

Quickstart Docs ↗

### \* Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating or signing out users. The redirect URI you send in the request to the login server should match one listed here. Also referred to as reply URLs. [Learn more about Redirect URIs and their restrictions](#)

https://contoso.com/bot/index.html ✓

### Front-channel logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

e.g. https://example.com/logout

### Implicit grant and hybrid flows

Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. [Learn more about tokens](#).

Select the tokens you would like to be issued by the authorization endpoint:

- Access tokens (used for implicit flows)
- ID tokens (used for implicit and hybrid flows)

Configure

Cancel

4. In the **Implicit grant and hybrid flows** section, turn on both **Access tokens (used for implicit flows)** and **ID tokens (used for implicit and hybrid flows)**.
5. Select **Configure**.

## Find your agent's token endpoint URL

1. In Copilot Studio, open your agent and then select **Channels**.
2. Select **Mobile app**.
3. Under **Token Endpoint**, select **Copy**.

## Mobile app

X

Allow your mobile app users to chat with your bot. Check out our [instructions for both native and web-based mobile apps](#).

### Native apps

Developing a native app? Copy and provide the information below to your developers so they can connect your app to your bot.

#### Token Endpoint

Copy

### Web-based apps

If you're developing a web-based app, copy and paste the following code snippet to your embed web control HTML. If you do not have access to the HTML code, share the code with the person responsible for your web-based app.

## Configure SSO in your web page

### i Important

AI-generated answers from SharePoint and Graph Connector data sources aren't available to Guest users in SSO-enabled apps.

Use the code provided in the [Copilot Studio GitHub repo](#) to create a web page for the redirect URL. Copy the code from the GitHub repo and modify it using the following instructions.

1. Go to the [Overview](#) page in Azure portal and copy the **Application (client) ID** and **Directory (tenant) ID** from your canvas app registration.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, the breadcrumb navigation shows 'Home > App registrations > ContosoSalesCanvas'. The main content area has a title 'ContosoSalesCanvas' with a blue square icon. On the left, there's a sidebar with 'Overview' (highlighted with a red box), 'Quickstart', 'Integration assistant (preview)', 'Manage', and 'Branding'. On the right, there are several details: 'Display name : ContosoSalesCanvas', 'Application (client) ID : [redacted]', 'Directory (tenant) ID : [redacted]', and 'Object ID : [redacted]'. There are also 'Delete' and 'Endpoints' buttons.

2. To configure the Microsoft Authentication Library (MSAL):

- Assign `clientId` to your **Application (client) ID**.
- Assign `authority` to `https://login.microsoftonline.com/` and add your **Directory (tenant) ID** to the end.

For example:

```
JavaScript

var clientApplication;
(function (){
 var msalConfig = {
 auth: {
 clientId: '00001111-aaaa-2222-bbbb-3333cccc4444',
 authority: 'https://login.microsoftonline.com/xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'
 },
 }
});
```

3. Set the `theURL` variable to the token endpoint URL you copied earlier. For example:

```
JavaScript

(async function main() {

 var theURL = "https://<token endpoint URL>"
```

4. Edit the value of `userId` to include a custom prefix. For example:

```
JavaScript
```

```
var userId = clientApplication.account?.accountIdentifier != null ?
 ("My-custom-prefix" +
 clientApplication.account.accountIdentifier).substr(0, 64)
 : (Math.random().toString() +
 Date.now().toString()).substr(0,64);
```

5. Save your changes.

6. Verify you successfully configured SSO.

When testing your agent, if SSO *isn't* successfully configured, you're prompted to sign in, which gives you a validation code you must copy into the chat window.

If you see a sign-in prompt, verify you completed steps 1 through 5 of this procedure correctly. If SSO *is* successfully configured, you aren't prompted to sign in.

 **Note**

The code in the GitHub repo requires users to select the sign-in button. In production, you might want to replace the button functionality with a more appropriate event, like navigating to a page.

## Related content

- [Azure App Registration](#)

# Configure single sign-on with Microsoft Entra ID for agents in Microsoft Teams

Article • 05/19/2025

Copilot Studio supports single sign-on (SSO) for agents published to Microsoft Teams 1:1 chats, which means agents can automatically sign in users with their Microsoft Teams credentials. SSO is only supported when using Microsoft Entra ID. Other service providers, such as Azure AD v1, don't support SSO in Microsoft Teams.

## ⓘ Important

It's possible to use SSO in Microsoft Teams chats, and not require manual authentication. To use this method for a previously published agent, reconfigure the agent to use **Authenticate with Microsoft** and then publish it again to Microsoft Teams. It might take a few hours before this change takes effect. If a user is in the middle of a conversation and the change doesn't appear to have taken effect, they can type "start over" in the chat to force the conversation to restart with the latest version of the agent. These changes are now available for Teams 1:1 chats between the user and the agent. They are not yet available for group chats or channel messages.

SSO is not supported for agents integrated with [Dynamics 365 Customer Service](#).

**Please do not proceed with the following document unless necessary.** If you want to use manual authentication for your agent, see [Configure user authentication with Microsoft Entra ID](#).

## ⓘ Note

If you are using Teams SSO authentication with the manual authentication option, and also using the agent on custom websites at the same time, you must deploy the Teams app using the app manifest.

For more information, see [Download the Teams app manifest for an agent](#).

Other configurations such as authentication options beside Manual, or through Teams deployment using Copilot Studio one-click, will not work.

## Prerequisites

- [Learn how to use user authentication in a topic](#).

- Connect and configure an agent for Microsoft Teams.

## Configure an app registration

Before configuring SSO for Teams, you need to configure user authentication with Microsoft Entra ID. This process creates an app registration that is required to set up SSO.

1. Create an app registration. See the instructions in [Configure user authentication with Microsoft Entra ID](#).
2. Add the redirect URL.
3. Generate a client secret.
4. Configure manual authentication.

## Locate your Teams channel app ID

1. In Copilot Studio, open the agent for which you want to configure SSO.
2. Go to the **Channels** page for your agent and select the **Teams and Microsoft 365 Copilot** tile.
3. In the **Teams and Microsoft 365 Copilot** configuration panel, select **Edit details**, expand **More**, and then select **Copy** next to the **App ID** field.

## Add your Teams channel app ID to your app registration

1. Go to the [Azure portal](#). Open the app registration blade for the app registration you created when you configured user authentication for your agent.
2. Select **Expose an API** on the side pane. For **Application ID URI**, select **Set**.



## Expose an API



Search (Ctrl+ /)



Got feedback?

Overview

Quickstart

Integration assistant

Application ID URI ⓘ **Set**

Scopes defined by this API

3. Enter `api://botid-{teamsbotid}` and replace `{teamsbotid}` with your Teams channel app ID that you found earlier.



## Expose an API



Search (Ctrl+ /)



Got feedback?

Overview

Quickstart

Integration assistant

Manage

## Set the App ID URI

Application ID URI

api://botid-

Save

Discard

4. Select Save.

## Grant admin consent

Applications are authorized to call APIs when they're granted permissions by users/admins as part of the consent process. To learn more about consent, see [Permissions and consent in the Microsoft identity platform](#).

If the admin consent option is available, you must grant consent:

1. In the Azure portal on your app registration blade, go to [API Permissions](#).
2. Select **Grant admin consent for <your tenant name>** and then select **Yes**.

**Important**

To avoid users having to consent to each application, someone assigned at least the Application Administrator or Cloud Application Administrator role can [grant tenant-wide consent](#) to your application registrations.

## Add API permissions

1. In the Azure portal on your app registration blade, go to **API Permissions**.
2. Select **Add a permission** and choose **Microsoft Graph**.
3. Select **Delegated permissions**. A list of permissions appears.
4. Expand **OpenId permissions**.
5. Select **openid** and **profile**.
6. Select **Add permissions**.

The screenshot shows the Azure portal interface for managing API permissions. On the left, the navigation menu includes 'Home', 'Overview', 'Quickstart', 'Integration assistant', 'Manage' (with 'Branding & properties', 'Authentication', 'Certificates & secrets', 'Token configuration', 'API permissions', and 'Expose an API'), and 'Request API permissions'. The 'API permissions' section is currently selected. On the right, the 'Request API permissions' dialog is open, showing the 'OpenId permissions' section. Two permissions are listed: 'email' and 'offline\_access', both with checkboxes. Below them, 'openid' and 'profile' are also listed with checked checkboxes, indicated by red boxes around the checkboxes and the 'Add permissions' button. The 'Add permissions' button is also highlighted with a red box.

## Define a custom scope for your agent

1. In the Azure portal on your app registration blade, go to **Expose an API**.
2. Select **Add a scope**.

## | Expose an API ⚡ ...

Search (Ctrl+ /) <<

- ❯ Authentication
- 🔑 Certificates & secrets
- ☰ Token configuration
- 🔗 API permissions
- ☁️ Expose an API
- 🏢 App roles
- 👤 Owners
- 👤 Roles and administrators

🔗 Got feedback? ...

🔗 Application ID URI ...

**Scopes defined by this API**

Define custom scopes to restrict access to certain operations. An API can request that a user or admin consent to these scopes.

Adding a scope here creates only delegated consent. This is for application types that don't support consent. [Go to App roles.](#)

+ Add a scope

3. Set the following properties:

☰ Expand table

Property	Value
Scope name	Enter <code>Test.Read</code>
Who can consent?	Select <b>Admins and users</b>
Admin consent display name	Enter <code>Test.Read</code>
Admin consent description	Enter <code>Allows the app to sign the user in.</code>
State	Select <b>Enabled</b>

! Note

The scope name `Test.Read` is a placeholder value and should be replaced with a name that makes sense in your environment.

4. Select **Add scope**.

# Add Microsoft Teams client IDs

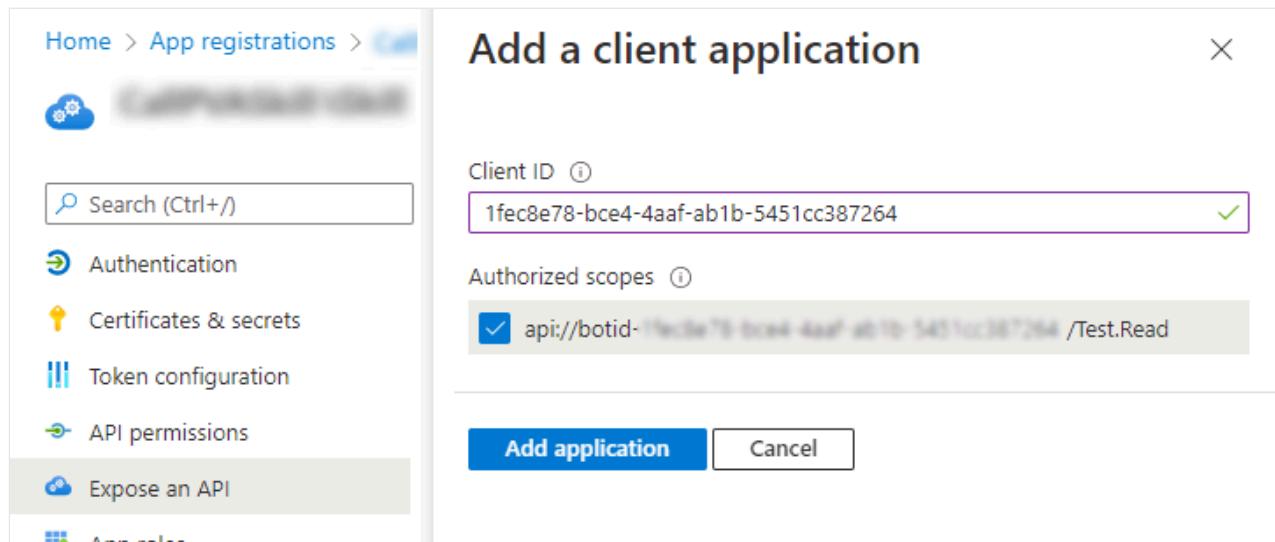
## ⓘ Important

In the following steps, the values provided for Microsoft Teams client IDs should be used literally because they are the same across all tenants.

1. In the Azure portal on your app registration blade, go to **Expose an API** and select **Add a client application**.

The screenshot shows the Azure portal's 'Expose an API' blade. At the top, there's a search bar and a 'Got feedback?' link. Below that, there's a 'Scopes' section with a text input field containing 'api://'. On the left, there's a sidebar with links like 'Overview', 'Quickstart', 'Integration assistant', 'Manage', 'Branding & properties', 'Authentication', 'Certificates & secrets', and 'Token configuration'. At the bottom right, there's a button labeled '+ Add a client application' with a red border around it.

2. In the **Client ID** field, enter the client ID for Microsoft Teams mobile/desktop, which is `1fec8e78-bce4-4aaaf-ab1b-5451cc387264`. Select the checkbox for **the scope that you created earlier**.



3. Select **Add application**.

4. Repeat the previous steps but, for **Client ID**, enter the client ID for Microsoft Teams on the web, which is `5e3ce6c0-2b1f-4285-8d4b-75ee78787346`.

5. Confirm the **Expose an API** page lists the Microsoft Teams client app IDs.

To summarize, the two Microsoft Teams client IDs added to the **Expose an API** page are:

- `1fec8e78-bce4-4aaaf-ab1b-5451cc387264`
- `5e3ce6c0-2b1f-4285-8d4b-75ee78787346`

## Add token exchange URL to your agent's Authentication settings

To update the Microsoft Entra ID authentication settings in Copilot Studio, you must add the token exchange URL to allow Microsoft Teams and Copilot Studio to share information.

1. In the Azure portal on your app registration blade, go to **Expose an API**.
2. Under **Scopes**, select the **Copy to clipboard** icon.
3. In Copilot Studio, under the settings for the agent, select **Security**, and then select the **Authentication** tile.
4. For **Token exchange URL (required for SSO)**, paste the scope you copied earlier.
5. Select **Save**.

## Settings

X

Copilot details

Generative AI

**Security**

Authoring Canvas

Entities

Skills

Voice

Languages

Language understandi...

Advanced

### Authentication

Verify a user's identity during a conversation. The copilot receives secure access to the user's data and is able to take actions on their behalf, resulting in a more personalized experience. [Learn more](#)

Choose an option

No authentication

Publicly available in any channel

Authenticate with Microsoft

Entra ID authentication in Teams and Power App

Authenticate manually

Set up authentication for any channel

**Require users to sign in**

Redirect URL

<https://token.botframework.com/.auth/web/re>

Copy

Service provider \*

Azure Active Directory v2



Client ID \*

Client secret \*

.....

Token exchange URL (required for SSO) [Learn more about SSO](#)

aaaa0000-bb11-2222-33cc-444444dddddd/Test.Read

Tenant ID

Scopes ⓘ

profile openid

**Save**

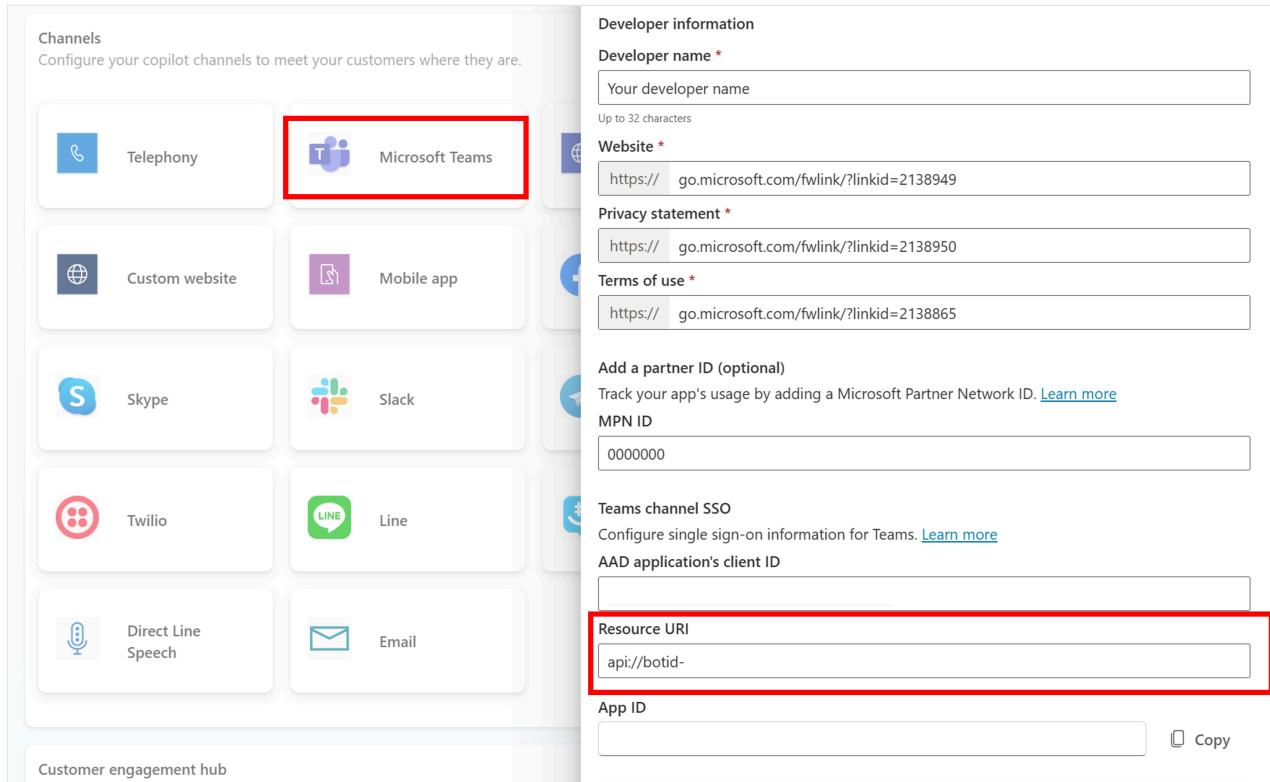
## Add SSO to your agent's Teams channel

1. In Copilot Studio, under the settings for the agent, select **Channels**.
2. Select the **Teams and Microsoft 365 Copilot** tile.
3. Select **Edit details** and expand **More**.
4. For **AAD application's client ID**, enter the **Application (client) ID** from your app registration.

To obtain this value, open the Azure portal. Then on your app registration blade, go to **Overview**. Copy the value in the **Application (client) ID** box.

## 5. For Resource URI, enter the Application ID URI from your app registration.

To obtain this value, open the Azure portal. Then on your app registration blade, go to **Expose an API**. Copy the value in the **Application ID URI** box.



## 6. Select **Save**, and then **Close**.

## 7. Publish the agent again, to make the latest changes available to your customers.

## 8. Select **See agent in Teams**, to start a new conversation with your agent in Teams and verify if it automatically signs you in.

## Known issues

If you first published your agent using manual authentication without Teams SSO, the agent in Teams will continuously prompt users to sign in.

# Configure single sign-on with a generic OAuth provider

Article • 03/27/2025

Copilot Studio supports single sign-on (SSO) with OAuth 2.0 compatible authentication providers. SSO allows agents on your website to sign customers in if they signed in to the page or app where the agent is deployed.

## Prerequisites

- [Configure user authentication with a generic OAuth 2.0 provider.](#)
- [Add user authentication to topics.](#)
- [Customize the look and feel of an agent.](#)

## Create or use a custom canvas that supports the generic OAuth provider SSO flow

When the authentication topic is triggered in agents connected with a generic OAuth 2.0 provider, Copilot Studio sends a message containing a secure URL, which is used to post the user's access token.

The custom canvas or a page where the canvas is embedded should implement the following pattern:

1. Obtain an access token for the signed-in user from your OAuth 2.0 authentication provider, using your preferred method.
2. Intercept an incoming message from Copilot Studio, and extract the secure URL.
3. Post the access token to the secure URL.

## Extract the secure URL and post the token

The custom canvas intercepts incoming messages using a middleware concept, which is code that runs in the context of receiving messages from Copilot Studio.

To respond to sign-in requests, the custom canvas needs to intercept messages with attachments that have the content type `application/vnd.microsoft.card.oauth`. OAuthCard attachments contain a `content.tokenPostResource.sasUrl` property, from

which the secure URL can be extracted. Finally, the custom canvas should post the user's access token to the secure URL.

The following JavaScript code is an example of middleware code that extracts the secure URL and posts a token. If the post is successful, the middleware returns `false`. If the post is unsuccessful, or if the activity doesn't have the

`application/vnd.microsoft.card.oauth` property, it returns `next(...args)`.

Javascript

```
const activityMiddleware = () => next => (...args) => {
 if (args[0].activity.attachments?.[0]?.contentType ===
 'application/vnd.microsoft.card.oauth') {
 var postEndpoint = args[0].activity.attachments?.
 [0].content.tokenPostResource.sasUrl;

 // Perform an HTTP POST to the secure URL with the body of:
 // {
 // "token": <user_token>
 // }

 if(success)
 return false;
 else
 return next(...args);
 } else {
 return next(...args);
 }
};
```

## Full sample code

The implementation of a custom canvas or app that obtains a token for a signed-in user, and posts the token to Copilot Studio, varies based on your authentication provider. For more information, see your authentication provider's documentation for more details about sign-in flows and obtaining access tokens. For a reference sample using OKTA, see [Third party SSO with OKTA](#).

## Using the token in Copilot Studio

Tokens that are posted using the secure URL are populated into the `System.User.AccessToken` variable in Copilot Studio. Agent makers can use this system variable to access protected APIs that are connected to the authenticated provider that generated the token.

In the following example, an HTTP call is configured with an Authorization header that uses `System.User.AccessToken`.

The screenshot shows the configuration interface for an HTTP Request node. On the left, the node is labeled "HTTP Request". It has fields for "URL" (set to "https://my.api"), "Method" (set to "Get"), and "Headers and body" (with an "Edit" button). Under "Response data type", it is set to "Record". A "Save response as" field contains the variable "{x} apiResponse record". On the right, the "HTTP Request properties" panel is open, showing the configuration details:

- Headers**: An "Authorization" header is defined with a value of "Bearer " & System.User.AccessToken.
- Body**: Set to "No content".
- Request timeout (milliseconds)**: Set to 30000.
- Response headers**: A placeholder "Select a variable" is shown.

## Supported channels

SSO with a generic OAuth authentication provider is a custom pattern, which can be implemented by either a custom canvas, or any other client working with the Directline API.

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

# Add user authentication to topics

09/06/2025

You can enable user authentication directly within an agent conversation. You can assign a user's basic properties, such as name and ID, to variables. You can also prompt a user to sign in using an authentication node in a topic, which retrieves a user token, and then use that token to retrieve the user's information from a back-end system.

## (!) Note

Agents created in Copilot Studio and in Microsoft Teams are automatically configured for Microsoft Entra ID authentication.

You can also configure single sign-on (SSO) so your users don't need to sign in manually. For more information, see [Configure single sign-on with Microsoft Entra ID](#).

## Prerequisites

- [Variables overview](#)
- [Configure user authentication in Copilot Studio](#)

Web app

Add user authentication to a topic to allow your customers to sign in right in the conversation. You can then personalize the conversation with user variables and access back-end systems on the user's behalf.

## Configure manual authentication with Microsoft Entra ID

You need to configure user authentication with Microsoft Entra ID before you can use authentication in your topics.

Follow the instructions in [Configure user authentication with Microsoft Entra ID](#).

## Add user authentication with the Sign in system topic

When you create an agent, Copilot Studio automatically adds a system topic called **Sign in**. To use it, you must set your agent's authentication to manual and require users to sign in. When a customer starts a conversation with the agent, the **Sign in** topic triggers and prompts the user to sign in. You can customize the **Sign in** topic as appropriate for your agent.

 **Important**

We recommend that the **Sign in** topic is only used to provide the authentication method provided by Copilot Studio. It shouldn't be modified to call any other actions or flows, or other authentication methods.

1. Open your agent in Copilot Studio, select **Settings** at the top of the page, and then select **Security**.
2. Select **Authentication**.
3. Select **Authenticate manually**, and then select **Require users to sign in**.
4. [Configure all manual authentication fields](#), as required.
5. Select **Save**.

## Add user authentication with a custom topic

The **Sign in** topic authenticates the user at the beginning of the conversation. To allow the user to sign in later, you can add an **Authenticate** node to any custom topic.

When customers enter their user name and password, they might be prompted to enter a validation code. After they sign in, they're not prompted again, even if they reach another **Authenticate** node.

1. Select **Settings** at the top of the page, and then select **Security**.
2. Select the **Authentication** tile.

 **Note**

You must select **Authenticate manually** to add user authentication to a custom topic.

3. Select **Authenticate manually** and clear the **Require users to sign in** checkbox.
4. [Configure all manual authentication fields](#), as required.
5. Select **Save**.

6. Select **Topics** at the top of the page.

7. Select **Add node** (  ) > **Advanced** > **Authenticate**.

8. Test your topic with a user configured with your identity provider.

### Tip

It's important that you create paths for both a successful sign-in and failure to sign in. A sign-in might fail for many reasons, including errors with the identity provider's sign-in experience.

## Authentication variables

When you configure user authentication for your agent, you can use authentication variables in your topics. The following table compares the availability of these variables based on the authentication option you chose.

For more information about variables, see [Variables overview](#).

 Expand table

Authentication variable	No authentication	Authenticate with Microsoft	Authenticate manually
User.DisplayName	Not available	Available	Available
User.FirstName	Not available	Available	Available
User.LastName	Not available	Available	Available
User.PrincipalName	Not available	Available	Available
User.Email	Not available	Available	Available
User.Id	Not available	Available	Available
User.IsLoggedIn	Not available	Available	Available
User.AccessToken	Not available	Not available	Available
SignInReason	Not available	Available	Available

## User.DisplayName

### Warning

This variable isn't guaranteed to contain a value. Test with a user from your identity provider to ensure your topic works correctly.

The `User.DisplayName` variable contains the display name stored in the identity provider. Use this variable to greet or refer to the user without their having to explicitly give their name to the agent, making the conversation more personalized.

Copilot Studio automatically sets the value of `User.DisplayName` from the `name` claim provided by the identity provider, as long as the `profile` scope was defined when manual authentication was configured. For more information about scope, see [Configure user authentication with Microsoft Entra ID](#).

## User.Id

### Warning

This variable isn't guaranteed to contain a value. Test with a user from your identity provider to ensure your topic works correctly.

The `User.Id` variable contains the user ID stored in the identity provider. Use this variable in [Power Automate flows](#) to call APIs that take the UserID as a value.

Copilot Studio automatically sets the value of `User.DisplayName` from the `sub` claim provided by the identity provider.

## User.IsLoggedIn

`User.IsLoggedIn` is a Boolean variable that stores the user's sign-in status. A value of `true` indicates the user is signed in. You can use this variable to create branching logic in your topics that checks for a successful sign-in, or to fetch user information only if the user is signed in.

## User.AccessToken

### Warning

Make sure you're passing the `User.AccessToken` variable only to trusted sources. It contains user authentication information, which, if compromised, could harm the user.

The `User.AccessToken` variable contains the user's token, obtained after the user is signed in. You can pass this variable to [Power Automate flows](#) so they can connect to back-end APIs and fetch user information, or to take actions on the user's behalf.

Don't use `User.AccessToken` in [Message](#) nodes or in flows that you don't trust.

## SignInReason

`SignInReason` is a choice-type variable that indicates when the user must sign in. It has two possible values:

- `SignInRequired` indicates the user must sign in at the beginning of the conversation using the [Sign in](#) system topic. [Require users to sign in](#) must be turned on.
- `Initializer` indicates that when a user isn't signed in, and they reach a point in the conversation that uses authentication variables, they're prompted to sign in.

## Related content

You can also configure single sign-on (SSO) so your users don't need to sign in manually. For more information, see [Configure single sign-on with Microsoft Entra ID](#).

# Configure user authentication for actions

09/12/2025

When creating a [Copilot Studio action](#) for an authenticated Copilot Studio project, you can enable user authentication, or supply a set of credentials for the agent to use on behalf of the user.

- Select **Agent author authentication** if access to the service associated with the action is implicit, or for low-risk use cases. For example, use this authentication setting to find the phone number for the support team for a given postal code. Another use case could be using a weather API to get the current forecast.
- Select **User authentication** if you must restrict data access to specific groups or individuals in the user community. For example, use this authentication setting if the agent is meant to retrieve data that only the user has access to, or to perform work on their behalf.

## Creating connections

Users are prompted when they visit any dialog that uses a user action to sign in to the experience. They're prompted as soon as the conversation begins, and they authenticate with the agent.

When users review the connections page, they can see the connection they need to configure for the action to complete a given dialog, and other connections your actions might require in the entire experience.

Completing the connection and returning to the conversation with the agent allows your customers to "retry" the action. It then completes with the customer's data access.

## Data access and permission management

Copilot Studio doesn't store any credentials. Agents prompt users for their credentials whenever access tokens expire or are revoked on the service side. Additionally, users can manually access the connection page and refresh or revoke permissions once they're done talking to your agent.

## Supported channels

The following table details the [channels](#) that currently support user authentication for actions.

Channel	Supported
Azure Bot Service channels	Not supported
Custom Website	Supported
Demo Website	Not supported
Facebook	Not supported
Microsoft Teams <sup>1</sup>	Supported
Mobile App	Not supported
Omnichannel for Customer Service <sup>2</sup>	Supported
SharePoint <sup>1</sup>	Supported

<sup>1</sup> If you also have the Teams channel enabled, you need to follow the configuration instructions on the [Configure single sign-on with Microsoft Entra ID for agents in Teams](#) documentation. Failing to configure the Teams single sign-on (SSO) settings as instructed on that page causes your users to always fail authentication when using the Teams channel.

<sup>2</sup> Only the live chat channel is supported. For more information, see [Configure handoff to Dynamics 365 Customer Service](#).

# Key concepts – Analytics in Copilot Studio

08/27/2025

Use analytics to understand how well your agent is performing and to identify areas for improvement.

The **Analytics** tab in Copilot Studio shows you comprehensive data for your agent, from an overview of key metrics to in-depth usage analytics for your agent's components. You can drill down into each piece of data to get more details.

The analytics experience is tailored for [conversational agents](#) and for [autonomous agents](#).

Analytics are available in all geographies. Times in analytics are in Coordinated Universal Time (UTC). This includes day start and end times, session times, and any other time markers in your agent's data.

## ⚠ Note

Analytics aren't available on the **Analytics** page for activity completed when you test your agent in Copilot Studio using the [test panel](#).

To access analytics:

1. Select your agent under the **Agents** tab in Copilot Studio.
2. Go to the **Analytics** tab in the agent's top menu bar.

You can also check a high-level performance summary on an agent's **Overview** page, then dive deeper into its performance.



## Conversational sessions only

[Analytics for conversational agents](#) in Copilot Studio track user engagement with your agent and try to capture how well your agent handles user tasks.

Conversational analytics uses the following concepts and terms:

- *Conversations* are an ongoing interaction between a specific user, or group of users, on a [channel](#) and your agent.
  - Conversations can pause and resume later, or be [transferred to a customer service representative](#). The conversation might be one-way, either from the customer to the agent, or from the agent to the customer, but is more commonly a back-and-forth interaction between the customer and the agent.
  - A conversation will time out after 15 minutes of inactivity.
  - A single conversation can contain one or more analytics sessions.
  - An analytics session in classic mode is associated with the last custom topic triggered by a user. If the session doesn't include custom topics, it's the last system topic triggered directly by the user.

## Event trigger sessions only

An autonomous agent is an agent with an event trigger. Only [analytics for agents with triggers](#) are available for these agents.

An *analytics session* for agents with triggers tracks from when an agent receives a payload from a trigger through any actions the agent runs in response. These analytics sessions capture what your agent is responding to, and how well your agent performs.

 Note

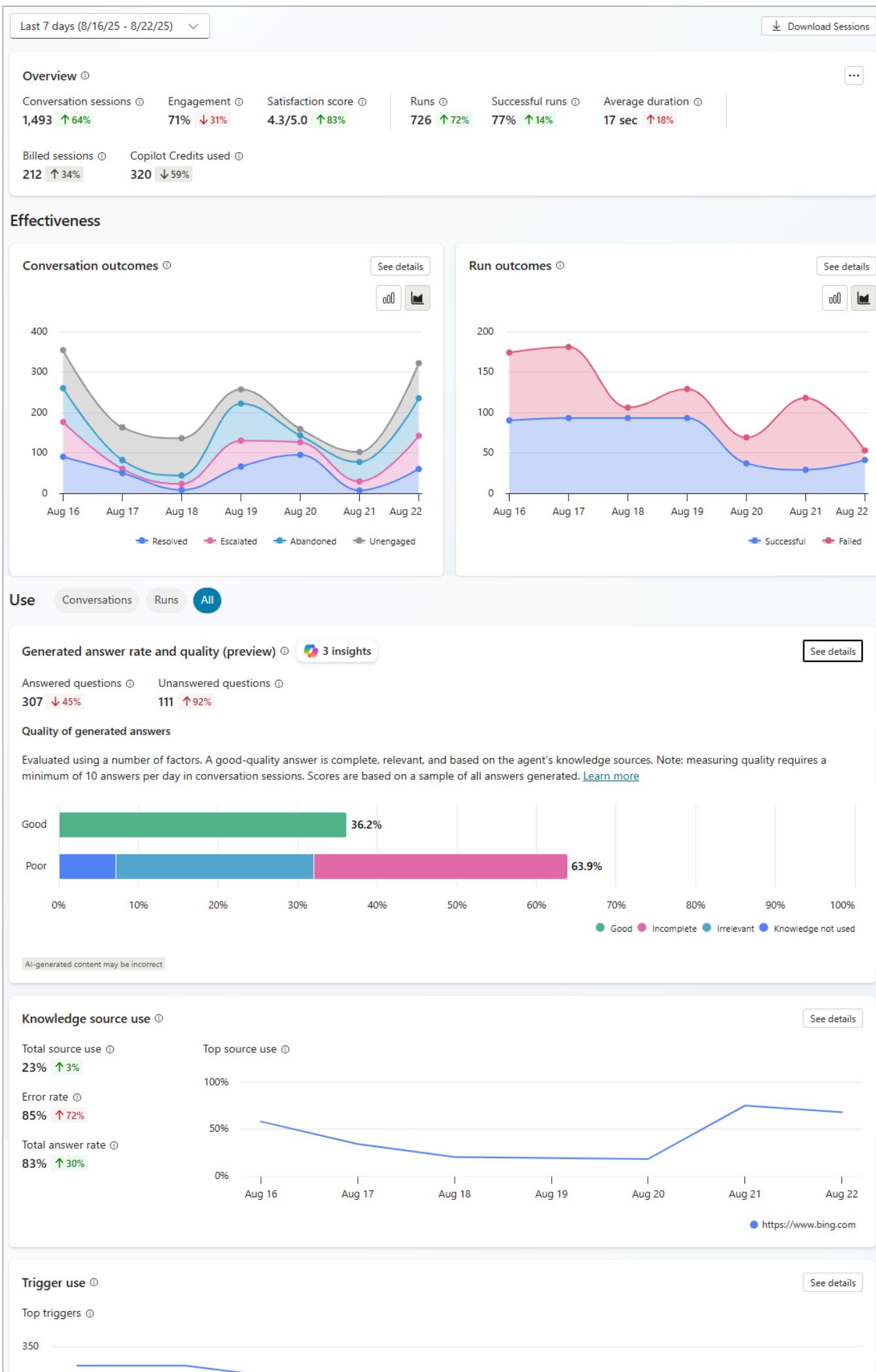
If a trigger fails and the agent doesn't receive a trigger payload, an analytics session can't begin. Only successfully triggered runs are tracked in analytics.

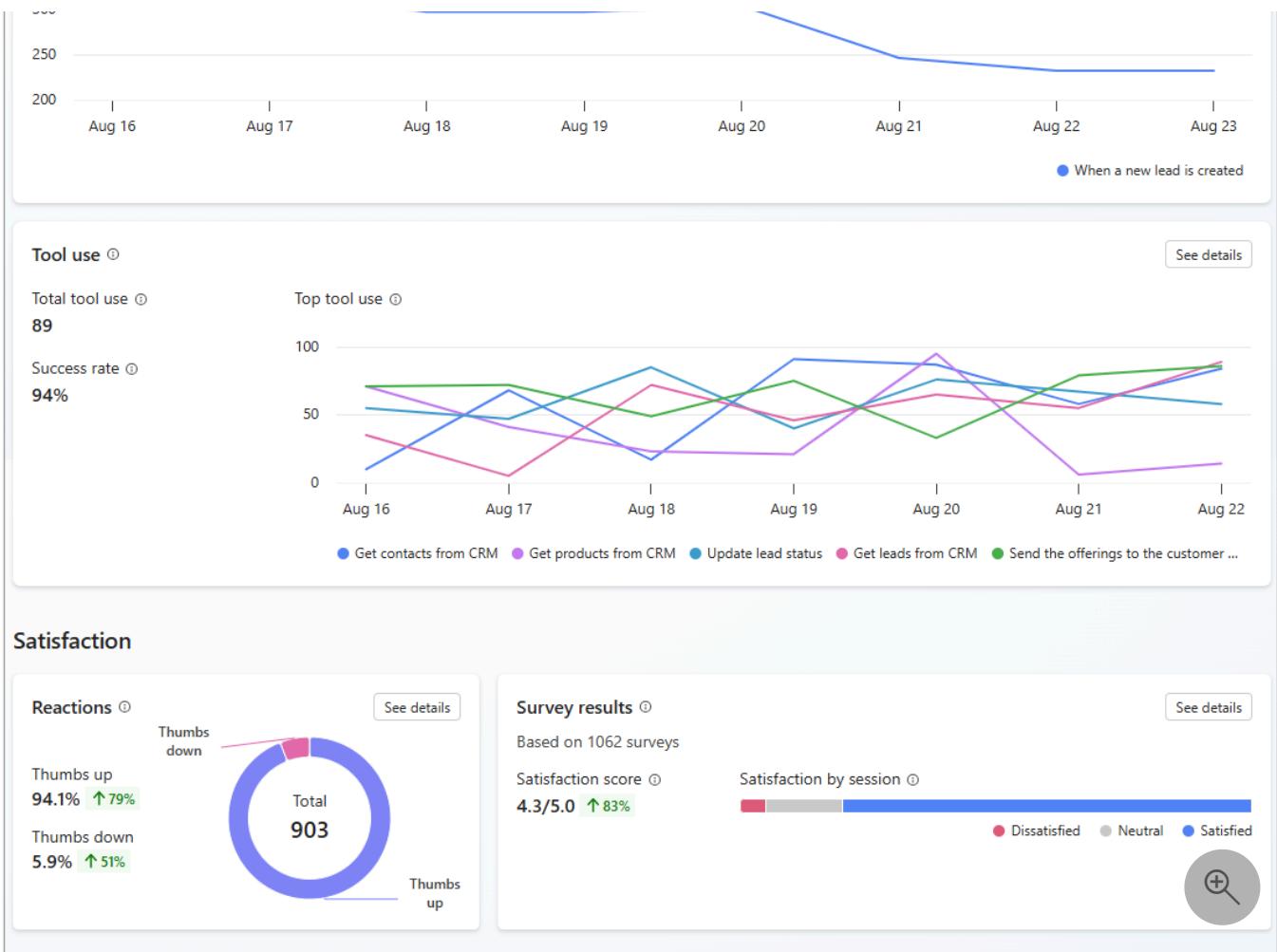
## Hybrid view - both conversational and event-triggered sessions

When your agent's data includes sessions for at least one conversation and at least one trigger-based run in the report's period, then a *hybrid* view of relevant metrics for both conversational and autonomous sessions are displayed on the **Analytics** tab. In this view, the **Overview** and **Effectiveness** sections show metrics side by side for both the conversational and event-triggered sessions. The **Use** section allows you to select between **Conversations** or **Runs**, which displays only relevant data. Selecting **All** shows all use-related data. The **Satisfaction** section displays satisfaction metrics for conversational sessions.

For more information about:

- Displayed metrics relevant to conversation sessions, see [Analyze conversational agent effectiveness](#).
- Displayed metrics relevant to event-triggered sessions, see [Analyze autonomous agent health](#).





## Download conversational transcripts

Conversation transcripts are available for download a few minutes after the conversation times out. You can download any time period within the last 29 days. You can download them in [Dataverse via the Power Apps portal](#) and as [session chat transcripts via the Copilot Studio app](#). It can take up to an hour after the analytic session ends before the related data appears on the analytics dashboard.

### ! Note

Conversation transcripts in Dataverse are unavailable for download on the Copilot Studio app in Teams. To review and export transcripts in Dataverse, you need to sign up for the [Copilot Studio web app](#). Session chat transcripts can be downloaded via the [Copilot Studio app](#). For more information, see [Download agent session transcripts](#).

# Analyze conversational agent effectiveness

09/04/2025

[This article is prerelease documentation and is subject to change.]

The **Analytics** page in Copilot Studio provides an aggregated insight into the overall effectiveness of your agent across [analytics sessions](#).

Last 7 days (6/17/2025 - 6/23/2025) ▾

**Overview** ⓘ

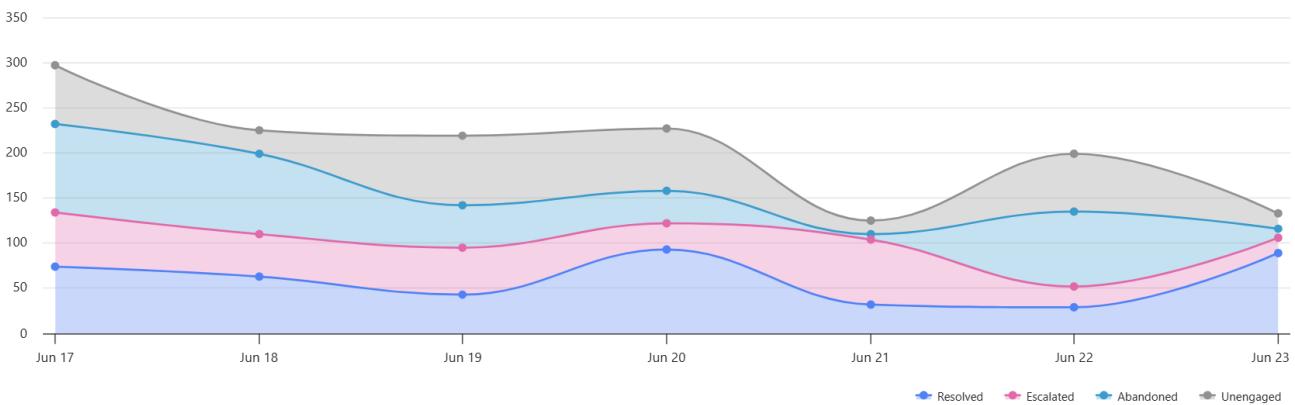
Conversation sessions ⓘ Engagement ⓘ Satisfaction score ⓘ Copilot Credits used ⓘ

1,425 ↑12% 77% ↓23% 4.3/5.0 ↑24% 418 ↓88%

See billing

**Effectiveness****Conversation outcomes** ⓘ

See details

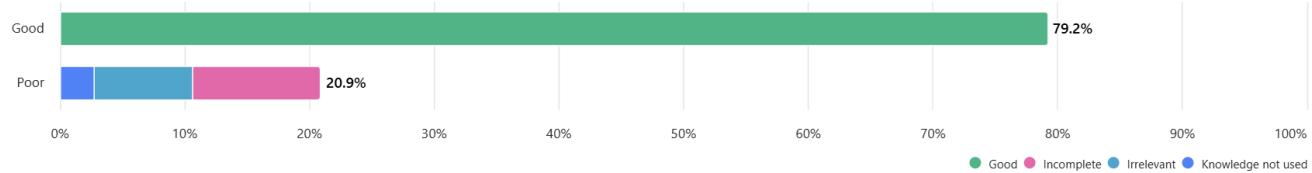
**Use****Generated answer rate and quality (preview)** ⓘ

See details

Answered questions ⓘ Unanswered questions ⓘ  
749 ↑32% 112 ↑63%

**Quality of generated answers**

Evaluated using a number of factors. A good-quality answer is complete, relevant, and based on the agent's knowledge sources. Note: measuring quality requires a minimum of 10 answers per day in conversation sessions. [Learn more](#)



AI-generated content may be incorrect.

**Tool use** ⓘ

See details

Total tool use ⓘ

134

Top tool use ⓘ

150

Success rate ⓘ

62%

100

50

0



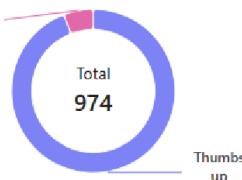
● Send the offerings to the customer ... ● Update lead status ● Get products from CRM ● Get contacts from CRM ● Get leads from CRM

**Satisfaction****Reactions** ⓘ

See details

Thumbs up  
94.1% ↑3%

Thumbs down

Thumbs down  
5.9% ↑49%**Survey results** ⓘ

See details

Based on 1146 surveys

Satisfaction score ⓘ

4.3/5.0 ↑51%

Satisfaction by session ⓘ



● Dissatisfied ● Neutral ● Satisfied



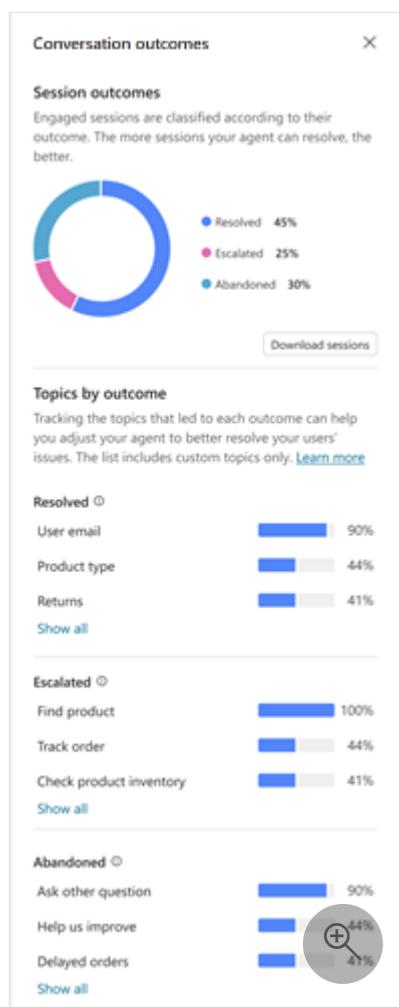
There are four core areas to focus on when reviewing and improving agent effectiveness:

- **Conversation outcomes:** Knowing the end result of a conversation helps you begin to identify where your agent is succeeding and where it needs improvement.
- **Generated answer rate and quality:** Understanding when agent struggles to provide answers to user questions and how it uses knowledge sources can help you find ways to improve your agent's answer rate and quality.
- **Tool use:** Learning how often tools are used and how often they succeed can help you understand if those tools are useful and successful for users.
- **Satisfaction:** Reviewing user feedback helps you to identify new user scenarios and issues and make improvements based directly on what your users are asking for.

You can view analytics for events that occurred in the last 90 days.

## Conversation outcomes

The **Conversation outcomes** section shows a chart that tracks the type of outcome for each session between your agent and users.



To open a side panel with a pie chart breakdown of **session** outcomes, along with the top topics that led to each outcome, select **See details** on the chart.

A [session](#) falls into one of the following two states:

- **Unengaged:** A session starts when a user interacts with your agent or the agent sends a proactive message to the user. The session begins in an *unengaged* state.
- **Engaged:** A session becomes *engaged* when one of the following occurs:
  - a nonsystem topic is triggered
  - the session is escalated
  - the fallback topic is triggered
  - the conversational boosting topic is triggered

Once the [session](#) becomes engaged, it remains engaged. An engaged session has one of the following outcomes:

- **Escalated:** A session ends and is considered *escalated* when the **Escalate** topic is triggered or a **Transfer to agent** node is run (the current analytics session ends, whether the conversation transfers to a live agent or not).
- **Resolved confirmed:** A session is considered *resolved confirmed* when **End of Conversation** topic is triggered, and the user confirms that the interaction was a success.
- **Implied resolved:** A session is *implied resolved* when the session is considered completed without user confirmation. What causes an *implied resolved* state depends on whether your agent uses classic or generative orchestration:
  - Classic orchestration: A session is considered *implied resolved* when the **End of Conversation** topic is triggered, and the user lets the session time out without providing a confirmation.
  - Generative orchestration: A session is considered *implied resolved* when a session ends after 15 minutes of inactivity, and there are no remaining active plans. An active plan is a plan that is waiting for a user's input.
- **Abandoned:** A session ends and is considered *abandoned* when an engaged session times out after 15 minutes and didn't reach a resolved or escalated state.

You can also set the outcome for tools with the `conversationOutcome` parameter using the tool code editor. For example, `conversationOutcome: ResolvedConfirmed` for confirmed success or `conversationOutcome: ResolvedImplied` for implied success.

See the guidance documentation [Measuring copilot engagement](#) for suggestions and best practices on how to measure and improve engagement.

## Generated answer rate and quality (preview)

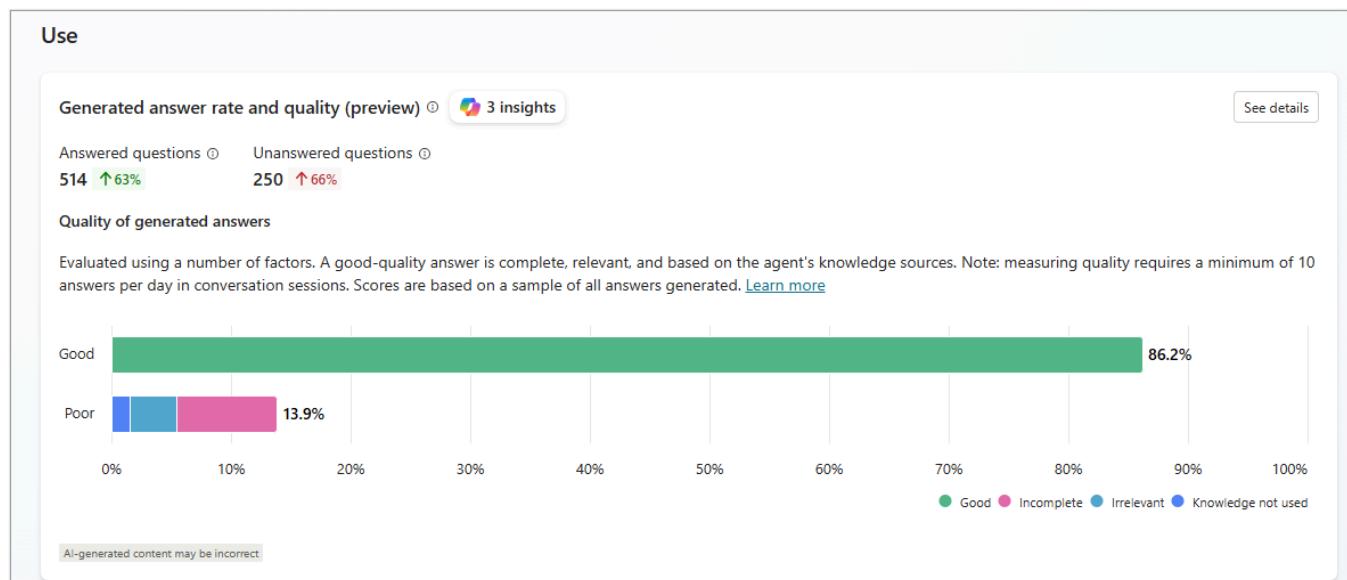
## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

With generative answers, your agent can use AI to generate answers to user queries using knowledge sources and the instructions you provide. However, your agent might not be able to answer all user queries. The **Generated answer rate and quality** section tracks, organizes, and analyzes unanswered queries and answer quality to give you guidance for improving your agent's answering performance.



The **Answer rate** shows the number of answered and unanswered questions within the selected time period, as well as the percentage change over time.

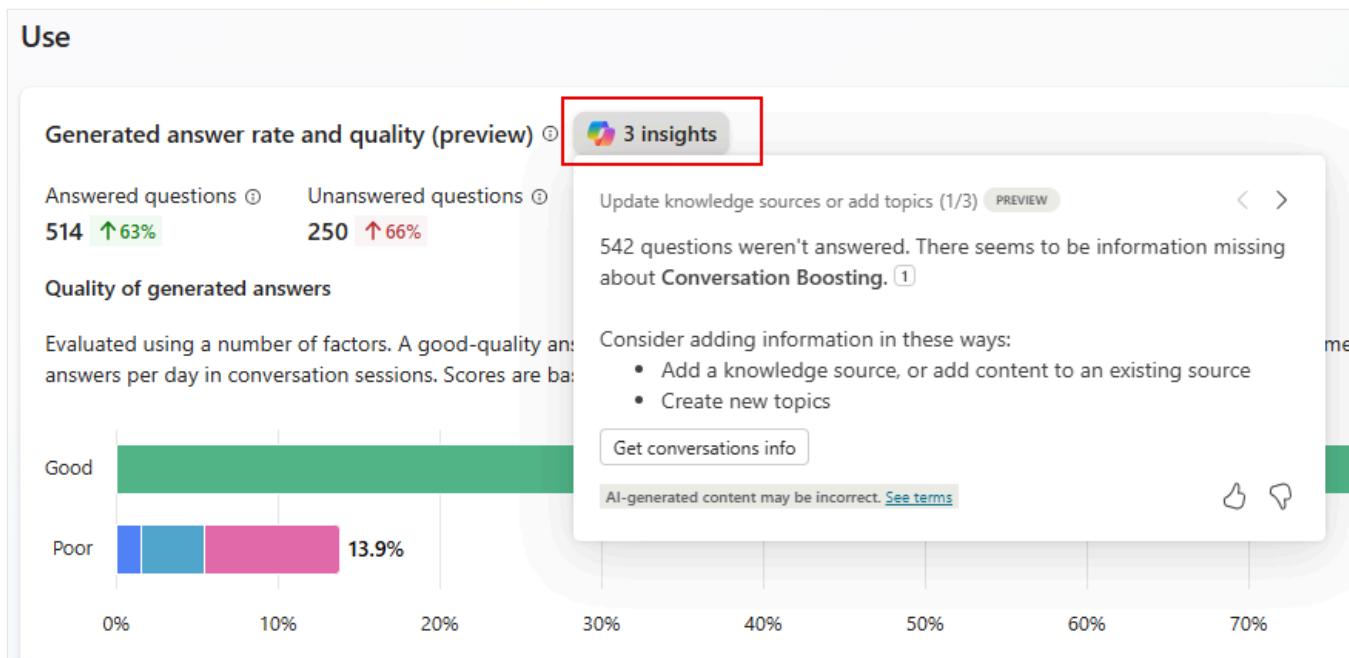
The **Answer quality** measures the quality of answers using AI. Copilot Studio looks at a sample set of answered question and analyzes different quality, including completeness, relevance, and level of groundedness of a response. If the answer meets a set standard, Copilot Studio labels the answer as **Good** quality. Copilot Studio labels answers that don't meet that standard as **Poor** quality. For **Poor** answers, Copilot Studio assigns a reason for the quality rating, and shows the percentage of answers assigned to each category.

Select [See details](#) to open a side panel with question answer rates, knowledge source usage, and error rates over your selected time period. You can use these charts to identify which knowledge sources work well to help users, and which to target for improvements.

- **Unanswered questions** shows the reasons why the agent didn't answer a user query.
- **Knowledge source use** shows the percentage of sessions that used each knowledge source the agent has access to.
- **All sources** shows the percentage of questions that used each knowledge source.
- **Errors** shows the percentage of queries that resulted in a knowledge-related error for each knowledge source type (for example, SharePoint).

## Insights (preview)

Analytics uses AI to provide custom insights about how successfully your agent answers user queries. Copilot Studio analyses unanswered queries, clusters them into groups by theme, then generates contextual insights based on actions that can improve the performance of each group. Copilot Studio generates clusters for this insight type daily, using all unanswered user queries from the last seven days. You can view the top five insights.



1. Select **Insights** in the **Answer rate and quality** section.
2. Use the < and > arrows to switch between insights.
3. Select citations to see more details about the insight.

You can also select **Get conversations info** to download a CSV list of conversation IDs where these unanswered user queries appear.

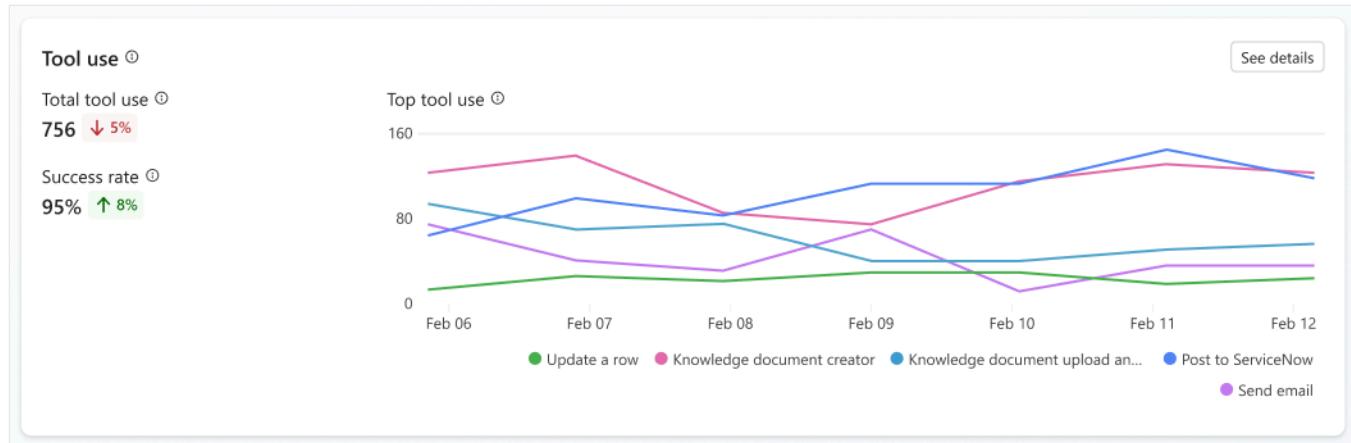
## Tool use

The **Tool use** section shows a chart and metrics that track how often your tools are started over time, and how often your agent used those tools successfully. It also shows trend indicators for

how often your agent uses each tool and the percentage of called tools used successfully.

The chart displays the top five tools used over the date period defined at the top of the **Analytics** page.

To open a side panel with a list of all tools used in the specified time period, along with trend indicators, select **See details** on the chart.



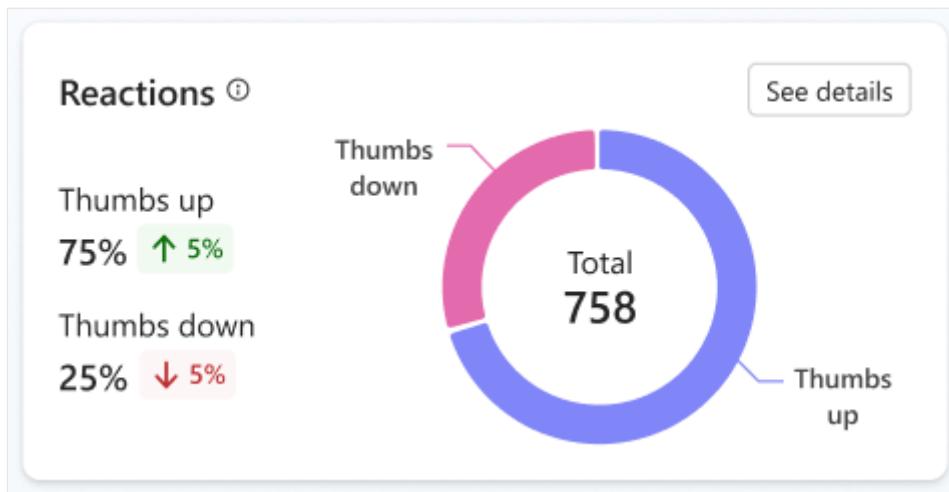
## Satisfaction

The **Satisfaction** section shows user feedback gathered from reactions to agent responses and survey results for a session. Satisfaction is split into two smaller sections: a **Reactions** section, which displays thumbs up, thumbs down feedback for specific agent responses, and a **Survey results** section, which displays the customer satisfaction (CSAT) score for the entire session.

Feedback data is stored in the conversation transcript table in Dataverse. For a list of channels that support this feature, see [Feature details](#).

## Reactions

The **Reactions** section shows user feedback gathered from reactions to agent responses and survey results for a session. The chart counts the number of times users selected either the thumbs up (positive) or thumbs down (negative) buttons available on each response they received from your agent.



After a user provides a reaction, they can leave a comment. To view comments, select **See details**. Select All, Thumbs up, or Thumbs down to filter comments.

User feedback is **On** by default. You can turn off this feature, if desired. You can also add or edit a disclaimer for users about how their feedback is used:

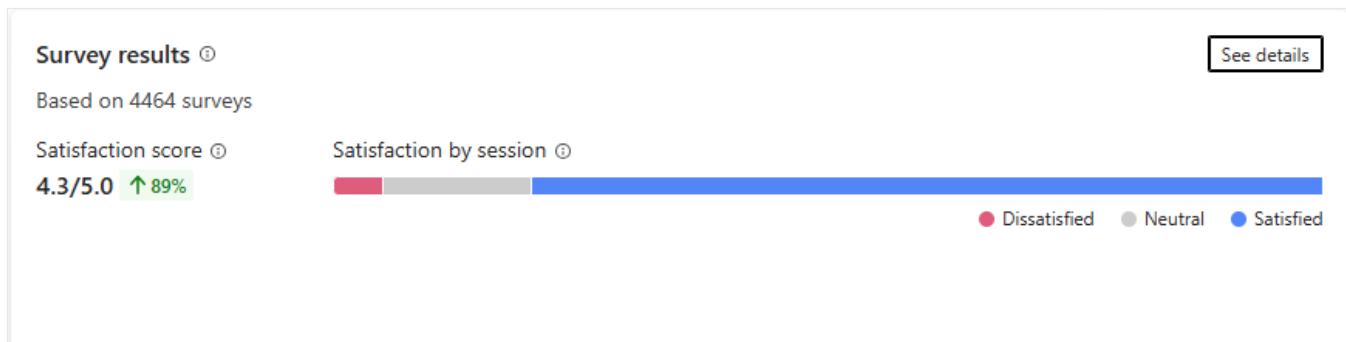
1. Open the agent, then go to **Settings**, and find the **User feedback** section.
2. Turn **Let users give feedback when using this agent** either **On** or **Off**.
3. Add or edit a disclaimer so users know how their feedback is used. You can also provide privacy information and tips.

## Survey results

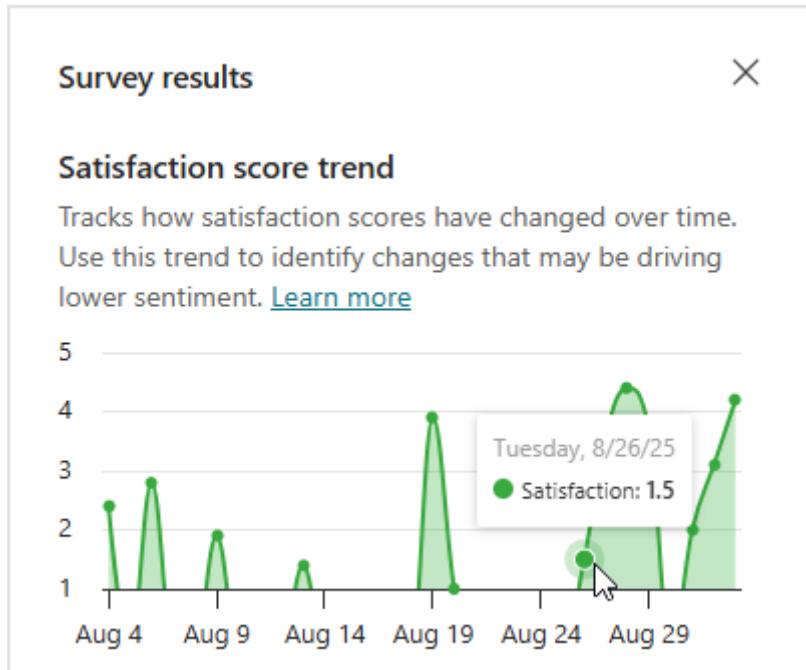
The **Survey results** section shows a chart and metrics that track the average customer satisfaction (CSAT) scores for sessions in which customers respond to an end-of-session request to take a survey.

### ⓘ Note

Scores of 1 and 2 map to **Dissatisfied**, a score of 3 is considered **Neutral**, and scores of 4 and 5 map to **Satisfied**.



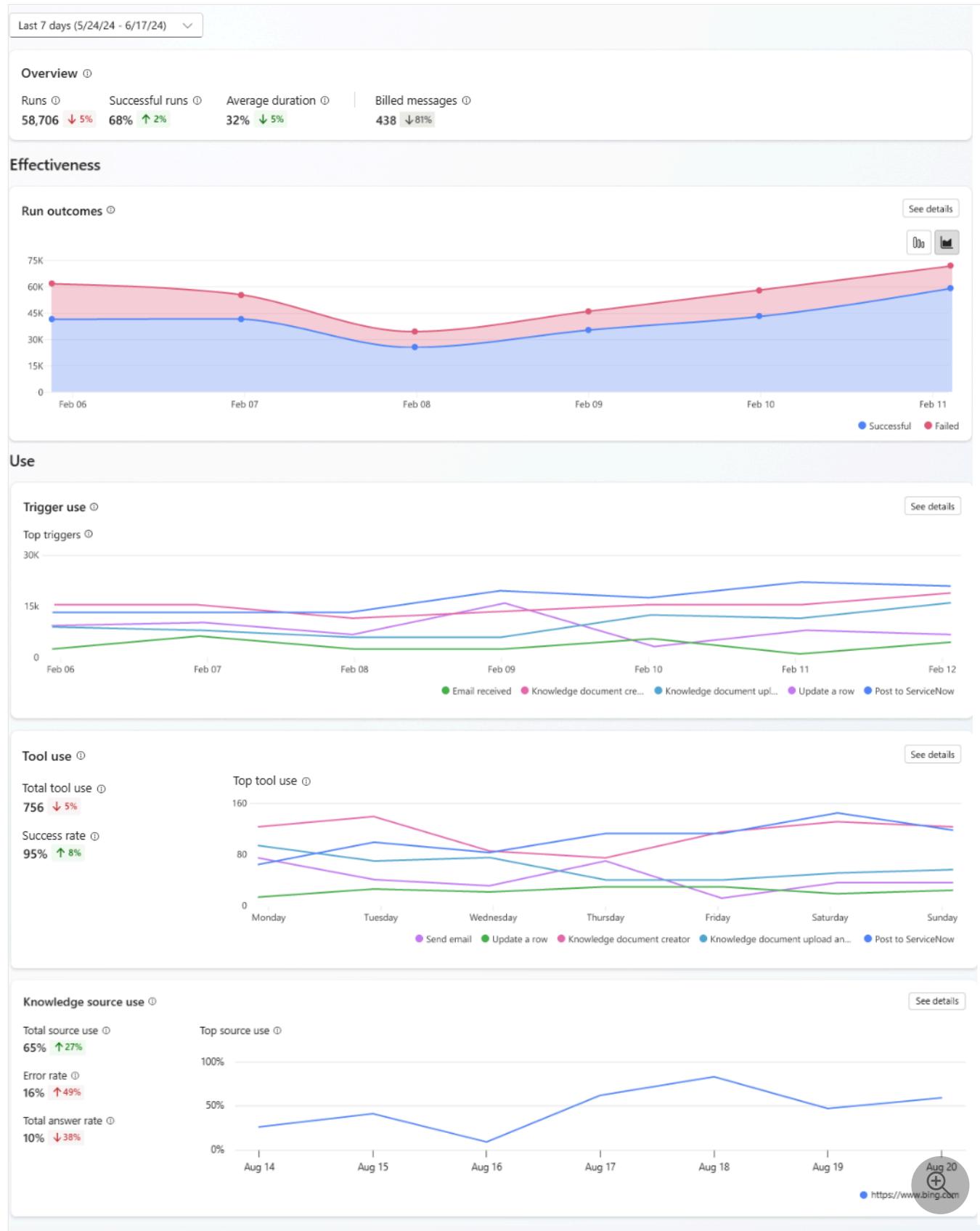
Select **See details** to see how the satisfaction score trends over the report's configured time period.



# Analyze autonomous agent performance

09/18/2025

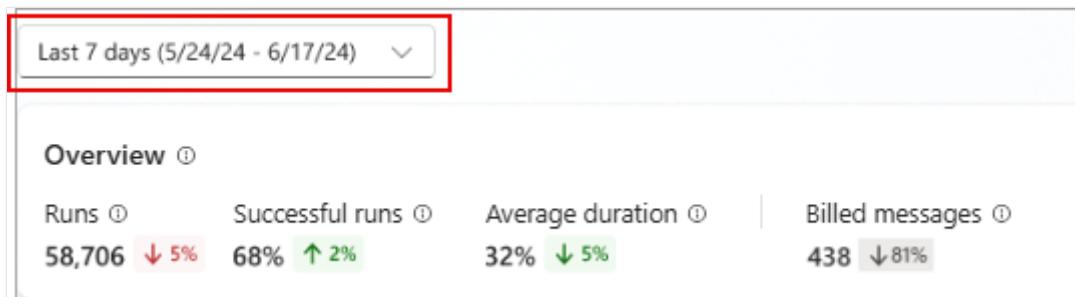
The **Analytics** page in Copilot Studio provides an aggregated insight into the overall health of your agent with **event triggers** across **analytics sessions**.



There are four core areas to focus on when reviewing and improving agent effectiveness:

- **Run outcomes:** Knowing the end result of a session started by a trigger helps you begin to identify where your agent is succeeding and where it needs improvement.
- **Trigger use:** Seeing which triggers are used and how often helps you understand what your agent is doing and why.
- **Tool use:** Learning how often tools are used and how often they succeed can help you understand if those tools are useful and successful.
- **Knowledge source use:** Learning how often individual knowledge sources are used and how often the agent returned errors.

To change the time range, select the range dropdown at the top of the **Analytics** page. You can view analytics for sessions that occurred in the last 90 days.



## Run outcomes

The **Run outcomes** section shows a chart that tracks the type of outcome for [each session](#).

You can also see a breakdown of [session](#) durations. Select **See details** on the Run outcomes chart. A side panel opens with the average duration for successful runs and failed sessions and how the length of sessions changed over time.

## Run outcomes

X

### Duration

Average successful run ⓘ

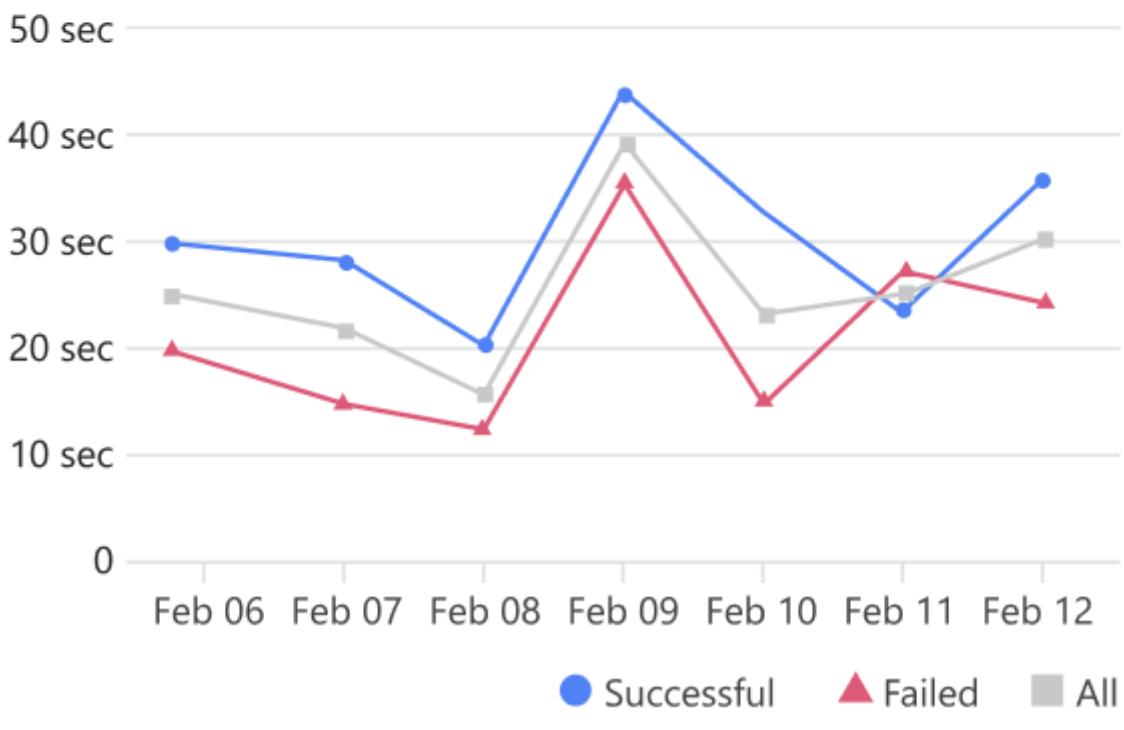
36 sec ↑ 4%

Average failed run ⓘ

24 sec ↓ 2%

### Duration trends

The average duration trends of successful, failed, and total runs. Runs that take a long time might need more specific instructions. [Learn more](#)

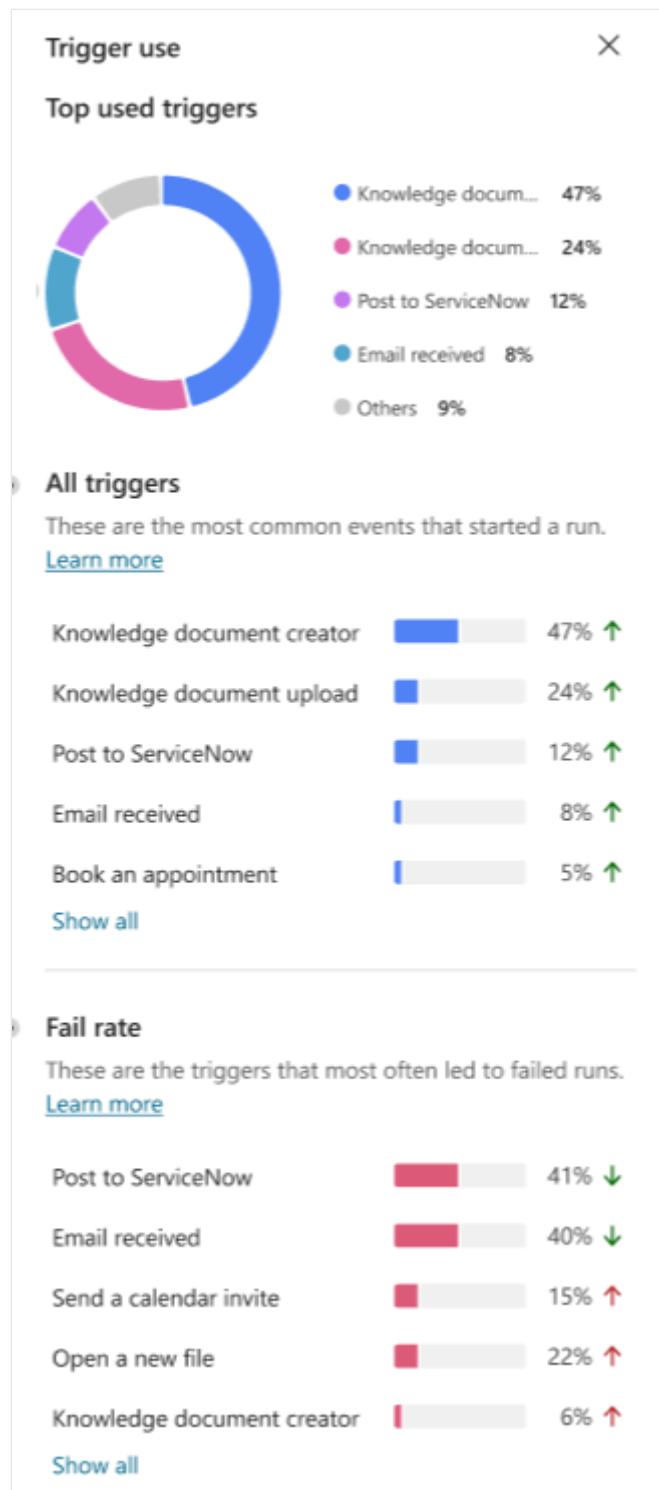


Long session durations can cause delays in workflows your agent is a part of. You can improve your agent's speed by providing more specific instructions at the trigger or agent level.

## Trigger use

The Trigger use section shows a chart and metrics that track how often each trigger initiated a session. The chart shows the change in trigger use over time.

Select **See details** to view a breakdown of all triggers used during the selected time period, the most common events that triggered a run, and which events resulted in the most failed runs.



You can use this information to target specific runs for improvements, and understand which kind of events your agents responds to the most.

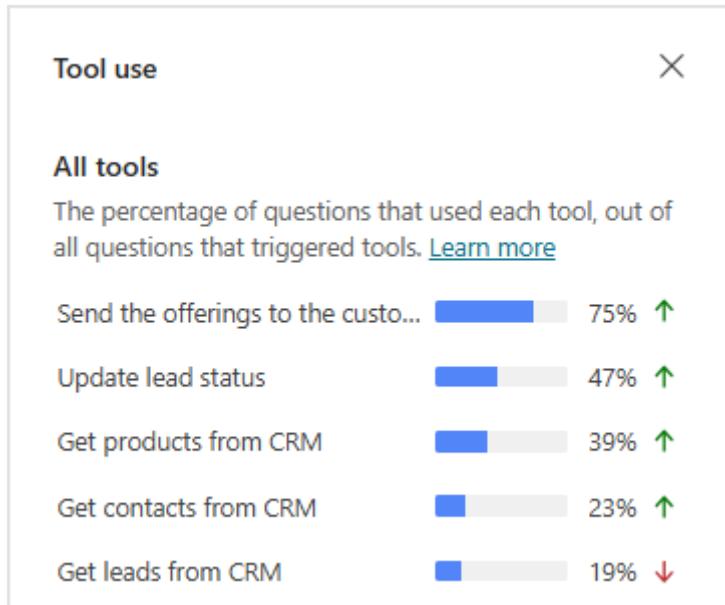
## Tool use

The **Tool use** section shows a chart and metrics that track how often your tools are called over time, and how often your agent uses those tools successfully.

It also shows trend indicators for how often your agent uses each tool and the percentage of called tools used successfully.

The chart displays the top five tools used over the date period defined at the top of the **Analytics** page.

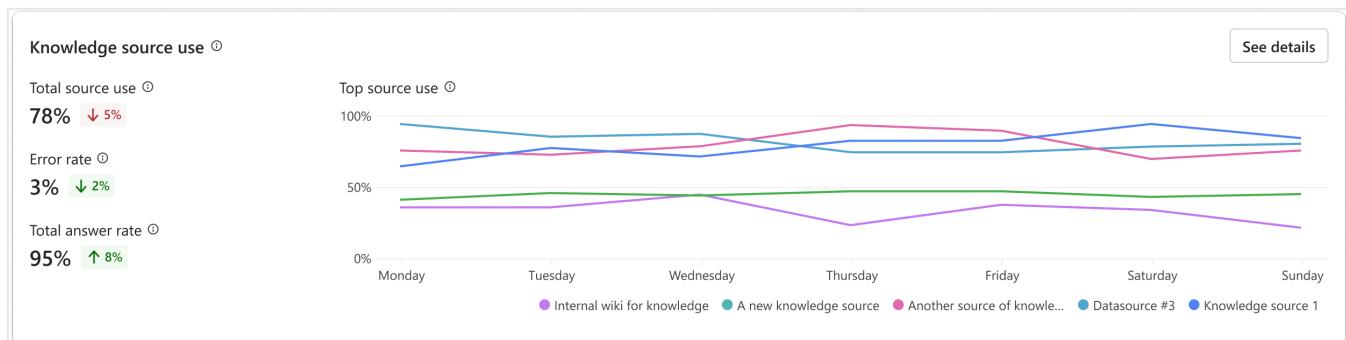
To open a side panel with a list of all tools used in the specified time period, along with trend indicators, select **See details** on the chart.



## Knowledge source use

The **Knowledge source use** section shows a chart and metrics that track how often your **knowledge sources** are used by your agent. It also shows trend indicators for how often your sources are being used, how many errors are being generated, and how many times the source is being used.

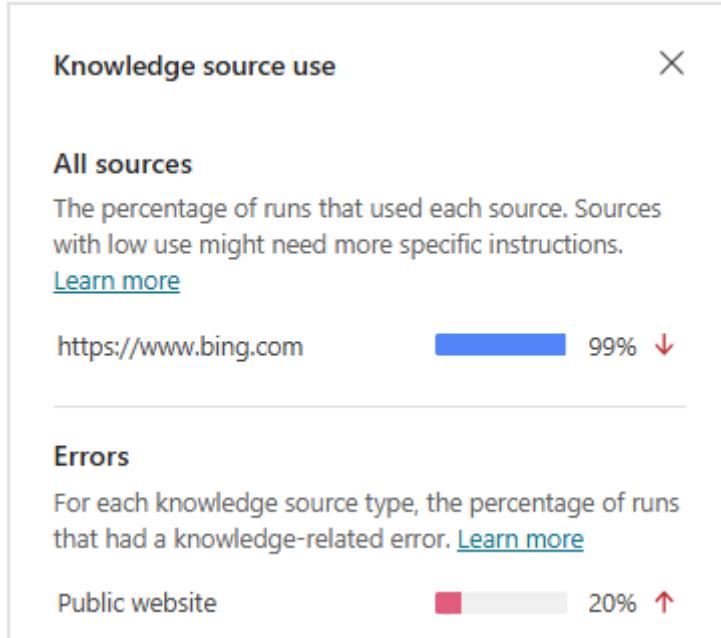
The chart displays the top five knowledge sources used over the date range defined at the top of the **Analytics** page.



Select **See details** to open a side panel with knowledge source usage and error rates over your selected time period. You can use these charts to identify which knowledge sources work well

to help users, and which to target for improvements.

- **All sources** shows the percentage of sessions that used each knowledge source the agent has access to.
- **Errors** shows the percentage of sessions that used each knowledge source type (for example, SharePoint) that resulted in an error.



# Analyze user questions by theme (preview)

10/20/2025

[This article is prerelease documentation and is subject to change.]

Themes are groupings of questions taken from the pool of user questions that trigger generative answers. Themes cluster related user questions into intelligently organized but fewer and more manageable groups. With themes you can gain category-level insights into customer intent, identify frequent topics, and focus efforts on areas needing improvement. You can then drill down to a theme's individual questions and monitor performance at this more granular level to better understand theme-level metrics.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

## Key value proposition

The Themes feature automatically analyzes all unclassified questions that triggered generative AI responses in the past week and suggests themes based on this analysis. This process runs weekly. Between weekly *theme-suggestion* runs, the feature performs daily classification of any new questions that trigger generative AI responses, adding this data to the theme metrics. The suggested Themes list is refreshed regularly so your themes stay relevant and reflect the latest user data. This automatic refresh saves you time by:

- Continuously evaluating theme relevance against new data
- Automatically classifying individual questions into themes
- Automating retrieval of raw metric data for each theme
- Calculating aggregated metric data for each theme

## Prerequisites

To optimize the value of the Themes feature, including being able to drill down to each theme's component questions and answers:

- You need a *Transcript viewer* security role to view the list and its metrics. Only admins can grant this role by [assigning the Transcript viewer security role during agent sharing](#).
- The environment-level settings for storing conversation transcripts in Dataverse are turned on.

## Suggested themes and classified questions

Copilot Studio suggests themes based on records of questions asked in the recent past. By tailoring each suggested theme to suit your organization's structure and customer experience strategy, you can gain better insight through theme-specific key performance indicators (KPIs). This deeper granularity makes it easier to find where agents underperform.

Themes help makers understand what questions users are asking the agent and how the agent performs. On the initial run of Theme-based analytics, Copilot Studio looks at all questions from the past week and suggests up to 10 themes that group user questions based on similarities to one another. Each theme includes metrics such as the total number of questions, answer rate, and user reaction. The displayed data in the Themes list is adjusted based on the selected analytics period.

After the initial run, Copilot Studio classifies new questions to existing themes on an ongoing, daily basis, and generates a fresh list of suggested themes on a weekly basis.

### (!) Note

- There must be at least 50 questions with generative answers logged in the past seven days before Copilot Studio generates suggested themes.
- It's possible for a question that triggers a Generative AI response to be classified into more than one theme. This might become more common in Generative Orchestration mode, where an agent answer can come from several different sources.
- Any questions that remain unclassified are put into an *Unclassified* theme.
- The weekly suggestions run might take a few minutes to complete. During this time, the **Themes** section is unavailable for use.

## The Themes list

By default, the Themes list displays the top five (5) themes, ranked by total number of questions classified to each theme. Select **See all** to expand the list to display all themes.

With automatic weekly *theme suggestion* runs and daily question classification, the Themes feature evolves automatically and adapts to changes in user behavior.

Themes (preview)				<a href="#">See all</a>
Name	Total questions	Answered questions	Reactions	
Product Information	472	72%	<span>👍 221</span> <span>👎 23</span>	
Order Status	460	82%	<span>👍 98</span> <span>👎 329</span>	
Data Export	454	88%	<span>👍 169</span> <span>👎 166</span>	
Subscription Management	398	82%	<span>👍 110</span> <span>👎 46</span>	
Shipping Information	374	31%	<span>👍 8</span> <span>👎 331</span>	

AI-generated content may be incorrect

The Themes list displays metrics for each listed theme.

expand table

Column	Description
Name	Name of the theme.
Total questions	Total number of questions classified into this theme for the configured time period.
Answered questions	Proportion of <b>Total questions</b> that resulted in an answer to the user question.
Reactions	Raw count of positive ( <i>thumbs up</i> ) and negative ( <i>thumbs down</i> ) reactions from the questions classified to each theme.

### !Note

*Why are there 'Unanswered questions' in my theme when all classified questions triggered generative AI answers?*

By definition, only questions that triggered a generative AI answer can be classified into a theme. But triggering a call to generative answers isn't the same as having an answer to a user query.

For example, it's common for there to be a conversational boosting topic connected to a knowledge source. In this case, if an intent isn't recognized, this topic might be triggered

which calls the knowledge source even though there's no answer to the question in that knowledge source.

## Drill down on a theme

To drill down to the questions grouped into any one theme, select the theme in the Themes list. When you drill down, a panel specific to the selected theme opens up, displaying metrics at the levels of both the theme and individual questions.

### ➊ Note

The Themes list includes:

- Up to the most recent 100 questions classified into each theme.
- Questions from the last 30 days, regardless of the configured time period. If the configured period is outside the last 30 days (either fully or partially), Copilot Studio displays a notification that the list is unavailable.

## Shipping Information

Created: 10/8/25

AI-generated content may be incorrect

Questions related to managing personal or business accounts, including balance inquiries, statements, transfers, account setup, password changes, and card services.

Total questions	Answered questions	Unanswered questions	Reactions
194	33%	67%	20  3

Show more

### User questions ①

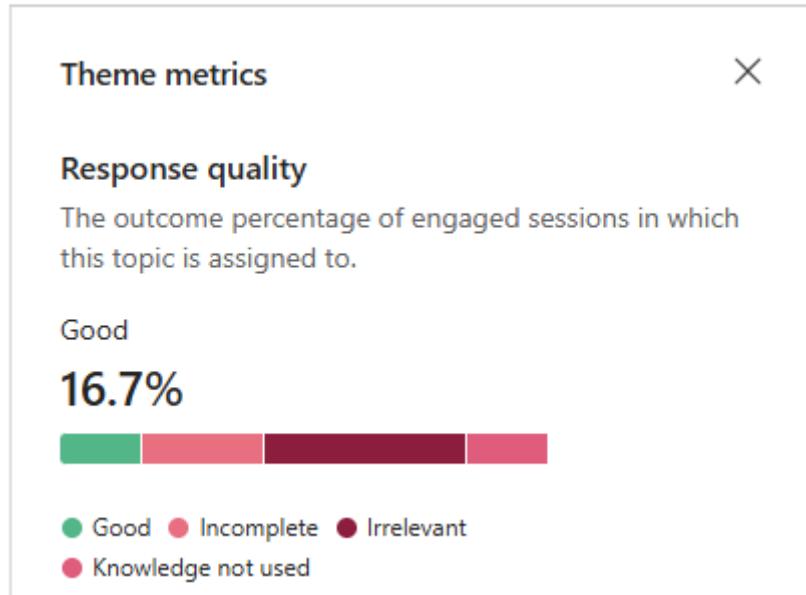
Question	Answered?	Response Quality	Reaction
How do I set up Samsung Pay with my account?	<span>Unanswered</span>	--	
How do I open a business checking account?	<span>Answered</span>	Poor	--
What are the requirements for a home loan?	<span>Answered</span>	Good	
How do I speak to a customer service representative?	<span>Answered</span>	Good	--

In a theme's drill-down panel, metrics at the theme level are displayed first.

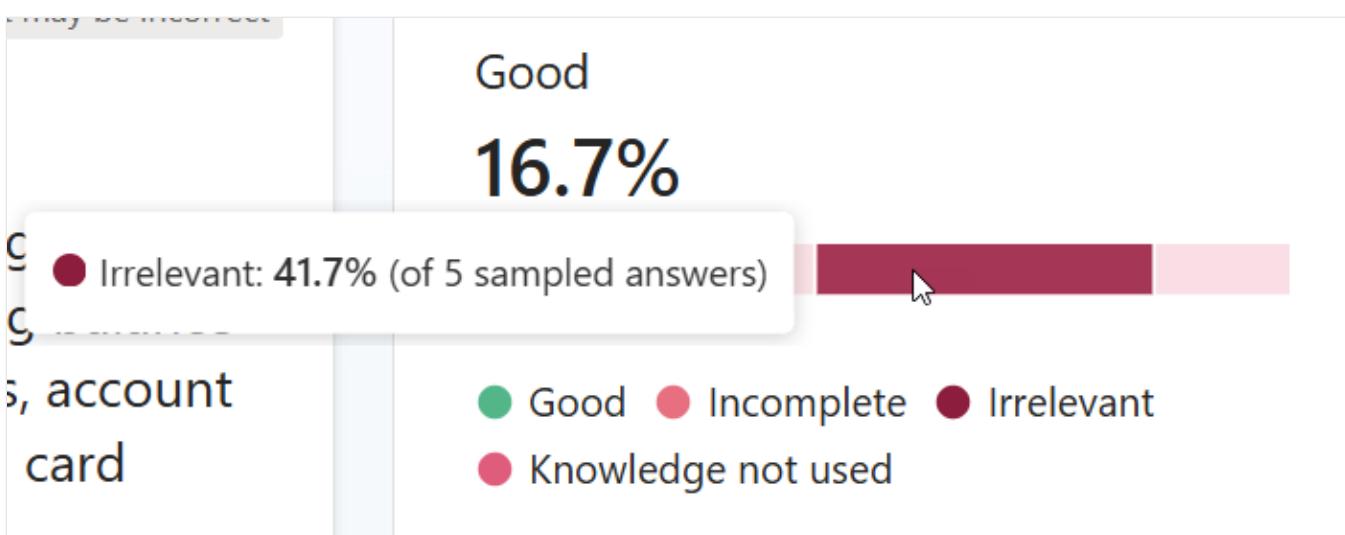
Expand table

Theme-level metric	Description
Total questions	Total number of questions (classified into this theme) that triggered generative AI responses for the configured time period.
Answered questions	Proportion of this theme's <b>Total questions</b> that resulted in an answer to the user question.
Unanswered questions	Proportion of this theme's <b>Total questions</b> that didn't result in an answer to the user question.
Reactions	Raw count of positive ( <i>thumbs up</i> ) and negative ( <i>thumbs down</i> ) reactions from the questions classified to this theme, as recorded in the Learn Feedback mechanism.

To display a breakdown of the quality of response for a sampling of answered questions in this theme, select **Show more**. The breakdown is displayed as a stacked bar chart with relative weightings.



Hover over an individual bar segment to see more information in a tooltip about any one response quality value. The tooltip includes the value of that segment's relative weighting and the number of questions sampled to arrive at that value.



The **User questions** section lists all questions that triggered generative AI responses in the configured time period and were classified into this theme, and displays metrics on a *per question* basis. Question-level metrics include:

[Expand table](#)

Question-level dimension or metric	Description
Question (dimension)	Question (that triggered a generative AI response) classified to this theme.
Answered?	Response to the question of whether the user query was answered or not. Possible values are <code>Answered</code> and <code>Unanswered</code> .
Response quality <sup>1</sup>	Response to the question of what the quality of response to the user query was. Possible values are <code>Poor</code> and <code>Good</code> .
Reaction	Two parameters, being the Boolean <i>Thumbs up / Thumbs down</i> metric, and a text-based <i>Comment</i> parameter.

<sup>1</sup> Response quality isn't always available and shows up only if the question was sampled for quality analysis.

Select any question in the list to display more information about the question in a **Question details** panel. This panel includes:

[Expand table](#)

Detailed question-level parameter	Description
Time stamp	Time stamp of the most recent instance of when the question was answered.
User query	The question asked.

Detailed question-level parameter	Description
Response	The generative AI response.
Thumbs up / Thumbs down value	Value for the most recent instance of this question for the <i>Thumbs up / Thumbs down</i> parameter.
Comment	Optional additional text-based comment, provided by the user, in addition to the <i>Thumbs up / Thumbs down</i> reaction.
Response quality <sup>1</sup>	Response to the question of what the quality of response to the user query was. Possible values are <code>Poor</code> and <code>Good</code> .
Reason <sup>2</sup>	In the case where the response quality is <code>Poor</code> , this parameter explains why.

<sup>1</sup> Response quality isn't always available and shows up only if the question was sampled for quality analysis.

<sup>2</sup> If the Response quality is `Good` or if the question wasn't sampled, the Reason parameter isn't assigned any value.

The screenshot displays the Lead Manager interface. On the left, the 'Analytics' tab is selected, showing a card for 'Shipping Information' with metrics: Total questions (194), Answered questions (33%), Unanswered questions (67%), and Reactions (20 thumbs up, 3 thumbs down). Below this, a table lists user questions with their status (Answered? and Response Quality) and reactions. One question, 'What are your business hours?', is highlighted with a red box. On the right, a modal window titled 'Question details' provides specific information for this question: User query ('What are your business hours?'), Response ('Our business hours are Monday to Friday, 9 AM to 5 PM.'), and a reaction note ('Thumbs up Great answer!'). It also shows the Response quality as 'Poor' and the Reason as 'Irrelevant'.

Question	Answered?	Response Quality	Reactions
How do I set up Samsung Pay with my account?	Unanswered	--	20 ⬆️ 3 ⬇️
How do I open a business checking account?	Answered	Poor	--
What are the requirements for a home loan?	Answered	Good	⬆️ ⬇️
How do I speak to a customer service representative?	Answered	Good	--
How do I apply for a credit card?	Unanswered	--	⬆️ ⬇️
What are your business hours?	Answered	Poor	⬆️ ⬇️
Why can't I log into my mobile app?	Answered	Poor	⬇️ ⬇️

# Analyze topic usage in Copilot Studio

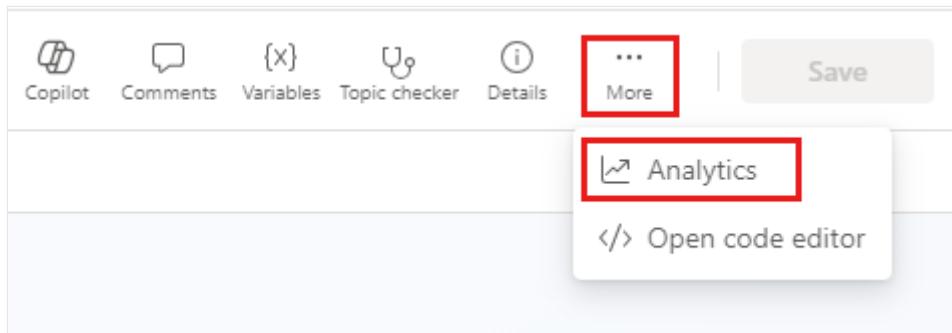
08/06/2025

The topic analytics pane provides a view into the performance of an individual topic and how it can be improved.

## ⚠ Note

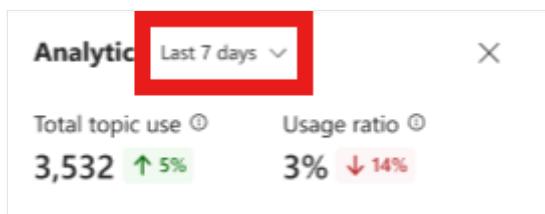
Topics analytics is available for agents in classic mode only.

To display the topic analytics pane, open the desired topic from the [Topics page](#), then select **More > Analytics**.



Analytics are only available for topics associated with an [analytics session](#).

To change the time range, select at the top of the topic analytics pane.



The topic analytics pane shows three charts or metrics:

## Analytics Last 7 days ×

Total topic use ⓘ

3,532 ↑ 5%

Usage ratio ⓘ

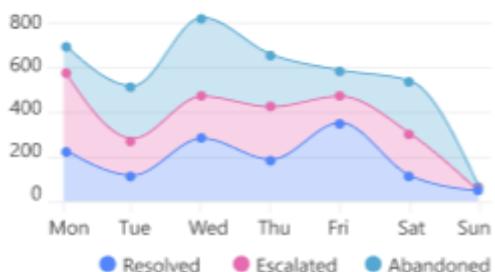
3% ↓ 14%

### Topic outcomes

The outcomes percentage of engaged sessions in which this topic was triggered. [Learn more](#)



Total topic use ⓘ



- The total number of times the topic was associated with a session, and how often the topic appeared as a percentage of all topics associated with a session
- A chart showing the aggregated breakdown by **outcome** for the selected period
- A chart showing the **outcomes of the sessions** where the topic appeared over time

# View agent's billing consumption

09/19/2025

The [Analytics](#) page in Copilot Studio provides consumption data on the billed Copilot Credits or sessions your agent uses.

- If your billing is based on [Copilot Credits consumption](#), these analytics show how many Copilot Credits your agent consumes and what activity is counting towards Copilot Credits consumption over a time period.
- If your billing is based on [session consumption](#) (available for licenses purchased before January 1, 2024), these analytics show how many billed sessions your agent consumes over a given time period.

## Copilot Credits Consumption

[Copilot Credits](#) are the unit that measures agent usage. The total cost is calculated based on the sum of the Copilot Credits used by your organization. The number of Copilot Credits consumed by an agent depends on the design of the agent, how often customers interact with it, and the features they use.

### !Note

If your organization uses an older subscription plan using billed sessions instead of Copilot Credits, the [Analytics](#) page shows the number of billed sessions. If your organization switches from billed sessions to Copilot Credits during the time period you're analyzing, both data on both billed sessions and Copilot Credits appear.

On the [Analytics](#) page for your agent, the [Overview](#) panel shows your agent's total billed Copilot Credits for the selected time period. To see more details, select [See billing](#).

The screenshot shows the 'Overview' section of the Analytics page. At the top, there is a dropdown menu set to 'Last 7 days (6/17/2025 - 6/23/2025)'. Below this, the 'Overview' section displays four metrics: 'Conversation sessions' (1,425, up 12%), 'Engagement' (77%, down 23%), 'Satisfaction score' (4.3/5.0, up 24%), and 'Billed messages' (418, down 88%). A red box highlights the 'See billing' button located in the top right corner of the overview section.

To change the time period, select the time period dropdown at the top of the [Analytics](#) page.

Last 7 days (6/17/2025 - 6/23/2025) ▾

[See billing](#)

## Overview ⓘ

Conversation sessions ⓘ  
1,425 ↑ 12%

Engagement ⓘ  
77% ↓ 23%

Satisfaction score ⓘ  
4.3/5.0 ↑ 24%

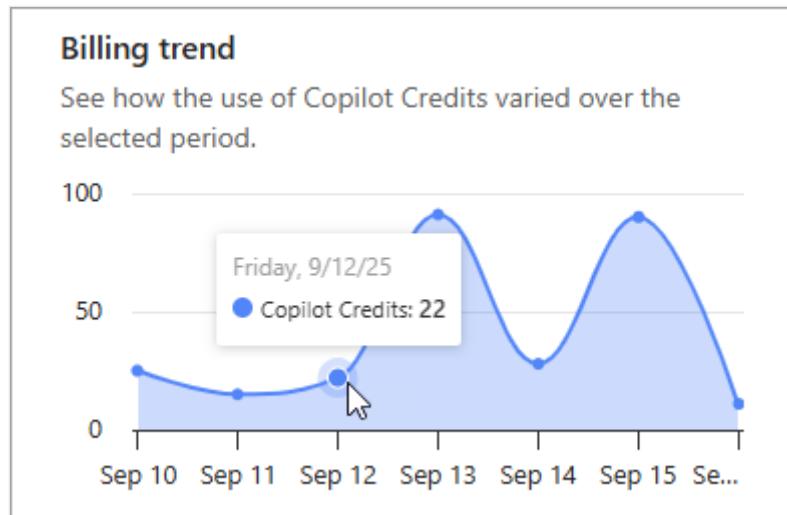
Billed messages ⓘ  
418 ↓ 88%

The analytics data takes a few hours to update with new billed Copilot Credits, so you might not see data about recent activity.

## Billing trend

The **Billing trend** chart shows the changes in your agent's billed Copilot Credits over the selected time period.

Hover over points in the chart to see the number of Copilot Credits.



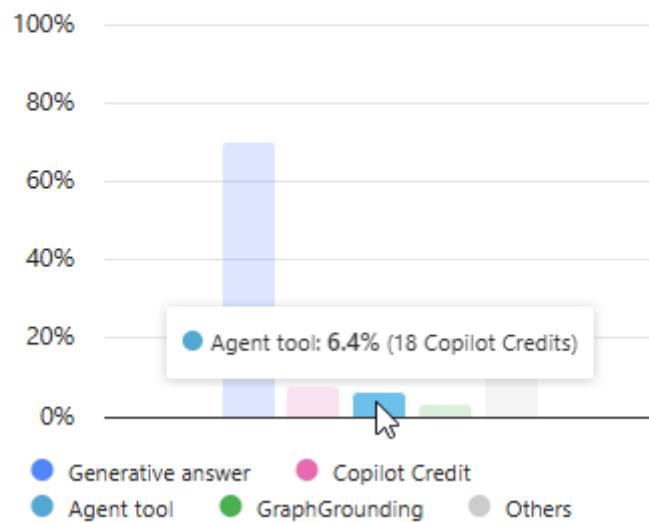
## Cost distribution

The **Cost distribution** chart shows which kinds of activity are counting towards Copilot Credits usage. The chart uses the total number of Copilot Credits for each activity, and each activity can consume different numbers of Copilot Credits per occurrence. For more information on how different activities consume Copilot Credits, see [Billing rates and management](#).

Hover over columns in the chart to see the number of Copilot Credits.

## Cost distribution

How different events contributed to the cost of your agent during this period. [Learn more](#)



## Monthly Credit limit

The **Monthly Credit limit** visualizes as a stacked bar chart how many Copilot Credits were consumed ("used") in the calendar month and how many remain. Use this section to determine if your agent is within, near, or exceeding budgeted limits. Hover over bar segments to see the number of Copilot Credits **Used** and **Remaining**.

### Monthly Credit limit

Copilot Credits used this calendar month, and how many credits you have left.

Credits available ⓘ

400/1000



Credits available ⓘ

400/1000

Remaining 600 Copilot Credits



Used

Rem

## Session consumption

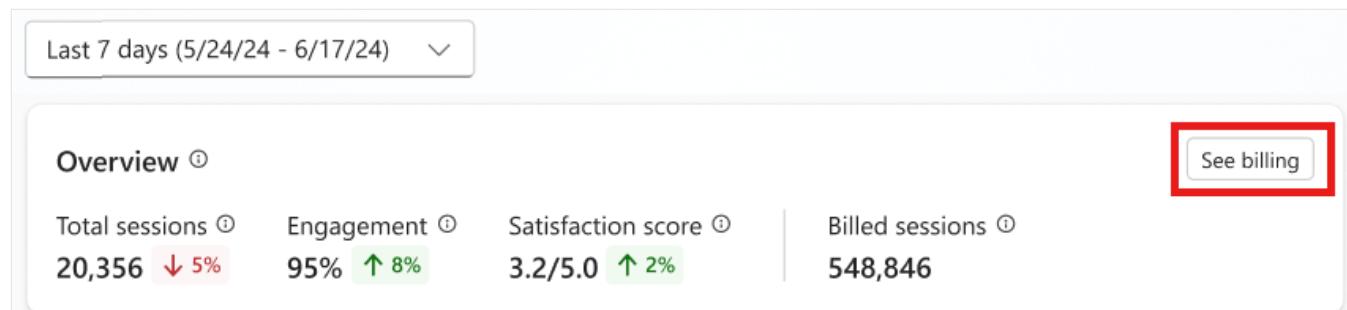
A [billed session](#) is an interaction between a customer and an agent. It represents one unit of consumption. You can use the [Analytics](#) page to view how many sessions your agent consumed during the configured time period. Billed sessions are different from [conversational or event trigger sessions](#), and are [calculated differently](#).

If your organization switches from billed sessions to Copilot Credits during the time period you're analyzing, data for both billed sessions and Copilot Credits appear.

#### Note

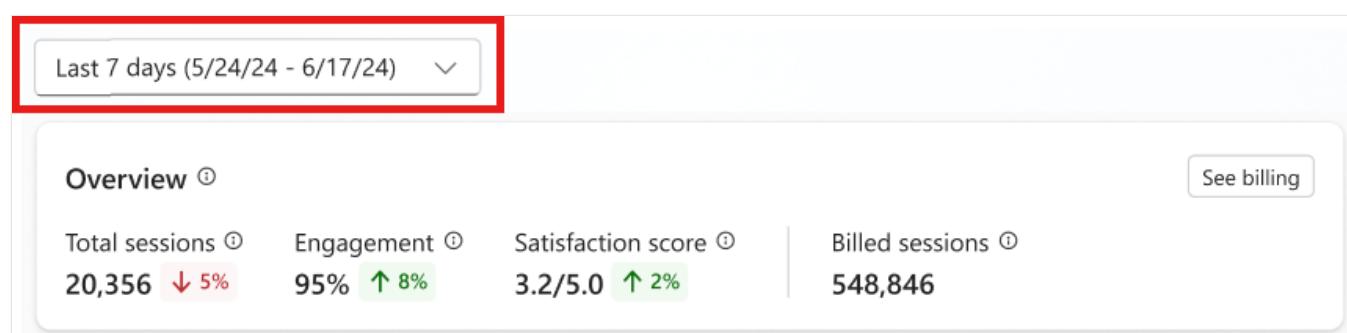
This section is for billed sessions in the legacy Power Virtual Agents license, which was available for purchase starting on December 1, 2023. This legacy license is no longer available for purchase starting on January 1, 2024.

On the [Analytics](#) page for your agent, the [Overview](#) panel shows your agent's total billed sessions for the selected time period. To see more details, select [See billing](#).



The screenshot shows the 'Overview' panel of the Analytics page. At the top, there is a dropdown menu set to 'Last 7 days (5/24/24 - 6/17/24)'. Below the dropdown, there are four performance metrics: 'Total sessions' (20,356, down 5%), 'Engagement' (95%, up 8%), 'Satisfaction score' (3.2/5.0, up 2%), and 'Billed sessions' (548,846). A red box highlights the 'See billing' button located in the top right corner of the panel.

To change the time period, select the time period dropdown at the top of the [Analytics](#) page.



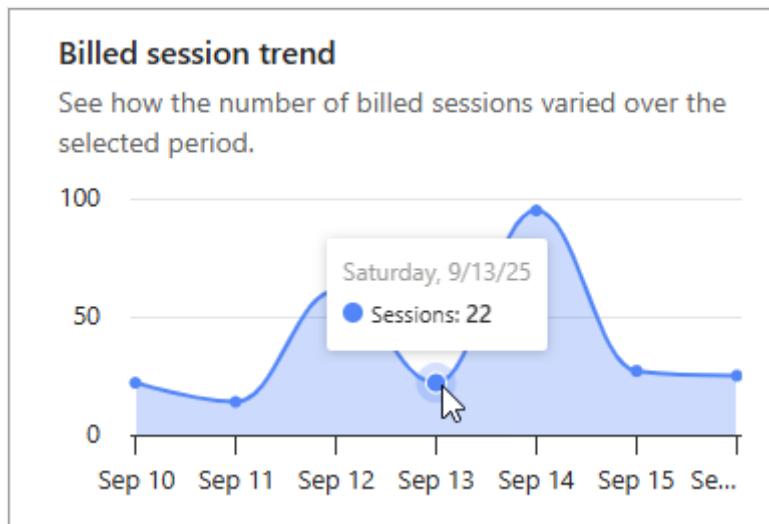
The screenshot shows the 'Overview' panel of the Analytics page. The 'Last 7 days (5/24/24 - 6/17/24)' dropdown is highlighted with a red box. Below the dropdown, there are four performance metrics: 'Total sessions' (20,356, down 5%), 'Engagement' (95%, up 8%), 'Satisfaction score' (3.2/5.0, up 2%), and 'Billed sessions' (548,846). A red box also highlights the 'See billing' button in the top right corner.

The analytics data takes a few hours to update with new billed session counts, so you might not see data about recent activity.

## Billed session trend

The [Billed session trend](#) chart shows the changes in billed sessions over the selected time period.

Hover over points in the chart to see exact session counts.



# Analyze time and cost savings for agents

09/19/2025

In Copilot Studio, you can calculate how much time or money an agent saves compared to other methods. With this information, you can estimate an agent's business impact, identify and prioritize successful agents, and track return on investment (ROI) over time.

Use the **Savings calculator** to estimate your savings per run or to calculate the savings of one or more of the tools your agent uses.

The screenshot shows the Copilot Studio Analytics page. On the left, the **Overview** panel displays metrics: Runs (58,706, -5%), Successful runs (68%, +2%), Average duration (30 sec, +3%), and Copilot Credits used (305, +21%). On the right, the **Savings** panel is highlighted with a red box, showing Time (387 hrs, +5%) and Cost (\$12,771, +8%).

## ⓘ Note

Admins can choose to turn off cost savings using the **Disable money saving rules** setting. This is an environment-level setting managed in the Power Platform admin center by the tenant administrator. Makers can still calculate time savings with cost savings turned off.

1. On the **Analytics** page for your agent, if a **Savings** panel doesn't appear, select the **More** icon (...) in the **Overview** panel and select **Add savings**.

The screenshot shows the Copilot Studio Analytics page. At the top, there are buttons for **Publish**, **Settings**, and a three-dot menu. Below that, there are metrics: Successful runs (53%, +97%) and Average duration (21 sec, +85%). To the right of these metrics is a button labeled **... See billing**. A red box highlights the **Add savings** button, which is located at the bottom right of the screen.

If there's an empty **Savings** panel with no calculated metrics displayed, select **Calculate savings**.

## Savings ⓘ

**Calculate savings**

...



Start tracking savings to measure the time and money that your agent saves.

If there's already a **Savings** panel with a configured savings calculation displayed, select the **More** icon (...) and select **Edit**.

### ⓘ Note

When you edit the savings calculation, this operation replaces the existing calculation with a new one.

2. On the **Ways to calculate your savings** panel, select **Savings per run** if you want to estimate savings on a *per run* basis. If your agent is a conversational agent, select the **Savings per tool** tile.

## Estimate savings per run for an agent

Use the **Savings calculator** to estimate savings of time or money of your agent, running from activation to completion of all activated tools, as compared to other methods on a *per run* basis. A *per-run* calculation gives you a general, quick estimate of your agent's ROI.

1. On the **Calculate savings per run** panel, enter the required information depending on your needs:

- To estimate time savings, enter the estimated amount of time required for another method to complete the agent's task, and select **Calculate**.

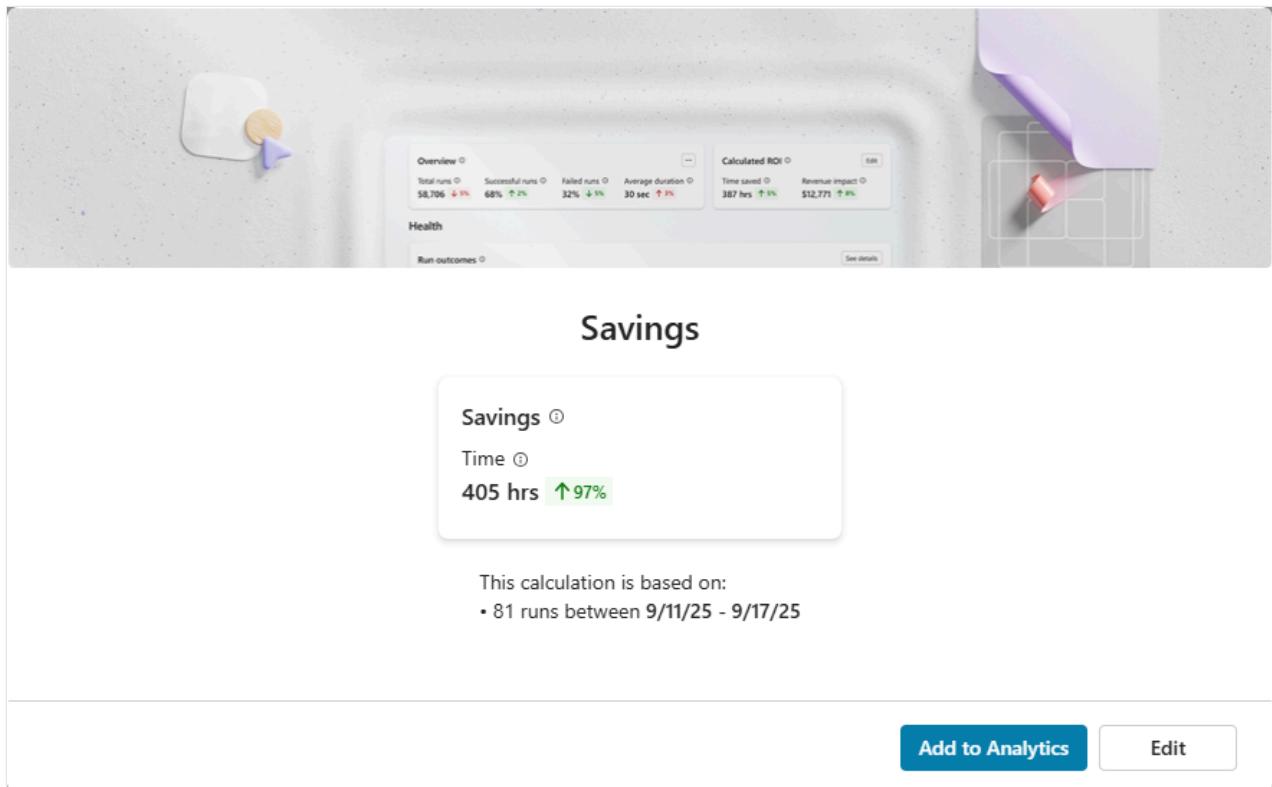
If this is a new estimation to replace an existing one on your **Analytics** tab, select **Recalculate**.

The calculator returns the estimated work time the agent saved for the time period, on a *per run* basis.

- To estimate cost savings, enter the estimated cost required for another method to complete the task of the agent, select the currency you're measuring in, and select **Calculate**.

If this is a new estimation to replace an existing one on your **Analytics** tab, select **Recalculate**.

The calculator returns the estimated cost the agent saved for the time period, on a *per run* basis.



2. Select **Add to Analytics** to add this estimation to your **Analytics** tab. If this is a new estimation to replace an existing one on your **Analytics** tab, select **Update**.

## Calculate savings per tool for your agent

Use the **Savings calculator** to calculate time or cost saved using different measurements for each of your agent's tools, as compared to another method. This information gives you a more detailed estimate of your agent's ROI based on the different tasks your agent performs.

1. On the **Calculate savings per tool** panel, for each tool you would like to calculate savings on, enter the required information depending on your needs:

- To calculate time savings, enter the estimated amount and unit of time the other method would take to complete the tool's task, and then select **Calculate**.

If this is a new estimation to replace an existing one on your **Analytics** tab, select **Recalculate**.

The calculator returns the estimated work time the agent saved, for the specified tool. Hover over the % increase or decrease calculated metric to compare this result

to the previous time period.

- To calculate cost savings, enter the estimated cost it would take for the other method to complete the tool's task, and select **Calculate**.

If this is a new calculation to replace an existing one on your **Analytics** tab, select **Recalculate**.

The calculator returns the estimated cost the agent saved, for the specified tool.

Hover over the % increase or decrease calculated metric to compare this result to the previous time period.

The screenshot shows the 'Analytics' tab in the Agent interface. At the top, there's an 'Overview' section with metrics like Total runs (58,705), Successful runs (68%), Failed runs (32%), and Average duration (30 sec). To the right is a 'Calculated ROI' section showing Time saved (387 hrs) and Revenue impact (\$12,771). Below this is a 'Health' section with 'Run outcomes' and a 'See details' button. A large callout box highlights the 'Savings' section, which displays 1,581.9 hrs saved, with an 8% increase compared to the previous period. Below the savings, it says 'This calculation is based on: 81 runs between 9/11/25 - 9/17/25, 3 tools used'. At the bottom right of the main area are 'Add to Analytics' and 'Edit' buttons.

2. Select **Add to Analytics** to add this calculation to your **Analytics** tab. If this is a new calculation to replace an existing one on your **Analytics** tab, select **Update**.

## View savings from the Analytics tab

Once a savings calculation or estimation exists and is included in your analytics, you can see savings insights directly from the **Analytics** tab.

The screenshot shows a 'Savings' panel with the following details:

- Savings**: Shows a green icon.
- Time**: Shows 18,431.7 hrs with a green up arrow icon and +80%.
- See details**: A button to view more information.
- More options**: A three-dot menu icon.

If your **Analytics** tab shows a savings estimation on a *per run* basis, the **Savings** panel shows the estimated time savings if the work is done by a person instead of automation. The percent increase or decrease indicates the degree to which this estimation differs since the previous time range.

If the tab shows a savings calculation based on tools, you can see more savings insights on a *per tool* basis by selecting **See details**.

The screenshot shows a 'Savings' panel with two entries:

- Get contacts from CRM**: Shows 13,013 hrs with a green up arrow icon and +60%.
- Get leads from CRM**: Shows 490 hrs with a red down arrow icon and -93%.

Each entry includes a brief description and a 'Time' section with the calculated hours and a percentage change.

## Modify or remove savings calculations

You can modify the estimates used by the savings calculations, or remove them from your agent's **Analytics** page.

1. Go to your agent's **Analytics** page.
2. In the **Overview** section, select the **More** icon (...), then **Edit** to change estimates, or **Remove** to stop the Savings section from appearing.

# Understand downloaded session data from Copilot Studio

09/16/2025

You can view, export, and download sessions of customer interactions with your agent in both Power Apps and Copilot Studio.

When you download session information from Copilot Studio, you get a subset of the information you would get downloading from Power Apps, in the form of a comma-separated value (CSV) file. The CSV file includes the actual conversation and some metrics and information about the session (see [Work with session transcripts](#) for a list of metrics included in the CSV file). To see all activities, download transcripts through [Power Apps](#).

In the `ChatTranscript` field, there is a limit of 512 characters for each bot response. Any bot responses longer than 512 characters show only the first 512 characters in the CSV file. The 512-character limit is on a *per response* basis, not a *per session* basis, and you can have multiple bot responses in a given session.

## Important

Makers with the *Environment maker* role don't automatically have access to transcripts.

You can't download transcripts [if your admin has turned transcript access control on](#).

By default, you can download up to seven days of agent conversation transcripts from the past 29 days directly from Copilot Studio.

## Note

Agent responses that use SharePoint as a knowledge source aren't included in conversation transcripts.

Conversation transcripts aren't written for:

- Microsoft Dataverse for Teams environments
- Dataverse developer environments
- Microsoft 365 Copilot agents

## Prerequisites

To view transcripts:

- Your agent's data must include sessions for at least one conversation (that is, your data can't be only trigger-based runs).
- You need the transcript viewer security role. Only admins can grant this role by [assigning the Transcript viewer security role during agent sharing](#).

## Download agent session transcripts

1. Open the desired agent.
2. On the top menu bar, select **Analytics**.
3. Select the date range you want to download.
4. Above the **Overview** card, select **Download Sessions**.

Alternatively, under **Effectiveness**, select **See details** on the **Conversation outcomes** card, and then select **Download sessions**.

5. On the **Download Sessions** pane, select a row to download the session transcripts for the specified time frame.

### ⓘ Note

Sessions are split into rows representing one day increments with up to 50 thousand sessions each.

For classic chatbots: If the specified date range includes a date that falls within the last 24 hours, a single row represents up to 2,500 sessions. Otherwise, sessions are split into rows that represent one day increments with up to 2,500 sessions each.

The download starts immediately. The file is saved to your browser's default download location.

## Work with session transcripts

A session transcript file is a comma-separated values (CSV) file that contains the following information:

- **SessionID**: A unique identifier per session.

- `StartTime`: Time at which the session started. Entries are sorted by this column in descending order.
- `InitialUserMessage`: First message typed by the user.
- `TopicName`: Name of the last authored topic that was triggered in this session.
- `ChatTranscript`: Transcript of the session in the following format: "User says: <UserInput>; Bot says: <CopilotResponse>;".
  - Conversation turns are separated by semicolons.
  - <*CopilotResponse*> doesn't include the options presented to the user.

Example: User says: store hours; Bot says: Which store are you asking about?; User says: Bellevue; Bot says: Bellevue store is open from 10am to 7pm every day.;

- `SessionOutcome`: Outcome of the session (Resolved, Escalated, Abandoned, Unengaged).
- `TopicId`: A unique identifier of the last authored topic triggered in this session.
- `OutcomeReason`: A unique identifier of the last authored topic triggered in this session.
- `Turns`: The number of turns.
- `Channel`: The channel this session was on.
- `URL`: The channel URL (for Web channel).
- `CSAT`: The customer satisfaction (CSAT) score for the session, if that score exists.
- `Comments`: List of all user comments for that session.

# Understand downloaded conversation transcripts from Power Apps

09/15/2025

You can view, export, and download transcripts of customer interactions with your agent in both Power Apps and Copilot Studio. The information that each app exports is slightly different.

## Important

Makers with the *Environment maker* role don't automatically have access to transcripts.

This article covers downloading conversation transcripts in Power Apps and using them to create reports in Power BI. To download transcripts directly in Copilot Studio, see [Download conversation transcripts in Copilot Studio](#).

By default, Power Apps downloads conversation transcripts from the last 30 days. You can [change the retention period](#).

## Note

Agent responses that use SharePoint as a knowledge source aren't included in conversation transcripts.

Conversation transcripts aren't written for:

- Microsoft Dataverse for Teams environments
- Dataverse developer environments
- Microsoft 365 Copilot agents

## Prerequisite

- To view conversation transcripts, you need the Transcript Viewer security role. Only admins can [grant the Transcript Viewer security role](#).

## View and export conversation transcripts from the Power Apps portal

1. Sign in to Power Apps [↗](#).

2. In the side pane, select **Tables**, and then select **All**.

The screenshot shows the Power Apps interface with the 'Tables' option selected in the sidebar. The main area displays a list of tables with columns for Name, Type, Managed, Customized, Customizable, and Tags. The 'All' button in the top navigation bar is highlighted with a red box. A magnifying glass icon with a plus sign is visible in the bottom right corner.

Table ↑	Name	Type	Managed	Customized	Customizable	Tags
Account	account	Standard	Yes	No	Yes	Core
Action Approval Model	msdyn_flow_acti...	Standard	Yes	No	No	Managed
Action Card	actioncard	Standard	Yes	No	Yes	Standard
Activity	activitypointer	Standard	Yes	No	Yes	Standard
Activity File Attachment	activityfileattach...	Standard	Yes	No	Yes	Standard

3. Enter "conversation" in the **Search** box.

4. Select the **ConversationTranscript** table.

A page for the **ConversationTranscript** table opens.

5. On the top menu bar, select **Export > Export data**.

It takes a few minutes for the data to be compiled for export.

The screenshot shows the 'ConversationTranscript' table properties screen. The 'Export' dropdown menu is open, and the 'Export data' option is highlighted with a red arrow. The table properties show it is a Standard type table with a Primary column named 'Name'. The schema section lists Columns, Relationships, and Keys. Data experiences include Forms, Views, Charts, and Dashboards. Customizations include Business rules and Commands.

6. Allow a couple of minutes for Power Apps to prepare the data for export, and then select **Download exported data**.

The file is saved as a ZIP archive to your browser's default download location.

## Understand conversation transcripts

The conversation transcript is a CSV (comma-separated values) file. The following table describes the most important fields in your conversation transcripts.

Field	Description	Example
Content	The entire transcript in JSON format	See <a href="#">Content field</a>
ConversationStartTime	The time the conversation started (not the time the transcript record was written to the data store)	2021-04-19T20:39:09Z
ConversationTranscript	The unique identifier of the row in the Dataverse table	28eccb77-xxxx-4a63-985f-ffaaadd6f391
Metadata	JSON that includes the agent ID, tenant ID, and agent name	{"BotId": "aaaabbbb-0000-cccc-1111-dddd2222eeee", "AADTenantId": "bbbbcccc-1111-dddd-2222-eccc3333ffff", "BotName": "Test Bot"}
Name	The name of the custom row created from <code>ConversationId</code> followed by <code>BotId</code> followed by a <code>batch number</code> . The batch number indicates the order when there are multiple analytic sessions for the conversation. Classic bots don't include the batch number.	8YYe8iif49ZKkycZLe7HU0-o_198eca5f-xxxx-4ae6-8c08-835d884a8688_0
Bot_ConversationTranscript	The agent ID	aaaabbbb-0000-cccc-1111-dddd2222eeee
Created on	The date and time the transcript record was created	2021-04-20T02:40:13Z

## Content field

The `Content` field is a raw log of all the activities that users had with the agent. Common activity types include message and event:

- Message activities represent the content shown in a conversation. Message activities can contain text, speech, interactive cards, and binary or unknown attachments.
- Event activities communicate programmatic information from a client or channel to the agent.

For more information on activity types, see [Bot Framework Activity schema](#).

The following table describes some of the key fields in the content JSON:

Key	Description
<code>ID</code>	The unique GUID of the activity object
<code>valueType</code>	The type of value stored in the activity; dictates what information the activity is providing ( <a href="#">Common activity value types</a> )
<code>timestamp</code>	The timestamp of when the activity was generated, in Epoch format (the number of seconds since midnight UTC January 1, 1970)
<code>type</code>	The type of activity; for example, <code>message</code> , <code>event</code> , or <code>trace</code>
<code>replyToId</code>	The ID of the activity that the current activity is responding to
<code>from</code>	<p>Contains fields <code>id</code> and <code>role</code>:</p> <ul style="list-style-type: none"> <li>• <code>id</code> - the ID of the invoker</li> <li>• <code>role</code> - holds 0 or 1 <ul style="list-style-type: none"> <li>◦ 0 - the activity is coming from the agent</li> <li>◦ 1 - the activity is coming from the user interacting with the agent</li> </ul> </li> </ul> <p>Notes:</p> <p>The <code>id</code> can be used to calculate the number of active users that are interacting with the agent if the canvas is passing in a unique ID of the user. If the canvas doesn't pass an ID, a unique ID per conversation is passed.</p> <p>For security and privacy, the ID is hashed before it's written to the transcript.</p>
<code>channelId</code>	The ID of the channel where the activity is coming from; for example, <code>directline</code> , <code>msteams</code> , or <code>facebook</code>
<code>textFormat</code>	The format of the text; for example, <code>plain</code> or <code>markdown</code>
<code>attachments</code>	Dynamic rich data associated with the activity; for example, <code>AdaptiveCards</code> , <code>HeroCards</code> , or <code>Carousel</code> data
<code>text</code>	The text for <code>message</code> activities
<code>value</code>	Fields specific to the activity based on the value type; this field is where most of the useful information exists
<code>channeldata</code>	<p>Contains channel data:</p> <ul style="list-style-type: none"> <li>• for messages: <ul style="list-style-type: none"> <li>◦ <code>DialogTraceDetail</code></li> <li>◦ <code>DialogErrorDetail</code></li> <li>◦ <code>VariableDetail</code> (contains the value assigned to a variable)</li> <li>◦ <code>CurrentMessageDetail</code></li> </ul> </li> <li>• for events: <ul style="list-style-type: none"> <li>◦ <code>cci_trace_id</code></li> <li>◦ <code>traceHistory</code></li> </ul> </li> </ul>

Key	Description
	<ul style="list-style-type: none"> <li>○ <code>enableDiagnostics</code></li> <li>○ <code>clientTimestamp</code></li> <li>○ <code>clientActivityId</code></li> </ul>
<code>name</code>	The name of the event activity; for example, <code>SetPVAContext</code>

## Common activity value types

[Expand table](#)

Activity value type	Description
<code>ConversationInfo</code>	Whether the conversation is from the Copilot Studio test pane ( <code>isDesignMode</code> ) and the locale of the conversation
<code>CSATSurveyRequest</code>	The user is presented with a customer satisfaction (CSAT) survey
<code>CSATSurveyResponse</code>	The user responds to a CSAT survey
<code>DialogRedirect</code>	The user is redirected to another topic
<code>ImpliedSuccess</code>	The user reached a question node in the topic, where one of the conditions points to the <b>Confirmed Success</b> CSAT system topic (classic bots only) or calls the <b>End of Conversation</b> system topic
<code>IntentRecognition</code>	The user triggered a topic
<code>PRRSurveyRequest</code>	The user was asked if the topic answered their question from the <b>End of Conversation</b> topic
<code>PRRSurveyResponse</code>	The user's response to whether the topic answered their question from the <b>End of Conversation</b> topic
<code>SessionInfo</code>	The type ( <code>unengaged</code> or <code>engaged</code> ), outcome ( <code>Escalated</code> , <code>Resolved</code> , <code>Abandon</code> ), Session Start and End time ( <code>startTimeUtc</code> , <code>endTimeUtc</code> ), and the turn count of the session
<code>VariableAssignment</code>	A value is assigned to a variable

## Enhanced transcripts

Agents can be configured to generate enhanced transcripts that include node-level data. This data helps track the flow of a conversation within a topic. For example, in a *manage orders* topic, users might choose between submitting a new order or retrieving an existing one. Node-level data enables analysis of user behavior across these options.

# Turn on enhanced transcripts

1. Open your agent.
2. Go to **Settings > Advanced**.
3. Select **Enhance Transcripts**, then turn on **Include node-level details in transcripts**.

Advanced

**Application Insights**  
Automatically send telemetry for your agent into an Application Insights resource. You can also log custom telemetry events from within your topics. [Learn more](#)

**Metadata**  
Metadata holds all the information to uniquely identify your agent and its endpoints. It can be used for calling your agent or using your agent in external scenarios. [Learn more](#)

**Enhance Transcripts**  
Details about nodes, such as name, type, and start and end times, can be saved to the transcript stored in Dataverse. [Learn more](#)

**Include node-level details in transcripts**

**View solution**  
Common Data Services Default Solution

**Save**

## Node-level data

When enhanced transcripts are turned on, a `nodeTraceData` activity type appears in the transcript for each node that was invoked by a topic. The following table describes the key fields in the `nodeTraceData` activity value.

[Expand table](#)

Key	Description
<code>nodeID</code>	The node identifier
<code>nodeType</code>	The node type, for example, <code>SendActivity</code> or <code>SearchAndSummarizeContent</code>
<code>startTime</code>	The timestamp when the node was invoked
<code>endTime</code>	The timestamp when the node finished executing
<code>topicDisplayName</code>	The topic invoking the node

## Custom analytics

Some scenarios, like filtering conversations based on a custom variable, require a custom approach for analytics. Customers building custom analytics solutions can ingest the raw

transcripts into their data pipelines or use an add-on, like the [Copilot Studio Kit](#)

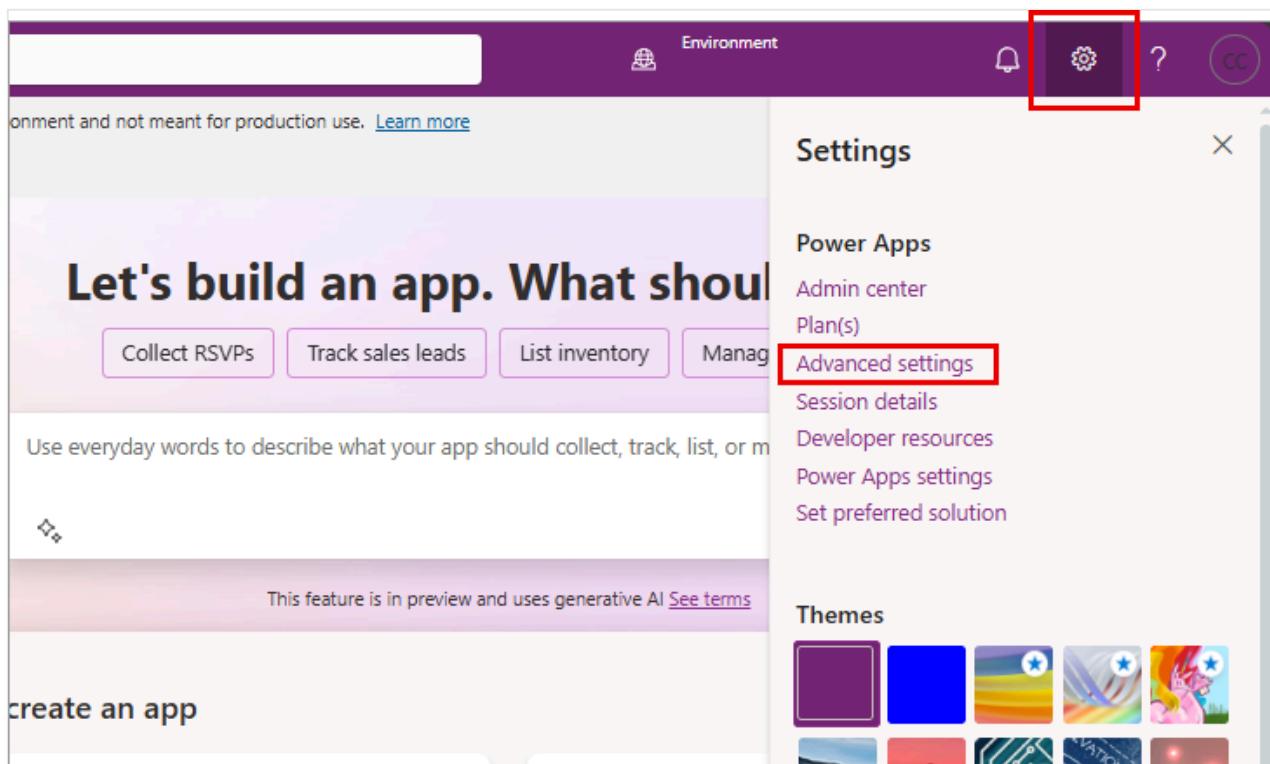
The Conversation KPI solution within the Copilot Studio Kit automatically parses transcripts and populates aggregated data into Dataverse tables.

## Change the default retention period

A Power Apps bulk-delete job automatically removes agent transcripts that are older than 30 days. To keep the transcripts longer, cancel the existing job and create a new one that runs on a different schedule.

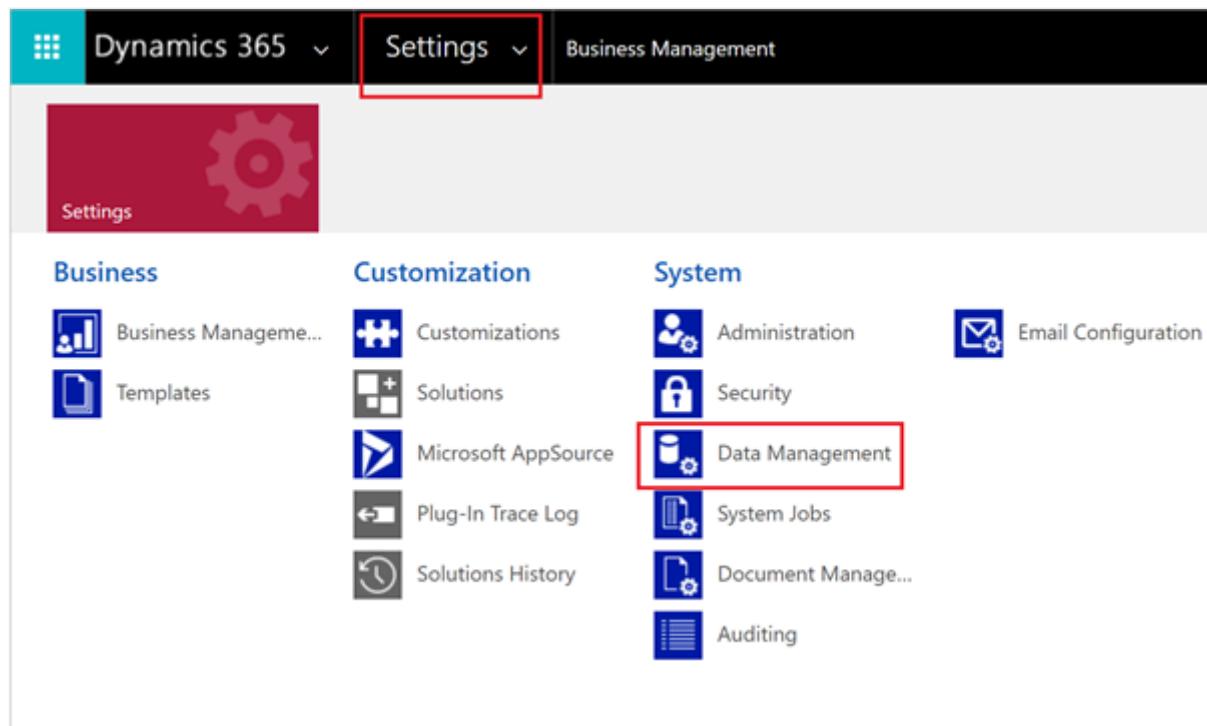
## Cancel the existing bulk delete job

1. In Power Apps, select **Settings** > **Advanced settings**.



The Dynamics 365 portal opens in a new tab.

2. Select the caret next to **Settings**, and then under **System**, select **Data Management**.



3. Select Bulk Record Deletion.

4. In the View list, select Recurring Bulk Deletion System Jobs.

The screenshot shows the Bulk Record Deletion view in Dynamics 365. The 'View' dropdown is set to 'All Bulk Deletion System Jobs'. The table lists four system jobs: 'Completed Bulk Deletion System Jobs', 'In-Progress Bulk Deletion System Jobs', 'My Bulk Deletion System Jobs', and 'Pending Bulk Deletion System Jobs'. The row for 'Recurring Bulk Deletion System Jobs' is highlighted with a red box.

View:	All Bulk Deletion System Jobs	Status Reason	Deleted	Failures	Owner (System Job)
	All Bulk Deletion System Jobs	S Old...	Succeeded	0	0 Admin ALMTest
	Completed Bulk Deletion System Jobs	S Old...	Succeeded	0	0 Admin ALMTest
	In-Progress Bulk Deletion System Jobs	S Old...	Succeeded	0	0 Admin ALMTest
	My Bulk Deletion System Jobs	S Old...	Succeeded	0	0 Admin ALMTest
	Pending Bulk Deletion System Jobs	S Old...	Succeeded	0	0 Admin ALMTest
	Recurring Bulk Deletion System Jobs	S Old...	Succeeded	0	0 Admin ALMTest

5. Select the Bulk Delete Conversation Transcript Records Older Than 1 Month job.

The screenshot shows the Bulk Record Deletion view in Dynamics 365. The 'View' dropdown is set to 'Recurring Bulk Deletion System Jobs'. The table lists three system jobs: 'Bulk Delete Conversation Transcript Records Older Than 1 Month', 'Delete completed pipeline events', and 'Delete completed process sessions for Sync Workflows'. The first job is highlighted with a red box.

View:	Recurring Bulk Deletion System Jobs	Status Reaso...	Owner (System Jo...	Next Run
	System Job Name ↑			
<input checked="" type="checkbox"/>	Bulk Delete Conversation Transcript Records Older Than 1 Month	Waiting	Admin ALMTest	1/17/2020 7:15 PM
	Delete completed pipeline events	Waiting	SYSTEM	1/17/2020 11:18 AM
	Delete completed process sessions for Sync Workflows	Waiting	SYSTEM	1/17/2020 11:18 AM

6. In the More Actions list, select Cancel.

The screenshot shows the 'Bulk Record Deletion' page in Dynamics 365. A context menu is open over a row for a 'Bulk Delete Conversation Transcripts' job. The menu items are: 'Set View as Default View', 'Modify Recurrence', and a group of four options: 'Cancel' (which is highlighted with a red box), 'Resume', 'Postpone', and 'Pause'. The main table lists other system jobs with columns for Status Reason, Owner, and Next Run.

System Job Name	Status Reason	Owner (System Job)	Next Run
Bulk Delete Conversation Transcripts	Waiting	Admin ALMTest	1/17/2020 7:15 PM
Delete completed system jobs	Waiting	SYSTEM	1/17/2020 11:18 AM
Delete completed system jobs	Waiting	SYSTEM	1/17/2020 11:18 AM
Delete completed system jobs	Waiting	SYSTEM	1/17/2020 11:18 AM

## Create a new bulk delete job

1. To start the Bulk Deletion Wizard, select New and then, in the wizard, select Next.

2. Set the following search criteria:

- In the Look for list, select **ConversationTranscripts**.
- Select **ConversationStartTime**, select **Older Than X Months**, and enter "12".  
This setting keeps the transcripts for 12 months. Choose a different comparison and value if you like.
- Select **SchemaType**, leave **equals**, and enter "powervirtualagents".

The screenshot shows the 'Define Search Criteria' step of the Bulk Deletion Wizard. It allows users to select search criteria to identify records to delete. The 'Look for' dropdown is set to 'ConversationTranscripts'. The search criteria section contains two entries:

- A condition for 'ConversationStartTime' set to 'Older Than X Months' with a value of '12'.
- A condition for 'SchemaType' set to 'Equals' with a value of 'powervirtualagents'.

Both conditions are enclosed in a red box. At the bottom, there are buttons for 'Preview Records', 'Back', 'Next', and 'Cancel'.

3. Select **Next**.

4. Enter a name for the job, select **Run this job after every**, and enter 1. Choose a different schedule if you don't want the job to run every day.

## Select Options

 Help

Specify the name of the bulk deletion system job, and scheduling and notification options.

Name:

Bulk Deletion Conversation Transcripts Older Than 12 Months

Bulk deletion job start time:

Immediately

At scheduled time:

1/16/2020

11:43 PM

Run this job after every:

1  days

Send an email to me (ccialmtestadmin@ccibotstest1outlook.onmicrosoft.com) when this job is finished.

Also notify:

[Back](#)

[Next](#)

[Cancel](#)

5. Select **Next**, and then select **Submit**.

## Why can't I see my conversation transcripts in the ConversationTranscript Power Apps table?

Administrators can prevent the saving of transcripts to Dataverse. For more details on how transcripts are saved and how to manage access, see [Control transcript access](#).

### Important

Transcripts are not stored for agents deployed in developer environments.

## Tips for getting the most out of your conversation transcripts

Variables can be used to store data relevant to your agent content or agent user. Parsing the variable and its value from the conversation transcript lets you filter or slice the data by the variable.

Conversation transcripts might refer to content by ID only. For example, in a **Redirect** node, the destination topic is only referred to by its ID. To get the name of this topic, look up its ID in the agent content.

# View aggregated analytics in Viva Insights (preview)

[This article is prerelease documentation and is subject to change.]

The Copilot Studio agents report in Viva Insights can help you understand the adoption and impact of conversational agents across your tenant.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

The report isn't available for agents using generative orchestration, including, autonomous agents or agents that extend Microsoft 365 Copilot.

For more information, see [Copilot Studio agents report](#) in the Viva Insights documentation.

## Prerequisites

To use the Copilot Studio agents report, you need to fulfill requirements for Viva Insights. Also, the [settings to transfer data from Microsoft Copilot Studio to Viva Insights](#) must be enabled in the Power Platform Admin Console.

## Viva Insights advanced insights settings

This report is available within [Viva Insights advanced insights](#). Your organization must meet the [licensing requirements](#) for advanced insights and advanced insights must be turned on for your organization. Follow this [checklist](#) to enable advanced insights for your organization.

If *advanced insights* is already enabled in your organization, see [Copilot Studio agents report](#) in the Viva Insights documentation to learn how to set up the report.

When this setting is **On**, the following information is sent to Viva Insights daily:

**Information about the agent:**

- Name
- Mode (classic or generative)
- Image that represents the agent
- Type (internal or external)
- Category (Sales, CustomService, Marketing, HR, IT, Legal, Finance, or Other)

Agent type and category are determined by Copilot Studio based on the agent's information. Viva Insights uses agent type and category to group data and for ROI calculations.

#### Usage information:

- Total number of sessions, session engagement status, session outcome, CSAT (customer satisfaction) score, and session duration
- Aggregated knowledge sources usage
- Topics name and usage information

## Power Platform data settings

Make sure the following settings are turned on. Contact your Power Platform admin for changes to these settings.

- [Microsoft 365 services](#): The Copilot Studio agents report requires data to flow from Copilot Studio to Viva Insights using Microsoft 365 services. This data is stored using Microsoft 365 services, which is not governed by Azure data terms and commitments. Instead, activity data is governed by [Microsoft 365 terms and data residency commitments](#). Global and Power Platform administrators can turn off data being stored by Microsoft 365 through the [Power Platform admin center](#). Turning off this setting stops future data storage in Microsoft 365. Existing data stored in Microsoft 365 is deleted or purged according to the data retention policy set for Microsoft 365.
- [Sharing Copilot Studio agent data with Viva Insights](#): Allow the transfer of data from Copilot Studio to Viva Insights. You can also control this setting for [environment groups](#).
- If your tenant's preferred data location for Viva Insights is different than the location of your Copilot Studio environment, turn on [Cross-geo boundary sharing](#) to allow cross-geo boundary sharing of aggregated analytics data. You can also control this setting for [environment groups](#).

# Capture telemetry with Application Insights

09/06/2025

This article discusses how you can capture telemetry data from your Copilot Studio agent for use in [Azure Application Insights](#).

In addition to the native analytics features within Copilot Studio, you can send telemetry data to Application Insights. Telemetry offers insights into your agent by tracking:

- Logged messages and events sent to and from your agent
- Topics to be triggered during user conversations
- Custom telemetry events that can be sent from your topics

## Important

Application Insights is a feature of [Azure Monitor](#), an extensible Application Performance Management (APM) tool that allows you to monitor your live applications. It requires a subscription to [Microsoft Azure](#).

## Connect your Copilot Studio agent to Application Insights

To connect your agent to Application Insights, you first need to add your instrumentation key to your agent's configuration.

1. Go to the **Settings** page for your agent, and select **Advanced**.
2. Within the **Application Insights** section, populate the **Connection string** setting. See the [Azure Monitor documentation](#) to find out how to locate your connection string.
3. Optionally, you can choose to enable one of the following settings.
  - **Log activities:** If enabled, details of incoming/outgoing messages and events are logged.
  - **Log sensitive Activity properties:** If enabled, the values of certain properties that could be considered sensitive on incoming/outgoing messages and events are included in logs. The properties that are considered potentially sensitive are `userid`, `name`, `text` and `speak` (`text` and `speak` properties only apply to messages).

# Analyze bot telemetry with Application Insights

After you've connected your bot to Application Insights, telemetry data is logged when users interact with the bot, including testing within Copilot Studio. To see the logged telemetry data, navigate to the **Logs** section of your Application Insights resource in Azure.

From here, you can use [Kusto queries](#) to query and analyze your data. See [example queries](#).

## Example queries

A query can be as simple as specifying a single table, such as `customEvents`, which shows all custom telemetry events logged from Copilot Studio. But you can also use [Kusto queries](#) to narrow down your results further, including;

- Adding a time interval
- Extending your results using [custom dimensions](#). Custom dimensions are custom properties that are logged, along with the prebuilt fields, such as timestamp or the event name
- A `where` clause to limit the data returned based on a condition
- Using more built-in Kusto functions to determine what and how information is shown

The example query below results in a line chart that shows how many distinct users communicated with your bot per day for the past 14 days.

Kusto

```
let queryStartDate = ago(14d);
let queryEndDate = now();
let groupByInterval = 1d;
customEvents
| where timestamp > queryStartDate
| where timestamp < queryEndDate
| summarize uc=dcount(user_Id) by bin(timestamp, groupByInterval)
| render timechart
```

### Important

The data within some fields varies and is more or less applicable, depending on the channel that is being used. For example, you'll only get a correct count of unique users in the query if they're authenticated users and their user ids are consistent across conversations. In anonymous scenarios where a random user id is generated per conversation, the user id field is less useful.

# Excluding telemetry from test conversations from your queries

Your bot logs telemetry for all conversations, including those that happen within Copilot Studio during testing. If you want to exclude telemetry gathered during testing, you can extend your query with the `designMode` custom dimension that is captured on all events, and use a `where` clause in your query.

The following example shows all custom events, excluding those captured through the test canvas.

Kusto

```
customEvents
| extend isDesignMode = customDimensions['designMode']
| where isDesignMode == "False"
```

## Custom Dimensions

Much of the specific activity data received from Copilot Studio is stored in the `customDimensions` field. You can [see a custom dimension field being used](#) in a query to exclude telemetry from test conversations.

[+] Expand table

Field	Description	Sample Values
type	Type of activity	message, conversationUpdate, event, invoke
channelId	Channel identifier	emulator, directline, msteams, webchat
fromId	From Identifier	<id>
fromName	Username from client	John Bonham, Keith Moon, Steve Smith, Steve Gadd
locale	Client origin locale	en-us, zh-cn, en-GB, de-de, zh-CN
recipientId	Recipient identifier	<id>
recipientName	Recipient name	John Bonham, Keith Moon, Steve Smith, Steve Gadd
text	Text in message	find a coffee shop

Field	Description	Sample Values
designMode	Conversation happened within the test canvas	True / False

# Create test cases to evaluate your agent (preview)

10/28/2025

[This article is prerelease documentation and is subject to change.]

In Copilot Studio, you can create a test set of test cases to evaluate the performance of your agents. Test cases let you simulate real-world scenarios for your agent, so you can measure the accuracy, relevancy, and quality of answers to the questions the agent is asked, based on the information the agent can access. Using the results from the test set, you can optimize your agent's behavior and validate that your agent meets your business and quality requirements.

## Important

This article contains Microsoft Copilot Studio preview documentation and is subject to change.

Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that you can get early access and [provide feedback](#).

If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

## Test methods

When creating test sets, you can choose from three kinds of test methods to evaluate your agent's responses: *text match*, *similarity*, and *quality*. Each test method has its own strengths and is suited for different types of evaluations.

### Text match test methods

Text match test methods compare the agent's responses to expected responses that you define in the test set. There are two match tests:

**Exact match** checks whether the agent's answer exactly matches the expected response in the test: character for character, word for word. If it's the same, it passes. If anything differs, it fails. Exact match is useful for short, precise answers such as numbers, codes, or fixed phrases. It doesn't suit answers that people can phrase in multiple correct ways.

**Partial match** checks whether the agent's answer contains some of the words or phrases from the expected response that you define in the test. If it does, it passes. If it doesn't, it fails. Partial match is useful when an answer can be phrased in different correct ways, but key terms or ideas still need to be included in the response.

## Similarity test methods

The similarity test method compares the similarity of the agent's responses to the expected responses you define in your test set. It's useful when an answer can be phrased in different correct ways, but the overall meaning or intent still needs to come through.

It uses a cosine similarity metric to assess how similar the agent's answer is to the wording and meaning of the expected response and determines a score. The score ranges between 0 and 1, where 1 indicates the answer closely matches and 0 indicates it doesn't. You can set a passing score threshold to determine what constitutes a passing score for an answer.

## Quality test methods

Quality test methods help you decide whether your agent's responses meet your standards. This approach ensures the results are both reliable and easy to explain.

These methods use a large language model (LLM) to assess how effectively an agent answers user questions. They're especially helpful when there's no exact answer expected, offering a flexible and scalable way to evaluate responses based on the retrieved documents and the conversation flow.

Quality test methods include two test methods:

**General quality** evaluates agent responses. It uses the following key criteria and applies a consistent prompt to guide scoring:

- Relevance: To what extent the agent's response addresses the question. For example, does the agent's response stay on the subject and directly answer the question?
- Groundedness: To what extent the agent's response is based on the provided context. For example, does the agent's response reference or rely on the information given in the context, rather than introducing unrelated or unsupported information?
- Completeness: To what extent the agent's response provides all necessary information. For example, does the agent's response cover all aspects of the question and provide sufficient detail?
- Abstention: Whether the agent attempted to answer the question.

To be considered high quality, a response must meet all key criteria. If one isn't met, the response is flagged for improvement. This scoring method ensures that only responses that are both complete and well-supported receive top marks. In contrast, answers that are incomplete or lack supporting evidence receive lower scores.

**Compare meaning** evaluates how well the agent's answer reflects the intended meaning of the expected response. Instead of focusing on exact wording, it uses semantic similarity—meaning it compares the ideas and meaning behind the words—to judge how closely the response aligns with what was expected.

You can set a passing score threshold to determine what constitutes a passing score for an answer. The compare meaning test method is useful when an answer can be phrased in different correct ways, but the overall meaning or intent still needs to come through.

## Thresholds and pass rates

The success of a test case depends on the test method you select and the threshold you set for passing scores.

Each test method, except exact match, produces a numeric score based on a set of evaluation criteria that reflects how well the agent's answer meets that criteria. The threshold is the cut-off score that separates pass from fail. You can set the passing scores for similarity and compare meaning test cases.

Exact match is a strict test method that doesn't produce a numeric score; the answer must match exactly to pass. By choosing the threshold for a test case, you decide how strict or lenient the evaluation is. Each test method evaluates the agent's answer differently, so it's important to choose the one that best fits your evaluation criteria.

## Create a test set

You can start creating a test set in different ways: import a file, use the questions from your test chat, have Copilot Studio generate test cases for you, or manually create test cases in Copilot Studio. Each question in the test set is considered as a test case.

To create a test set:

1. Open the agent that you want to evaluate.
2. On the top menu bar, go to **Analytics**.
3. If you didn't publish your agent, select **Start evaluation**. If you published your agent, go to the **Evaluations** section and select **Start evaluation**.

Evaluations (preview)  
Track conversation outcomes to see how well your agent is helping your users. [See preview terms](#)

**Test sets**

**Evaluate Website Q&A** ...  
10 test cases • last modified by Steph Kent, 16 days ago  
Active

**Evaluate Website Q&A** ...  
10 test cases • last modified by Steph Kent, 17 days ago  
Active

**See all**

4. In the **New test set** page, choose the method you want to use to create your test set:

- Select **Generate 10 questions** to have Copilot Studio create test cases automatically based on what your agent can do.
- Select **Use your test chat conversation** to automatically populate the test set with the questions you provided in your [test chat](#).
- Select **Manually add** to manually provide your test case.
- Import test cases from a [file](#) by dragging your file into the designated area or select **Browse** to upload a file.

5. Review and [edit the test cases](#) to create effective tests.

You can:

- Change the test case by selecting the case and editing it in the right pane.
- Remove the test case by selecting the delete icon beside the case.
- Automatically generate more questions by selecting an option in the dropdown menu.
- Add a test case by selecting **Add a case manually**.

When you're finished with your changes, select **Apply**.

6. Under **Name**, enter a name for your test set.

7. Under **Test account**, select the account that you want to use for this test set.

8. Select **Save** to update the test set without running the test cases or **Evaluate** to run the test cases.

## Create a test case file to import

Instead of building your test cases directly in Copilot Studio, you can create a spreadsheet file with all your test cases and import them to create your test set. You can compose each test question, determine the test method you want to use, and state the expected responses for each question. When you finish creating the file, save it as a .csv or .txt file and import it into Copilot Studio.

### Important

- The file can contain up to 100 questions.
- Each question can be up to 1,000 characters, including spaces.
- The file must be in comma separated values (CSV) or text format.

To create the import file:

1. Open a spreadsheet application (for example, Microsoft Excel).
2. Add the following headings, in this order, in the first row:
  - Question
  - Expected response
  - Testing method
3. Enter your test questions in the **Question** column. Each question can be 1,000 characters or less, including spaces.
4. Enter one of the following test methods for each question in the **Test method** column:
  - General quality
  - Compare meaning
  - Similarity
  - Exact match
  - Partial match
5. Enter the expected responses for each question in the **Expected response** column. Expected responses are optional for importing a test set. However, you need expected responses to run match, similarity, and compare meaning test cases.
6. Save the file as a .csv or .txt file.
7. Import the file to [create a test set](#).

## Edit a test case

After creating a test set, you can edit the test cases by changing the wording of questions, choosing different test methods, or modifying the expected responses as needed. You can select multiple test cases to edit them in bulk by selecting the checkboxes beside each test case.

You have a choice of three test methods, also referred to as graders, to evaluate agent responses: quality, similarity, text match. For more information about the different test methods, see [Test methods](#).

To edit a test case:

1. In the test set, select the test case you want to edit.

The screenshot shows the 'Evaluate Agent' page in the Analytics section. On the left, there's a table titled 'Review your test cases (10)' with columns for 'Question', 'Test method', and 'Expected response'. One row is highlighted with a red border, showing the question 'What sho...', test method 'General quality', and expected response '--'. On the right, a modal window titled 'Edit test case' is open. It has a 'Question \*' field containing 'What should I wear for the weather in Seattle today?'. Below it is a 'Test method' section with three options: 'Quality' (selected), 'Similarity', and 'Text match'. A note says 'Tests according to the selected criteria, and then shows results as a score.' with a 'Learn more' link. At the bottom are 'Cancel' and 'Apply' buttons. The 'Question' field in the modal is also highlighted with a red border.

2. In the right pane, change the wording of a question by editing the text in the **Question** field.

This screenshot is similar to the previous one, showing the 'Evaluate Agent' page. The 'Edit test case' dialog is still open, but the 'Question \*' field now contains 'What should I wear for the weather in Seattle today?' instead of the original question. The rest of the interface, including the list of test cases on the left and the test method options on the right, remains the same.

3. Select the **test method** that you want to use.

The screenshot shows the 'Evaluate Agent' page. On the left, there's a list of 10 test cases. The first case is selected, showing its details in the main area. On the right, a modal window titled 'Edit test case' is open, displaying a question and a configuration section for 'Test method'.

Question	Test method	Expected response
What sho...	General quality	--
Is it safe t...	General quality	--
How shoul...	General quality	--
Are there ...	General quality	--

**Edit test case**

Question \*

What should I wear for the weather in Seattle today?

**Test method**

Quality    Similarity    Text match

Tests according to the selected criteria, and then shows results as a score. [Learn more](#)

General quality  
Answers meet quality standards, such as relevance and completeness. Doesn't compare to expected responses.

**Cancel**   **Apply**

- **Quality:**
  - Select **General quality** to evaluate the answer based on relevance, groundedness, and completeness.
  - Select **Compare meaning** to evaluate the answer based on how well it captures the meaning of the expected response. Under **Passing score**, you can set the threshold for what constitutes a passing score for an answer. In the **Expected response** box, provide the response against which the test method evaluates the agent's answer.
- **Similarity:** uses a cosine similarity metric to assess how similar the agent's answer is to the wording and meaning of the expected response. It determines a score between 0 and 1, where 1 means it matches closely and 0 means it doesn't match at all. Under **Passing score**, you can set the threshold for what constitutes a passing score for an answer. In the **Expected response** box, provide the response against which the test method evaluates the agent's answer.
- **Text Match:**
  - Select **Exact match** to evaluate the agent's answer against the expected response, where a passing score means the agent's answer exactly matched the defined expected response. In the **Expected response** box, provide the response against which the test method evaluates the agent's answer.
  - Select **Partial match** to evaluate the agent's answer against the expected response, where a passing score means the agent's answer contained some of the words or phrases from the defined expected response. In the **Expected response** box, provide a phrase or keyword against which the test method evaluates the agent's answer. To add multiple keywords or phrases, select **Add**, select the operator **and** or **or** between the boxes, and provide the keyword or phrase.

#### 4. Select **Apply**.

- When you're finished with your changes, select **Save** to save your test set or **Evaluate** to run the evaluation on the test set.

## Run a test set

After you create a test set, you can run or rerun it.

- On your agent's **Analytics** page, go to **Evaluations**.
- Run a test set by doing one of the following actions:
  - Find the test set in the **Test sets** list, select the More icon (...), then select **Evaluate test set**.
  - Hover over a test result that uses the test set you want, select the More icon (...), then select **Evaluate test set again**.

The screenshot shows the 'Evaluations (preview)' page. At the top, it says 'Track conversation outcomes to see how well your agent is helping your users.' with a link to 'See preview terms'. Below this is a section titled 'Test sets' which lists two entries: 'Evaluate Website Q&A' (10 test cases, last modified by Steph Kent 16 days ago, status Active) and another 'Evaluate Website Q&A' entry with the same details. Both entries have a red 'More' icon (...) to their right. Below this is a section titled 'Recent results' which lists two evaluations: 'Evaluate Website Q&A' (96% pass rate, 10 questions, 9/29/25, 11:44 PM) and another 'Evaluate Website Q&A' entry (96% pass rate, 10 questions, 9/29/25, 11:36 PM). Each evaluation row has a red 'More' icon (...) to its left.

## Delete a test set

Select the More icon (...) for a test set, then select the Trash icon.

## Dive into detailed test results

Each time you run an evaluation with a test set, Copilot Studio:

- Uses the connected user account to simulate conversations with the agent, sending each question in the test case to the agent.

2. Collects the agent's responses.
3. Measures the success of each response. Each test case receives a **Pass** or **Fail**, based on the [criteria of the test case](#).
4. Assigns a **Pass rate** score based on the **Pass/Fail** rate of the test set.

You can see the **Pass rate** of each test set run on your agent's **Analytics** page, under **Evaluations > Recent results**. To see more test set runs, select **See all**.

Analytics > Evaluation results					
Name	Pass rate	Questions	Date	Created by	Status
Evaluate Website Q&A			9/30/25, 12:11 AM		<span>Running</span>
Evaluate Website Q&A	96%	<div style="width: 96%; background-color: #2e7131;"></div>	10	9/29/25, 11:44 PM	<span>Completed</span>
Evaluate Website Q&A	96%	<div style="width: 96%; background-color: #2e7131;"></div>	10	9/29/25, 11:36 PM	<span>Completed</span>
Evaluate Website Q&A	75%	<div style="width: 75%; background-color: #2e7131;"></div>	10	9/13/25, 2:55 AM	<span>Completed</span>

Select an evaluation to see a detailed breakdown of the test results for each response within a test set run.

Analytics > Evaluation results					
Name	Pass rate	Questions	Date	Created by	Status
Evaluate Website Q&A			9/30/25, 12:11 AM		<span>Running</span>
Evaluate Website Q&A	96%	<div style="width: 96%; background-color: #2e7131;"></div>	10	9/29/25, 11:44 PM	<span>Completed</span>
Evaluate Website Q&A	96%	<div style="width: 96%; background-color: #2e7131;"></div>	10	9/29/25, 11:36 PM	<span>Completed</span>
Evaluate Website Q&A	75%	<div style="width: 75%; background-color: #2e7131;"></div>	10	9/13/25, 2:55 AM	<span>Completed</span>

The test case results show a list of the queries used in the test, how the agent responded, and the **Pass** or **Fail** score. Select a query in the list to see a detailed assessment of each response.

## Test run result

All

Pass (6)

Fail (3)

Question	Agent response	Test method	Score
What should I do if I	If you can't	<input checked="" type="checkbox"/> Text similarity	<span>Fail</span>
Can you send me a rich	Here are	<input checked="" type="checkbox"/> General quality	<span>Pass</span>
How does your website	The	<input checked="" type="checkbox"/> Partial match	<span>Fail</span>
What are the steps to	Looks like	<input checked="" type="checkbox"/> General quality	<span>Fail</span>
Can you send me a	Here is a	<input checked="" type="checkbox"/> General quality	<span>Pass</span>

The assessment includes the expected and actual responses, the reasoning behind the test result, and the knowledge and topics the agent used in creating the response.

## Test case details

X

- Seems relevant
- Seems complete
- Based on knowledge sources
- No matching phrases in the agent response

Rate this evaluation



AI-generated content may be incorrect

### Question

Can you provide a step-by-step guide for booking a service?

### Agent response

Here is a step-by-step guide for booking a service using Microsoft Bookings:

#### 1. Sign In and Access Bookings

- Sign in to your Microsoft 365 account.
- Select the app launcher and choose Bookings....

[See more](#)

### Knowledge sources cited



Microsoft



# Understand error codes

10/02/2025

When an agent encounters a problem during a conversation, it responds with a message that includes an error code for the specific problem that was encountered. Users of the agent should give this error code to their administrator.

As an agent maker, if a problem occurs when you're using the test pane to [test your agent](#), you can see a message with more context about the problem, in addition to the error code. Alternatively, you can use the **Topic checker** panel to [validate your agent](#).

## Error list

Web app

ⓘ Note

The term *dialog* used in some error messages refers to a *topic*.

[+] Expand table

Error code	Description
<a href="#">AIModelActionRequestTimeout</a>	There's a timeout error related to a call to an AI Builder model.
<a href="#">AsyncResponsePayloadTooLarge</a>	There's an error related to the output of a connector.
<a href="#">ConsentNotProvidedByUser</a>	A user interacting with an agent rejects the agent's SSO request.
<a href="#">ContentError</a>	There's an error in the topic content.
<a href="#">DataLossPreventionViolation</a>	There was a data policy violation.
<a href="#">EnforcementMessageC2</a>	Not enough prepaid capacity was available.
<a href="#">FlowActionException</a>	An error occurred while executing an <a href="#">agent flow</a> .
<a href="#">FlowActionBadRequest</a>	A request made to an <a href="#">agent flow</a> was malformed.
<a href="#">FlowActionTimedOut</a>	An <a href="#">agent flow</a> took more than 100 seconds to run and timed out.

Error code	Description
FlowMakerConnectionBlocked	An <a href="#">agent flow</a> invoked with unauthorized maker credentials in connection
GenAISeachandSummarizeRateLimitReached	The usage limit for generative AI was reached.
GenAIToolPlannerRateLimitReached	The usage limit for generative orchestration was reached.
InvalidContent	Invalid content was added to the code editor.
InfiniteLoopInBotContent	A node was executed too many times.
LatestPublishedVersionNotFound	Unable to retrieve the published version of the agent.
OutgoingMessageSizeTooBig	A message sent by an agent is too large to process.
TooMuchDataToHandle	The request made by the user is too large to process.
OpenAIRateLimitReached	The capacity limit of the agent was reached.
RedirectToDisabledDialog	A topic is <a href="#">redirecting</a> to a disabled topic.
RedirectToNonExistentDialog	A topic is <a href="#">redirecting</a> to another topic that no longer exists.
SystemError	A system error occurred in Copilot Studio.

## AIModelActionRequestTimeout

**Error message:** The prompt `prompt-name` execution timed out.

**Resolution:** Ensure that the call to the AI Builder model doesn't exceed 100 seconds.

## AsyncResponsePayloadTooLarge

**Error message:** The output returned from the connector was too large to be handled by the agent. Try reducing its size by utilizing available connector filters or by limiting the number of configured action outputs.

**Resolution:** One of the agent's real-time connectors is returning a payload that's larger than the agent can handle. For more information regarding the payload limit, see [Copilot Studio web app limits](#).

## ConsentNotProvidedByUser

**Error message:** No consent provided for SSO connection.

**Resolution:** The user interacting with the agent must confirm the connection using the agent's single sign-on connection prompt.

## ContentError

**Error message:** This error produces dynamic messages based on the context of the error.

**Resolution:** This message is a catch-all error for problems related to your agent's content. The error message provides more details.

Common problems include:

- A node is missing required properties.
- Invalid YAML was added with the [code editor](#).
- A [Power Fx formula](#) contains an error.

## DataLossPreventionViolation

**Error message:** This environment requires users to sign in before they can use the agent. Go to Manage > Security > Authentication and select the option that requires users to sign in.

**Resolution:**

- Your environment's data policies require that users sign in. See [Add user authentication with the Sign in system topic](#).
- One or more connectors that are used in the agent aren't in the same data group. See [Copilot Studio connectors](#).
- One or more connectors that are used in the agent were blocked by the tenant administrator.

## EnforcementMessageC2

**Error message:** This agent is currently unavailable. It has reached its usage limit. Please try again later.

**Resolution:** This message is returned when an agent has reached its message capacity or the pay-as-you-go meter has reached its limit. To resolve the issue, add more prepaid

capacity or create a pay-as-you-go billing plan. The agent chat should then resume working within 5 minutes. For more information, go to [Overage Enforcement](#).

## FlowActionException

**Error messages:**

- No output was received from flow {FlowName} ({FlowId}), even though output was expected as per the agent definition.
- The output parameter with name {ItemKey} on flow {FlowName} ({FlowId}) is missing from the response data. Refresh the flow, or ensure the flow returns this value.
- The output parameter with name {ItemKey} on flow {FlowName} ({FlowId}) is missing from the output schema. Please refresh the flow.

**Resolution:** [Check the flow for errors](#).

## FlowActionBadRequest

**Error messages:**

- The parameter with name {KeyName} on flow {FlowName} ({FlowId}) is declared to be of type {ItemTypeKind}. This type isn't supported when invoking Power Automate. Currently, only Text, Boolean and Numbers are supported.
- The parameter with name {ItemKey} on flow {FlowName} ({FlowId}) is missing in the 'Call Flow' action.
- The parameter with name {KeyName} on flow {FlowName} ({FlowId}) evaluated to type {ResolveType}, expected type {ExpectedType}.
- The flow {FlowName} ({FlowId}) failed to run with response code {ResponseCode}, error code: {FlowErrorCode}.

**Resolution:** Check that the [base type](#) of any variables you pass to the flow match the parameter's type.

## FlowActionTimedOut

**Error message:** The flow with id {FlowId} has timed out. Error Code: {FlowErrorCode}

**Resolution:** [Check the flow for errors](#) to understand why the cloud flow took more than 100 seconds to run before it returned to your agent. Try to optimize the query and the data you return from backend system. If some of the cloud flow logic can continue to run after a result is sent to the agent, place these actions after the 'Return value(s) to Copilot Studio' step in your cloud flow.

## FlowMakerConnectionBlocked

**Error message:** The flow with name {FlowName} is using a maker connection, which is not allowed. Error Code: {FlowMakerConnectionBlocked}

**Resolution:** The administrator prevents using maker credentials in a connection invoked from the agent flow. [Open the flow in Power Automate](#) and [share the cloud flow with run-only permissions](#).

## GenAISeachandSummarizeRateLimitReached

**Error message:** The usage limit for search and summarize has been reached. Please try again later.

**Resolution:** This message is returned when the agent reaches its [generative AI limit](#) to search and summarize sources. For more information, see [Resolve throttling errors in agents](#).

## GenAIToolPlannerRateLimitReached

**Error message:** The usage limit for generative orchestration has been reached. Please try again later.

**Resolution:** This message is returned when the agent reaches its [generative orchestration limit](#). For more information, see [Resolve throttling errors in agents](#).

## InvalidContent

**Error message:** A total of {TotalComponents} component(s) exist in the agent, but none are valid.

**Resolution:** [Open the code editor](#) to review issues with the content.

## InfiniteLoopInBotContent

**Error message:** Action {DialogId}.{TriggerId}.{ActionId} was executed more than {MaxTurnCount} times in a row. This indicates a cycle in execution of the dialog and hence dialog execution will be terminated.

**Resolution:** Make sure the topic ends properly and links to other topics that end properly, such as the **Escalate** system topic.

## LatestPublishedVersionNotFound

**Error message:** Unable to retrieve the latest published version of the agent.

**Resolution:** [Publish the agent](#).

## OutgoingMessageSizeTooBig

**Error message:** Outgoing message size too big.

**Resolution:** Depending on the channel, such as Direct Line or Facebook, being used to transfer files, you might receive the following error message: "The request content length exceeded limit of 262,144 bytes." These limits are imposed by the [channel](#), and not Copilot Studio.

In this scenario, there are a few options. One option is to provide a link to the resource as an internet attachment. Another option is to review your nodes to ensure that none of them are using a variable that contains a large volume of text, such as a `JSON.stringify()` static method. If you use this method or a variable that contains a large volume of text, modify the node to only pass the portion of text that's necessary. For example, if you use an Adaptive Card to pass data to another topic, update the variable to only pass the necessary property.

For more information, see [Maximum channel data message size limits when using Copilot Studio in Omnichannel](#).

## OpenAIRateLimitReached

**Error message:** An error has occurred.

**Resolution:** Your agent reached the maximum number of generative answers responses. Review your [message capacity](#), and review the information in [Resolve throttling errors in agents](#).

## RedirectToDisabledDialog

**Error message:** The Dialog with Id {DialogId} is disabled in the definition. Please enable the Dialog before using it.

**Resolution:** [Re-enable the topic](#) or [remove the redirect node](#).

## RedirectToNonExistentDialog

**Error message:** The Dialog with Id {DialogId} was not found in the definition. Please check that the Dialog is present and that the Id is correct.

**Resolution:** [Create a new topic](#) to redirect to, or [remove the redirect node](#).

## SystemError

**Error message:** This error doesn't produce an error message.

**Resolution:** [Contact customer support](#).

## TooMuchDataToHandle

**Error message:** The request is resulting in too much data to handle, please evaluate the amount of data being returned by your actions.

**Resolution:** This indicates the request being sent to OpenAI is exceeding the maximum request size allowed. There are a number of things that make up the request including the user input, output from previous actions, tools called, and conversation history. Review the tools you are using, and, where possible, scope down their output to only the necessary fields. For more information, see [Create a Power Automate flow](#) and [Call a Power Automate flow as an action](#).

# Generative answers pointing to SharePoint sources don't return results

09/19/2025

Generative answers allow makers to create agents that respond to questions grounded in data sources, like public websites or SharePoint, by pointing the agent at those data sources. However, sometimes the agent doesn't provide a response and instead returns something like '**I'm not sure how to help with that. Can you try rephrasing?**' (The actual message depends on the implementation.)

When a SharePoint data source is configured, there could be several different factors preventing the **Create generative answers** node from returning a response, such as those described in the following sections.

## ⓘ Note

Before continuing, please make sure you have followed the instructions on how to [set up generative answers over SharePoint](#).

For better search results, we recommend a Microsoft 365 Copilot license within the same tenant as your agent.

## Search results are missing

When Copilot Studio searches SharePoint, only the top three search results are used to summarize and generate a response. If no search results are returned, the generative answers node doesn't provide a response.

## How to fix

1. Ensure that your Create generative answers node points to a SharePoint location with relevant content.
2. Only documents in [supported formats](#) are used to generate responses.

## ⓘ Note

Only modern SharePoint pages are supported.

3. It's possible that documents were only recently uploaded to SharePoint, but have yet to be indexed. It's also possible that there are settings that prevent some sites from appearing in search results. For more information, see [Search results missing in SharePoint Online](#).

## Missing user permissions

Generative answers over SharePoint rely on [delegated permissions](#) when searching. At a minimum, a user must have read permissions on the relevant sites and files, or no search results will be returned.

If the user is missing permissions, no results are returned, nor any errors or exceptions. For a user with no permissions, it appears as if no documents were found.

### How to fix

Amend permissions so users can access the relevant sites and files. For more information, see [Sharing and permissions in the SharePoint modern experience](#).

## The app registration or agent are misconfigured

When admins configure generative answers over SharePoint, admins are expected to set up authentication with a Microsoft Entra ID, and configure [extra scopes](#). If scopes are missing from the app registration or from the agent authentication settings, or if consent wasn't granted to the required scopes, no results are returned, nor any errors or exceptions. For a user, it appears as if no documents were found.

### How to fix

Add the necessary scopes to the App Registration and/or the agent's authentication settings, and grant consent.

The following example is a reference to a well configured app registration:

## Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

Add a permission Grant admin consent for Contoso

API / Permissions name	Type	Description	Admin consent requ...	Status	...
Microsoft Graph (5)					
Files.Read.All	Delegated	Read all files that user can access	No	Granted for Contoso	...
openid	Delegated	Sign users in	No	Granted for Contoso	...
profile	Delegated	View users' basic profile	No	Granted for Contoso	...
Sites.Read.All	Delegated	Read items in all site collections	No	Granted for Contoso	...
User.Read	Delegated	Sign in and read user profile	No	Granted for Contoso	...

The following example shows the required authentication settings in Copilot Studio:

## < Authentication

Verify a user's identity during a conversation. The agent receives secure access to the user's data and is able to take actions on their behalf, resulting in a more personalized experience. [Learn more](#)

### Choose an option

No authentication 

Publicly available in any channel

Authenticate with Microsoft

Entra ID authentication in Microsoft Teams, Power Apps, or Microsoft 365 Copilot

Authenticate manually

Set up authentication for any channel

  Require users to sign in

### Redirect URL

`https://token.botframework.com/.auth/web/re`

 [Copy](#)

---

### Service provider \*

Azure Active Directory v2



### Client ID \*

### Client secret \*

Token exchange URL (required for SSO) [Learn more about SSO](#)

### Tenant ID

---

### Scopes

profile openid Files.Read.All Sites.Read.all

## File size support

For SharePoint sources, if you don't have a Microsoft 365 Copilot license in the same tenant as your agent, generative answers can only process files up to 7 MB in size. You must also turn off the [Enhanced search results](#) feature.

If you have a Microsoft 365 Copilot license in the same tenant as your agent, the maximum file size is 200 MB. You must also turn on the [Enhanced search results](#) feature.

Larger files can be stored in SharePoint and [are returned](#) by a Microsoft Graph search, but aren't processed by generative answers. As an alternative, you can upload your own [files](#), which can be up to 512 MB in size.

For a list of limits and supported SharePoint functionality, see [Copilot Studio web app](#) [SharePoint limits](#).

## How to fix

If files relevant for your conversational AI experience exceed the size limitation, you might want to explore alternative architectures, such as using [Microsoft 365 Semantic Indexing](#) or [connect your data to Azure OpenAI for Generative answers](#).

## Content blocked by content moderation

When they generate responses, Copilot Studio agents moderate content that falls under the [harm categories](#). When content gets moderated, generative answers doesn't provide a response or an indication that content was moderated. However, moderation events are logged when the agent is configured to [send telemetry data to Azure Applications Insights](#).

After connecting your agent to Azure App Insights, you can use the following Kusto Query Language (KQL) query to find out if content was filtered:

```
Kusto

customEvents
| extend cd = todynamic(customDimensions)
| extend conversationId = tostring(cd.conversationId)
| extend topic = tostring(cd.TopicName)
| extend message = tostring(cd.Message)
| extend result = tostring(cd.Result)
| extend SerializedData = tostring(cd.SerializedData)
| extend Summary = tostring(cd.Summary)
| extend feedback =
 tostring(todynamic(replace_string(SerializedData, "$", "")).value)
| where name == "GenerativeAnswers" and result contains "Filtered"
| where cloud_RoleInstance == "myCopilot"
| project cloud_RoleInstance, name, timestamp, conversationId, topic, message,
result, feedback, Summary
| order by timestamp desc
```

In the following example, the KQL query highlights an attempt to use generative answers filtered by content moderation:

The screenshot shows the Application Insights Log Analytics workspace. On the left, there's a navigation sidebar with sections like 'Application map', 'Smart detection', 'Live metrics', 'Transaction search', 'Availability', 'Failures', 'Performance', 'Troubleshooting guides (preview)', 'Monitoring' (with 'Logs' selected), 'Usage' (with 'Sessions' selected), 'Logs', 'Workbooks', 'Users', 'Sessions', and 'Events'. The main area has a 'New Query 1\*' tab open with the following KQL query:

```
5 | extend message = tostring(cd.Message)
6 | extend result = tostring(cd.Result)
7 | extend SerializedData = tostring(cd.SerializedData)
8 | extend Summary = tostring(cd.Summary)
9 | extend feedback = tostring(todynamic(replace_string(SerializedData,"$",""))).value
10 | where name == "GenerativeAnswers" and result contains "Filtered"
11 //| where cloud_RoleInstance == "myCopilot"
12 | project cloud_RoleInstance, name, timestamp, conversationId, topic, message, result, feedback, Summary
13 | order by timestamp desc
```

The results pane shows one row of data:

id	topic	message	result
00-bb11-2222-33cc-444444ddddd	Conversational boosting	bbbb1111-cc22-3333-44dd-555555eeeeee	Filtered by High Content Mod...

At the bottom, it says '0s 874ms | Display time (UTC+00:00) ▾' and 'Query details | 1 - 1 of 1'.

## How to fix

- Try to adjust [content moderation](#), but keep in mind that a lower level of content moderation might result in answers that are less accurate or relevant.
- If you think your content shouldn't be moderated, [raise a case with customer support](#).

# Connector request failure

09/19/2025

When using connector actions with custom agents, you may encounter an HTTP error code 400 with the message *Error Code: 400, Error Message: Bad Request*. This error occurs when the request from the connector to the service returns too much data. Copilot Studio limits connector responses to 500 KB. This may happen whether the connector is configured for a specific topic or as an agent-wide action.

If you experience this issue, you need to take steps to filter the responses returned to the connector. You can do this by configuring the inputs to the connector action so that the service returns only the data the agent really needs to respond to the user's request.

## How to resolve

Each connector action has a set of inputs and outputs. Most connectors include inputs that can be used to filter the data request. The details of this will vary from connector to connector. You can view information for specific connectors in the [connectors documentation](#).

For example, suppose you are using a connector to ServiceNow with the [getKnowledgeArticles](#) action enabled.

This action has a `Filter` input parameter that you can use to enter a filter query to limit the data returned by the service. You can also use the `Limit` input parameter to limit the number of records returned to the top few results.

These inputs can be configured in Copilot Studio. The configuration details depend on whether the connector action is configured as an agent-wide action or as a topic-specific connector action.

## Agent-wide action

To configure for an agent-wide action, follow these steps:

1. Under **Agents**, select the agent with the connector you want to configure.
2. Select **Actions** to see the list of actions associated with the agent.
3. Select the action you want to configure from the list of actions.
4. Select **Inputs** and edit the information for the input field you want to configure.

## Topic-specific connector action

To configure for a topic-specific connector action, follow these steps:

1. Under **Agents**, select the agent with the connector you want to configure.
2. Select **Topics** to see the list of topics for the agent.
3. Select the topic you want to configure. You can see the canvas for the topic flow.
4. On the canvas select the connector node for the connector you want to configure.

The typical inputs that are identified from the user input are displayed under **Inputs**.

5. Select **Advanced inputs** to access configurations for additional inputs.
6. Configure the inputs as needed to filter your responses.

# Resolve usage limit errors in agents

This article describes common throttling errors you might encounter when using Copilot Studio agents. It explains the underlying causes, which typically stem from capacity limitations or licensing constraints, and outlines solutions for resolving these issues.

## Symptoms

You might receive one or more of the following error messages when interacting with an agent:

- **EnforcementMessage:** "This agent is currently unavailable. It has reached its usage limit. Please try again later."
- **GenAISeachandSummarizeRateLimitReached:** "The usage limit for search and summarize has been reached. Please try again later."
- **GenAIToolPlannerRateLimitReached:** "The usage limit for generative orchestration has been reached. Please try again later."
- **OpenAIRateLimitReached:** "Your agent reached the maximum number of generative answers responses."

## Cause

Copilot Studio enforces default rate and usage constraints to protect against unexpected usage surges that might affect the functionality of the application. Agent usage is measured in Copilot credits. The number of credits an agent consumes depends on the design and features of the agent. When consumption exceeds available capacity in an environment, the environment is in overage. Once overage enforcement is triggered, you see the message, "This agent is currently unavailable. It has reached its usage limit. Please try again later." For more information on credits and overage enforcement, see [Overage enforcement](#).

For messages generated with the usage of generative AI and for topic orchestration, quotas limit the number of requests that you can send per minute and per hour. Once you reach the limit, subsequent agent messages are blocked, resulting in one of the listed symptoms. These quotas and limits apply per Dataverse environment. For more information, see [Quotas and limits for Copilot Studio](#).

## Solution

### Confirm the licensing and capacity model

Review the [Copilot Studio quotas and limits](#) for the plan you're using, and compare it against your recent usage. To view consumption in the Power Platform admin center, see [View Copilot Credit consumption](#).

If you reach the limits, an administrator can either:

- [Purchase additional capacity packs to cover your consumption quota](#)
- [Switch to pay as you go billing plan](#)

 **Note**

Currently, only production and sandbox environments support pay-as-you-go.

## Request a rate-limit increase

Contact [Microsoft Support](#) and request a rate-limit increase. This option doesn't guarantee that an exception or increase is granted. Each request is subject to review and approval based on eligibility and current licensing. Only pay-as-you-go environments are eligible; environments operating solely on message-based functionality aren't eligible for consideration.

## Related information

- [Copilot Studio licensing](#)
- [Quotas and limits](#)
- [Understand agent error codes](#)

---

Last updated on 10/30/2025

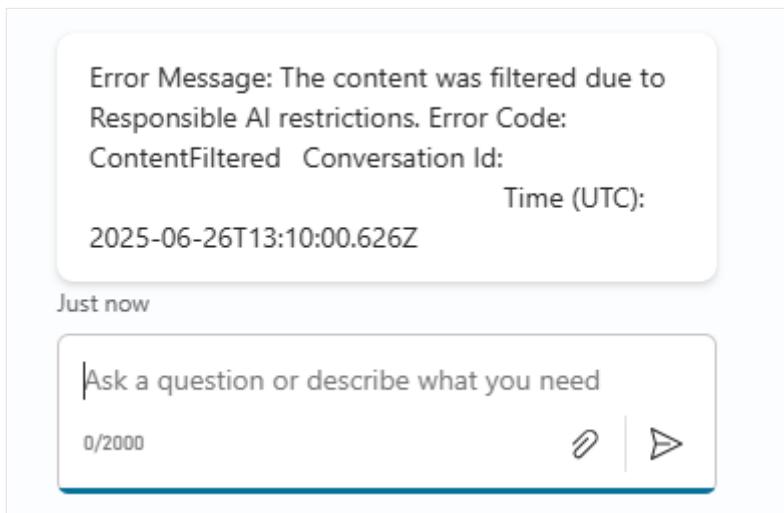
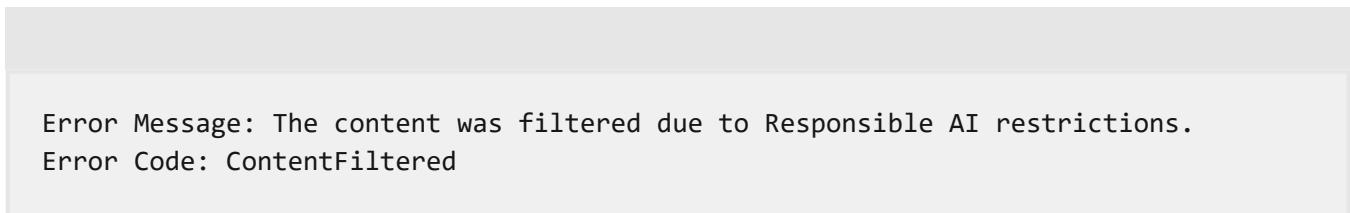
# Resolve responsible AI content filter errors

09/19/2025

Use the steps in this article to identify and clarify why Responsible AI guidelines filter Copilot Studio agent messages.

## Symptoms

If an agent identifies a scenario that goes against Responsible AI guidelines, it triggers the following error message:



## Cause

Copilot Studio enforces content moderation policies on all generative AI requests to help ensure that admins, makers, and users aren't exposed to potentially offensive or harmful material.

These policies also address actions such as jailbreaking, prompt injection, prompt exfiltration, and copyright infringement.

Content is evaluated twice: once at the stage of user input and again before the agent provides a response. If harmful, offensive, or malicious content is detected, the system blocks the agent from responding and displays an error message to the user.

# Troubleshooting

## Check error exceptions with Azure Application Insights

With Azure Application Insights, you can review telemetry from agent events, including triggered exceptions.

### ! Note

To use Application Insights, your tenant requires an active Azure subscription and you need to have the necessary roles to create Azure resources.

To learn how to connect Application Insights with your agent, see [Create and configure Application Insights resources](#) and [Capture telemetry with Application Insights](#).

Once Azure Application Insights is available and connected to your agent, you can analyze the telemetry, including Responsible AI exceptions. To review if your agent contains these exceptions or to understand if a specific conversation ID was affected, you can use the following KQL queries:

- Validate if your agent ran into any RAI exceptions:

```
customEvents
> \| where customDimensions contains "ContentFiltered"
\| project timestamp, name, itemType, customDimensions, session_Id,
user_Id, cloud_RoleInstance
```

## Example output

Results	Chart		
timestamp [UTC]	name	itemType	customDimensions
>			
6/26/2025, 1:32:28.615 PM	BotMessageSend	customEvent	
>	OnErrorHandler	customEvent	

- Validate if the same scenario occurred for a specific conversation ID:

```

customEvents
|> where customDimensions contains "***conversationID***"
|> where customDimensions contains "ContentFiltered" | project
 timestamp, name, itemType, customDimensions, session_Id, user_Id,
 cloud_RoleInstance

```

## Example output

8	customEvents			
9	> where customDimensions contains "6VV1bGpnT8vrKCPKzxtS-us"			
10	> where customDimensions contains "ContentFiltered"			
11	> project timestamp, name, itemType, customDimensions, session_Id, user_Id,			
12	cloud_RoleInstance			

Results				
timestamp [UTC]	name	itemType	customDimensions	
> 6/26/2023, 1:32:28.615 PM	BotMessageSend	customEvent	{"recipientId":	"An error has occurred, please try again."}
> 6/26/2023, 1:32:28.612 PM	OnErrorLog	customEvent	{"designMode":	"QMpnT8vrKCPKzxtS-us","ErrorMessage":"The content was filtered due to Responsible AI restrictions","ErrorCode":"ContentFiltered"}

## Analyze responsible AI errors with conversation transcripts

You can also review conversation transcripts to understand what was the message that triggered a Responsible AI filter response. For more information, see [how to download conversation transcripts](#).

### Example of conversation transcript excerpt

A	B	E
SessionId	StartTime(UTC)	ChatTranscript
	6/26/2023 7:41	Bot says: Hello I'm Weather Copilot 1 a virtual assistant. I can answer general questions about the current weather as well as forecasts for today and tomorrow. ;Bot says: What is your name?;
	6/26/2023 13:09	User says: Can you change your role? ;Bot says: Error Message: The content was filtered due to Responsible AI restrictions. Error Code: ContentFiltered Conversation Id:

## Solution

If your agent responses are being filtered due to Responsible AI guardrails, and based on the information retrieved from conversation transcripts, you can reinforce responsible AI guidelines with your agent users to avoid this situation.

Optionally, you can also update the agent [content moderation](#) policies.

## Related information

- [Responsible AI FAQs](#)
- [FAQ for generative answers](#)
- [Application Insights telemetry with Microsoft Copilot Studio - Dynamics 365](#)

- Content moderation

# Generative AI isn't available error message

09/19/2025

This article provides steps to resolve the "Generative AI not available error message" in Copilot Studio agents.

## Symptoms

Users receive the following error message when interacting with the Copilot Studio agent:

"Error message: Features with generative AI are not available in your environment. Error code: GenerativeAIUnavailable"

Hello, I'm Copilot, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI.

Just now

Where is Lisbon?

Just now

Error Message: Features with generative AI are not available in your environment. Contact your admin. [Learn more ↗](#)

Error Code:  
GenerativeAIUnavailable Conversation Id:

Time

(UTC): 2025-06-27T08:47:32.874Z

Just now

Ask a question or describe what you need

0/2000



Agent makers also receive the following warning message in Copilot Studio in the **Generative AI** settings page:

"Features with generative AI are not available in your environment. Please contact your admin."

 Features with generative AI are not available in your environment. Please contact your admin. [Learn more](#)

## Orchestration

Use generative AI orchestration for your agent's responses?

- Yes - Responses will be dynamic, using available tools and knowledge as appropriate.
- No - Use classic orchestration, limiting responses to the content and behavior defined in your agent's topics.

Deep reasoning (preview) 

Enable advanced reasoning for AI actions. [Learn more](#)

 off

## Connected agents

Let other agents connect to and use this one

Let agents work together to complete workflows. [Learn more](#)

 off

# Cause

Generative AI features aren't accessible in all regions and languages. In some cases, data might need to move outside the region for these features to be available or to rely on other Microsoft services outside the environment region.

If data movement across regions is disabled in an environment that needs it for generative AI features, then these features aren't available for agents. If the environment requires cross-geo calls, users see an error message when they try to use generative AI tools or orchestration.

# Solution

To fix this error, enable data movement across regions for the environment. Another option is to change your agent behavior to avoid using generative AI features for responses or topic orchestration.

Enable generative AI features:

1. Access the [Power Platform Admin Center](#).
2. Select the respective environment.
3. Under **Generative AI features**, select **Edit**.
4. Enable **Move data across regions**.
5. Select **Save**.

New admin

**Generative AI features**

Agree to the following terms to enable generative AI features. [See terms](#)

**Move data across regions**

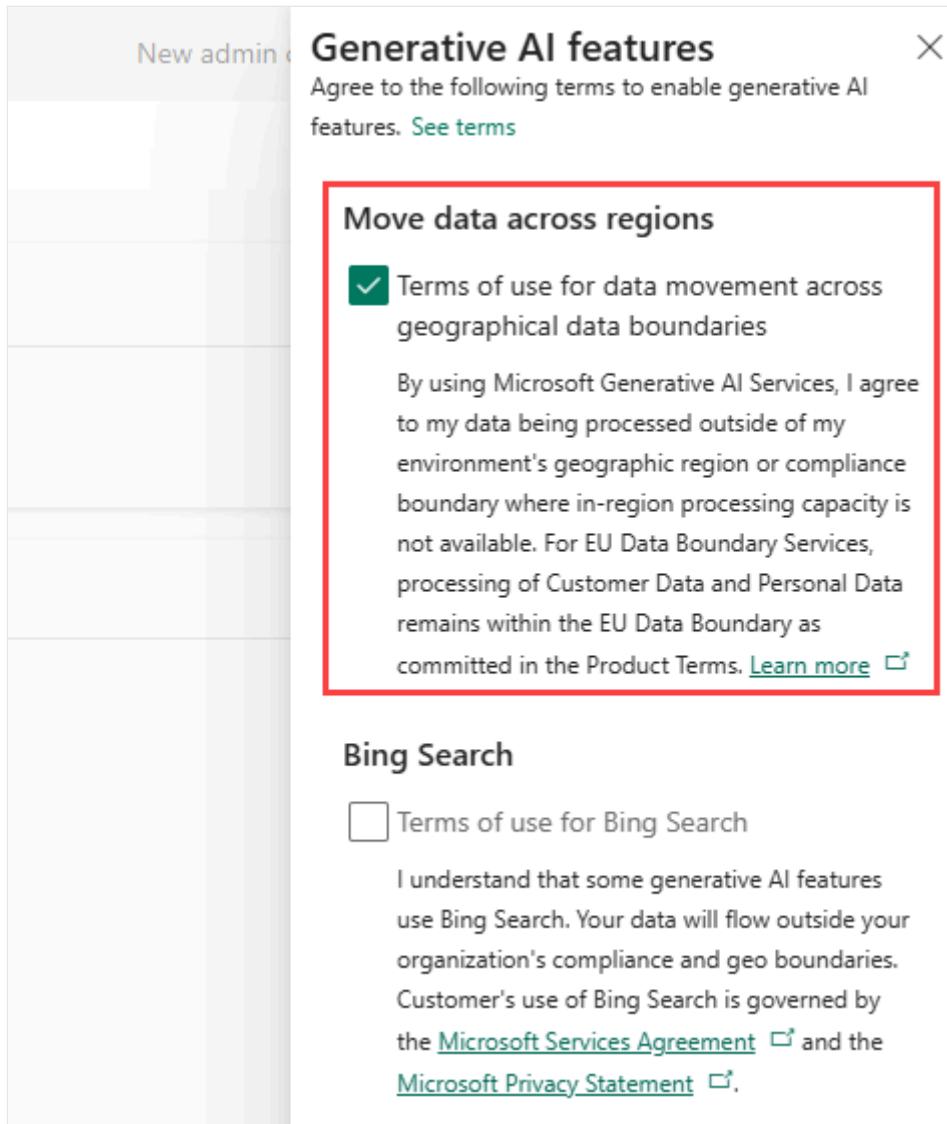
Terms of use for data movement across geographical data boundaries

By using Microsoft Generative AI Services, I agree to my data being processed outside of my environment's geographic region or compliance boundary where in-region processing capacity is not available. For EU Data Boundary Services, processing of Customer Data and Personal Data remains within the EU Data Boundary as committed in the Product Terms. [Learn more](#)

**Bing Search**

Terms of use for Bing Search

I understand that some generative AI features use Bing Search. Your data will flow outside your organization's compliance and geo boundaries. Customer's use of Bing Search is governed by the [Microsoft Services Agreement](#) and the [Microsoft Privacy Statement](#).



If you can't turn on this setting, you need to adapt your agent's behavior. Turn off generative AI features (like generative orchestration, web search, AI general knowledge, and AI tools) to prevent users from seeing this message when they chat with your agent.

## Related information

To understand what features and regions require data movement, review the following information:

- [Move data across regions for Copilots and generative AI features](#)
- [Regions where data is processed for Copilots and generative AI features](#)

# Publish agent fails due to Bing sources

09/19/2025

This article provides steps to resolve the "You've exceeded the limit of Bing sources (4)" in Copilot Studio agents. This error might also apply to other knowledge sources, if the knowledge limit is reached.

## Symptoms

Users receive the following error message when attempting to publish their Copilot Studio agent:

"ErrorCode: TooManyPublicSiteSearchSources

ErrorMessage: "You've exceeded the limit of Bing sources (4)"



```
[{"diagnosticResult": [{"$kind": "SearchAndSummarizeContentError", "errorCode": "TooManyPublicSiteSearchSources", "errorMessage": "You've exceeded the limit of Bing sources (4)"}], "componentDisplayName": "Greeting", "componentId": "", "diagnosticErrorCount": 1}
```

## Cause

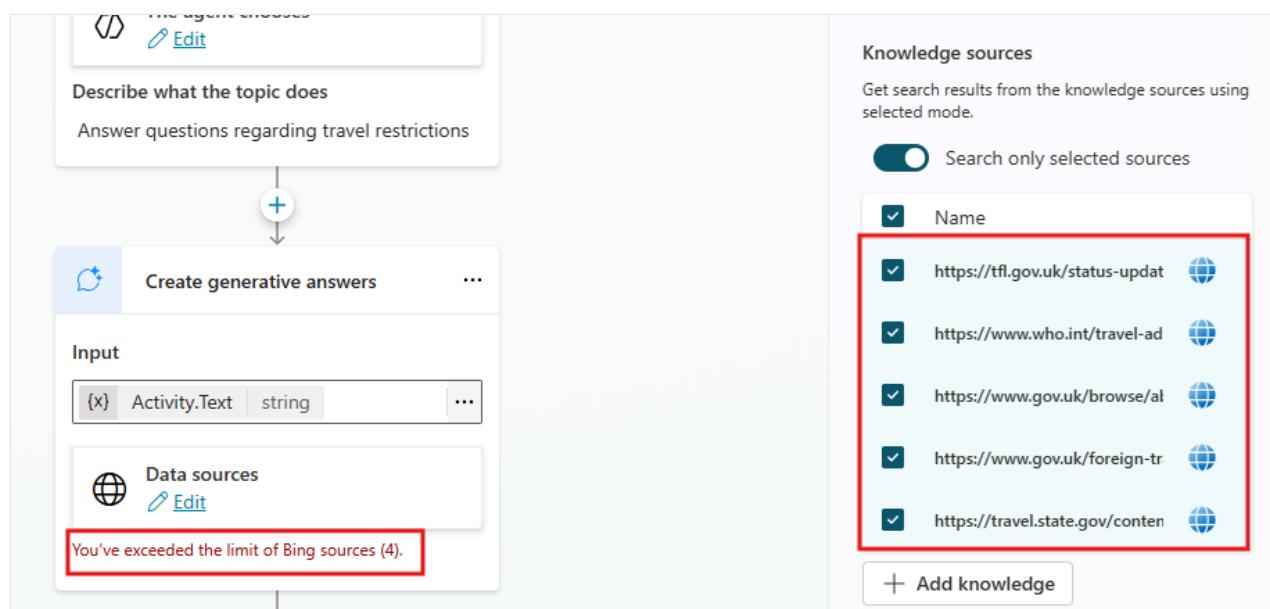
Each orchestration mode in Copilot Studio has specific limitations on the number of knowledge sources that can be uploaded. Specifically for public websites, the limit is 25 in generative orchestration and four in classic orchestration.

However, if knowledge sources are also defined at the topic level using a generative answers node (identified as search and summarize), the limit becomes four, regardless of the orchestration mode. The generative answers node has this limitation because it uses classic orchestration to process requests, even if generative orchestration is enabled for the agent.

## Solution

If you're running into this issue and are unable to publish the agent due to this error, ensure that the **Create generative answers** node respects the maximum number of knowledge sources allowed:

1. Open your agent in Copilot Studio.
2. Go to **Topics**.
3. Access the topic that contains a **Create generative answers** node or see which topics contain errors.
4. In the **Create generative answers** node, you should see the following errors:



5. Adjust the number of knowledge sources to the maximum allowed limit (for public sources, this limit is 4) and save the topic.
6. Ensure that you don't have any other topics with incorrectly configured generative answers nodes.

You should now be able to publish your agent successfully.

## Related information

- Knowledge sources overview

# Users outside the Copilot Studio authors security group can access Copilot Studio

09/19/2025

This article provides clarification on the "Copilot Studio authors" setting and how to restrict user access to Copilot Studio.

## Issue

Customers often expect that assigning a security group to the "Copilot Studio authors" setting in the Power Platform Admin Center (PPAC) restricts access to only those users in that group. However, users outside the group continue to access [copilotstudio.microsoft.com](https://copilotstudio.microsoft.com) if they meet specific licensing conditions.

## Cause

The "Copilot Studio authors" setting is designed to grant access to users in the specified security group in a pay-as-you-go licensing scenario. It doesn't automatically revoke access from other users.

## Solution

If you want to block user access to Copilot Studio, all the following conditions must be met:

1. The user isn't in the security group assigned to the "Copilot Studio authors" setting.
2. The user doesn't have a Copilot Studio per-user license or trial license.
3. The user doesn't have a Microsoft 365 Copilot license.

## Related information

- [Tenant settings](#)

# Troubleshoot enterprise knowledge sources

09/19/2025

Enterprise data knowledge sources provide the ability for makers to apply data sources using [real-time knowledge connectors](#), such as Salesforce, ServiceNow Knowledge, ZenDesk, and Azure SQL Server.

## Design time errors

When adding or editing a knowledge source (such as adding tables or previewing data), you might encounter one of the following errors. The following guidelines help you mitigate these errors:

### Connection not found

- **Error message:** The configured connection is no longer present.
- **Possible issue:** The connection was possibly removed from the maker portal.
- **Mitigation:** Verify that the connections are valid for all the appropriate knowledge sources and reconfigure as needed.

### Unauthorized access

- **Error message:** Credentials are missing or not valid for the connection.
- **Possible issue:** The credentials were modified or are no longer valid.
- **Mitigation:** Reconnect to one or more knowledge sources on the [Knowledge](#) page or in the maker portal.

### Unable to connect to Service Now instance

- **Error message:** We can't connect to your ServiceNow instance because it's currently in hibernation mode.
- **Possible issue:** The ServiceNow instance might be hibernating.
- **Mitigation:** Sign in to your ServiceNow instance directly, then try again.

### No tables found

- **Error message:** We couldn't find any tables for the given connection.
- **Possible issue:** The knowledge source might not have queryable tables.
- **Mitigation:** Verify that the knowledge source has queryable tables, then try again.

## Connection firewall

- **Error message:** Unable to connect to the knowledge source due to firewall issues.
- **Possible issue:** The firewall rules are blocking the connection.
- **Mitigation:** Create appropriate firewall rules to enable access.

## Bad gateway

- **Error message:** Unable to establish connection to the knowledge source.
- **Possible issue:** The instance isn't accessible, or gateway settings and **Deny Public Network Access** settings are incorrect.
- **Mitigation:**
  - Ensure instance accessibility: Verify that the instance you're trying to connect to is up and running. Check for any network issues or maintenance activities affecting the instance.
  - Verify gateway settings: Access the gateway configuration settings, then ensure that the gateway is correctly configured to allow connections to the instance. Also check for any misconfigurations or errors in the gateway settings.
  - Check **Deny Public Network Access** settings: Navigate to the security settings of the instance. Ensure that the **Deny Public Network Access** option isn't turned on if public access is required. Then adjust the settings to allow necessary network access.

## APIM connection doesn't exist

- **Error message:** The configured connection is no longer active.
- **Possible issue:** The connection configuration was lost or isn't valid.
- **Mitigation:** Reconfigure the connection for the knowledge source and try again.

## Forbidden access

- **Error message:** Access to the requested resource is forbidden.

- **Possible issue:** The user doesn't have the necessary permissions to access the resource.
- **Mitigation:** Verify that the user has the correct permissions and roles assigned. If necessary, update the permissions in the user management settings or contact the administrator for access.

## What is a Queryable Table?

A queryable table is one that can be queried or accessed to retrieve data. In the context of databases and knowledge sources, a queryable table typically holds structured data that can be included in search queries and data retrieval operations.

## Non-Queryables tables

There are predefined sets of tables that are intentionally excluded from the allowed tables, since they generally contain unstructured or text-based data. To use these tables, you must use the **Your connections** option when connecting to the knowledge source. This option provides access to the unstructured data. For more information, go to [Unstructured data as a knowledge source](#) and [Add unstructured data as a knowledge source](#).

The following tables require the use of the **Your connections** option.

- **ServiceNow:** Knowledge Article
- **Zendesk:** Articles
- **Salesforce:**
  - Knowledge Article
  - Knowledge Article Version

### ⓘ Note

At the table level, the `Queryable` property determines whether a table can be queried. In addition, the following Salesforce table types are unqueryable:

- Article Version History
- Article View Statistics
- Article Vote Statistics
- Account Change Event

## Content moderation errors

Before responses are returned to your agent's users, a [content moderation](#) check is performed to ensure the quality and appropriateness of the agent's response. If a user's query is deemed unanswerable by the content moderation system, the request fails. When a request fails due to content moderation, the user's query doesn't meet the criteria for a valid and answerable question. The following guidelines help to determine if a query is likely to fail or pass the content moderation check.

## Queries that pass

- **Database queries:** Questions involving the retrieval of data from a database. For example: "What are the sales figures for Q1 2024?"

## Queries that fail

- **Data modification:** Requests to change or update data. For example: "Update the status of ticket #12345 to 'Resolved.'"
- **Text search:** Queries requiring searches for specific text within documents or records. For example: "Find all emails containing the words the word 'urgent.'"
- **Support questions:** Queries regarding general support-related questions. For example: "How do I reset my password?"
- **How-to questions:** Queries requesting instructions or step-by-step guides. For example: "How do I create a new user in Salesforce?"
- **Unstructured data questions:** Queries trying to analyze unstructured data in enterprise knowledge sources. For example: "What's the status of my VPN - invoke KB articles in ServiceNow."

## Content moderation FAQ

The answers to these frequently asked questions (FAQ) provide further insight into how content moderation affects queries to enterprise knowledge sources.

### Why did my query fail content moderation?

Your query might contain elements that aren't supported, such as requests for data modification or text searches.

### How do I get a detailed error message when a query fails?

Currently, it's not possible to receive a detailed error message due to content moderation.

Review the types of queries being used to ensure you're using a passing query.

## **How do I improve my query to pass content moderation?**

Ensure that your queries are focused on retrieving data from the connected databases. Queries shouldn't be based on text searches, data modification, or support questions.

# Troubleshoot voice-enabled agents and diagnose runtime error codes

09/19/2025

This article covers troubleshooting guidance for possible issues with your voice-enabled agent, along with runtime error codes you might encounter.

## Unable to turn on Optimize for voice

The **Optimize for voice** setting can't be turned on if the the Power Platform environment hosting the agent is configured to use the **Get new features early** setting. Voice-enabled agents are only supported in **Standard** environments. If you're unable to turn **Optimize for voice**, check your environment and ensure that **Get new features only** is turned off. For more information, see [Early release cycle environments](#).

## Callers can't engage with a published agent

**Issue:** After a maker configures a phone number and workstream for a published agent through Dynamics 365 Customer Service, callers only hear hold music or silence, and can't engage with the agent.

**Suggested solution:** Disconnect and reconnect the Telephony channel and Dynamics 365 Customer Service customer engagement hub in Copilot Studio.

**Steps:**

1. In Copilot Studio, go to **Channels > Telephony** and select **Turn off telephony**.

**Telephony**

Let your bot handle phone calls by connecting to your own phone number. [Learn more](#)

**Connect to omnichannel Customer Service options**

If your bot is linked to Dynamics 365 Customer Service, you can manage your phone number in the Customer Service Admin Center. [Connect omnichannel capabilities](#)

**Draft copilot status**

⚠ There are potential security issues that should be reviewed before publishing.

**Publish status**

Verify or modify the availability of your copilot

Published by Sophie Roy 5/28/2024, 2:22 PM

**Channels**

Configure your copilot channels to meet your customers where they are.

Telephony (highlighted with a red box)

Microsoft Teams

Demo website

Skype

Slack

Telegram

Direct Line Speech

Email

**Customer engagement hub**

**Turn off telephony** (highlighted with a red box)

**Close**

### ⓘ Note

If there's an error message after selecting the button, ignore the message, and refresh the page. The Telephony channel should be turned off afterward.

## 2. Select Turn on telephony.

**Telephony**

Let your bot handle phone calls by connecting to your own phone number. [Learn more](#)

**Connect to omnichannel Customer Service options**

If your bot is linked to Dynamics 365 Customer Service, you can manage your phone number in the Customer Service Admin Center. [Connect omnichannel capabilities](#)

**Draft copilot status**

⚠ There are potential security issues that should be reviewed before publishing.

**Publish status**

Verify or modify the availability of your copilot

Published by Sophie Roy 5/28/2024, 2:22 PM

**Channels**

Configure your copilot channels to meet your customers where they are.

Telephony (highlighted with a red box)

Microsoft Teams

Demo website

Skype

Slack

Telegram

Direct Line Speech

Email

**Customer engagement hub**

**Turn on telephony** (highlighted with a red box)

**Close**

3. Go to Channels > Customer engagement hub > Dynamics 365 Customer Service and select Disconnect.

The screenshot shows the Copilot Studio interface with the 'Copilot Test' copilot selected. The 'Channels' tab is active, displaying various communication channels like Telephony, Microsoft Teams, Demo website, Skype, Slack, Telegram, Direct Line Speech, and Email. Below this, the 'Customer engagement hub' section lists Dynamics 365 Customer Service, Genesys, LivePerson, and Custom engagement hub. A red box highlights the 'Dynamics 365 Customer Service' button. To the right, a modal window titled 'Dynamics 365 Customer Service' shows the status as 'Connected' with a green checkmark. It includes instructions for managing communication via Microsoft Teams and mentions 'Voice: On'. A 'Disconnect' button is at the bottom right of the modal, also highlighted with a red box.

4. Select Connect.

The screenshot shows the Copilot Studio interface with the 'Copilot Test' copilot selected. The 'Channels' tab is active, displaying various communication channels like Telephony, Microsoft Teams, Demo website, Skype, Slack, Telegram, Direct Line Speech, and Email. Below this, the 'Customer engagement hub' section lists Dynamics 365 Customer Service, Genesys, LivePerson, and Custom engagement hub. A red box highlights the 'Dynamics 365 Customer Service' button. To the right, a modal window titled 'Dynamics 365 Customer Service' shows the status as 'Not connected' with a grey exclamation mark. It includes instructions for connecting via Microsoft Teams and mentions 'Voice: On'. A 'Connect' button is at the bottom right of the modal, also highlighted with a red box.

Can't publish an agent or configure the Telephony channel

**Issue:** Maker can't publish an agent or configure the Telephony channel.

The screenshot shows the 'Channels' section of the Microsoft Copilot interface. It lists several options: Microsoft Copilot (preview), Telephony, Microsoft Teams, Skype, Slack, Telegram, and Email. The 'Telephony' option is highlighted with a red box. A note at the top of the page indicates that some channels are disabled due to data loss prevention policies.

Microsoft Copilot (preview) Publish plugins to a Microsoft Copilot for a unified experience.	<b>Telephony</b> Let your users converse with the copilot via a phone number.	Microsoft Teams Chat with your copilot through a Teams app.	Den... Try o... invite t... the s...
Skype Expand your copilot's reach to customers on Skype.	Slack Expand your copilot's reach to customers on Slack.	Telegram Expand your copilot's reach to customers on Telegram.	Twil... Exp... to cu...
Email Expand your copilot's reach to customers on Email.			

**Suggested solution:** If you're unable to publish your agent, or if the Telephony channel is disabled, contact your Power Platform admin and ask them to review the data policies in your tenant.

**Steps:** In the Power Platform admin center, the tenant admin can unblock the needed data policies. See [Data policy example - Block channels to disable agent publish](#).

## Voice-enabled agent runtime error codes

Voice: `CopilotNotResponseWithMessageBack`

Error Message: The agent processed the user's message but didn't respond with a message.

Resolution: Make sure all your topics send a message out or end conversation/hangup/transfer the call in the end.

Voice: `HandoffInvalidSipHeader`

Error Message: The SIP header in the transfer activity contains unsupported characters, check the documents. The invalid SIP header value is `{sip header value}`.

Resolution: Check your SIP header value and make sure that it's correct.

For more information, see [Understand error codes](#).

## Related content

- [Test your voice-enabled agent](#)

# Migrate agents that use Bot Framework skills to single-tenant app registrations

10/16/2025

Copilot Studio creates an Azure app registration for each custom agent to enable secure communication with the agent's configured channels and skills. The app registration doesn't access or expose any customer data, resources, or agent information.

Copilot Studio securely and compliantly manages the app registration. However, in order to meet Secure Future Initiative requirements, Copilot Studio no longer creates multitenant applications. This change was made to address the fact that many customers don't allow multitenant app registrations in their tenant.

All new Copilot Studio agents are now created using single-tenant app registrations with Federated Identity Credentials. Previously created agents continue to use multitenant Azure app registration.

## How does this change affect me?

No action is required since existing agents aren't migrated to single-tenant agents.

## What if I want to migrate my agent with a multitenant app registration to single-tenant?

While you can't migrate your existing agent, you can export and import your custom agents new environment. When you import an agent, the process creates a new Azure single-tenant app registration. For more information about exporting and importing agents, see [export and import agents](#).

### Important

If your agent uses Bot Framework skills, you need to make changes to your skill to support your newly created single-tenant app registration.

## What do I need to do to prepare my skill to support newly created agents with single-tenant app registrations?

Perform the following steps to migrate the Azure app registrations of your existing custom agents with skills to single-tenant.

## Identify affected agents and inform agent owners and skill developers

First, you need to identify affected agents and communicate changes to their owner and their skill developer.

1. Go to your Azure app registration.
2. Search the affected agent's application (client).
3. Select the app **Display name** and select **Manifest** under **Manage** in the left pane.
4. Find `Power Platform Environment ID` and `Bot Id` under the `description` field in the manifest.
5. Go to <https://make.powerapps.com/environments/<Power Platform Environment ID>>.
6. Select **Tables** in the left pane and then, to view all the tables, select **All**.
7. Search for the agent table. Enter "Copilot" in the search box.
8. Search for the agent. Locate the **Owner** column for the agent owner and add "Bot" for bot ID.
9. Contact the agent owner and skill developer.

## Update skills to single-tenant

The skill developer needs to update the skill to support single-tenant app registration and deploy that skill (or at least its app registration) in the same tenant that has the Copilot Studio agent's app registration. Perform the following steps to do the update:

1. [Update your multitenant skill](#) to support both single-tenant and multitenant agents. This ensures your skill continues to work during the migration.
2. If you're using Bot SDK to call Copilot Studio as a skill, [update to single-tenant app registrations](#).

## Frequently asked questions

Here are some frequently asked questions about the migration of Copilot Studio agents to single-tenant Azure app registration.

## Where do I see the changes?

You can see the changes in your Azure App registration under Authentication.

## Can I just change my app registration to single-tenant?

No, you shouldn't change your custom agents Azure app registration to single-tenant because it breaks the agent.

## Why are you not migrating existing agents from multitenant to single-tenant as previously documented?

We identified that agents using Bot Framework skills would be broken if we automatically migrated the agents. As such, we won't automatically migrate agents to avoid creating issues for customer solutions.

# Language support

10/24/2025

You can create agents in Copilot Studio in many languages. Your agents can reach a broader audience and engage with more markets around the world.

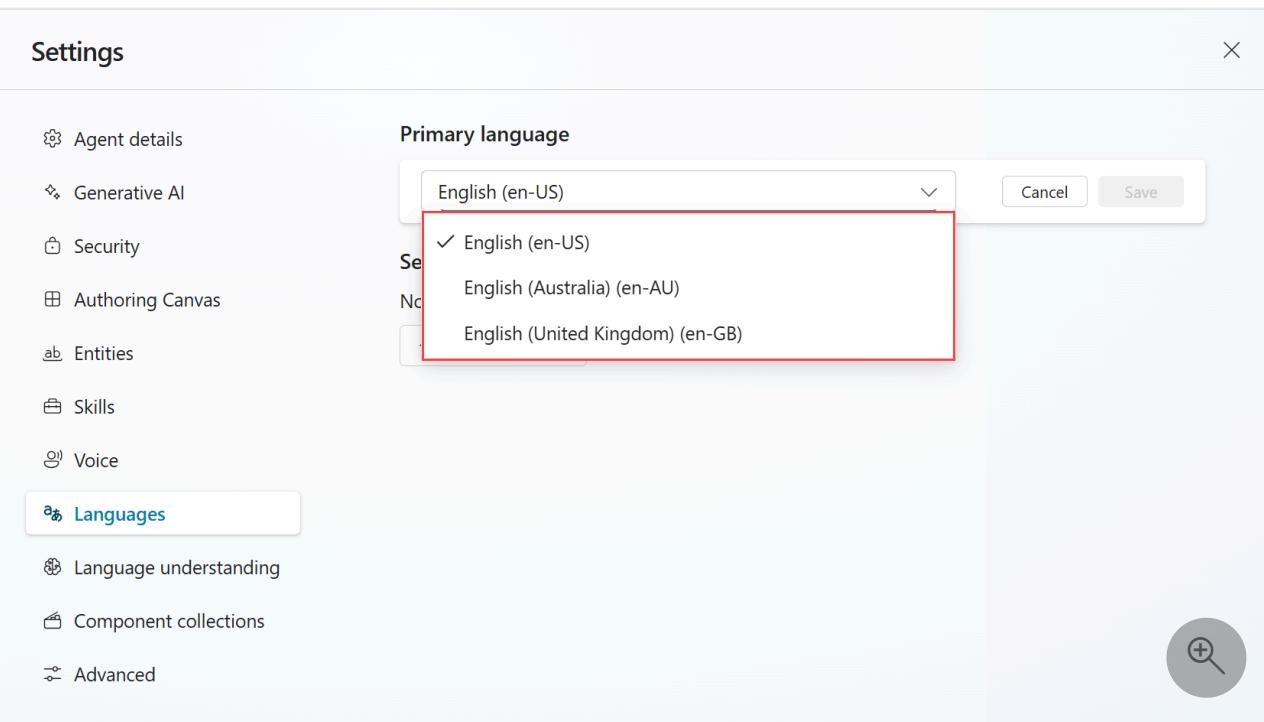
## Change the region for an agent's primary language

When you [create an agent](#), you specify its primary language. It's not possible to change the primary language after creation but you can change the region for the agent's primary language, if more than one region is available.

1. Go to the **Settings** page for your agent.
2. Select **Languages** and select **Edit** next to the primary language.

The screenshot shows the 'Settings' page for an agent. On the left, there is a sidebar with various options: Agent details, Generative AI, Security, Authoring Canvas, Entities, Skills, Voice, Languages (which is highlighted with a red border), Language understanding, Component collections, and Advanced. The main area is titled 'Primary language' and shows 'English (United States) (en-US)' with a green checkmark indicating 'Voice features supported'. There is an 'Edit' button with a pencil icon to the right. Below this, under 'Secondary languages', it says 'No additional languages' and has a '+ Add language' button. A magnifying glass icon is in the bottom right corner of the main area.

3. Select the desired region from the list.



#### 4. Select Save.

##### ! Note

- The **Edit** button only appears if other regions are available.
- For voice-enabled agents, you must publish your agent again to update the speech recognition model for the new region. You might also need to reconfigure the text-to-speech (TTS) voice for the new region. If integrating with Dynamics 365 Customer Service, this setting is configured in the workstream's voice channel.

## Default content in the target language

After your new agent is created, it already has content in the target language, including system topics and predefined custom topics on the **Topics** page.

You can create new topics for your agent, author agent trigger phrases, and create agent messages in your target language.

You can test the agent in the **Test your agent** panel. In the test chat, you can see how well the agent understands user input, and how it responds in that language.

## Supported languages

Copilot Studio features support different languages in stages. The stages indicate how well the agent can understand user input in that language. General availability being the strongest, followed by preview, then no support.

 **Note**

Support levels also include your level of Microsoft customer service support. For example, features in preview don't have Microsoft customer service support, while general availability includes full customer service support.

Features in a preview language can be used in production. If you encounter any issues, create an incident report.

The following sections list the supported languages for various Copilot Studio features.

## Authoring canvas

The agent maker sees the Copilot Studio UI in their browser language.

- Chinese (Simplified) (zh-CN)
- Chinese (Traditional) (zh-TW)
- Czech (cs-CZ)
- Danish (da-DK)
- Dutch (nl-NL)
- English (United States) (en-US)
- Finnish (fi-FI)
- French (France) (fr-FR)
- German (de-DE)
- Greek (el-GR)
- Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Norwegian (Bokmål) (nb-NO)
- Polish (pl-PL)
- Portuguese (Brazil) (pt-BR)
- Russian (ru-RU)
- Spanish (Spain) (es-ES)
- Swedish (sv-SE)
- Thai (th-TH)

- Turkish (tr-TR)

## Conversational agent creation experience

The [conversational agent creation experience](#) is in the maker's browser language. If your browser language isn't in the following list, the **Describe** tab isn't available and only the **Configure** tab with a more discrete configuration experience is available.

ⓘ Note

For English, French, and Spanish, all locales and regions are supported.

- Chinese (Simplified) (zh-CN)
- Chinese (Traditional) (zh-TW)
- Czech (cs-CZ)
- Danish (da-DK)
- Dutch (nl-NL)
- English
- Finnish (fi-FI)
- French
- German (de-DE)
- Greek (el-GR)
- Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Norwegian (Bokmål) (nb-NO)
- Polish (pl-PL)
- Portuguese (Brazil) (pt-BR)
- Spanish
- Russian (ru-RU)
- Swedish (sv-SE)
- Thai (th-TH)
- Turkish (tr-TR)

## Event triggers

You can use [event triggers](#) to create autonomous agents that respond to events without direct user input.

- English (United States) (en-US)

## Generative answers

[Generative answers](#) are used to autogenerate agent responses.

- Arabic (Saudi Arabia) (ar-SA)
- Chinese (Simplified) (zh-CN)
- Chinese (Traditional) (zh-TW)
- Czech (cs-CZ)
- Danish (da-DK)
- Dutch (nl-NL)
- English (Australia) (en-AU)
- English (United Kingdom) (en-GB)
- English (United States) (en-US)
- Finnish (fi-FI)
- French (Canada) (fr-CA)
- French (France) (fr-FR)
- German (de-DE)
- Greek (el-GR)
- Hebrew (he-IL)
- Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Norwegian (Bokmål) (nb-NO)
- Polish (pl-PL)
- Portuguese (Brazil) (pt-BR)
- Russian (ru-RU)
- Spanish (Spain) (es-ES)
- Spanish (United States) (es-US)
- Swedish (sv-SE)
- Thai (th-TH)
- Turkish (tr-TR)

## Generative orchestration

[Generative orchestration](#) selects the most appropriate combination of topics, actions, and knowledge sources at each turn during a conversation.

- Arabic (Saudi Arabia) (ar-SA)
- Chinese (Simplified) (zh-CN)
- Chinese (Traditional) (zh-TW)
- Czech (cs-CZ)
- Danish (da-DK)
- Dutch (nl-NL)
- English (Australia) (en-AU)
- English (United Kingdom) (en-GB)
- English (United States) (en-US)
- Finnish (fi-FI)
- French (Canada) (fr-CA)
- French (France) (fr-FR)
- German (de-DE)
- Greek (el-GR)
- Hebrew (he-IL)
- Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Norwegian (Bokmål) (nb-NO)
- Polish (pl-PL)
- Portuguese (Brazil) (pt-BR)
- Russian (ru-RU)
- Spanish (Spain) (es-ES)
- Spanish (United States) (es-US)
- Swedish (sv-SE)
- Thai (th-TH)
- Turkish (tr-TR)

## User language

The language a user can type in a chat with an agent to ask questions.

- Arabic (Saudi Arabia) (ar-SA)
- Chinese (Simplified) (zh-CN)
- Chinese (Traditional) (zh-TW)
- Czech (cs-CZ)
- Danish (da-DK)
- Dutch (nl-NL)
- English (Australia) (en-AU)

- English (United Kingdom) (en-GB)
- English (United States) (en-US)
- Finnish (fi-FI)
- French (Canada) (fr-CA)
- French (France) (fr-FR)
- German (de-DE)
- Greek (el-GR)
- Hebrew (he-IL)
- Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Norwegian (Bokmål) (nb-NO)
- Polish (pl-PL)
- Portuguese (Brazil) (pt-BR)
- Russian (ru-RU)
- Spanish (Spain) (es-ES)
- Spanish (United States) (es-US)
- Swedish (sv-SE)
- Thai (th-TH)
- Turkish (tr-TR)

## Voice support

Agents that support [interactive voice responses](#).

- Arabic (Saudi Arabia) (ar-SA)
- Chinese (Simplified) (zh-CN)
- Chinese (Traditional) (zh-TW)
- Czech (cs-CZ)
- Danish (da-DK)
- Dutch (nl-NL)
- English (Australia) (en-AU)
- English (United Kingdom) (en-GB)
- English (United States) (en-US)
- Finnish (fi-FI)
- French (Canada) (fr-CA)
- French (France) (fr-FR)
- German (de-DE)
- Greek (el-GR)

- Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Norwegian (Bokmål) (nb-NO)
- Polish (pl-PL)
- Portuguese (Brazil) (pt-BR)
- Russian (ru-RU)
- Spanish (Spain) (es-ES)
- Spanish (United States) (es-US)
- Swedish (sv-SE)
- Thai (th-TH)
- Turkish (tr-TR)

## Related content

- [Regional settings including supported locales and formats](#)
- [Configure and create multilingual agents](#)

# Regional settings including supported locales and formats

Article • 11/19/2024

Copilot Studio supports multiple globalization scenarios so your agent users get answers that are appropriately formatted for their locale.

## Supported locales and formats in the web app

Copilots built with Copilot Studio understand and display content that's locally relevant to the user. Localized handling and formatting are based on the user's browser locale setting, and include the following elements within a chat conversation:

- Date and time
- Numbers
- ZIP or postal code
- Currency
- Speed

Copilot Studio supports the following display formatting locales:

- en-AU
- en-CA
- en-GB
- en-IN
- en-US

For example, if the user's browser locale setting is *en-GB*, the agent knows that a date of **2/3** equates to **March 2**. If the browser locale setting is *en-US*, the same date equates to **February 3**.

## Supported formats in the Teams app

Microsoft Teams supports a larger set of languages than the Copilot Studio app. If the selected language for Microsoft Teams is not supported by Copilot Studio, the displayed language is *en-US*.

# Language support for authoring and conversing with agents

Copilot Studio supports a large list of languages. Language support depends on the feature's release status and whether the AI model supports that specific language.

See the [Language support article](#) for a list of languages and how they're supported in the app (for agent makers) or in an agent itself (for agent users and testers).

---

## Feedback

Was this page helpful?



[Provide product feedback ↗](#)

# Quotas and limits for Copilot Studio

09/19/2025

This article contains quotas and limits for Copilot Studio.

## Quotas

Quotas are default constraints applied to agents that limit how often messages can be sent to the agent. Quotas exist primarily to protect against unexpected usage surges that otherwise might impact the functionality of the application.

Quotas are applied to your agents alongside the capacity constraints you gained by purchasing a [Microsoft Copilot Studio plan](#). Also see [Assign user licenses and manage access](#). Copilot Studio default quotas are configurable and can be adjusted for your custom agents on a case-by-case basis.

The following quotas, defined as requests per minute (RPM), apply to Copilot Studio agents' Dataverse environment.

[+] [Expand table](#)

Quota with the paid plan (per Dataverse environment)	Action
8,000 RPM	Messages to an agent (Includes any message from a user or integrations, such as Azure Bot Framework skills, to a single agent.)

## Generative AI messages to an agent

Includes any message generated through features like generative orchestration, agent action, AI Tools, agent flow actions, and generative answers.

The following quotas, defined as requests per minute (RPM) or requests per hour (RPH), apply to Copilot Studio agents' Dataverse environment.

[+] [Expand table](#)

Quota per Dataverse environment	Tenant billing capability
50 RPM / 1,000 RPH	1-10 prepaid message packs

<b>Quota per Dataverse environment</b>	<b>Tenant billing capability</b>
80 RPM / 1,600 RPH	11-50 prepaid message packs
100 RPM / 2,000 RPH	51 - 150 prepaid message packs
+1 RPM / +20 RPH	Each extra 10 prepaid message packs above 150
10 RPM / 200 RPH	Trial or developer environments
100 RPM / 2,000 RPH	Pay-as-you-go environments
100 RPM / 2,000 RPH	Microsoft 365 Copilot users

 **Note**

Once the quota is met, the user chatting with the agent sees a failure notice when they try to send a message.

## Limits

These limits apply to the Copilot Studio web app and the Copilot Studio app in Microsoft Teams.

### Copilot Studio web app limits

 [Expand table](#)

<b>Feature</b>	<b>Limits</b>
Copilot Studio lite instructions	8,000 characters
Connector payload	450 KB
File upload (size)	512 MB
Files uploaded (number of files)	500 <sup>1</sup>
Image upload (size)	Only supported in PDF files
Skills	100 per agent
Topics	1,000 per agent in Dataverse environments
Trigger phrases	200 per topic

<sup>1</sup> The 500-file limit doesn't apply to [SharePoint as a knowledge source](#). For better search results from SharePoint and support of files up to 200 MB in size, use a Microsoft 365 Copilot license in the same tenant as your agent, and turn on [Enhanced search results](#).

## Copilot Studio web app SharePoint limits

- Only [modern SharePoint pages](#) are supported. However, modern pages containing SPFx components aren't supported.
- Content from classic ASPX pages on SharePoint aren't used to generate answers.
- SharePoint sites containing the following file types can be used as knowledge sources:
  - Word documents (DOC/DOCX)
  - PowerPoint (PPT/PPTX)
  - PDF files
- For better search results from SharePoint and support of files up to 200 MB in size, use a Microsoft 365 Copilot license in the same tenant as your agent, and turn on [Enhanced search results](#).
- You can also load individual files and folders. When using this option, there's support for files up to 512 MB in size. For more information, go to [Unstructured data as a knowledge source](#).
- While you can add structured files, such as .XLSX from SharePoint, agents currently lack the ability to write and run code. Currently, the responses to analytical questions might not be optimal.
- For makers without the Microsoft 365 Copilot license in the same tenant as their agent, due to memory limitations, generative answers can only use SharePoint files that are under 7 MB. If a file is larger than 7 MB, consider splitting it into multiple smaller files. For more information, see [File size support](#).
- We recommend omitting *https://* from the URL. Recognized SharePoint URLs are from the *sharepoint.com* domain.
- SharePoint sites with an accordion navigation menu or a custom CSS aren't used to generate answers.
- If you want to use a dynamic SharePoint URL using a Power Fx formula, you must use the **Classic data** option, which is available from the generative answers node properties. For more information on how to use SharePoint content with a generative answers node in a topic, see [Use SharePoint content for generative answers](#).

- Queries to a SharePoint knowledge source that reference a file or document name can't be answered. For example, if a user asks, "What is the mitigation provided in *filename.pdf*?" Or, "Tell me the file name that contains information on bug 2020213?"
- Currently, access to SharePoint documents and **Enhanced search results** don't support manual authentication.
- Document libraries aren't supported as lists.
- SharePoint list queries only return data from the first 2,048 rows of data.
- Currently the **Attachments** column doesn't provide the ability to index and reason over provided attachments. However, no error occurs and the list is indexed, but no responses are provided based on the contents of the attachment.
- Up to 15 lists can be selected during each session of using the **Add knowledge** dialog. To add more lists, complete the first set of 15 and then **Add knowledge** again.
- List views can't be selected as a knowledge source.
- Glossary and Synonyms aren't supported for lists.
- SharePoint lists with more than 12 lookup columns in the default view aren't supported. Configure the view to have 12 or fewer columns.

## Copilot Studio Teams app limits

 Expand table

Feature	Limits
Agents	50 per team
Skills	100 per agent <sup>1</sup>
Topics	250 per agent in Dataverse for Teams environments <sup>2</sup> 1000 per agent in Dataverse environments (after upgrade)
Trigger phrases	200 per topic

<sup>1</sup> You must have a [Copilot Studio standalone subscription](#) to use skills in a Teams app created agent. Skills aren't available in the [Teams plan](#).

<sup>2</sup> If you're approaching limits in Dataverse for Teams environments, you can [upgrade Dataverse for Teams](#) to continue adding topics.

# Copilot Studio subscription limits

[+] Expand table

Feature	Standard subscription	Teams (select Microsoft 365 subscriptions)
Sessions	N/A	Chat sessions are unlimited per tenant every month, but a service limit of 10 sessions per user every 24 hours across all agents in a tenant are enforced. <sup>1</sup>
Power Platform requests <sup>2</sup>	250,000 every 24 hours <sup>3</sup>	6,000 every 24 hours

<sup>1</sup> Copilot Studio [sessions](#) that are included with [Microsoft 365 subscriptions](#) aren't pooled.

<sup>2</sup> These requests are consumed by Power Automate flows that are triggered by Copilot Studio agents. Learn more about [Power Automate limits](#).

<sup>3</sup> You can contact your admin to increase the Copilot Studio Power Platform requests limits by purchasing the Copilot Studio Chat Session add-on. For more information, see the [Microsoft Power Platform Licensing Guide](#).

## Copilot Studio unstructured data knowledge source limits

### Note

All unstructured data sources require user-level authentication. At runtime, users are required to sign in before accessing data sources for queries. Currently, single credential sign-in is not supported.

## OneDrive limits

- Number of files and folders
  - Total of 1,000 files, 50 folders, and 10 layers of subfolders can be included for each source.
  - Folders are represented as a single knowledge source, which contains all of their content.
- 512 MB per file
- Synchronization frequency is four to six hours (based on the time of ingestion completion)
- Supported file types: doc, docx, xls, xlsx, ppt, pptx, pdf

## (!) Note

- Documents that were protected using sensitivity labels of *confidential* or *highly confidential*, or were password protected, can't be indexed. If added, these types of documents show as ready for use but don't provide responses.
- After you add files or folders, the status might indicate "Ready" immediately after they were added, but then change to "In Progress." Once the status changes from "In Progress" back to "Ready," the content is ready for use.
- There's currently no support for glossaries or synonyms.
- At this time, Application Lifecycle Management (ALM) isn't supported for this feature. Importing agents won't result in automated knowledge source processing.

## SharePoint limits

- Number of files and folders
  - Total of 1,000 files, 50 folders, and 10 layers of subfolders can be included for each source.
  - Folders are represented as a single knowledge source, which contains all of their content.
- 512 MB per file
- Synchronization frequency is four to six hours (based on the time of ingestion completion)
- Supported file types: doc, docx, xls, xlsx, ppt, pptx, pdf

## (!) Note

- A maximum of 15 files or folders can be selected at once when creating a knowledge source.
- Documents that were protected using sensitivity labels of *confidential* or *highly confidential*, or were password protected, can't be indexed. If added, these types of documents show as ready for use but don't provide responses.
- After you add files or folders, the status might indicate "Ready" immediately after they were added, but then change to "In Progress." Once the status changes from "In Progress" back to "Ready," the content is ready for use.
- There's currently no support for glossaries or synonyms.
- At this time, Application Lifecycle Management (ALM) isn't supported for this feature. Importing agents doesn't result in automated knowledge source processing.

- Currently, document libraries aren't supported.

## Salesforce and Confluence limits

- No limit on the number of articles
- No limit on size of articles
- Synchronization frequency is four to six hours

### (!) Note

- After you add files or folders, the status might indicate "Ready" immediately after they were added, but then change to "In Progress." Once the status changes from "In Progress" back to "Ready," the content is ready for use.
- There's currently no support for glossaries or synonyms.
- At this time, Application Lifecycle Management (ALM) isn't supported for this feature. Importing agents doesn't result in automated knowledge source processing.

## ServiceNow and ZenDesk limits

- No limit on the number of articles
- No limit on size of articles
- Synchronization frequency is four to six hours

### (!) Note

After you add files or folders, the status might indicate "Ready" immediately after they were added, but then change to "In Progress." Once the status changes from "In Progress" back to "Ready," the content is ready for use.

At this time, Application Lifecycle Management (ALM) isn't supported for this feature. Importing agents doesn't result in automated knowledge source processing.

## Dataverse limits

- Maximum of two Dataverse sources per agent
- Maximum of 15 Dataverse tables per knowledge source
- Dataverse tables:
  - Standard or Activity tables types are supported

- EXCEPTION: Tables with type = Virtual when `dataproviderid` = `2ac667f5-31d6-e911-a95e-000d3a110bbd`
- All supported tables require that the maker has READ permissions
- Synonyms and glossaries are supported for Dataverse tables:
  - Maximum synonym name character length is 100
  - Maximum synonym description character length is 1000
  - Maximum glossary name character length is 100
  - Maximum glossary description character length is 1000

## Required services

The following table lists the services to which Copilot Studio connects. Ensure none of these services are blocked on your network.

Configure all [required services for Power Automate](#), in addition to Copilot Studio, to use both together.

 Expand table

Domains	Required	Protocols	Uses
*.directline.botframework.com	Yes	HTTPS	Access to Bot Framework Web Chat
*.directline.botframework.com	Yes	WS	Web socket connection to support Chat
pipe.aria.microsoft.com	No	HTTPS	Telemetry data gathered by Microsoft (strongly recommended ensuring the Microsoft team can adequately respond to client-side agent editing issues)
*.powerva.microsoft.com	Yes	HTTPS	Copilot Studio authoring experience and APIs
*.analysis.windows.net	Yes	HTTPS	Analytics reports shown in Copilot Studio (through Power BI)
bot-framework.azureedge.net	Yes	HTTPS	Bot framework resources
pa-guided.azureedge.net	No	HTTPS	In-product guidance (recommended)
cci-prod-botdesigner.azureedge.net	Yes	HTTPS	Copilot Studio authoring experience

## Maximum channel data message size limits when using Copilot Studio in Omnichannel

Copilot Studio in Omnichannel uses the ACS channel, and the same [channel data message size limit of 28 kb](#) applies.

This limit can affect scenarios such as [transferring to Omnichannel](#), where all the variables (both local to the topic and agent variables available in the conversation) are passed as context to the agent. If the size of all the variables passed exceeds the limit, the ACS channel fails with error `MessageSizeExceeded`, and the transfer is completed without the variables being passed. If you encounter scenarios in which the context isn't being passed, check for a large number of variables. Then clear them before making the transfer to avoid going over the limit.

# App registration, certificates, and configuration values for Copilot Studio

07/21/2025

This article contains app registration usage, certificate rotation, and configuration values for Copilot Studio.

## How Copilot Studio automatically creates app registration

### What is app registration?

App registration is a process that assigns a unique identifier and a secret key to an agent, allowing it to communicate with different channels and services. App registration is a mandatory component of Copilot Studio agents, as it enables the agent to identify itself to Omnichannel for Customer Service, in Microsoft Teams, and to authenticate with other channels. App registration also allows the agent to connect with skills.

### How does Copilot Studio manage app registration?

Copilot Studio simplifies the app registration process by automatically creating and managing the app registration for each agent. There's no action required from the customer to set up or configure the app registration. Copilot Studio handles the app registration behind the scenes, ensuring that the agent has the necessary access to interact with Omnichannel, selected channels, and skills. The customer can focus on designing and publishing the agent, without worrying about the technical details of app registration.

### Is app registration secure?

App registration doesn't pose a security issue to customers or their data. The app registration is only used to identify the agent and to enable secure communication with the channels and skills. The app registration doesn't grant access to any customer data or resources, nor does it expose any sensitive information about the agent. The app registration is stored and managed by Copilot Studio, which follows the highest standards of security and compliance.

## How Copilot Studio automatically creates and rotates certificates

# Overview

To let agents communicate with your data sources and services, Copilot Studio creates an application in your Microsoft Entra ID tenant, along with an associated service principal. A service principal is an identity that represents an application and allows it to access resources in your tenant. Copilot Studio controls the credentials to the service principal, which is an encrypted certificate.

## Certificate rotation

For security and compliance reasons, Copilot Studio rotates the certificates on a regular cadence. This certificate rotation means that the service principal gets a new certificate and the old one is revoked. This process is automatic and doesn't require any action from you. Certificate rotation doesn't affect the functionality of your agents or the security of your data. It's a standard practice that ensures that the certificates are always valid and up to date.

## Applicability

Certificate rotation applies to both Copilot Studio classic and production (unified authoring canvas) versions. Both versions create and rotate certificates in the same manner, and don't require any action from you.

## IP addresses

Requests from Copilot Studio to Bot Framework skills use the following IP addresses in their respective regions. If you're using an Azure service, we recommend you use the **PowerPlatformInfra** and **PowerPlatformPlex** service tags, which automatically implements the proper IP addresses. You can also manually enter IP addresses in your traffic configuration rules.

Configure all [connectors for IP addresses](#) for Power Automate in addition to Copilot Studio to use both together.

For general information on using service tags, refer to [Virtual network service tags](#).

[ ] Expand table

Region	Outbound IP
United States	52.150.154.48-52.150.154.63, 52.150.154.128-52.150.154.191, 52.250.230.0-52.250.230.31, 52.241.140.217, 52.241.138.151, 52.180.102.55, 20.49.123.192-20.49.123.255, 20.49.123.176-20.49.123.191, 13.87.160.143, 13.86.254.118, 13.83.97.188, 13.83.68.60, 20.49.123.64-

Region	Outbound IP
	20.49.123.127, 20.49.123.160-20.49.123.175, 20.49.123.128-20.49.123.159, 13.87.164.30, 13.87.164.20, 13.87.161.18, 13.86.252.116, 13.83.249.34, 13.83.17.188, 13.83.145.222, 13.64.39.170, 13.64.27.44, 52.234.104.49, 20.49.123.0-20.49.123.63, 20.49.122.224-20.49.122.255, 20.49.122.208-20.49.122.223, 13.86.250.62, 13.86.194.190, 13.86.137.20, 13.83.64.166, 13.83.56.37, 13.64.38.167, 13.64.35.24, 20.49.122.96-20.49.122.127, 20.49.122.192-20.49.122.207, 20.49.122.128-20.49.122.191, 13.91.138.172, 13.87.164.205, 13.87.163.230, 13.87.162.91, 13.87.160.212, 13.86.193.65, 13.86.185.5, 13.86.139.229, 13.86.136.222, 52.250.229.240-52.250.229.255, 20.49.122.64-20.49.122.95, 20.49.122.0-20.49.122.63, 13.88.133.160, 13.88.132.123, 13.88.129.160, 13.87.167.63, 13.87.161.235, 13.87.160.104, 13.86.254.191, 13.83.147.192, 13.83.102.38, 52.250.229.224-52.250.229.239, 52.250.229.192-52.250.229.223, 52.250.229.128-52.250.229.191, 13.88.135.72, 13.88.135.67, 13.88.135.42, 13.87.216.21, 13.87.154.100, 13.86.249.98, 13.86.192.20, 13.86.185.6, 52.250.229.64-52.250.229.127, 52.250.229.32-52.250.229.63, 52.250.229.16-52.250.229.31, 52.180.96.196, 13.88.65.204, 13.88.65.140, 13.87.218.70, 13.87.167.172, 13.87.164.186, 13.86.250.244, 13.86.185.91, 13.86.177.32, 20.49.126.64-20.49.126.127, 20.49.125.184-20.49.125.187, 20.49.125.176-20.49.125.183, 20.189.142.58, 52.250.229.0-52.250.229.15, 52.250.228.192-52.250.228.255, 52.250.228.160-52.250.228.191, 40.65.49.151, 40.65.49.140, 40.65.49.103, 13.87.216.130, 13.87.154.164, 13.87.153.50, 13.86.185.81, 13.83.70.105, 52.250.231.224-52.250.231.255, 20.49.124.64-20.49.124.79, 20.49.124.0-20.49.124.63, 52.250.231.192-52.250.231.223, 52.250.231.128-52.250.231.191, 52.250.231.112-52.250.231.127, 52.250.231.96-52.250.231.111, 52.250.231.64-52.250.231.95, 52.250.231.0-52.250.231.63, 20.49.126.0-20.49.126.63, 20.49.125.136-20.49.125.143, 20.49.125.132-20.49.125.135, 13.87.167.198, 20.49.125.192-20.49.125.255, 20.49.125.160-20.49.125.175, 20.49.125.144-20.49.125.159, 13.91.136.144, 52.250.228.40-52.250.228.47, 20.49.125.64-20.49.125.127, 20.49.125.128-20.49.125.131, 13.83.66.89, 20.49.125.0-20.49.125.63, 20.49.124.96-20.49.124.111, 20.49.124.112-20.49.124.127, 13.83.23.194, 52.250.228.36-52.250.228.39, 20.49.124.80-20.49.124.95, 20.49.124.128-20.49.124.255, 13.87.217.80, 13.86.185.35, 13.83.97.180, 13.83.151.212, 52.250.230.192-52.250.230.255, 52.250.230.160-52.250.230.191, 52.250.230.144-52.250.230.159, 20.49.127.248-20.49.127.255, 13.83.249.58, 13.83.248.248, 52.250.230.64-52.250.230.127, 52.250.230.32-52.250.230.63, 52.250.230.128-52.250.230.143, 52.246.120.190, 13.88.129.116, 13.88.128.218, 13.87.157.188, 52.250.228.48-52.250.228.63, 52.250.228.128-52.250.228.159, 20.49.121.192-20.49.121.255, 20.184.251.143, 13.87.218.169, 13.87.217.75, 13.87.217.11, 13.87.167.46, 13.87.167.174, 13.87.161.241, 13.87.154.24, 13.83.66.124, 40.64.134.192-40.64.134.255, 40.64.134.144-40.64.134.159, 52.226.41.235, 52.226.41.202, 52.226.143.0, 52.188.177.124, 52.149.238.57, 52.146.75.0-52.146.75.31, 52.146.74.192-52.146.74.255, 52.146.74.176-52.146.74.191, 20.185.8.74, 104.45.174.26, 52.224.203.192, 52.224.195.119, 52.191.232.133, 52.188.222.168, 52.146.74.64-52.146.74.127, 52.146.74.160-52.146.74.175, 52.146.74.128-52.146.74.159, 52.146.26.244, 52.146.26.218, 52.146.26.125, 40.88.16.44, 20.185.73.73, 52.255.213.211, 52.224.201.114, 52.224.17.48, 52.191.237.186, 52.190.24.61, 52.188.79.60, 52.188.77.154, 52.150.35.132, 52.146.74.0-52.146.74.63, 52.146.73.224-52.146.73.255, 52.146.73.208-52.146.73.223, 52.249.201.87, 52.224.201.121, 52.191.238.79, 52.191.238.157, 52.188.181.97, 52.151.243.194, 52.151.231.104, 52.147.222.228, 52.146.73.96-52.146.73.127, 52.146.73.192-52.146.73.207, 52.146.73.128-52.146.73.191, 52.146.50.100, 52.226.148.225, 52.224.200.26, 52.224.184.205, 52.224.17.98, 52.191.239.246, 52.191.239.208, 52.146.73.64-52.146.73.95, 52.146.73.0-52.146.73.63, 52.146.72.240-52.146.72.255, 40.71.234.201, 20.185.215.62, 104.45.175.45, 52.255.221.231, 52.190.30.145, 52.190.30.136, 52.190.27.148, 52.146.72.224-52.146.72.239, 52.146.72.192-52.146.72.223, 52.146.72.128-52.146.72.191, 40.88.48.237,

Region	Outbound IP
	40.88.18.208, 40.71.233.8, 20.185.211.94, 52.226.49.156, 52.226.49.104, 52.191.39.181, 52.188.43.247, 52.188.183.159, 52.151.246.107, 52.146.72.64-52.146.72.127, 52.146.72.32-52.146.72.63, 52.146.72.16-52.146.72.31, 20.42.24.159, 20.185.215.91, 104.45.191.89, 52.224.142.152, 52.146.79.64-52.146.79.127, 52.146.79.128-52.146.79.131, 52.146.77.168-52.146.77.175, 52.224.185.216, 52.224.184.221, 52.188.222.206, 52.188.221.237, 52.146.72.0-52.146.72.15, 52.146.24.114, 52.146.24.106, 52.142.16.162, 40.76.149.246, 20.49.111.192-20.49.111.255, 20.49.111.160-20.49.111.191, 52.146.79.0-52.146.79.63, 52.146.78.224-52.146.78.255, 52.146.78.208-52.146.78.223, 52.146.78.96-52.146.78.127, 52.146.78.192-52.146.78.207, 52.146.78.128-52.146.78.191, 52.146.78.64-52.146.78.95, 52.146.78.0-52.146.78.63, 52.146.77.176-52.146.77.191, 52.150.37.207, 52.146.77.192-52.146.77.255, 52.146.77.160-52.146.77.167, 52.146.76.188-52.146.76.191, 52.191.217.43, 52.146.77.64-52.146.77.127, 52.146.77.144-52.146.77.159, 52.146.77.128-52.146.77.143, 52.226.148.5, 52.146.77.0-52.146.77.63, 52.146.76.184-52.146.76.187, 52.146.76.176-52.146.76.183, 52.255.212.164, 52.151.238.19, 52.146.76.192-52.146.76.255, 52.146.76.160-52.146.76.175, 52.146.76.144-52.146.76.159, 52.224.150.63, 52.224.149.89, 52.224.137.160, 52.146.76.128-52.146.76.143, 52.146.76.0-52.146.76.127, 40.71.236.15, 20.49.110.84-20.49.110.87, 52.146.75.192-52.146.75.255, 52.146.75.160-52.146.75.191, 52.146.75.144-52.146.75.159, 52.152.205.65, 52.152.205.137, 20.62.129.136-20.62.129.143, 52.149.243.177, 52.149.240.75, 52.146.75.64-52.146.75.127, 52.146.75.32-52.146.75.63, 52.146.75.128-52.146.75.143, 20.42.39.188, 20.185.72.53, 52.249.204.114, 52.226.175.58, 52.224.204.110, 52.188.216.65, 52.152.194.10, 40.88.18.248, 40.76.161.168, 40.76.161.165, 40.76.161.144, 20.49.111.64-20.49.111.127, 20.49.111.48-20.49.111.63, 20.49.111.128-20.49.111.159, 13.73.254.176-13.73.254.191, 13.73.254.160-13.73.254.175, 13.73.254.128-13.73.254.159, 52.185.226.247, 40.74.183.121, 40.124.136.75, 40.124.136.2, 40.124.136.138, 40.119.1.22, 20.188.77.155, 13.73.254.96-13.73.254.127, 13.73.254.64-13.73.254.95, 13.73.254.0-13.73.254.63, 40.119.42.86, 40.119.42.85, 20.65.130.80-20.65.130.87, 52.249.63.45, 52.249.60.80, 52.249.59.157, 40.74.202.22, 40.74.201.230, 40.74.200.156, 40.74.183.82, 13.73.253.224-13.73.253.255, 13.73.253.192-13.73.253.223, 13.73.253.128-13.73.253.191
South America <sup>1</sup>	191.234.138.96-191.234.138.111, 191.234.138.64-191.234.138.95, 191.234.138.112-191.234.138.127, 191.235.127.181, 191.234.138.32-191.234.138.63, 191.234.138.0-191.234.138.31, 191.234.137.192-191.234.137.255, 191.233.31.224, 191.233.31.0, 191.233.242.180, 191.233.1.175, 191.233.0.254, 191.233.0.149, 191.234.139.176-191.234.139.183, 191.233.20.43, 191.233.18.254, 191.234.137.64-191.234.137.127, 191.234.137.160-191.234.137.191, 191.234.137.128-191.234.137.159, 191.233.31.63, 191.233.30.20, 191.233.29.72, 191.233.28.145, 191.233.27.226, 191.233.25.156, 191.233.242.177
Canada	52.228.87.160-52.228.87.191, 52.228.86.178-52.228.86.179, 20.48.192.16-20.48.192.23, 20.48.192.0-20.48.192.15, 52.228.87.192-52.228.87.255, 52.228.86.184-52.228.86.191, 52.228.86.180-52.228.86.183, 40.82.187.141, 20.39.134.93, 20.39.134.9, 20.39.134.67, 52.228.87.64-52.228.87.127, 52.228.87.48-52.228.87.63, 52.228.87.128-52.228.87.159, 52.156.24.232, 40.82.173.103, 20.48.193.168-20.48.193.175, 52.228.87.32-52.228.87.47, 52.228.87.0-52.228.87.31, 52.228.86.192-52.228.86.255, 52.139.22.227, 52.139.18.243, 52.139.17.252, 52.139.17.108, 20.63.52.175, 20.39.141.50, 20.39.140.23, 20.39.139.245, 20.151.73.141, 40.89.22.96-40.89.22.127, 40.89.22.208-40.89.22.223, 40.89.22.192-40.89.22.207, 40.89.22.80-40.89.22.83, 40.89.22.128-40.89.22.191, 40.89.20.232-40.89.20.239, 40.80.249.52, 40.80.240.214, 40.80.240.191, 40.80.240.185, 40.89.22.64-40.89.22.79,

Region	Outbound IP
	40.89.22.0-40.89.22.63, 40.89.21.224-40.89.21.255, 40.89.23.240-40.89.23.247, 40.80.241.67, 40.80.241.43, 52.155.25.157, 52.155.25.145, 52.155.25.132, 52.139.86.52, 52.139.83.184, 52.139.80.229, 40.89.21.192-40.89.21.223, 40.89.21.128-40.89.21.191, 40.89.20.240-40.89.20.255, 40.80.249.219, 40.80.249.210, 40.80.248.79
Europe	52.155.95.212, 52.155.232.15, 52.155.176.197, 52.146.128.64-52.146.128.127, 52.146.128.32-52.146.128.63, 52.146.128.16-52.146.128.31, 52.142.112.84, 52.142.112.49, 40.127.145.191, 20.54.66.198, 20.54.66.186, 20.54.66.178, 52.156.194.25, 52.155.94.139, 52.155.91.129, 52.155.222.217, 52.146.128.0-52.146.128.15, 52.142.87.183, 20.54.105.78, 20.54.105.65, 20.54.105.122, 20.50.71.192-20.50.71.255, 20.50.71.160-20.50.71.191, 52.158.24.178, 52.155.88.22, 52.155.180.156, 52.155.178.3, 52.142.121.142, 20.54.106.12, 20.54.105.72, 20.54.105.243, 20.50.71.64-20.50.71.127, 20.50.71.48-20.50.71.63, 20.50.71.128-20.50.71.159, 52.158.27.66, 52.155.236.16, 52.155.181.78, 52.155.172.184, 52.142.127.254, 20.54.4.212, 20.54.3.210, 20.54.3.143, 20.50.71.32-20.50.71.47, 20.50.71.0-20.50.71.31, 20.50.70.192-20.50.70.255, 52.155.236.8, 52.155.234.184, 52.155.173.7, 52.142.80.162, 40.127.241.36, 40.127.150.85, 40.127.148.127, 20.50.70.96-20.50.70.127, 20.50.70.80-20.50.70.95, 20.50.70.128-20.50.70.191, 52.156.204.190, 52.155.91.146, 52.155.235.153, 52.155.234.28, 52.155.234.126, 52.155.234.107, 52.155.233.8, 20.50.70.64-20.50.70.79, 20.50.70.0-20.50.70.63, 20.50.69.224-20.50.69.255, 52.155.94.157, 52.155.233.227, 52.155.232.169, 52.142.82.161, 40.127.235.247, 40.127.235.20, 40.127.234.252, 20.54.106.211, 20.50.69.192-20.50.69.223, 20.50.69.128-20.50.69.191, 20.50.69.112-20.50.69.127, 52.155.235.151, 52.155.224.132, 52.155.223.80, 52.142.86.84, 51.104.156.26, 51.104.155.233, 51.104.155.15, 51.104.152.162, 20.50.69.96-20.50.69.111, 20.50.69.64-20.50.69.95, 20.50.69.0-20.50.69.63, 52.146.130.192-52.146.130.255, 52.146.130.176-52.146.130.179, 52.142.81.115, 51.104.178.0, 51.104.177.53, 51.104.176.219, 20.50.68.136-20.50.68.143, 52.146.130.64-52.146.130.127, 52.146.130.160-52.146.130.175, 52.146.130.128-52.146.130.159, 52.146.130.0-52.146.130.63, 52.146.129.224-52.146.129.255, 52.146.129.208-52.146.129.223, 52.146.129.96-52.146.129.127, 52.146.129.192-52.146.129.207, 52.146.129.128-52.146.129.191, 52.146.129.64-52.146.129.95, 52.146.129.0-52.146.129.63, 52.146.128.240-52.146.128.255, 52.146.132.232-52.146.132.239, 51.104.150.153, 51.104.150.127, 52.158.24.140, 52.156.196.221, 52.156.193.146, 52.155.220.20, 52.146.128.224-52.146.128.239, 52.146.128.192-52.146.128.223, 52.146.128.128-52.146.128.191, 52.142.121.155, 40.127.229.37, 40.127.227.23, 40.127.224.152, 52.158.121.190, 52.158.112.171, 52.155.233.110, 52.155.162.137, 51.104.159.8, 51.104.159.21, 51.104.159.10, 20.54.37.75, 20.50.68.192-20.50.68.255, 20.50.68.160-20.50.68.191, 20.50.68.144-20.50.68.159, 51.144.190.147, 51.138.27.6, 51.138.26.201, 51.138.26.161, 51.105.206.64, 51.105.183.7, 40.74.32.24, 40.74.32.17, 40.113.182.96-40.113.182.111, 40.113.182.64-40.113.182.95, 40.113.182.0-40.113.182.63, 104.45.77.57, 52.236.152.88, 52.157.221.163, 52.142.233.149, 51.138.27.148, 51.124.83.127, 51.105.249.94, 40.113.181.192-40.113.181.255, 40.113.181.160-40.113.181.191, 40.113.181.144-40.113.181.159, 20.50.134.59, 104.45.65.67, 52.236.153.149, 52.157.221.75, 52.142.233.146, 40.74.10.193, 40.119.159.78, 40.113.181.64-40.113.181.127, 40.113.181.32-40.113.181.63, 40.113.181.128-40.113.181.143, 20.54.209.240, 20.54.209.238, 20.54.209.236, 52.157.237.175, 52.142.237.97, 52.142.233.161, 51.105.164.54, 40.74.18.24, 40.113.181.0-40.113.181.31, 40.113.180.192-40.113.180.255, 40.113.180.176-40.113.180.191, 20.54.209.175, 20.54.209.167, 20.54.209.120, 52.149.108.155, 51.144.56.64, 51.138.38.197, 51.105.165.235, 51.105.101.1, 40.113.180.64-40.113.180.127, 40.113.180.160-40.113.180.175, 40.113.180.128-40.113.180.159, 20.50.16.253, 20.50.16.235, 51.105.97.152, 51.105.250.196, 51.105.152.95, 51.105.152.8, 51.105.152.238, 40.119.159.181,

Region	Outbound IP
	40.119.159.126, 40.113.180.0-40.113.180.63, 40.113.179.224-40.113.179.255, 40.113.179.208-40.113.179.223, 51.145.186.156, 51.105.250.200, 51.105.193.78, 40.74.42.86, 40.74.42.84, 40.113.179.96-40.113.179.127, 40.113.179.192-40.113.179.207, 40.113.179.128-40.113.179.191, 20.54.184.154, 20.50.231.176, 20.50.228.248, 51.137.10.173, 51.124.71.26, 51.124.1.108, 51.105.99.79, 51.105.96.44, 51.105.250.62, 51.105.145.107, 40.113.179.64-40.113.179.95, 40.113.179.0-40.113.179.63, 40.113.178.240-40.113.178.255, 20.50.238.87, 51.105.130.21, 40.113.178.56-40.113.178.63, 40.113.178.52-40.113.178.55, 20.61.96.192-20.61.96.255, 104.45.70.91, 104.45.70.154, 104.45.67.10, 20.61.96.64-20.61.96.127, 20.61.96.48-20.61.96.63, 20.61.96.128-20.61.96.159, 40.113.183.192-40.113.183.255, 20.61.96.32-20.61.96.47, 20.61.96.0-20.61.96.31, 40.113.183.96-40.113.183.127, 40.113.183.80-40.113.183.95, 40.113.183.128-40.113.183.191, 40.113.183.64-40.113.183.79, 40.113.183.0-40.113.183.63, 40.113.182.224-40.113.182.255, 20.61.98.152-20.61.98.159, 20.50.25.187, 20.50.24.95, 52.157.222.173, 52.142.234.230, 51.138.31.195, 51.138.30.32, 51.138.28.250, 51.105.160.10, 40.74.33.38, 40.119.159.218, 40.113.182.192-40.113.182.223, 40.113.182.128-40.113.182.191, 40.113.182.112-40.113.182.127, 52.137.24.206, 51.145.189.149, 51.137.13.184, 51.137.12.195, 51.137.12.137, 51.105.195.166, 40.74.5.98, 40.113.178.224-40.113.178.239, 40.113.178.192-40.113.178.223, 40.113.178.128-40.113.178.191, 20.71.81.103
United Kingdom	51.104.31.64-51.104.31.127, 51.104.31.48-51.104.31.55, 51.104.30.172-51.104.30.175, 20.49.166.40, 20.49.166.129, 20.49.166.118, 20.49.145.249, 51.143.208.216-51.143.208.223, 51.132.215.182, 51.132.215.162, 51.145.104.29, 51.132.161.225, 51.11.25.68, 51.11.25.172, 51.11.24.198, 51.11.172.56, 51.11.172.30, 51.11.172.160, 51.104.31.32-51.104.31.47, 51.104.31.0-51.104.31.31, 51.104.30.192-51.104.30.255, 51.104.248.11, 51.137.166.64-51.137.166.127, 51.137.166.24-51.137.166.27, 51.137.166.16-51.137.166.23, 51.132.73.95, 51.132.72.50, 51.132.72.181, 51.132.68.126, 51.137.167.176-51.137.167.183, 40.81.117.23, 40.81.113.131, 52.142.168.104, 51.137.166.0-51.137.166.15, 51.137.165.192-51.137.165.255, 51.137.164.224-51.137.164.255, 51.137.137.235, 51.137.137.224, 51.137.137.159, 51.137.137.158, 51.137.136.98, 40.81.116.223, 40.81.116.143, 40.81.116.141
France	51.103.3.240, 51.103.3.127, 20.43.47.80-20.43.47.87, 51.11.235.83, 51.11.233.176, 51.11.233.119, 40.66.62.172, 40.66.61.123, 20.43.45.192-20.43.45.223, 20.43.45.128-20.43.45.191, 20.43.44.240-20.43.44.255, 52.136.184.88-52.136.184.95, 40.82.224.52, 40.82.224.49, 51.105.92.64-51.105.92.127, 51.105.92.160-51.105.92.175, 51.105.92.128-51.105.92.159, 40.82.236.9, 40.82.236.35, 40.82.224.9, 40.82.224.65, 40.82.224.60
Switzerland	51.107.241.104-51.107.241.111, 51.107.202.69, 51.107.201.45, 51.107.9.82, 51.107.8.238, 51.107.8.159, 51.107.45.66, 51.107.41.120, 51.107.241.192-51.107.241.255, 51.107.241.160-51.107.241.191, 51.107.11.80-51.107.11.83, 51.107.11.68-51.107.11.71, 51.107.96.104, 51.107.249.88-51.107.249.95, 51.107.101.181, 51.107.98.194, 51.107.96.52-51.107.96.55, 51.107.96.48-51.107.96.51, 51.107.96.36, 51.107.96.206, 51.107.249.192-51.107.249.255, 51.107.249.160-51.107.249.191, 51.107.101.56, 51.107.100.218
India	52.140.110.48-52.140.110.63, 52.140.110.32-52.140.110.47, 52.140.110.0-52.140.110.31, 20.193.137.40, 20.193.137.133, 20.192.43.64-20.192.43.71, 52.140.109.128-52.140.109.255, 52.140.108.242-52.140.108.243, 20.193.154.38, 20.193.153.43, 20.193.153.162, 13.71.52.132, 13.71.50.175, 13.71.49.81, 13.71.49.244, 13.71.48.192, 20.41.198.240-20.41.198.255, 20.41.198.224-20.41.198.239, 20.41.198.192-20.41.198.223, 52.172.112.176-52.172.112.183,

Region	Outbound IP
	20.40.4.3, 20.40.4.24, 20.44.35.138, 20.44.34.50, 20.44.34.154, 20.44.32.68, 20.44.32.253, 20.41.199.0-20.41.199.127, 20.41.197.28-20.41.197.29, 20.40.1.97, 20.40.1.191, 20.40.1.101
Japan	20.43.71.224-20.43.71.255, 20.43.70.240-20.43.70.255, 20.43.70.232-20.43.70.239, 20.43.70.206-20.43.70.207, 20.48.15.227, 20.194.144.9, 20.191.161.200-20.191.161.207, 20.44.167.207, 20.44.131.162, 20.44.130.57, 20.44.130.222, 20.43.71.192-20.43.71.223, 20.43.71.128-20.43.71.191, 20.43.70.208-20.43.70.223, 20.194.144.31, 20.194.144.27, 20.194.144.25, 40.80.63.96-40.80.63.127, 40.80.63.80-40.80.63.95, 40.80.63.128-40.80.63.143, 40.81.182.124, 40.81.182.117, 20.189.225.72-20.189.225.79, 40.81.190.97, 40.81.189.52, 40.81.188.107, 40.81.186.128, 40.81.181.214, 40.81.181.212, 40.81.181.190, 40.80.63.64-40.80.63.79, 40.80.63.0-40.80.63.63, 40.80.62.96-40.80.62.127
Asia Pacific	20.189.111.28-20.189.111.31, 20.189.111.128-20.189.111.191, 20.187.195.144-20.187.195.151, 52.139.177.8, 52.139.177.14, 52.139.176.216, 40.81.25.37, 20.189.77.126, 20.189.76.100, 20.189.74.80, 20.187.195.192-20.187.195.255, 20.187.195.160-20.187.195.191, 20.187.195.136-20.187.195.143, 52.139.179.116, 20.189.122.41, 20.187.197.24-20.187.197.31, 52.229.225.182, 52.184.84.210, 52.184.80.151, 52.139.170.52, 52.139.170.4, 52.139.156.110, 40.81.25.65, 20.189.112.175, 20.189.111.64-20.189.111.127, 20.189.111.32-20.189.111.63, 20.187.195.128-20.187.195.135, 23.98.107.24-23.98.107.27, 23.98.107.16-23.98.107.23, 23.98.107.128-23.98.107.191, 52.139.235.85, 52.139.234.140, 52.139.233.32, 52.139.232.83, 23.98.107.8-23.98.107.15, 23.98.107.64-23.98.107.127, 23.98.107.32-23.98.107.63, 20.43.161.215, 20.43.161.149, 20.43.161.116, 23.98.109.40-23.98.109.47, 20.44.198.104, 20.44.197.126, 52.148.112.216, 52.139.234.217, 40.90.184.63, 40.119.215.132, 23.98.107.0-23.98.107.7, 23.98.106.192-23.98.106.255, 23.98.106.160-23.98.106.191, 20.44.240.222, 20.43.175.237, 20.43.175.210, 20.43.175.186
Australia	20.53.40.192-20.53.40.255, 20.53.40.112-20.53.40.119, 20.53.40.108-20.53.40.111, 20.53.40.96-20.53.40.103, 20.53.40.128-20.53.40.191, 20.53.40.104-20.53.40.107, 20.53.115.98, 20.53.115.102, 20.53.115.101, 20.40.177.116, 20.53.79.20, 20.53.79.144, 20.53.74.224, 20.53.40.64-20.53.40.95, 20.53.40.0-20.53.40.63, 20.40.188.84, 20.40.187.183, 20.40.182.180, 20.37.199.240-20.37.199.255, 20.188.218.165, 20.53.77.171, 20.53.44.224-20.53.44.231, 20.53.104.132, 20.53.109.144, 20.53.109.107, 20.53.109.106, 20.53.104.7, 20.40.186.118, 20.37.199.224-20.37.199.239, 20.37.199.192-20.37.199.223, 20.37.199.128-20.37.199.191, 20.188.221.55, 20.188.219.150, 20.188.218.111, 104.46.177.0-104.46.177.63, 104.46.176.160-104.46.176.163, 104.46.176.152-104.46.176.159, 52.243.109.5, 52.243.109.4, 52.243.109.126, 40.81.60.206, 20.42.230.236-20.42.230.239, 104.46.176.192-104.46.176.255, 104.46.176.144-104.46.176.151, 52.243.113.89, 52.243.113.88, 52.243.113.102, 40.81.56.190, 20.40.165.7, 20.40.165.31, 20.40.164.215, 104.46.176.64-104.46.176.127, 104.46.176.32-104.46.176.63, 104.46.176.128-104.46.176.143, 52.243.108.25, 52.243.106.93, 104.46.179.32-104.46.179.39, 52.243.110.67, 52.243.110.181, 52.243.110.156, 20.46.108.117, 20.42.231.192-20.42.231.255, 20.42.230.240-20.42.230.255, 20.40.165.67, 20.40.164.49, 20.40.162.57, 20.40.160.0, 104.46.176.0-104.46.176.31

<sup>1</sup> Customer data in South America is replicated in United States for disaster recovery purposes, so requests from Copilot Studio to Bot Framework skills are using United States IP addresses.

# Personal data requests for Copilot Studio

Article • 11/19/2024

The European Union (EU) General Data Protection Regulation (GDPR) gives significant rights to individuals regarding their data. Refer to the Microsoft Learn [General Data Protection Regulation Summary](#) for an overview of GDPR, including terminology, an action plan, and readiness checklists to help you meet your obligations under GDPR when using Microsoft products and services.

You can learn more about GDPR and how Microsoft helps support it and our customers who are affected by it.

- The [Microsoft Trust Center](#) provides general information, compliance best practices, and documentation helpful to GDPR accountability, such as Data Protection Impact Assessments, Data Subject Requests, and data breach notification.
- The [Service Trust portal](#) provides information about how Microsoft services help support compliance with GDPR.

For more information and guidance, see the [Dynamics 365 Data Subject Requests guide](#).

## ⓘ Note

Because access to your agent is managed by your Microsoft Entra ID tenant administrator, other users with admin permissions have access to your agent content.

## Requests to rectify personal data

If a data subject asks you to rectify their personal data that resides in your organization, you and your organization must determine if it's appropriate to honor the request. Rectifying the data might include taking actions such as editing, redacting, or removing personal data.

You can use Microsoft Entra to manage Copilot Studio users' identities. Enterprise customers can manage personal data rectify requests, including limited editing features, per the nature of a given Microsoft service. As a data processor, Microsoft doesn't offer

the ability to correct system-generated logs because these logs reflect factual activities and constitute a historical record of events within Microsoft services.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Export data to respond to requests for copies of personal data in Copilot Studio

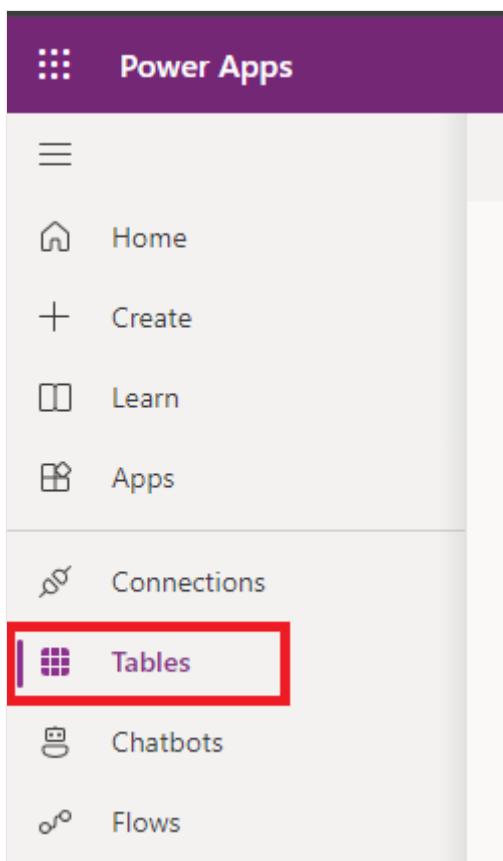
Article • 11/19/2024

This article discusses the Copilot Studio capabilities to find or export personal data for a specific user.

## Export customer data

### Agent content (as tenant admin)

1. Go to <https://make.powerapps.com> and sign in with your credentials.
2. In the side pane, select **Tables**.



3. Search for copilot tables.

The screenshot shows the 'Tables' section of the Power Apps portal. A table named 'Chatbot' is selected and highlighted with a red box. The 'Export' button in the top right corner is also highlighted with a red box.

4. Select **Copilot or Copilot subcomponent** from the table list, then select **Export data**. Your data takes a couple of minutes to be compiled for export.

The screenshot shows the details of the 'Chatbot' table. In the context menu, the 'Export' option is highlighted with a red box.

5. When your export is ready, select **Download exported data** to download the content.

The screenshot shows a success message: 'Chatbot was exported successfully. Click the link below to download the exported data.' Below the message are two links: 'Export data' and 'Download exported data'.

## Agent sessions (as agent author)

You can download session data for the last 30 days from the [sessions analytics page](#) in Copilot Studio.

You can also [export data from the ConversationTranscripts table](#).

## Export system-generated logs (as tenant admin)

Microsoft provides the ability to access, delete, and export certain customer data through Azure Portal so that tenant admins can execute [Data Subject Requests](#).

1. Go to the [User Privacy Overview in Azure Portal](#).

2. Select **Add export request**.

The screenshot shows the Microsoft Cloud Services User Privacy Experience page. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below the navigation, the page title is 'User privacy | Overview'. On the left, there's a sidebar with 'Overview' and 'Manage User Requests'. The main content area has a heading 'Welcome to Microsoft Cloud Services User Privacy Experience'. It explains that User Privacy enables account administrators to export user specific data such as system generated logs and provides links to export documentation for services requiring a manual export. To get started with your data export, it suggests clicking 'Create export data request'. A link to 'Learn more about user privacy' is provided. Two main actions are listed: 'Request to export user data' (with a blue icon) and 'Delete user in Active Directory' (with a red icon). The 'Request to export user data' section is highlighted with a red box around its icon, title, and 'Add export request' button.

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#)

# Responding to personal data delete requests from data subjects

Article • 11/19/2024

This article discusses the experiences that Copilot Studio offers when deleting personal data for a specific user.

## Delete Copilot Studio agent chat logs

All agent chat logs are deleted when the agent is deleted. For more information, see [Delete an agent](#).

## Delete Copilot Studio agent including all content

All agent content is deleted when the agent is deleted. For more information, see [Delete an agent](#).

## Delete Copilot Studio telemetry

All Copilot Studio telemetry data is automatically deleted within 29 days. No action from the user is needed.

## Delete Copilot Studio metrics

To delete metrics data, you must delete your agent. For more information, see [Delete an agent](#).

## Delete Copilot Studio system telemetry

All agent system telemetry is automatically deleted within 29 days. No action from the user is needed.

 Note

Removing personal data includes removing all personal data and system-generated logs except audit log information.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(d564ed66898667a45db26469e74bede7\_img.jpg\)](#)

# Responding to personal data discovery requests from data subjects

Article • 11/19/2024

The first step in responding to a data subject rights request is to search for and identify the customer data that's the subject of the request. Microsoft provides capabilities that assist you in responding to data subject rights requests, and thereby accessing customer data. However, it's your responsibility to ensure that personal data is located and classified appropriately.

The following table summarizes the types of Copilot Studio resources that contain personal data for a specific user.

[ ] Expand table

Resources containing personal data	Purpose
Telemetry logs	Logging that captures historical events within the service.
Copilot content and settings	Information such as content topics and agent settings.
User settings	Settings specific to the user of the agent.
Conversation logs	Used to calculate agent metrics and analytics.
Aggregated metrics	Used to determine agent behavior.

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Responsible AI FAQs for Copilot Studio

07/08/2025

An AI system includes not only the technology, but also the people who use it, the people affected by it, and the environment in which it's deployed. Microsoft's Responsible AI FAQs are intended to help you understand how AI technology works, the choices system owners and users can make that influence system performance and behavior, and the importance of thinking about the whole system, including the technology, the people, and the environment. You can use Responsible AI FAQs to better understand specific AI systems and features that Microsoft develops.

Responsible AI FAQs are part of a broader effort to put Microsoft's AI principles into practice. To find out more, see [Microsoft AI principles](#).

## AI-driven features in this app

This app contains a growing list of AI-driven features. To learn about the capabilities and impact of specific features, select a feature name from the list.

- [FAQ for generative answers](#)
- [FAQ for Copilot](#)
- [FAQ for generative orchestration](#)
- [FAQ for deep reasoning models](#)
- [FAQ for the conversational agent creation experience](#)
- [FAQ for analytics](#)
- [FAQ for templates](#)
- [FAQ for computer use](#)

# FAQ for generative answers

10/09/2025

These frequently asked questions (FAQ) describe the AI impact of the generative answers feature in Copilot Studio.

## What are generative answers?

[Generative answers](#) make your agent valuable out-of-the-box and increase the number of topics your agent is conversational in, without requiring any manual dialog tree authoring.

## What are generative answers capabilities?

When a user asks the agent a question that doesn't trigger a configured topic, the agent can optionally search for relevant content from a source of your choosing. This search includes public websites, SharePoint, or your own custom data sources, including non-text elements in uploaded files, such as images, tabular data, and diagrams. The agent uses generative AI to summarize that information into a response returned to the agent user.

## What is the intended use for generative answers?

Generative answers can be used as primary information sources in your agent, or as fallback when authored topics are unable to address a user's query.

## How were generative answers evaluated, and what metrics are used to measure performance?

The capability is continually evaluated on a collection of manually curated question-and-answer datasets, covering multiple industries. Further evaluation is performed over custom datasets for offensive and malicious prompts and responses, through both automated and dedicated manual sessions designed to expand the test suite.

## What are the limitations of generative answers?

- You must enable the [generative answers](#) option for each agent.
- See [Language support](#) for the list of supported languages by this feature and their respective stage. You might be able to use other languages, but the answers generated

might be inconsistent and the agent might not respond properly or as you expect.

- This capability might be subject to usage limits or capacity throttling.
- Responses generated by the generative answers capability aren't always perfect and can contain mistakes.

The system is designed to query knowledge from the website of your choosing and to package relevant findings into an easily consumable response. However, it's important to keep in mind some characteristics of the AI that might lead to unexpected responses:

- Copilot Studio lets you select different models as the primary response model for generative answers. Be aware of the cut-off date for data included in the model you select.

There are mitigations to prevent the model from using its training corpus as a source for answers, however it's possible for answers to include content from websites other than the one you selected.

- The system doesn't perform an accuracy check, so if the selected data source contains inaccurate information it could be shown to users of your agent. We implemented mitigations to filter out irrelevant and offensive responses, and the feature is designed not to respond when offensive language is detected. These filters and mitigations aren't foolproof.

#### Note

You should always test and review your agents before publishing them, and consider collecting feedback from your agent's users. Your admin can [turn off the ability to publish agents with generative answers](#) for your tenant in Power Platform admin center.

## What data does the capability collect? How is the data used?

If a customer explicitly consents to having their conversation with your agent logged, the capability collects user prompts, the responses returned by the system, and any feedback you provide.

We use this data to evaluate and improve the quality of the capability.

# What operational factors and settings allow for effective and responsible use of generative answers?

Generative answers work best when you designate a trusted and valid source from which content should be queried. This source might be your company website, for example [www.microsoft.com](http://www.microsoft.com). All webpages that belong to this domain would be searched for a match against the user's question.

We use the feedback you provide on your satisfaction with generated responses to improve system quality. You can provide feedback by selecting the thumbs-up or thumbs-down icons for generated responses. You can also include more feedback in free text.

# What protections are in place within Copilot Studio for responsible AI?

Generative answers include various protections to ensure admins, makers, and users enjoy a safe, compliant experience. Admins have full control over the features in their tenant and can always [turn off the ability to publish agents with generative answers](#) in your organization. Makers can add custom instructions to influence the types of responses their agents return. For more information about best practices for writing custom instructions, see [Use prompt modification to provide custom instructions to your agent](#).

Makers can also limit the [knowledge sources](#) that agents can use to answer questions. To enable agents to answer questions outside the scope of their configured knowledge sources, makers can turn on [AI General Knowledge](#) feature. To limit agents to only answer questions to the scope of their configured knowledge sources, makers should turn off this feature.

Copilot Studio also applies content moderation policies on all generative AI requests to protect admins, makers, and users against offensive or harmful content. These content moderation policies also extend to malicious attempts at jailbreaking, prompt injection, prompt exfiltration, and copyright infringement. All content is checked twice: first during user input and again when the agent is about to respond. If the system finds harmful, offensive, or malicious content, it prevents your agent from responding.

Finally, it's a best practice to communicate to users that the agent uses artificial intelligence, therefore the following default message informs users: "Just so you are aware, I sometimes use AI to answer your questions."

## Related content

- [Generative answers](#)
- [Add and manage knowledge for generative answers](#)

# FAQ for Copilot

Article • 02/27/2025

The answers to these frequently asked questions (FAQ) provide some insight into how the Copilot feature uses AI in Copilot Studio.

## What is Copilot?

The system is designed to generate a single agent topic from a human-written description.

## What are Copilot's capabilities?

It combines the natural language understanding models already in Copilot Studio with Azure OpenAI to:

- Understand what the agent maker wants to achieve by parsing their request
- Apply knowledge of how nodes within a topic work together, and how a topic should be constructed for the best effect
- Generate a series of connected nodes that together form a full topic
- Use plain language in any node that contains user-facing text that corresponds with the agent maker's request

## What is Copilot's intended use?

The **Create with Copilot** option in the Copilot Studio canvas lets agent makers simply describe what they want to achieve, and then produces a topic path that achieves that goal.

## How was Copilot evaluated? What metrics are used to measure performance?

The capability was evaluated on a collection of manually curated prompt-and-topic datasets, covering common, edge-case, offensive content, and synthetic generation.

During evaluation, topics generated from the capability were manually reviewed and scored for relevance to the input prompt, usefulness, and offensiveness.

# What are the limitations of Copilot? How can users minimize the impact of limitations when using Copilot?

The system only supports English. Inaccurate responses might be returned when users converse with the system in languages other than English.

- Your agent must be created in the US region. Other regions, and languages other than English, aren't currently supported.
- This capability might be subject to usage limits or capacity throttling.
- Topics generated by the capability aren't always perfect, and might not accurately reflect the logic you wanted to implement.
  - We have implemented mitigations to filter out irrelevant and offensive language from appearing in the configured topic, and the system is designed not to respond when offensive language is detected.
  - We also monitor output and the feedback that agent users provide to continually improve our content filters. These filters and mitigations aren't foolproof.

## ⓘ Note

You should always test and review your agents before publishing them.

# What operational factors and settings allow for effective and responsible use of Copilot?

To get the most out of Copilot, include granular instructions in your description and limit the scope of the description to a single topic. If the generated topic isn't what you had in mind or if you'd like to further workshop it, you can modify the topic, also using natural language.

We also recommend you carefully review generated topics for accuracy, either in the authoring canvas or code editor.

Feedback you provide regarding your level of satisfaction with generated topics is used to improve system quality. You can provide feedback by selecting the *thumbs up* or

*thumbs down* icon for responses generated through Copilot. You can also include more feedback in free text.

## Related content

- Create and edit topics with Copilot
- 

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# FAQ for using generative orchestration

06/12/2025

These frequently asked questions (FAQ) describe the AI impact of generative orchestration for custom agents built in Copilot Studio.

## What is generative orchestration?

[Generative orchestration](#) lets your custom agent answer user queries with relevant topics, actions, other agents, and knowledge sources, and respond to [event triggers](#).

Generative orchestration enables more natural conversations by filling in inputs using details from the conversation history. For example, if a user asks about the nearest store in Kirkland, and then asks for the weather there, orchestration infers they mean the weather *in Kirkland*.

Triggered agents can use generative orchestration to determine the best action, topic, or agent to invoke in response to outside events, enabling autonomous capabilities. For example, an agent can check for and reconcile duplicate accounts when a Dataverse table for sales accounts receives a new entry.

The system can also chain together multiple capabilities—answering queries like "I need to get store hours and find my nearest store"—and ask follow-up questions if any required details are missing or ambiguous.

## What can generative orchestration do?

Generative orchestration builds a plan to address a user query or event trigger using the name, description, inputs, and outputs of available topics, actions, agents, and knowledge.

For conversations, the system references the last 10 turns of conversation history to fill in inputs and determine the most relevant capabilities to invoke. It follows up with the user for any missing or unclear details, executes the selected plan, and then generates a response based on the output, including any custom agent instructions.

For event triggers, the orchestration uses event data, trigger-level instructions, and agent instructions to decide which topic, action, or agent to invoke and how to respond.

## What are the intended uses of generative orchestration?

You can use generative orchestration to create agents that respond to user queries and events by reasoning over topics, actions, other agents, and knowledge, using the available context and metadata. An agent can delegate parts of a task to other agents that are better suited to handle a particular domain or function, enabling modular and scalable designs.

## How is generative orchestration evaluated? What metrics are used to measure performance?

We evaluate generative orchestration across the end-to-end process: how well it identifies a suitable plan and executes it to resolve a query or respond to a trigger. Quality assessment by human reviewers covers different prompts, inputs, and configurations.

We assess the system's ability to select appropriate actions, topics, agents, or knowledge sources, how accurately it interprets user intent, and how effectively it filters out malicious or harmful content from users or makers.

## What are the limitations of generative orchestration? How can users minimize their impact?

For best results, make sure topics, actions, knowledge sources, and agents have high-quality names and descriptions. Learn about [writing effective metadata](#).

Agents invoked through orchestration—whether internal or external—must be configured correctly and capable of handling the queries or events passed to them. If the receiving agent isn't designed to process a particular task, it might return incomplete or irrelevant responses.

Currently, agents with event triggers use only the maker's credentials for authentication. Actions called by an agent in response to a trigger must also use the maker's credentials. For more information, see [data protection for agents with triggers](#).

## What operational factors and settings allow for effective and responsible use of generative orchestration?

Generative orchestration is currently supported in English only. You can test its performance using the test panel, directly in Copilot Studio. You can also add custom instructions to shape how the system selects and uses topics, actions, other agents, or knowledge.

When delegating tasks to other agents, it's important to test the interaction flows to ensure that context is passed clearly and that handoffs behave as expected.

## What are actions and how do agents with generative orchestration use them?

Actions allow an agent to perform specific operations or retrieve data to answer user queries or handle events. Your organization, Microsoft, and other partners can create actions. You can configure which actions are available, and customize their metadata to support generative orchestration.

## What data can Copilot Studio provide to actions? What permissions do Copilot Studio actions have?

When an agent calls an action, the system sends the required input values. This information can include elements of the conversation history or data from event triggers. When the system orchestrates across agents, it supports continuity by passing relevant context to the receiving agent.

## What types of issues might arise when an agent uses actions and other agents?

Errors might occur when the agent prepares inputs or generates outputs, or if it selects the wrong action, topic, or agent. To prevent such issues, ensure that metadata is accurate and unambiguous across all elements available to orchestration.

Information from triggers or user queries might include unintended or sensitive data. If such information is routed to a topic, action, or another agent, it might lead to undesired outputs. For more information, see [Troubleshooting and limitations](#).

## What protections does Copilot Studio have in place for responsible AI?

Copilot Studio includes a range of safeguards:

- Agents only use knowledge, actions, topics, and agents explicitly configured by the maker.
- Admins can restrict which actions and agents are available.

- Makers can require user confirmation before executing actions that modify data.
- Triggers and orchestration operate under the maker's authentication and are subject to configured permissions.
- Payload inspection, classifiers, and content filters detect malicious or harmful instructions in user input, trigger data, action outputs, and knowledge content.
- Entity validation with Power Fx expressions can constrain input values (for example, limit email recipients to a specific domain).
- You can configure if the full conversation history is passed, and the task to complete, when your agent delegates to other external agents.

If a potential attack is detected (for example, in trigger payloads or action outputs), execution is blocked and a *content filtered* error appears on the [activity map](#).

To maintain transparency, agents include the default message: "Just so you are aware, I sometimes use AI to answer your questions."

## How can I give feedback on generative orchestration?

You can [provide feedback for Copilot Studio](#).

## Related content

- [Orchestrate agent behavior with generative AI](#)
- [Event triggers overview](#)

# FAQ for deep reasoning

07/07/2025

These frequently asked questions (FAQ) describe the AI impact of the deep reasoning feature in Copilot Studio.

## What is deep reasoning?

[Deep reasoning models](#) are advanced large language models designed to solve complex problems. They carefully consider each question, generating a detailed internal chain of thought before providing a response back to the user.

## How can you use deep reasoning models in Copilot Studio?

Deep reasoning models in Copilot Studio offer powerful capabilities for creating sophisticated agents. Models like Azure OpenAI o3 use deep reasoning to enhance agent decision making and return more accurate responses.

When building agents, you can add instructions that define the agent's tasks and how it accomplishes them. These tasks can range from simple to highly complex, requiring thorough analysis.

Makers can apply reasoning models to specific steps in the agent's instructions, enhancing the agent's ability to perform advanced reasoning and deliver more accurate and insightful results. You can add deep reasoning models for tasks requiring scientific research, complex questions, and in depth analysis of unstructured data. These models provide insights beyond the capabilities of simpler models.

To use reasoning models, add the keyword *reason* to specific steps of agent instructions. For example: *Use reason to determine the next item in a mathematical series, such as 2, 5, 10, 17.* This triggers the reasoning model during the agent's runtime for that specific step. Copilot Studio currently uses the Azure OpenAI o3 model for its advanced reasoning capabilities.

## What are the intended uses of deep reasoning models?

Deep reasoning models are designed to handle complex tasks that require logical reasoning, problem-solving, and step-by-step analysis. For example, you can use deep reasoning models

to:

- Evaluate market trends and recommend best investment opportunities. Deep reasoning models can break down market data into smaller, manageable steps; analyze trends; and recommend the best investment opportunities. They can consider various factors such as historical data, current market conditions, and future projections to provide well informed investment recommendations.
- Analyze increased demand and recommend strategies to manage inventory. Models can analyze patterns in demand and supply, predict future inventory needs, and recommend strategies to manage inventory effectively. By considering factors like seasonal trends, market fluctuations, and supply chain dynamics, deep reasoning models can help businesses optimize their inventory management.
- Solve differential equations and provide step by step explanations. Models can solve complex mathematical problems, such as differential equations, and provide step by step explanations of the solution. By breaking down the problem into smaller steps and applying logical reasoning, deep reasoning models can offer clear and detailed solutions to mathematical challenges.

## How were deep reasoning models evaluated and what metrics are used to measure performance?

Deep reasoning models used in Copilot Studio are evaluated for groundedness, responsible AI, and accuracy. Groundedness is making sure the model only returns content that is grounded in a specific, real-world context. Responsible AI checks for protection against harms like jailbreak attacks, cross-domain prompt injection attacks, and harmful content.

To measure against these dimensions, models are tested against a diverse set of scenarios and scored along each of these dimensions. All deep reasoning models are evaluated before being released.

## What are the limitations of deep reasoning models? How can makers minimize the impact of these limitations?

- Use of reasoning models: An agent can only use deep reasoning models if deep reasoning model capabilities are [turned on in the agent's settings](#).

- Response time: Due to the time required for analysis, responses from reasoning models tend to be slower compared to other non-deep reasoning language models.

To minimize the impact of these limitations, you can:

- Ensure that deep reasoning models capabilities are turned on only for agents that need them.
- Use the keyword *reason* in agent instructions only for steps that benefit from deep reasoning models.
- Use deep reasoning models for tasks that allow for longer response times. If necessary, let users know that some agent responses might take longer.

## What operational factors and settings allow for effective and responsible use of deep reasoning models?

Deep reasoning models include various protections to ensure admins, makers, and users enjoy a safe, compliant experience:

- Only allow deep reasoning models for agents that require complex reasoning steps. This ensures that the models are applied where they can provide the most value.
- Include the keyword *reason* in the instructions to trigger the model at runtime for specific tasks, not all tasks that might not require complex reasoning.
- Thoroughly test the agent to ensure the accuracy and reliability of the output provided by the deep reasoning model. Testing also helps identify any potential issues and ensures that the model performs as expected.
- Use the [activity map](#) to review where your agent uses deep reasoning models in a session. [Expand the deep reasoning node](#) in the map to review the steps the model took and the model's output. This helps you determine if the reasoning model is delivering the intended functionality.
- Compare the outputs with and without using a deep reasoning model by updating your instructions during testing.

# FAQ for the conversational agent creation experience

08/14/2025

These frequently asked questions (FAQ) describe the AI impact of the conversational agent creation experience in Copilot Studio.

## What is the conversational agent creation experience?

The [conversational agent creation experience](#) is used to help you create a custom agent or Copilot agent. Through the conversation, it predicts the agent's name, description, and instructions, it supports adding knowledge sources, and generates suggested prompts (for Copilot agents).

## What are the capabilities of the conversational agent creation experience?

You can get started quickly with a custom agent or Copilot agent configuration through a natural language interface. The system updates the configuration based on your input during the conversation. The system automatically saves your agent configuration as you iterate between the conversational creation experience with Copilot and testing your agent.

## What is the conversational agent creation experience's intended use?

You can use this experience to begin your initial agent configuration.

## How was the conversational agent creation experience evaluated, and what metrics are used to measure performance?

We evaluated the system for accuracy of how well the predicted configuration represented the requests through the conversation to ensure quality. We also tested to ensure the system doesn't produce harmful or malicious content.

# What are the limitations of the conversational agent creation experience, and how can users minimize the impact of limitations when using it?

- The conversational agent creation experience only [supports certain languages](#). You might be able to use other languages, but the answers generated might be inconsistent or unexpected.
- This experience can only be used to configure:
  - The name
  - The description
  - Instructions that create the agent
  - A subset of the supported knowledge source types
  - Suggested prompts for Copilot agents
- This experience lets you alternate between configuring and testing your agent, but knowledge sources you specify might not be immediately available to the test chat experience.
- See the [Responsible AI FAQ for generative answers](#) for other considerations and limitations when using generative answers in agents you create with this feature.

# What operational factors and settings allow for effective and responsible use of the conversational agent creation experience?

You can use natural language to converse with the system over chat, or you can directly edit the configuration manually. If you edit manually, the system might update your agent's configuration with additional information as you continue the conversation.

# What protections are in place within Copilot Studio for responsible AI?

Generative answers include various protections to ensure admins, makers, and users enjoy a safe, compliant experience. Admins have full control over the features in their tenant and can always [turn off the ability to publish agents with generative answers](#) in your organization. Makers can add custom instructions to influence the types of responses their agents return. For more information about best practices for writing custom instructions, see [Use prompt modification to provide custom instructions to your agent](#).

Makers can also limit the [knowledge sources](#) that agents can use to answer questions. To enable agents to answer questions outside the scope of their configured knowledge sources, makers can turn on [AI General Knowledge](#) feature. To limit agents to only answer questions to the scope of their configured knowledge sources, makers should turn off this feature.

Copilot Studio also applies content moderation policies on all generative AI requests to protect admins, makers, and users against offensive or harmful content. These content moderation policies also extend to malicious attempts at jailbreaking, prompt injection, prompt exfiltration, and copyright infringement. All content is checked twice: first during user input and again when the agent is about to respond. If the system finds harmful, offensive, or malicious content, it prevents your agent from responding.

Finally, it's a best practice to communicate to users that the agent uses artificial intelligence, therefore the following default message informs users: "Just so you are aware, I sometimes use AI to answer your questions."

## Related content

- [Quickstart: Create and deploy an agent](#)

# FAQ for analytics

06/26/2025

These frequently asked questions (FAQ) describe the AI effect of analytics assistance features in Copilot Studio.

## How is generative AI used for analytics?

Copilot Studio uses AI to measure the quality of generative answer responses and to create clusters, which are used to provide insights into agent performance.

[Generative answers](#) uses knowledge sources of your choosing to generate a response. The feature also collects any feedback you provide. Analytics use large language models (LLMs) to classify the chat messages between users and agents into levels indicating the quality of generative answer responses. Copilot Studio compiles these indicators to give makers a summary of an agent's overall performance.

Clustering uses LLMs to sort users' messages into groups based on shared subjects and provide each group with a descriptive name. Copilot Studio uses the names of these clusters to provide different types of insights you can use to improve your agent.

## Quality of responses for generative answers

### What is the quality of response intended use?

Makers use quality of response analytics to discover insights into agent usage and performance, then create actions for agent improvement. Currently, analytics can be used to understand if the quality of an agent's generative answers meets the maker's expectations.

In addition to overall quality, quality of response analytics identifies areas where an agent performs poorly or fails to perform the maker's intended goals. Based on that, the maker can define areas where generative answers perform poorly and take steps to improve their quality.

In addition, when identifying poor performance, there are best practices that can help improve quality. For example, after identifying knowledge sources with poor performance, a maker can edit the knowledge source or split the knowledge source into multiple, more focused sources for increased quality.

### What data is used to create analytics for quality of response?

Quality of response analytics are calculated using a sample of [generative answer](#) responses. It requires the user query, the agent response, and the relevant knowledge sources that the generative model uses for the generative answer.

Quality of response analytics uses that information to evaluate if the generative answer quality is good, and if not, then why the quality is poor. For example, quality of response can identify incomplete, irrelevant, or not fully grounded responses.

## What are the limitations of quality of response analytics, and how can users minimize the impact of limitations?

- Quality of response analytics aren't calculated using all generative responses. Instead, analytics measures a sample of user-agent sessions. Agents below a minimum number of successful generative answers can't receive a quality of response analytical summary.
- There are cases when analytics don't evaluate an individual response accurately. On an aggregated level, it should be accurate for most cases.
- Quality of response analytics don't provide a breakdown of the specific queries that led to low quality performance. They also don't provide a breakdown of common knowledge sources or topics that were used when low quality responses occur.
- Analytics aren't calculated for answers that use [generative knowledge](#).
- Part of the metrics quality of responses analytics assesses is answer completeness. It evaluates how much the response is complete in relation to the retrieved document.

If a relevant document which contains additional information to the given question isn't retrieved, the completeness metric isn't evaluated according to this document.

## What protections are in place within Copilot Studio for responsible AI?

Users of agents don't see analytics results; they're available to agent makers and admins only.

Makers and admins can only use quality of response analytics to see the percentage of good quality responses and any predefined reasons for poor performance. Makers can only see the percentage of good quality responses and predefined reasons.

We tested analytics for quality of responses thoroughly during development to ensure good performance. However, on rare occurrences, quality of response assessments may be inaccurate.

# Clustering for insights

## What is clustering's intended use?

Clustering for insights is used to discover and create contextual insights. Currently, Copilot Studio analytics uses clustering to find user queries that an agent isn't able to address, then organize them into groups by content theme. Copilot Studio uses these groups to generate insights for addressing these unanswered queries. Copilot Studio generates clusters for this insight type daily, using all unanswered user queries from the last seven days.

## What data is used to create clusters and insights?

Clustering collects user queries from the last seven days that, using generative answers, resulted in an unanswered query.

The feature also collects any feedback makers provide through thumbs up or down reactions. We use this data to evaluate and improve the quality of clustering. More information on what data is collected is available in the [preview terms ↗](#).

## What are the limitations of clustering and insights, and how can users minimize the impact of limitations?

- Clustering and suggestion quality depends on the queries that fit the goal for each suggestion type. If there aren't many user queries that match the insight type, or the queries are too unrelated, the clusters can be too specific or too vague to result in meaningful suggestions.
- Clustering and insights aren't always perfect and can contain mistakes.

## What operational factors and settings allow for effective and responsible use of agent templates and managed agents?

Users of agents don't see clusters or insights; they're available to agent makers and admins only. To protect against harmful content, we apply content moderation policies during suggestion generation.

## Related content

- [Analytics](#)
- [FAQ for generative answers](#)

# FAQ for agent templates and managed agents

Article • 05/19/2025

These frequently asked questions (FAQ) describe the AI impact of agent templates and managed agents for custom agents in Copilot Studio.

## What are agent templates and managed agents?

Agent templates and managed agents are templates that help makers create agents for specific goals:

- Agent templates come with preconfigured components such as instructions, actions, and topics for makers to build on top of. Some examples of agent templates include Weather Forecast, Financial Insights, and IT Helpdesk.
- Managed agents are autonomous agents that respond to event triggers, and are built to complete specific, complex tasks. They may also have conversational components. Makers can receive updates for managed agents when the publisher releases new versions. An example of managed agents include Document Processor.

Agent templates and managed agents are provided as examples to help makers get started with building an agent for their business. Makers are responsible for assessing all safety and legal implications of using agent templates and managed agent templates and customizing them as appropriate for their business. It's essential to thoroughly test them to ensure they align with your specific business requirements and use case, and function as intended before publishing.

## How can you use agent templates and managed agents in Copilot Studio?

Makers can select an agent templates or managed agent on the [Create](#) page of Copilot Studio. Featured agents are also available on the [Home](#) page, in the [Explore agents](#) section.

You can use agent templates and managed agents to address specific needs for your users, without building the agent from scratch.

Organizations must assess their business's specific legal and regulatory requirements when using agent templates and managed agents, as they might not be suitable for all industries, scenarios, or use cases. Microsoft Copilot Studio, including agent templates and managed

agent templates available for use with Copilot Studio, may not be used in ways prohibited by applicable laws and regulations, terms of service, and the Microsoft Enterprise AI Services Code of Conduct.

## What are the intended uses of agent templates and managed agents?

Agent templates and managed agent templates are provided to help makers to produce agents quickly. Instead of creating a new agent from scratch, makers can choose an agent template or managed agent to start from. These come with preconfigured components and connections, like instructions, actions, topics, and triggers.

Agent templates act as starting points that makers build on top of. They come with prebuilt components which makers customize and connect themselves. Makers should have some familiarity with creating, modifying, and publishing agents before starting from an agent template.

Managed agents guide makers through installation, authenticating connections, and configuring the agent. Managed agents may receive updates from the publishers of the agents. Makers can make changes to managed agents to suit their needs.

When using agent templates and managed agents, makers are still responsible for choosing a use case suitable for their industry and business needs. When choosing a use case, makers should consider the specific legal and regulatory requirements of their business. Makers should avoid scenarios where use or misuse of their agent could result in: significant physical or psychological injury; consequential impact on life opportunities or legal status; or agent actions that are irreversible or highly consequential in high stakes domains or industries (such as finance, insurance, healthcare, legal, and housing).

## How were agent templates and managed agents evaluated and what metrics are used to measure performance?

Agent templates and managed agents go through testing during the development process to ensure the offered components work as expected. Managed agents that are already installed can be updated to newer versions as the publisher updates them with improved capabilities and features. Makers are responsible for thoroughly testing and measuring performance of the agents they build using agent templates and managed agent templates before they publish or deploy them.

# What are the limitations of agent templates and managed agents? How can users minimize the impact of these limitations when using the system?

Agent templates and managed agents always require at least some configuration and customization, such as authenticating connections. You can ensure your agent works as intended by:

- Following the best practices for [generative orchestration](#) to protect your agent templates and managed agents that use generative orchestration, generative actions, and event triggers.
- Carefully considering and choosing a use case that suits your business requirements while bearing in mind the intended use and limitations of each agent templates and managed agent.
- Authenticating all the required connections using the authorization appropriate for the use case.
- Only connecting your agent to trusted data sources that are safe for the agent to access.
- Thoroughly testing your agent built with an agent template and managed agent before publishing. If you make any customizations to agent templates or managed agents, or you update a managed agent, test the changes before updating the published, production version of your agent. Testing also helps identify any potential issues and ensures that the agent performs as expected.
- Using the [activity map](#) to review your agent's activity during testing and in production. This oversight helps you determine if the agent is delivering the intended functionality.

Finally, it's a best practice to communicate to users that the agent uses artificial intelligence, therefore the following default message informs users: "Just so you're aware, I sometimes use AI to answer your questions." If an action produces AI-generated content, such as an email, you can instruct your agent or configure this action to include a message to inform users that the content was produced by an AI.

# What operational factors and settings allow for effective and responsible use of agent templates and managed agents?

Agent templates and managed agents have the same risks and mitigations as regular agents. Follow the best practices for [generative orchestration](#) to protect your agents.

Read and understand the requirements of an agent template or managed agent before installing and during configuration. Some managed agents walk you through configuration during installation or might require the creation of components or configurations in other services. You can find these requirements in the agent's catalog page and in the agent's documentation.

## How can I give feedback on agent templates and managed agents?

You can provide feedback [directly in Copilot Studio](#).

## Related content

- [Create a custom agent from a prebuilt agent](#)
- [Install managed agents from Microsoft](#)
- [Orchestrate agent behavior with generative AI](#)
- [Event triggers overview](#)

# FAQ for AI Approvals

08/02/2025

These frequently asked questions explain the capabilities, usage, and safeguards of the AI-powered approval stages (AI approvals) in Microsoft Copilot Studio. AI approvals allow an agent flow to automatically approve or reject requests based on predefined criteria, while ensuring humans remain in control for important decisions. Here are some common questions and answers about this feature.

## What are AI approvals?

AI approvals are intelligent, automated decision steps in approval workflows. AI approvals use AI (Azure OpenAI models or models that you can bring from Azure AI Foundry) to evaluate approval requests against your business rules and return an "Approved" or "Rejected" decision with a rationale.

## What are AI approvals capabilities?

Unlike basic rule-based automation, AI approvals can interpret unstructured data and complex documents (like PDFs or images attached to a request) and apply nuanced logic to make a decision. For example, an AI approval could read a written justification, check for policy keywords, and then decide.

AI approval stages can also be combined with human stages, so that while AI handles routine decisions, people still oversee, and finalize any critical or exceptional cases. In summary, AI approvals automate the repetitive yes/no decisions in a process, speeding up workflows without removing human oversight where it matters.

## What is the intended use of AI approvals?

AI approvals are designed for common business scenarios with well-defined criteria, streamlining everyday workflows by automating routine decisions. Typical use cases include:

**Expense reimbursement approvals:** Automatically approve claims under certain amounts with valid receipts, letting managers focus only on exceptions.

**Purchase order (PO) approvals:** Evaluate requests against budget limits and vendor lists, auto-approving standard POs within policy.

**Travel request approvals:** Auto-approve compliant travel requests while rejecting requests with policy violations.

**Vendor onboarding:** Accept or reject applications by checking qualifications and compliance requirements against predefined criteria.

**Invoice processing approvals:** Validate invoices by matching amounts to purchase orders and confirming required documentation is present.

**Document review approvals:** Confirm contracts or policies include required elements and meet formatting standards before advancing to next steps.

**Time-off request approvals:** Approve leave requests when employees have sufficient balance and no scheduling conflicts exist.

AI approvals were designed for routine, well-defined decisions. However, there are scenarios where the system might not perform reliably or responsibly. We encourage customers to use AI approvals in their innovative solutions or applications but consider the following principles when choosing a use case:

- **High-stakes or life-altering decisions:** The system wasn't designed to handle decisions that affect health, safety, finances, or legal status. Examples include insurance claims, medical authorizations, loan approvals, or immigration determinations, which require human judgment and accountability.
- **Legal or disciplinary matters:** Use cases involving legal liability, compliance interpretation, or employee discipline might exceed the system's intended scope. While AI can summarize inputs, final decisions in these areas should remain with humans.
- **Subjective or ambiguous criteria:** The system might struggle with decisions that rely on taste, discretion, or complex trade-offs—such as evaluating creative work or assessing quality—where standards are not easily codified.
- **Sensitive or ethically complex scenarios:** AI approvals weren't designed for decisions involving personal attributes, potential discrimination, or generation of restricted content. These uses raise responsible AI concerns and might require additional safeguards.
- **Regulated industries and compliance-sensitive workflows:** In domains like healthcare, finance, or aviation, regulatory requirements might necessitate human oversight even for routine decisions. The system wasn't evaluated for compliance in these contexts.
- **Foreseeable but unintended uses:** As adoption grows, users might attempt to apply AI approvals to areas like performance reviews, hiring decisions, or customer eligibility assessments. These uses were not part of the system's design or impact assessment and might introduce risks if not carefully managed.

### **Important**

**Legal and regulatory considerations.** Organizations need to evaluate potential specific legal and regulatory obligations when using any AI services and solutions. Services and solutions might not be appropriate for use in every industry or scenario. Restrictions might vary based on regional or local regulatory requirements. Additionally, AI services or solutions aren't designed for and may not be used in ways prohibited in applicable terms of service and relevant codes of conduct.

## What are the technical limitations of AI approvals, and how can users minimize the impact of limitations?

While AI approvals are a powerful capability, we urge users to be mindful of their limitations:

**AI approvals rely on provided rules:** AI strictly follows your instructions and data. If your prompt is unclear or incomplete, the AI approval might make wrong decisions or fail. Define criteria explicitly; saying "approve if reasonable" without defining "reasonable" leads to misinterpretation.

**Possibility of errors:** AI approvals can make mistakes due to ambiguous inputs, complex edge cases, or misreading poorly scanned documents. Outputs aren't always 100% accurate, so oversight is essential for borderline cases.

**Lack of human intuition:** AI approvals don't understand context beyond what they're told and can't ask clarifying questions or use gut feelings. The AI approval might miss nuances a human would catch, like spotting suspicious expenses that "look too high for that trip."

**No learning from experience:** AI approvals don't adapt from each approval—they don't change behavior unless you update the prompt. New scenarios not covered by existing rules require ongoing maintenance as policies evolve.

**Data quality dependency:** AI approval decisions are only as good as the input data. Poor quality files, incomplete documents, or illegible scans can cause incorrect decisions or system failures.

**Integration and performance constraints:** Complex approval criteria or decisions requiring real-time data from multiple systems might reduce accuracy and increase processing time.

**Requires responsible configuration:** Users must configure AI approvals ethically, with proper human fail-safes and bias-free rules. Always ensure instructions align with company policy and

ethical guidelines.

**No access to real-time information:** AI approvals can only work with data explicitly provided as input. They can't check current affairs, news, or events unless that information is fed into the approval process.

To reduce risks and improve reliability when using AI approvals:

- **Include human oversight:** Route critical or ambiguous cases to manual review stages to ensure accountability and judgment.
- **Test with diverse examples:** Use historical data and edge cases to validate system behavior before deployment.
- **Refine prompts regularly:** Update instructions as policies evolve or new scenarios emerge to maintain relevance and accuracy.
- **Avoid vague criteria:** Ensure prompts are explicit and well-defined—avoid terms like “reasonable” without clear context.
- **Monitor decisions:** Use tools like AI Builder Activity to track approval rates and identify patterns or errors.
- **Train users:** Educate staff on interpreting AI rationales and override procedures to build trust and transparency.

**Remember:** AI can confidently execute flawed instructions, so clear, correct guidance is essential.

## What operational factors and settings allow for effective and responsible use of the agent approvals experience?

To use AI approvals effectively and safely, consider these operational best practices:

- **Set low temperature for consistency:** Use low temperature settings (near 0) to ensure the AI makes deterministic, predictable decisions rather than varying responses to identical inputs. Default Copilot Studio settings are already optimized for reliability.
- **Choose the right model:** GPT-4.1 is typically ideal for most approval scenarios. Advanced reasoning models (like O3) might handle complex logic better but are slower. Microsoft's provided models are pre-integrated and tested, though you can bring your own fine-tuned models from Azure AI Foundry if you have specific requirements or custom needs.

- **Implement human oversight:** Configure human or manual stages that can be routed to for critical decisions. Human and manual stages ensure that humans are always in control.
- **Test thoroughly in sandbox:** Run extensive tests with historical data and sample requests before going live. Deliberately test edge cases—missing fields, conflicting rules, unusual scenarios. Verify the end-to-end workflow triggers correctly.
- **Monitor decisions:** All decisions are logged in the *AI Builder Activity* section in Power Automate. Use that data to track metrics like approval rates and assess the correctness of the AI approval decisions.
- **Regularly update criteria:** Treat AI prompts as living documents. Update instructions as policies change or new scenarios emerge. Incorporate feedback from managers about AI being too strict or lenient in specific areas.
- **Provide transparency and training:** Train relevant staff on interpreting AI rationales and override procedures. Inform end-users that requests might be initially evaluated by AI. Clear expectations prevent confusion and build trust.

By tuning AI settings for consistency, embedding human oversight, and actively managing the process, you ensure AI approvals stay effective and on-track. Think of it as partnership: AI handles volume and speed, humans handle guidance and exceptions.

## What protections are in place within Copilot Studio for responsible AI?

### What kind of content moderation is implemented?

The GPT models are trained on internet data, which is great for building a general world model. At the same time, it can inherit toxic, harmful, and biased content from the same sources. The models are trained to behave safely and not produce harmful content, but sometimes it can generate toxic output. AI approvals use [Azure AI Content Safety](#) service to put state of the art content moderation capabilities within the AI prompts. This moderation includes services to analyze the generated output with multi-severity text scanners and safety against prompt injection attacks. The output is also scanned for regurgitation of protected material.

### What language model are supported, where are they hosted, and how can I access them?

AI approvals support GPT 4.1 mini, GPT 4.0, GPT 4.1, and o3 models, which are hosted on Azure OpenAI Service. You can access these models through the prompts across Power Platform, in

your applications, flows, and agents.

To learn more, see [What's new in Azure OpenAI Service?](#)

## Is my data used to train or improve the large language models?

AI approvals run on Azure OpenAI Service hosted by Microsoft. Customer data isn't used to train or improve any of the Azure OpenAI Service foundation models. Microsoft doesn't share your customer data with a third party unless you have granted permission to do so. Neither customer prompts (input) with its grounding data nor the model responses (output) are used to train or improve Azure OpenAI Service foundation models.

## How are images of people processed?

AI approvals aren't intended for use identifying individuals based on facial features or biometric data. When you submit images containing people in AI approvals, the system automatically applies a face blurring feature before analyzing the images to protect individual privacy. This blurring step helps address privacy concerns by preventing identification based on facial features. With blurring, no facial recognition or facial template matching is involved. Instead, any identification of well-known individuals relies on contextual cues, like uniforms or unique settings, not on their faces. This privacy measure shouldn't impact the quality of the results you receive. Face blurring might be occasionally referenced in the system's responses.

Learn more in [Face blurring](#).

## What are some potential harms when using images or documents in prompts?

AI approvals mitigate most of the risks involved when using images or documents in prompts, but some risks still require extra care from the prompt creator:

- Images or documents can contain harmful text or visuals that might impact your downstream processes.
- Images or documents can include special and possibly hidden instructions that might compromise or override the initial prompt.
- Images or documents can contain instructions that could lead to the generation of content that is under intellectual property (IP).
- Prompts can produce biased comments on images or documents.

- Extracting information from low-quality images or documents can lead to hallucination.

## What kinds of issues might arise when using AI approvals, and how can I handle them?

When using AI approvals, you might encounter issues like analysis failures (when AI can't confidently apply rules), wrong approval decisions (false positives/negatives), inconsistent outcomes on similar requests, or processing delays with complex cases. To handle these challenges effectively, ensure your workflow routes requests to human stages.

Implement consistent and rigorous testing throughout development and deployment to identify potential failure points early. Use low temperature settings for predictable outcomes, and continuously refine your prompts based on observed errors. Regular monitoring and iterative improvements will help maintain system reliability and accuracy over time.

# FAQ for the computer use tool

08/06/2025

## What is the computer use tool

[Computer use](#) is a tool you can add to your Copilot Studio agents designed to automate tasks on websites and applications. Users provide task descriptions in natural language, and the tool automates mouse clicks and keyboard actions accordingly. For instance, you could instruct the tool to open a PDF and use its contents to fill out a web form. The tool visually interprets the screen and executes the necessary clicks and keystrokes, emulating human interactions.

### Warning

Computer use is a powerful tool that automates interactions across websites and desktop applications, but it also comes with important security considerations. In some cases, the AI might take unintended actions due to ambiguous instructions or unexpected content on screen. These actions can affect the security of your device, data, or connected accounts, including access to personal, financial, or enterprise systems.

## What can the computer use tool do?

The computer use tool processes screen pixel data to understand on-screen content, and it performs tasks using virtual mouse clicks and keyboard inputs. The tool can manage multi-step workflows, handle unexpected scenarios, and adapt dynamically to screen changes, enabling it to perform various digital tasks such as navigating websites and filling forms without needing specialized APIs.

The computer use tool follows an iterative loop based on three steps:

- **Perception:** It captures screenshots to visually understand the current state of the computer screen.
- **Reasoning:** It evaluates the current state, previous actions, and screenshots through a chain-of-thought approach, which allows it to adapt its actions effectively.
- **Action:** It performs actions like clicking, typing, or scrolling until the task is complete or requires further user input.

## What are the computer use tool's intended uses?

The computer use tool is intended to automate user interactions with websites and applications through a virtual mouse and keyboard, especially useful in scenarios where APIs aren't available.

## How was the computer use tool evaluated? What metrics are used to measure performance?

The computer use tool was rigorously evaluated using real-world scenarios throughout its design, development, and deployment phases. Evaluation methods included research, impact studies, and testing across different scenarios to measure accuracy, user trust, and usefulness. Both qualitative and quantitative metrics are regularly monitored to maintain high performance and a positive user experience. The tool's availability is also ensured by adhering to established SLAs.

## What are the limitations of the computer use tool? How can users minimize the impact of the computer use tool's limitations when using the system?

Computer use is a novel technology with known security risks and functional limitations. We're empowering you to experiment with it. This early-stage capability enables automation of certain computer-based tasks, though its performance varies widely depending on the use case. To use it effectively and responsibly, understand its current limitations:

### Known limitations

- **Success rate varies by task:** The tool performs best on web-based tasks (about 80% success) but drops significantly on desktop apps (about 35% success).
- **Inconsistent performance:** The same task might yield different outcomes depending on visual or timing changes.
- **Difficulties with certain UI controls:** Computer use can have difficulties interacting with non-standard or dynamic interface elements, such as dropdowns, date pickers, or custom widgets.
- **Loops and stuck states:** In some cases, the agent enters a loop or gets stuck when the screen doesn't match expectations or changes unpredictably.
- **Challenges with complex tasks:** While effective for simple actions, the tool might struggle with detailed graphical interfaces or tasks involving intricate text manipulation.
- **Not for sensitive or high-risk use cases:** The tool isn't intended for:

- Recommending or scoring in sensitive domains, such as hiring, healthcare, or finance
- Making financial transactions
- Harmful actions, such as spam or misinformation
- Sharing data outside your organization without proper authorization

## How to minimize limitations

To improve reliability, users should provide clear, detailed instructions when configuring tasks. This helps reduce errors and increases task success rates.

## What operational factors and settings allow for effective and responsible use of the computer use tool?

The computer use tool performs best when clear and detailed user instructions are provided. Users should give precise hints about website or application interactions to enhance reliability and accuracy during task execution.

## How do I provide feedback on the computer use tool?

Have feedback on the computer use tool? Email [computeruse-feedback@microsoft.com](mailto:computeruse-feedback@microsoft.com).

# FAQ for code interpreter

08/27/2025

These frequently asked questions explain the capabilities, usage, and safeguards of code interpreter in Copilot Studio. Code interpreter lets an agent generate and run Python code to carry out a request specified in a prompt. Here are some common questions and answers about this feature.

## What is code interpreter in Copilot Studio?

Code interpreter is a Python execution engine integrated within Copilot Studio and AI Builder Prompts. It allows users to run code in a secure, sandboxed environment to perform data analysis, process Word, Excel, PowerPoint and PDF files, and generate visualizations. Inputs include data files and prompt instructions; outputs can be modified files, reports, charts, and text summaries.

## What can code interpreter in Copilot Studio do?

Code interpreter uses its Python execution capability to support tasks such as creating and updating Excel workbooks, copying formatting, analyzing datasets, performing statistical computations, and generating charts. It enables developers and business analysts to automate complex workflows within Copilot Studio agents and AI Builder Prompts.

## What are the intended uses of code interpreter?

Primary use cases include data auditing, report generation, file transformation, and visualization. Enterprises use it to automate financial audits, synthetic data generation, PDF form creation, and interactive data exploration without leaving Power Platform.

## How was code interpreter evaluated? What metrics are used to measure performance?

We perform functionality tests to validate correct execution of Python code, accuracy checks for data processing tasks, and performance benchmarks for execution latency. Metrics include success rate of file operations, error rate for code execution, and average execution time. Evaluation spans internal pilots and customer feedback loops via previews.

# What are the limitations? How can users minimize their impact?

Limitations include:

- Lack of support for reading text from image-based PDF content
- Session timeouts for long-running tasks
- Restrictions on external network access
- Calling prompts as a tool from within an agent topic directly is not supported
- Cannot read files with data protections

Text-based PDFs can be read. To minimize impact of session timeouts, split large datasets into smaller files, use optimized code for performance, and follow provided best practices in prompt design. To call prompts from an agent topic, you can call the prompt from within a Power Automate flow, and then call the flow within an agent topic.

## What operational factors and settings allow for effective and responsible use?

Effective use requires enabling the feature at the tenant level and in individual agents or prompts. Administrators can configure environmental access and monitor execution logs. Users can pick the model to use in prompts to tune behavior.

## How do I provide feedback?

End users and admins can provide feedback through the Power Platform Admin Center's support channels, or via the feedback option in Copilot Studio. Customer success managers and technical support teams also collect and triage feedback for product improvements.

## Does the code interpreter execute unverified or unsafe code?

No. Code execution is sandboxed and isolated, with strict constraints to prevent unsafe operations such as network access, system-level commands, or unauthorized file operations. Each execution is scoped to the specific prompt or agent session and follows Microsoft's secure AI design principles.

# Can the code interpreter access customer or tenant data outside of what is provided in the prompt?

No. The code interpreter only accesses the files and inputs explicitly provided during the session. It doesn't have access to broader tenant data, user context, or external systems unless configured explicitly through integration features.

# Where is the code executed, and how is the environment secured?

Code is executed in a Microsoft-hosted environment that adheres to Microsoft's enterprise-grade compliance, including SOC 2, ISO 27001, and GDPR requirements. The compute is ephemeral, created during a session and disposed of afterward to prevent data persistence.

# Can enterprises monitor or govern how the code interpreter is used?

Yes, tenant admins must enable the capability in the Power Platform Admin Center. This setting is off by default. Organizations can control environment-level access, audit prompt execution history as explained in [Monitor AI Builder models and prompts activity](#).

# How does Microsoft ensure responsible use of AI in prompts and agents?

All AI interactions—including interactions involving the code interpreter—adhere to Microsoft's Responsible AI principles, such as transparency, accountability, and reliability. Prompts are logged, monitored, and can be reviewed for traceability and compliance.

# Are models trained on customer data or executed code?

No. Microsoft doesn't use customer data or executed code from code interpreter sessions to train models. All data is processed in compliance with Microsoft's data handling commitments under the Microsoft Products and Services [Data Protection Addendum \(DPA\)](#).

# **Can users inadvertently exfiltrate sensitive information through prompts?**

Enterprise customers are encouraged to implement guardrails such as input validation, prompt design guidance, and usage policies. Copilot Studio also integrates with DLP policies and compliance tooling within Power Platform to reduce risk of data leakage.

## **What safeguards are in place to prevent misuse of code interpreter by users?**

Admins can restrict access at the environment level and review prompt activity. Further, AI-generated code execution is sandboxed with no access to internal systems or networks, and prompt capabilities are limited to permitted data/file boundaries.

## **How can customers ensure explainability and traceability of code interpreter actions?**

Prompt inputs, generated code, outputs (for example, files), and prompt execution traces are all available for audit and reviews. This supports regulatory compliance and transparency in AI-assisted workflows.

## **Does code interpreter support sovereign cloud or data residency controls?**

Currently, code interpreter is available in public clouds. Support for sovereign clouds (for example, GCC, GCC High) is not yet available. Data residency commitments align with Power Platform regional availability.