

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](#)

Extend agents with actions

[Create and configure agent connectors, actions, and extensions](#)
[Call a Power Platform connector](#)

Best practices guidance

[Topics best practices](#)
[Measure and improve agent engagement](#)

[Use Microsoft Bot Framework Skills](#)

[Use actions with custom agents](#)

[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 agent performance and usage](#)

[Analyze topic usage](#)

Related content

Copilot learning hub

Discover tools, guidance, and support for adopting, securing, and managing custom and Microsoft Copilot experience...

Power Platform

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

Power Apps

Quickly build low-code apps that modernize processes and solve tough business challenges in your organizati...

Power Automate

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

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.

See also

Dynamics 365

The next generation of CRM and ERP applications

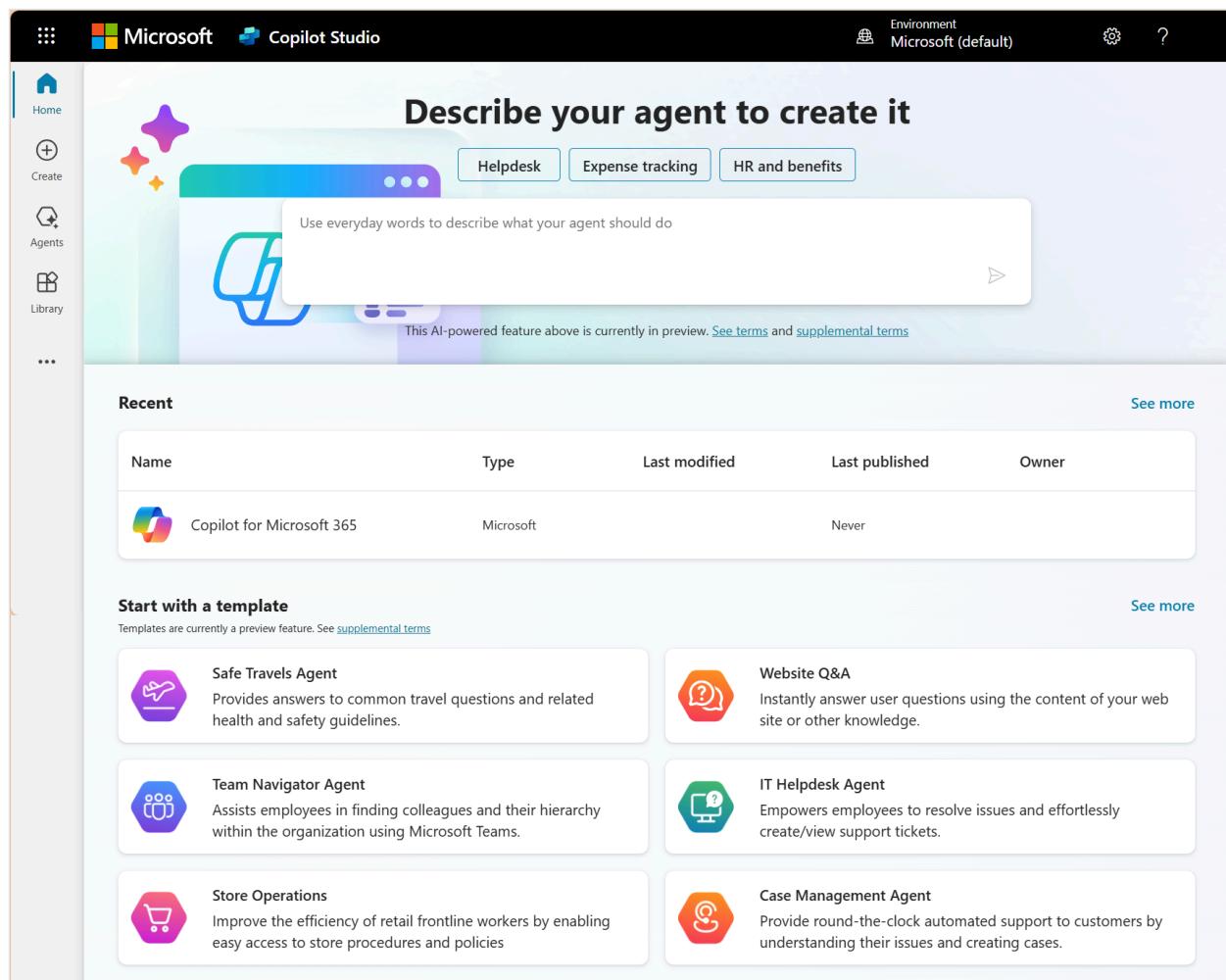
Microsoft Azure

Ever-expanding set of cloud computing services to help your organization meet its business challenges

Copilot Studio overview

Article • 11/19/2024

Copilot Studio is a graphical, low-code tool for both creating an agent—including building automation with Power Automate—and extending a Microsoft 365 Copilot with your own enterprise data and scenarios.



The screenshot shows the Microsoft Copilot Studio interface. At the top, there's a navigation bar with icons for Home, Create, Agents, Library, and three dots. The Microsoft logo and 'Copilot Studio' are also present. On the right, it says 'Environment Microsoft (default)' with a gear icon and a question mark. The main area has a title 'Describe your agent to create it' with a sub-section for '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'. A note at the bottom of this section states: 'This AI-powered feature above is currently in preview. See terms and supplemental terms.' A sidebar on the left lists 'Recent' items, showing one entry: 'Copilot for Microsoft 365' by Microsoft, last modified 'Never'. Below this is a section titled 'Start with a template' with six cards: 'Safe Travels Agent' (purple icon), 'Website Q&A' (orange icon), 'Team Navigator Agent' (blue icon), 'IT Helpdesk Agent' (green icon), 'Store Operations' (pink icon), and 'Case Management Agent' (red icon). Each card has a brief description and a 'See more' link.

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 triggers 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.

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 for interacting with customers.- You're familiar with agent services 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.	Explore the Copilot Studio demo ↗
Teams app	<ul style="list-style-type: none">- An employee or member of an organization who wants to use agents to answer common employee questions.- You want to use advanced	Open or add the Copilot Studio app in Teams ↗

Copilot Studio version	Use cases	More information
	<p>concepts, such as entities and variables, and have an internally available agent in Teams.</p> <ul style="list-style-type: none"> - You want to create and distribute an agent in the shortest time possible. 	

Plan your agent

Consider the following when planning your agent.

Extend Microsoft 365 Copilot with a Copilot 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.

The screenshot shows the Microsoft Copilot Studio interface with the following details:

- Header:** Weather Agent (with a gear icon), Overview, Knowledge, Topics, Actions, Activity, Analytics, Channel, Publish, Settings, ...
- Agent Status:** Your agent is ready! Here's what's next:
 - Add actions so your agent can do things for you
 - Build topics to focus and guide how your agent answers
 - Publish your agent so others can use it
- Agent Details:**
 - Name:** Weather Agent (with a sun and cloud icon)
 - Description:** Your go-to assistant for getting weather forecast.
 - Instructions:** You should remain friendly and polite at all times. Do not answer questions that are not related to the weather.
- Test your agent:**
 - Content:** 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 [See terms](#).
 - Input Field:** What can I ask?
 - Output Placeholder:** Ask a question or describe what you need
 - Feedback:** 0/2000 characters
 - Footer Note:** Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

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, and 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 and looking to create a fully customized end-to-end solution for your web or mobile app 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

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.](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

AI-based agent authoring overview

Article • 11/19/2024

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 Studio](#).
- **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 \(preview\)](#).

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's 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.

[+ Add knowledge](#)

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 additional 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.
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

AI features for Teams and Classic chatbots

Article • 11/19/2024

With copilots in the Copilot Studio app in Teams or **Classic** chatbots in the Copilot Studio web app, you can enable extra features that improve the core AI.

Agents include next-generation AI features that supersede or replace the features listed in this article.

You can use these AI features for backwards compatibility with older chatbots and support some scenarios. In most cases, you should [create or convert your agents with the unified authoring canvas](#) 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 [using the unified authoring canvas](#) in the Copilot Studio web app.

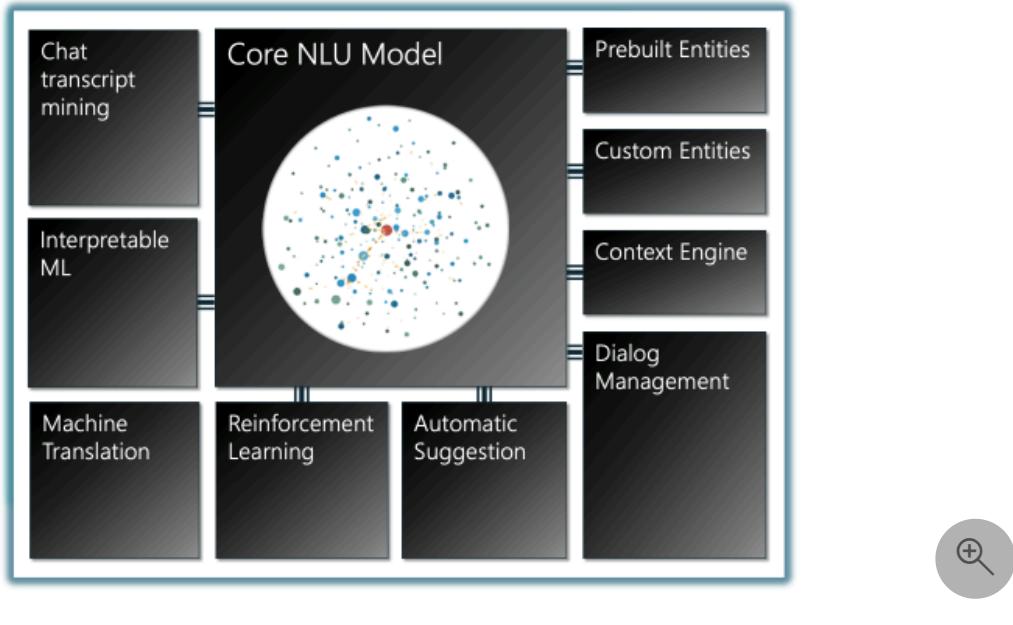
Prerequisites

You can use the AI features 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 (background)

A suite of AI capabilities. On a single service.



Copilot Studio hosts multiple AI models and AI capabilities on a single service. The core feature is a transformer-based natural language understanding (NLU) model.

Traditionally, intent triggering—how an AI model determines the intent of a question by using NLU—is formalized as a multi-class classification problem. The model is highly associated with known categories. But any change in these categories means you need to build a new AI model.

Copilot Studio, however, employs a language understanding model that uses an example-based approach, powered by a deep neural model. This type of large-scale model only needs to be trained once, with large amounts of data, using AI supercomputing. The AI is then used with a few examples without further training.

This model is part of the [AI at Scale](#) initiative by Microsoft. The way AI is developed and used is changing. In Copilot Studio, this model allows for an intuitive way for chatbot makers to create content confidently, without having to involve AI experts.

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 copilot's need to ask clarifying questions before triggering a topic.

Tip

Topic overlap detection is in general availability and supports [all languages supported in Copilot Studio](#).

After you enable advanced AI capabilities, you can view a list of overlapped topics. In the navigation menu, select **Analytics**, then select the **Topic triggering** tab.

 Microsoft Power Virtual Agents | Support Bot

≡ Chatbots Overview Topics Entities Analytics Publish Settings Details Channels Agent transfers Security Skills AI capabilities

Analytics

Updated about one hour ago

Summary Topic triggering Customer Satisfaction Sessions Billing



Overlapping topics

Trigger phrases that are too similar blur the boundaries between topics. To improve triggering, review overlapping topics and trigger phrases. [Learn more](#)

🕒 Updated 4 minutes ago

Similarity ⓘ ↓	Topics (5)	Overlaps
94	I stopped sharing the subscription with	1
94	I deactivated the subscription sharing f	1
91	Languages supported in Microsoft 365	1
91	Use Microsoft 365 in other languages?	1
90	Where can I get technical support?	1

The list shows each overlapping topic along with a similarity score. This score represents the overall overlapped status for a topic and the number of topics that overlap with the listed one. The AI determines the similarity score as it evaluates how semantically similar the overlapping trigger phrases are to each other. 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 its similarity score, topic name, or number of trigger phrase overlaps.

If you select an item in the list, the **Topic overlap details** pane opens.

The screenshot shows a 'Topic overlap details' pane. At the top, it says 'Languages supported in Microsoft 365'. Below that is a link to 'Rewrite or remove trigger phrases to distinguish between topics. [Learn more](#)'. The main section is titled 'Overlapping trigger phrases (1)'. It lists two trigger phrases: 'For which languages is Microsoft 365 available?' and 'Can I use Microsoft 365 in languages other than the one I originally purc...'. Each phrase has a trash can icon to its right. At the bottom are 'Save' and 'Cancel' buttons.

In this example, there's one trigger phrase in the "Languages supported in Microsoft 365" topic (*For which languages is Microsoft 365 available?*) that semantically overlaps with a trigger phrase in the "Use Microsoft 365 in other languages?" topic (*Can I use Microsoft 365 in languages other than the one I originally purchased?*). Here, the AI determines that both trigger phrases are semantically similar. The phrases have similar sentence structure, words, and grammar.

Using semantically similar trigger phrases for two different topics can lead to confusion. The copilot might not know which topic to open and asks follow-up questions to the copilot user.

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

On the **Topic overlap details** pane, you can select the link to go directly to a topic. You can also make changes to the trigger phrase or delete it. Select **Save** to apply any changes.

After you save your changes, the overlapping status automatically refreshes. You can manually refresh the topic overlap status, using 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 agent and users and gives suggestions based on unmatched user input.

Once enabled, go to [Analytics > Topic triggering \(preview\)](#). A list of potential topics are shown with the number of times a query about this topic was made by users of the agent. 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.

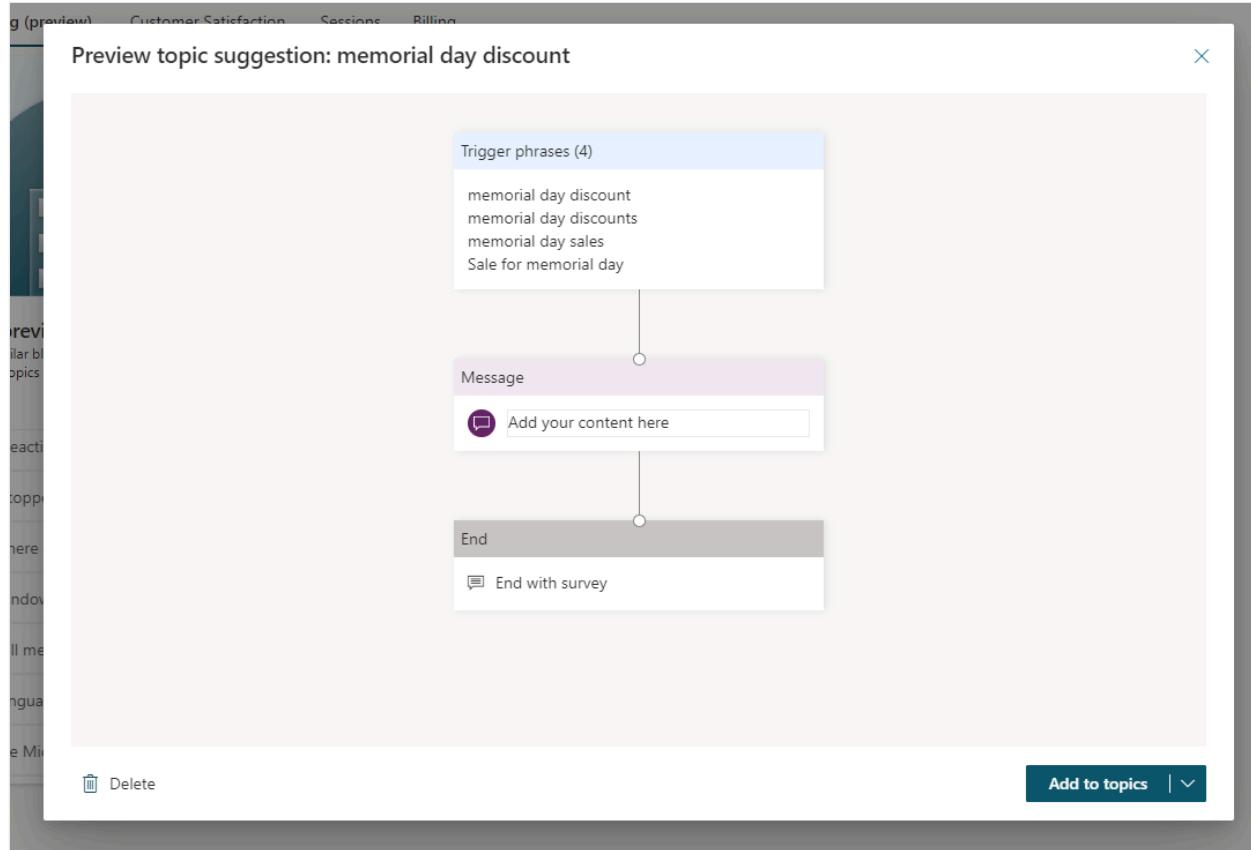
Updated 17 hours ago

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. The analyzer groups together queries that don't match an existing topic and shows them in the list. Your agent needs at least 100 new conversations from the last time any suggestion was generated to trigger 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 agent 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 agent, or add it to your list of topics by selecting **Add to topics**.



Enable or disable Generative AI

ⓘ Important

You can't enable these features for agents *not* marked **Classic**. This includes any agent created (or converted) in the web app after May 23, 2023, or any agents that aren't created with the unified authoring canvas.

The unified authoring canvas includes improvements to the types of AI used by Copilot Studio. We recommend you [create and convert your agents with the unified authoring canvas](#).

To enable these older AI capabilities in Copilot Studio:

1. Open your agent and select **Settings**:

- In the web app, the agent must be marked with **Classic**.

- In the Teams app, the copilot can be any copilot.
2. From the navigation menu, select **Generative AI**.
3. For each feature, select the checkbox to turn it on or off.
4. Select **Save** at the top of the tab.

Settings

Copilot details

Generative AI

Security

Authoring Canvas

Entities

Skills

Voice

Languages

Language understandi...

Advanced

Using generative AI in conversations

How should your copilot interact with people?

Classic – Use the topics you build to respond to trigger phrases—actions can only be called from inside a topic.

Generative - Use generative AI to respond with the best combination of actions, topics, and knowledge.

How strict should the content moderation be?

Low - More creative Medium - More balanced High - More precise

Save

🔍

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

What's new in Copilot Studio

Article • 12/16/2024

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.

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 Graph 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 loss prevention 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 ↗](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Get access to Copilot Studio

Article • 11/19/2024

This article includes details about Copilot Studio Licensing. You can also review the [Microsoft Power Platform Licensing Guide](#) to learn more.

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

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

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

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 subscriptions with Microsoft Power Platform and Teams capabilities, excluding 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 subscriptions, against the standalone Copilot Studio subscription. For a full, comparative list, see the [Microsoft Power Platform Licensing Guide](#).

Also see the [Quotas, limits, app registration, certificates, and configuration values](#) article for other capacity considerations.

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
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
Integrate Microsoft Bot Framework dialogs	Not available	Overview of Bot Framework Composer
Hand off agent conversation to a live representative	Not available	Hand off to a live agent

Upgrading 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.

The screenshot shows the 'Copilot' section of the Microsoft Teams interface. On the left, there's a sidebar with various options like Overview, Topics, Entities, Generative AI, Analytics, Publish, Settings, and AI integration tools. The 'Channels' option is highlighted with a red box. The main area is titled 'Channels' and contains six tiles representing different publishing channels:

- Microsoft Teams**: Chat with your chatbot through a Teams app.
- Demo website**: Try out your chatbot and invite team members to do the same.
- Custom website**: Activate your chatbot on your own website.
- Mobile app**: Add your chatbot to a native or web-based mobile app.
- Facebook**: Connect with your customers on Messenger.
- Skype**: Expand your chatbot's reach to customers on Skype. This tile has a circular button with a plus sign and a magnifying glass icon.

At the top, there are two status messages: one about data loss prevention policies and another about channel settings being unavailable due to the same policies. There are also 'Learn more' links and 'Download details' buttons.

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.

This screenshot shows a Microsoft Teams window with a floating promotional overlay for 'Power Virtual Agents (preview)'. The overlay features a diamond icon and the text 'Get the premium experience'. It encourages users to start a 60-day free trial to access premium features and provides a dropdown menu to choose their country/region. A checkbox for accepting terms and conditions is present, along with a 'Start free trial' button and a 'Close' button. The background shows a 'Test bot' channel in the Microsoft Teams interface.

① Note

The option to create a free trial may 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](#).

Feedback

Was this page helpful?



[Provide product feedback](#)

Sign up for a Copilot Studio trial

Article • 11/19/2024

This article explains how to sign up for Copilot Studio as an individual. If you're an administrator, see [Assign user licenses and manage access to Copilot Studio](#).

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.

Trial expiration

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.

Troubleshooting the sign-up process

In most cases, you're able to sign up for Copilot Studio using the steps in this article. Some issues might prevent you from signing up.

Personal email address

You can't sign up for Copilot Studio using a personal email address. You must use a work or school account.

Self-service sign-up disabled

If you receive a message that your sign-up couldn't be completed it means your organization's IT administrator has disabled 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](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Quickstart: Create and deploy an agent

Article • 11/19/2024

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 article helps you create an agent for the first time. Learn how to add knowledge to your agent, test content changes in real time, and deploy your agent to a test page you can share with others.

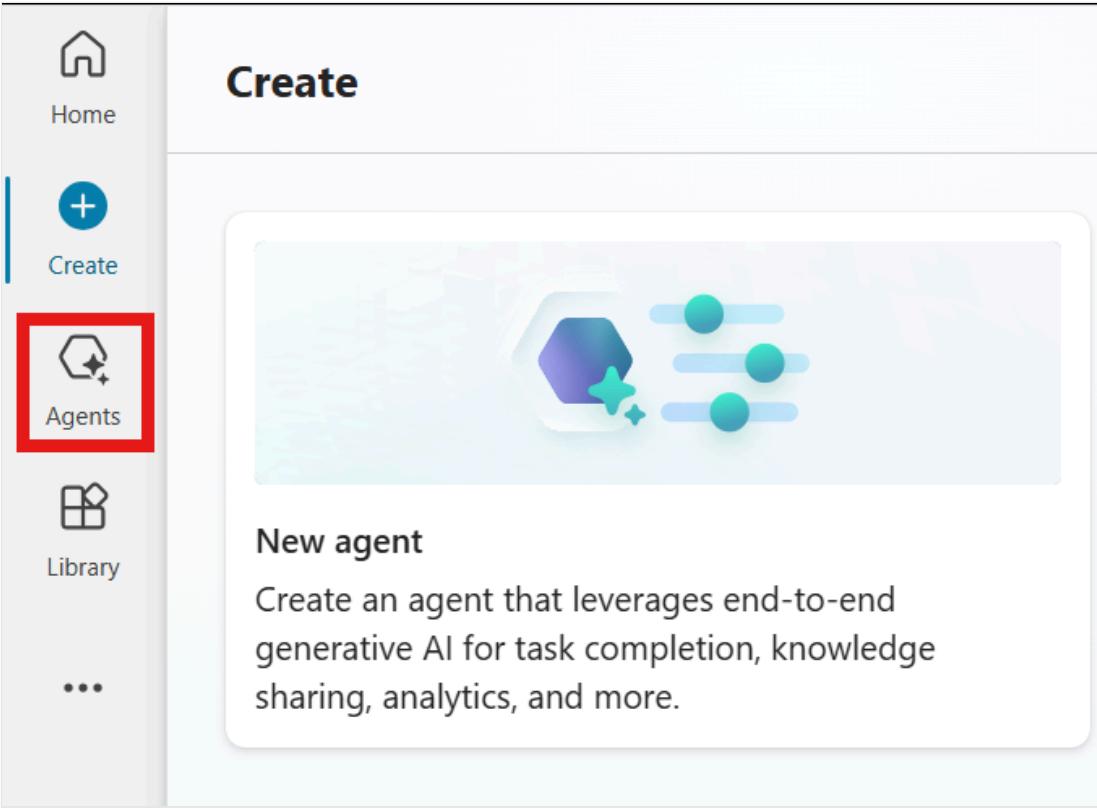
Web app

Create an agent

When you create an agent, you can describe what you want your agent to be and do, or add those details as separate pieces of information.

The examples in this quickstart create an agent called the Friendly Agent Tutor to help users learn about Copilot Studio. However, you can create your own unique agent using a publicly available website.

1. After [logging in or signing up](#) for Copilot Studio, you land on the **Home** page. Select **Create** in the left navigation.



The screenshot shows the 'Create' page of Copilot Studio. On the left, there's a vertical navigation bar with icons for 'Home', 'Create' (which is highlighted with a blue background), 'Agents' (which is highlighted with a red border), 'Library', and three dots. The main area has a large 'Create' button at the top. Below it is a placeholder card with a hexagonal icon and some teal circles. At the bottom, there's a section titled 'New agent' with the sub-instruction: 'Create an agent that leverages end-to-end generative AI for task completion, knowledge sharing, analytics, and more.'

2. On the **Create** page, select **New agent**.
3. Use the chat to describe your agent, using the provided questions for guidance.

Keep your description simple for now, but make sure you include information about what your agent helps users do and what conversation style and tone it uses. Copilot Studio uses your responses to fill in the details in name, description, instructions, and knowledge that define your agent.

For example: _Your name is Friendly Agent Tutor. You will help users learn how to create agents. You should talk to users like a kind, patient teacher.

 **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 do not 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 conversion. You can use words and phrases like *casual, formal, and understandable for children in grade three.*

 4. Add an image to represent your agent.
 - a. Select the agent icon in the top bar.
 - b. Select **Change icon**.
 - c. Choose an image from your device. The image file must be in PNG format and less than 30 KB in size.
 - d. Select **Save**.
 5. After you have a name, icon, description, and instructions, your agent is ready. Select **Create**. The **Overview** tab for your agent appears.

You now have an agent you can start testing! You can chat with your agent in the **Test your agent** chat.

Improve your agent

Now that you have an agent, you can start testing and improving it.

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

You need the **Overview** tab for most of the actions in this section.

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'll 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 your agent** chat. Ask your agent a question. For example, ask how to create an agent.

▶ Test your agent

... ⌂ ×

2 minutes ago

How do I add a knowledge source?

Just now

Adding a knowledge source to a virtual assistant can significantly enhance its ability to provide accurate and relevant information. Here's a step-by-step guide to help you through the process:

Step 1: Identify the Knowledge Source

Ask a question or describe what you need

0/2000



2. 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?

In the **Details** card, select **Edit**. Change your agent's instructions 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?

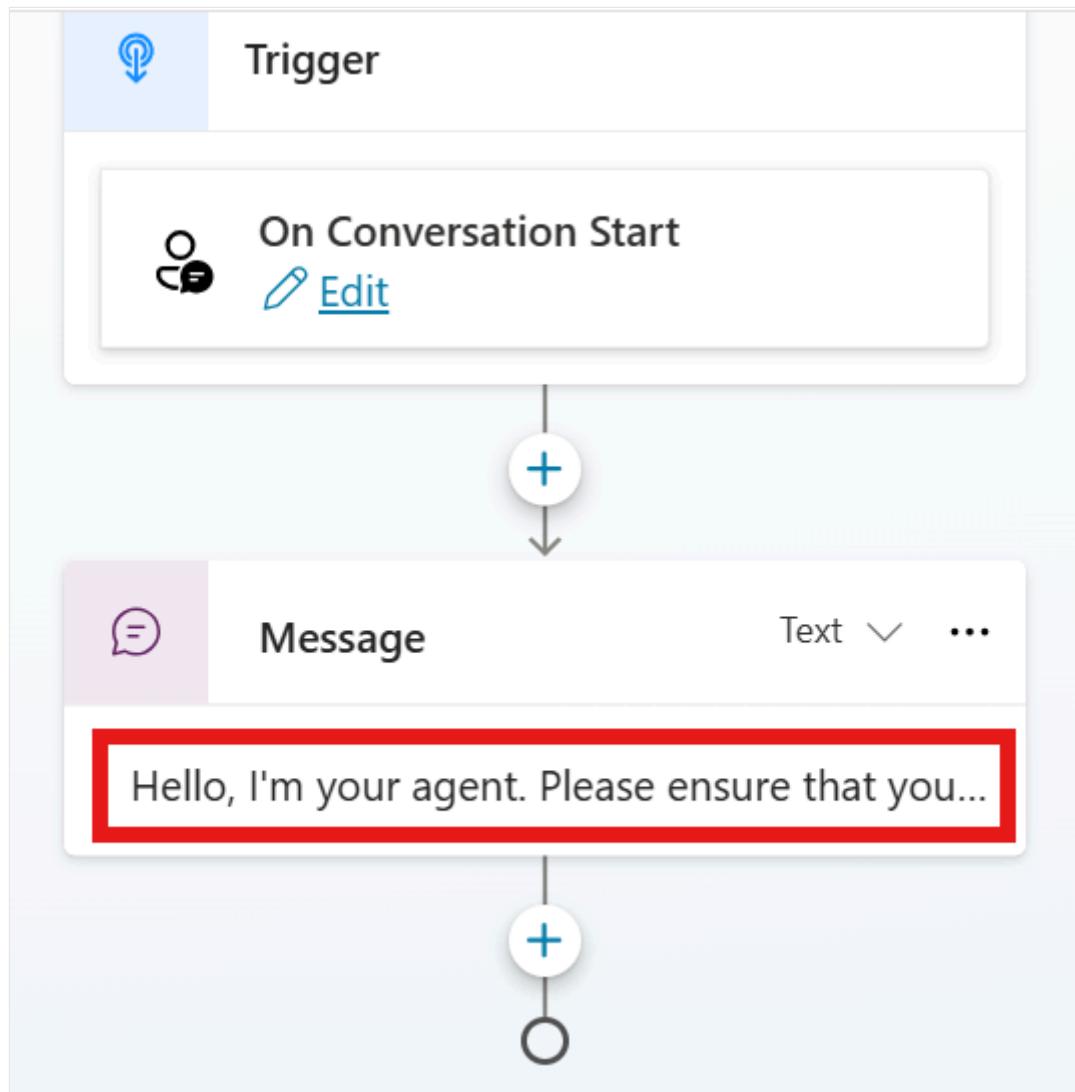
Change your agent's introduction

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

1. In the **Test your agent** chat, select on your agent's introductory message. This opens the message in the **Topics** tab.

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

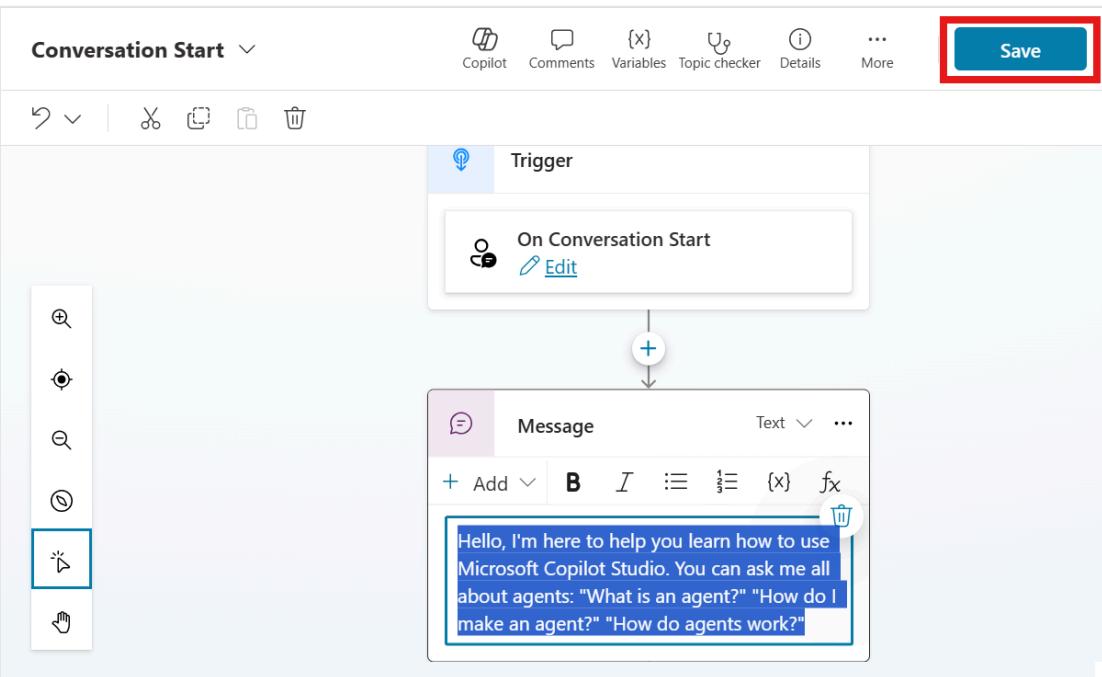
2. In the **Message** box, select the text of the message.



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

For the Friendly Agent Tutor, the introduction message says *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**.



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

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, description, or instructions:

1. In the **Overview** tab's **Details** section, select **Edit**.
2. Make your 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. In the **Overview** tab's **Knowledge** section, select **Add knowledge**.

 Edit

 Name
Friendly Copilot Tutor

Description
A copilot to answer questions about copilots.

Instructions
Maintain a patient, teaching tone with users. Explain important terms in your responses. Help users understand how their copilots work.

 Knowledge
Add data, files, and other resources to inform and improve AI-generated responses.
[+ Add knowledge](#)

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

 Microsoft Copilot Studio docs 

[See all](#)

2. Select the type of knowledge. This quickstart uses **Public websites** sources.
3. Enter the URL for the website, then select **Add** beside the textbox.
4. Name and describe the knowledge source so you can keep track of all your agent's sources.

 Add public websites 

How to choose websites:

- If your site is external, make sure it's indexed or found by Bing.
- Don't use sites with forums or comments from end users; this can reduce the relevancy of answers.
- Don't include query strings, more than two levels of depth, or the character "?" in your URL.

Public website link
 

Link	Name	Description
 https://learn.microsoft.com/mic...	Microsoft Copilot Studio docs	This knowledge source searches informati

This generative AI 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 the [Microsoft Privacy Statement](#)

5. Select **Add**.

To change existing knowledge sources:

1. In the **Overview** tab's **Knowledge** section, select the overflow menu.
2. Select **Edit** to change the knowledge source, or **Delete** to remove it from your agent's sources.

You now have an agent you can start testing! You can chat with your agent in the **Test your agent** chat.

Publish your agent

Once you're happy with the content authored in your agent, you can publish your agent to a website.

ⓘ Note

You may 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 **Go to demo website** in the overflow menu.
3. Send the URL to others to demonstrate it.

ⓘ Note

For more information on 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.

Feedback

Was this page helpful?



[Provide product feedback](#)

Upgrade to Copilot Studio unified authoring

Article • 11/19/2024

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.

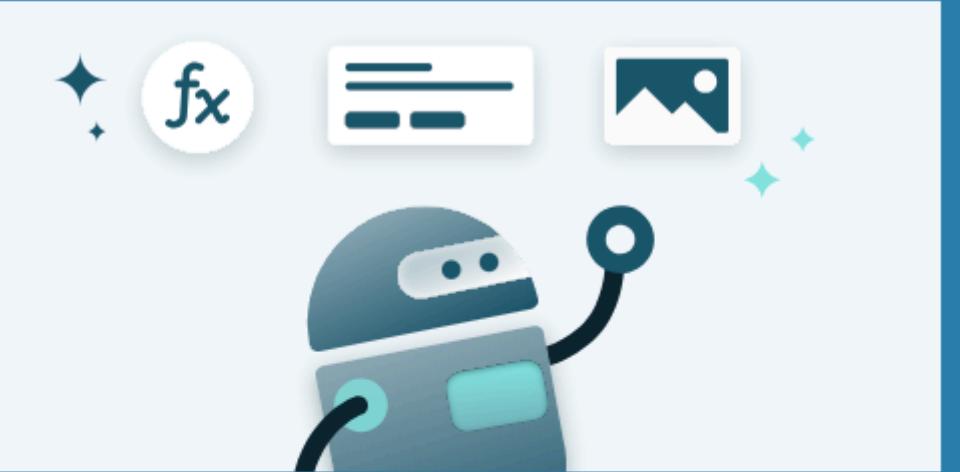
With the release of our latest version of Copilot Studio, now is the time to consider migrating from your classic Power Virtual Agents experience to the latest version of Copilot Studio. This guide aims to help readers understand some of the significant improvements within this release, and areas to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

[!\[\]\(99f78bec33c2da739e55956a838a5959_img.jpg\) Copy this chatbot](#) [!\[\]\(4b6568864191cec3e982494e479532d5_img.jpg\) 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 (not topics built using Bot Composer).
- 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 may 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 let you name the agent, specify the agent's language, and also get started with boosted conversations.

Additionally, you can configure the icon, choose to include (or not include) lesson topics, and also 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: Users are no longer able to create topics using Bot Composer. 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 causes 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 the latest Copilot Studio release, 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

We're launching a brand new Copilot Studio in a Day training on Microsoft Learn, to help get you started with structured learning of Copilot Studio. The Copilot Studio in a Day training also contains specific material for instructors. At this time, we have preview lab materials that can be accessed here, which provide readers with a selection of labs they can use to get started. Check out [Events on the Microsoft Events page](#). (Note: When booking, ensure which version the event is targeting.)

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

Limitations

When you clone the contents of a classic chatbot, any topic authored using Bot Framework Composer is created as an empty topic with none of the content or nodes in the original topic. You must add content to those topics after your agent is created. However, the cloned topic has the same input and output variables as the original.

For additional guidance, see [Cloning an existing Power Virtual Agents classic bot to a Copilot Studio agent](#) in this article.

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

There are a range of resources available to assist with business application technology deployments, and best practices around conversational projects using Copilot Studio. Check out the following links to learn more:

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

 **Warning**

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

Recommendations when 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

- [Key concepts for Bot Framework Composer users](#)
- [Find community help and support](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Find community help and support for Copilot Studio

Article • 11/19/2024

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

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. In the side pane, select **Help + Support**, and then select **New support request**.

The screenshot shows the Microsoft Power Platform support portal. On the left, there's a sidebar with various navigation options: Home, Environments, Advisor (Preview), Analytics, Billing, Resources, Help + support (which is highlighted with a red box), Data integration, Data (preview), Policies, and Admin centers. The main content area is titled 'Help + support' and has two tabs: 'Support requests' (which is underlined) and 'Known issues (Preview)'. Below these tabs, a message reads 'Welcome to the Power Platform support portal. If you have questions or need support to resolve an issue, create a new support request. [Learn more](#)'. A purple button labeled 'New support request' is also highlighted with a red box. In the bottom right corner of the main area, there's a circular 'Feedback' button with a plus sign.

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.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Key concepts - Copilot Studio security and governance

Article • 12/03/2024

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
DLP controls	Admins can use DLP 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 loss prevention 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
Run connectors and flows with user	Agent makers can configure connectors and flows to use the user's credentials by default.	Use actions with custom copilots

Control	Core scenario	Related content
credentials		(preview)
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 DLP	Admins can manage agent capabilities with triggers using DLP policies, ensuring protection against data exfiltration and other risks.	Data loss prevention example - Block event triggers in agents

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.

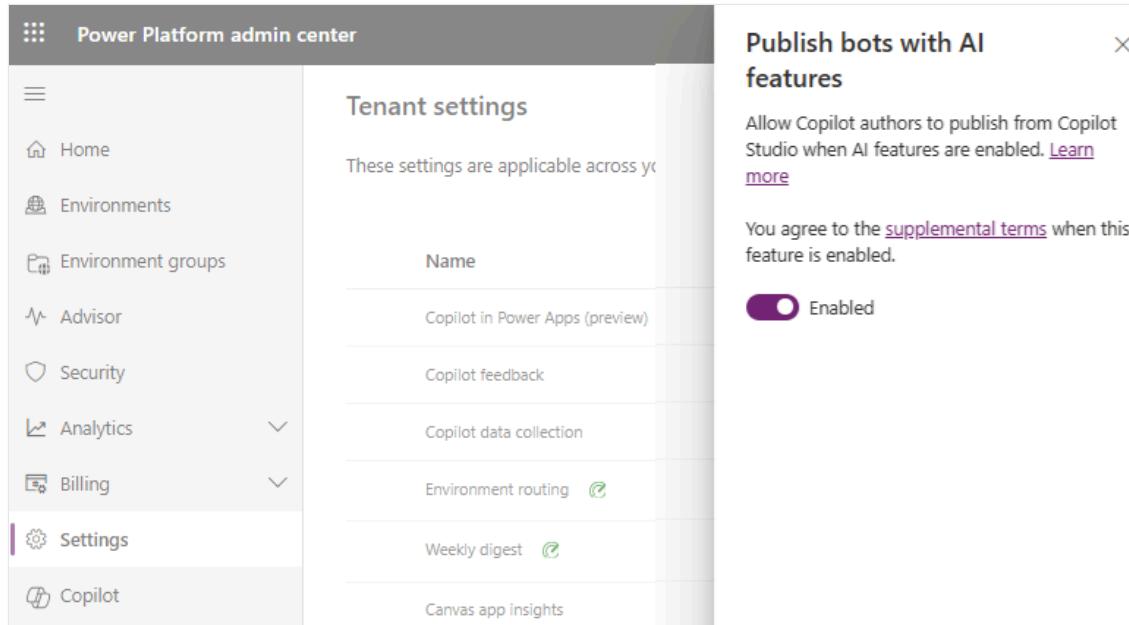
Learn more at [Copilot Studio compliance offerings](#).

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.



- [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](#).

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Security FAQs for Copilot Studio

Article • 12/02/2024

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 rotates the certificates on a regular cadence. This 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.

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

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

- For more information, see [Data loss prevention 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.

Microsoft Copilot Studio stores its data in Microsoft secured storage and in Microsoft Dataverse. When you apply the customer-managed key to these environments, only the data stores in Microsoft Dataverse are encrypted with your key. The non-Microsoft Dataverse data continues to be encrypted with the Microsoft-managed key. Learn more about [customer-managed keys](#).

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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

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?



[Provide product feedback ↗](#)

Data locations in Copilot Studio

Article • 08/26/2024

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 copilot to any of the Microsoft Azure datacenters around the world.

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

This article lists the Azure geographic locations and datacenters where your data will be stored when using Copilot Studio.

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.

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 Pacific	Southeast Asia (Singapore), East Asia (Hong Kong Special Administrative Region)

Azure geographic locations	Azure datacenters (regions)
Australia	Australia East (New South Wales), Australia Southeast (Victoria)
Canada	Canada Central (Toronto), Canada East (Quebec City)
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)
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)
Switzerland	Switzerland North (Zurich), Switzerland west (Geneva)
United Arab Emirates	UAE North (Dubai), UAE Central (Abu Dhabi)
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.
- **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 agents.
 - **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.

Related content

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

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Geographic data residency in Copilot Studio

Article • 08/19/2024

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.

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.

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, a copilot 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 copilot 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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Security and geographic data residency in Copilot Studio

Article • 08/19/2024

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 copilots that can operate within [specific geographic locations](#). This capability ensures that data generated and processed by these copilots 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 allows organizations to store data in multiple geographic locations while maintaining a unified management interface.

Key features include:

- **Data segmentation:** Data can be segmented based on geographic requirements, ensuring that sensitive information remains within the designated region.

- **Centralized management:** Despite the geographic distribution, administrators can manage all data and services from a central location, simplifying compliance and security management.

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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Ensure compliance with Copilot Studio

Article • 08/19/2024

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

1. 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.
2. Customer trust: Adhering to compliance standards demonstrates a commitment to data security, which can enhance customer trust and loyalty.
3. Risk management: Compliance helps in identifying and mitigating risks associated with data breaches and unauthorized access.
4. 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 copilots that handle protected health information when your organization is bound by HIPAA, as in the following scenarios where the copilot 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 ↗](#)

Key concepts - Authoring agents

Article • 11/19/2024

Creating an agent with Copilot Studio is easier than ever. Whether you're new to conversational AI or a seasoned developer, our intelligence platform is with you and your team every step of the way.

How agent conversations work

Copilot Studio uses [customized NLU model and AI capabilities](#) to understand what a user types, then responds with the best agent topic. An agent topic is a portion of a conversational thread between a user and the agent. Topics are linked together to form nodes. 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 support overhead is reduced 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. These questions use `if/else` arguments, or logic gates, to determine which store the customer wants. The final output of the topic shows the hours and contact information for that store's location.

However, you can't anticipate all the types of questions your customers ask. To help mitigate this, Copilot Studio incorporates powerful new AI-powered capabilities that use the latest advancements in NLU models. With your agent linked to a public or Bing-indexed website, your agent automatically generates responses. These responses are conversational, plain language, and don't depend on the agent builder to create topics for every eventuality.

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

Your agent uses AI powered by the Azure OpenAI GPT model, also used in Bing, to create agent 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.

Accessibility

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

Routing anchors

For complex scenarios, users who author content with keyboard navigation can use routing anchors.

Press `Enter` or `Space` on a route anchor to go into editing mode for that route. Once in the editing mode, you can `Tab` to other node anchors.

With an anchor selected, you can:

- Press `Delete` to delete the route. The target node becomes an orphaned node if this route is the only route leading to that node.
- Press `Escape` to come out of `Tab` mode and resume normal tabbing.
- Press `Enter` or `Space` on another node to reconnect the route to the new node. This node has the same source node, but a new target node.
- Press `Tab` to go to the next available location to drop this route.

At any time, pressing these hotkeys on a route anchor reads out the properties of the route:

- `Alt + Shift + A` reads out the source node.
- `Alt + Shift + B` reads out the target node.

ⓘ Note

If you're using a screen reader, the **Settings** navigation menu might not get read as "Collapsed", but it's an expandable control which can be selected and expanded. This is a known issue with our menu control, which will be fixed at a future date.

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Use entities and slot filling in agents

Article • 11/19/2024

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, the user might say "I tried to use my gift card but it doesn't work" and the agent knows to route the user 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. Watch the video to get started with entities.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4mTUN?postJs||IMsg=true> ↗

An entity can be thought of as a unit of information that represents a certain type of a real-world subject, like a phone number, zip 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.

Out of the box, Copilot Studio comes with 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 a user input and save it for later use.

To help understand that notion, the **Money** entity can be used 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 the prebuilt entities that are available is shown.

Name	Description	Method	Errors	Modified ↓
Age	Age of a person, place, or thing, ...	Prebuilt		
Boolean	Positive or negative responses, e...	Prebuilt		
City	City names, extracted as a string	Prebuilt		
Color	Primary colors and hues on the c...	Prebuilt		

2. Select the **Money** entity to open the details pane for the entity.

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 the ways it can be used to look for information related to money or currency from a user's 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 entity and saves it to a variable, it saves "1000" as a number even though the surrounding information was text.

Custom entities

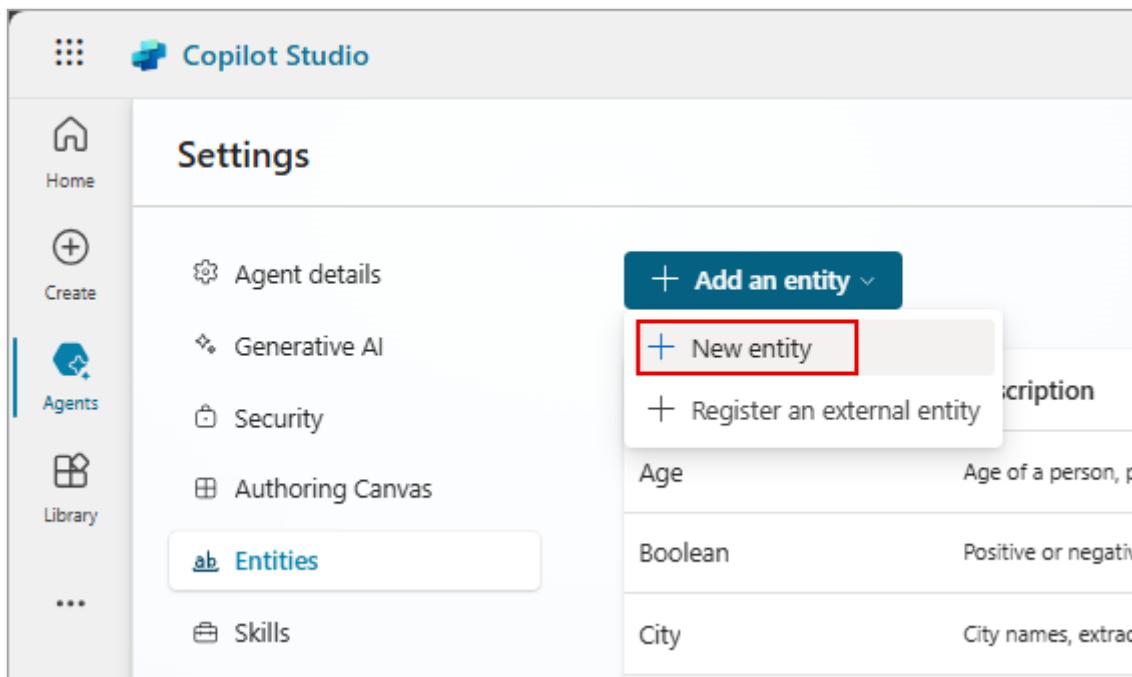
The prebuilt entities cover commonly used information types. On some occasions, such as when building an agent that serves a specific purpose, you might need to teach the

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 dialog.

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

Create an entity



Make your agent smarter with entities.

An entity is a piece of information (person, place, thing, or concept) that the agent might want to pick out from what your customer says.

[Learn more about entities](#)

Choose a method

Closed list

Define a list. Use when you have an exact list of items that is a manageable size (small to medium).

Example: List of sizes

Extra small
Small
Medium
Large

Regular expression (Regex)

Define a specific logical pattern, such as an ID number, credit card number, IP address, and more.

Example: Article ID pattern that starts with "kb" and ends with 6 digits, such as 'kb123456'.

`kb[0-9]{6}`

Selecting this option when creating an entity shows a pane where you can specify the name of the entity, an optional description, and enter the items you want included in the entity.

Camping activities

Name *

Camping activities

List items

Enter item

Add

Description

These are activities that users can ask about.

Method

List

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

Item

Synonyms

hiking

trekking, mountaineering

yoga

+ Synonyms

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](#)

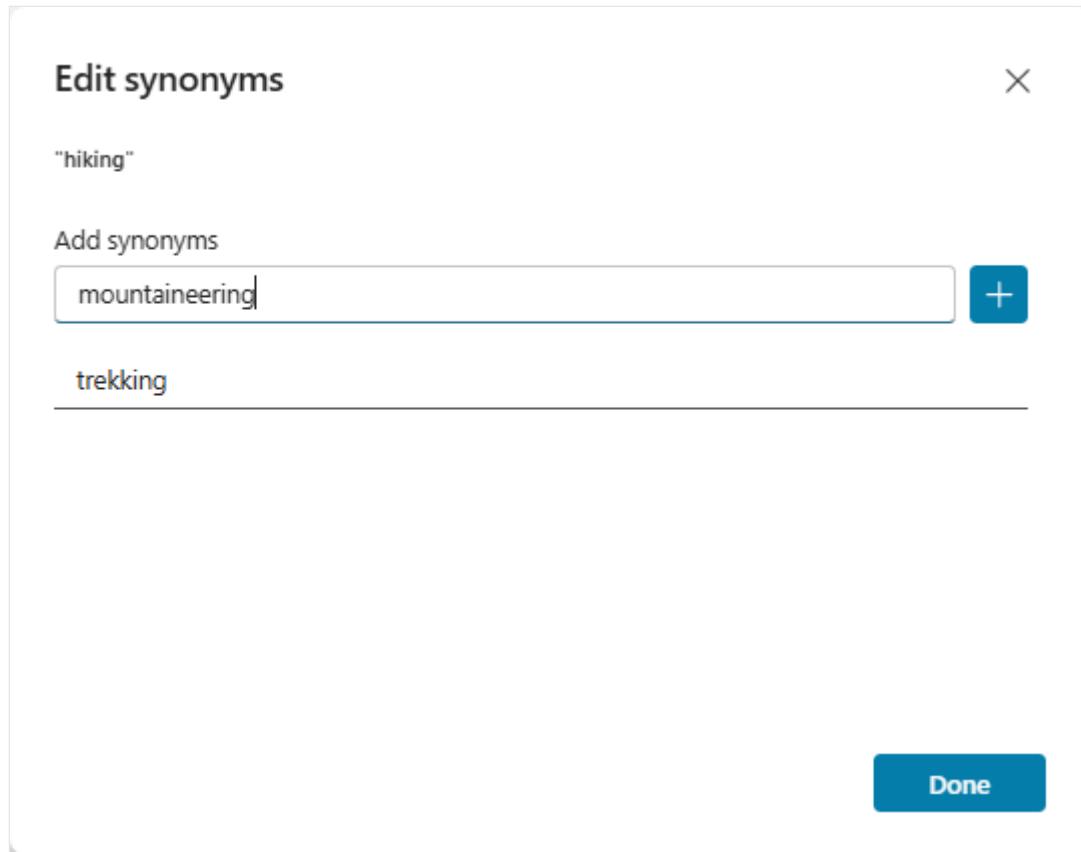
Save

Close

When you enter items, you can:

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

You can add synonyms to manually expand the matching logic for each item in the entity's list. For example, in the "hiking" item, you can add "trekking" and "mountaineering" as synonyms.



For each entity, you can also enable **Smart matching**.

Smart matching is part of the intelligence supported by the agent's language understanding model. With this option enabled, the agent will interpret the user's input using fuzzy logic, based on the listed items in the entity.

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

When you're finished creating or editing your entity, select **Save** to save and return to the list of entities. Select **Close** to discard your changes (you'll be given an option to go back to editing the entity in case you accidentally selected the wrong option).

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 a user's input, or if you need to allow for specific variations in how a user might format or enter their input in a conversation.

For example, you could use a regex entity to identify items such as a tracking ID, a license number, a credit card number, or an IP address from a string the user enters into the agent.

Create an entity



Make your agent smarter with entities.

An entity is a piece of information (person, place, thing, or concept) that the agent might want to pick out from what your customer says.

[Learn more about entities](#)

Choose a method

Closed list

Define a list. Use when you have an exact list of items that is a manageable size (small to medium).

Example: List of sizes

Extra small
Small
Medium
Large

Regular expression (Regex)

Define a specific logical pattern, such as an ID number, credit card number, IP address, and more.

Example: Article ID pattern that starts with "kb" and ends with 6 digits, such as 'kb123456'.

`kb[0-9]{6}`

Selecting this option when creating an entity will show a pane where you can specify the name of the entity, an optional description, and enter the regex pattern that should be used to match and extract entity items from the input.

Regex selection

Name *
Regex selection

Description
This is a regex to match on user selections.

Pattern *
Enter regular expression pattern

! The pattern of a regex type cannot be empty

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**

A Copilot Studio regex entity uses the [.NET 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, and additional information on the syntax and more examples can be found in the [Regular Expression Language - Quick Reference](#) topic.

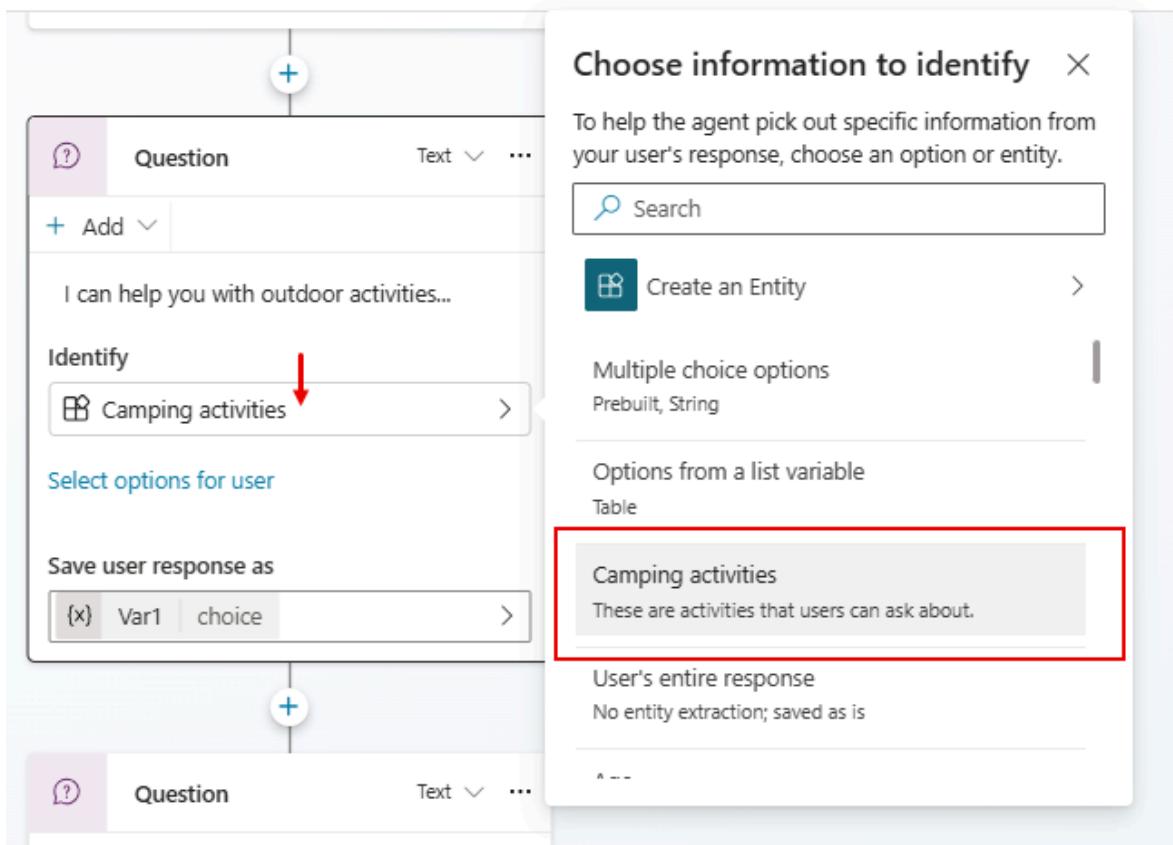
If you need to make the regular expression accommodate multiple patterns you can use the alternation operator `|` to concatenate regular expressions. For example, you could combine two regex patterns to look for both the old and new formats of a product ID code.

When you're finished creating or editing your entity, select **Save** to save and return to the list of entities. Select **Close** to discard your changes (you'll be given an option to go back to editing the entity in case you accidentally selected the wrong option).

Use entities in a conversation

Now that you've done the work of giving the agent the knowledge about outdoor gear by creating that product category entity and a few other custom entities, you can start to use them when constructing an agent conversation.

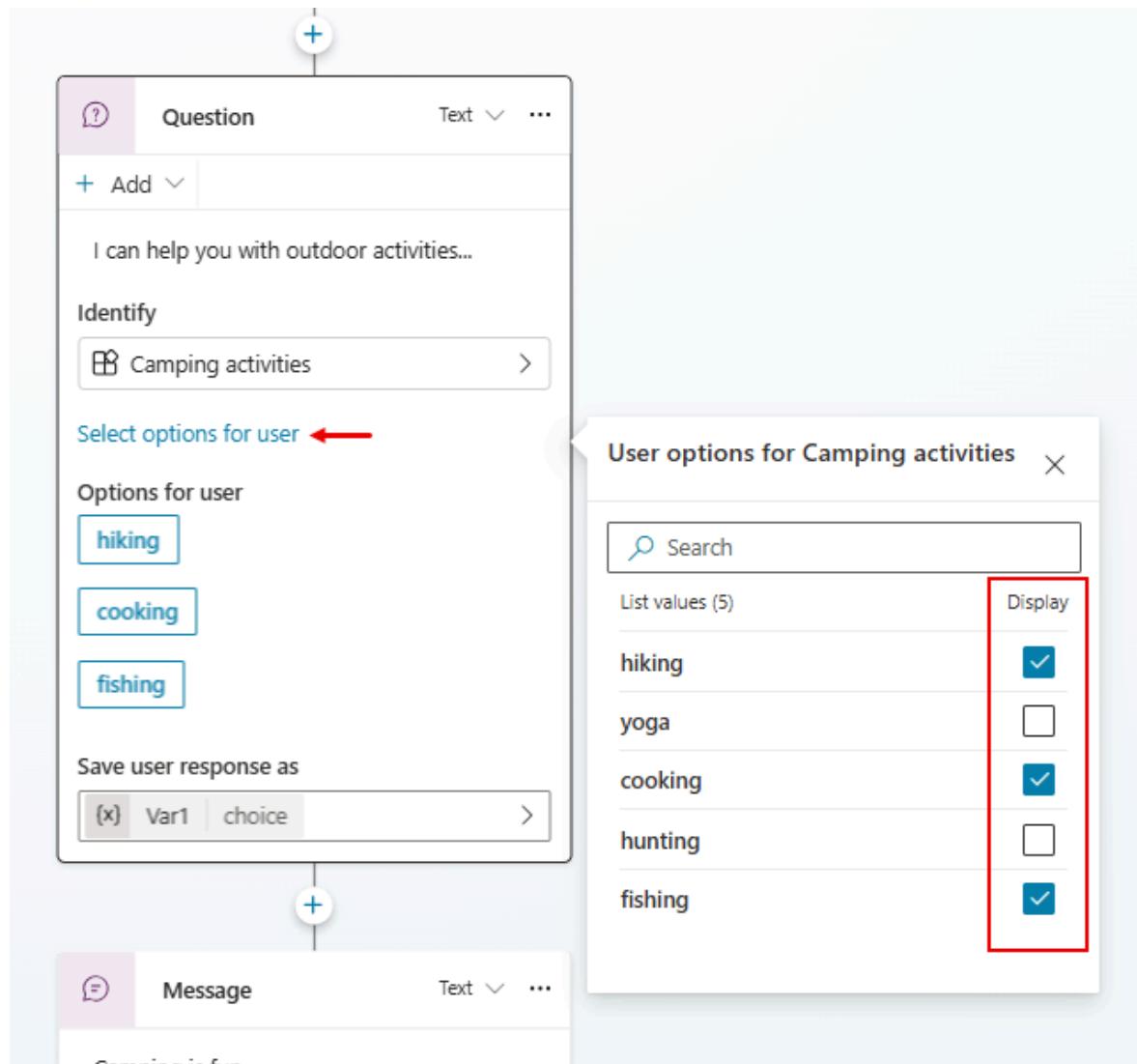
1. Go to the [Topics](#) page for the agent you want to edit.
2. Open the topic for an entity you want to collect.
3. Select the **Add node icon**  and then select **Ask a question**.
4. Under **Identify**, select the entity you created in [Custom entities](#).



The screenshot shows the Microsoft Bot Framework designer interface. On the left, there's a main panel with a 'Question' node, a text input field containing 'I can help you with outdoor activities...', and a 'Identify' section where 'Camping activities' is selected. A red arrow points from the text 'These are activities that users can ask about.' in the 'Choose information to identify' dialog to the 'Camping activities' entry in the 'Identify' list. The right panel is titled 'Choose information to identify' and contains a search bar, a 'Create an Entity' button, and several options: 'Multiple choice options' (Prebuilt, String), 'Options from a list variable' (Table), and a highlighted box containing 'Camping activities' with the subtext 'These are activities that users can ask about.' This box is also outlined in red. Below it are other options: 'User's entire response' (No entity extraction; saved as is) and a collapsed section indicated by '...'. The overall interface is light-colored with blue and grey UI elements.

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. However, in Copilot Studio, slot filling means placing the extracted entity value into a variable.

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

The topic is successfully triggered and the agent asks for the type of activity, also showing the button choices specified when authoring the **Ask a question** node. In the dialog tree, tracking between topics also shows the agent is running to the question node you just edited.

The screenshot shows a conversational AI authoring interface. On the left, a 'Trigger' panel lists several 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?'. Below this is a 'Question' panel containing the response 'I can help you with outdoor activities...' and a 'Select options for user' section. Under '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 message history shows the AI responding with its identity and a welcome message. A user message 'I'm bored.' is shown with a response 'I can help you with outdoor activities...'. Below the message history is a note: '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.

In tracking between topics, it shows 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. In the watch window, it shows the variable value is "Hiking".

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

Variables

Browse Test

Topic (1)

VarCampType choice

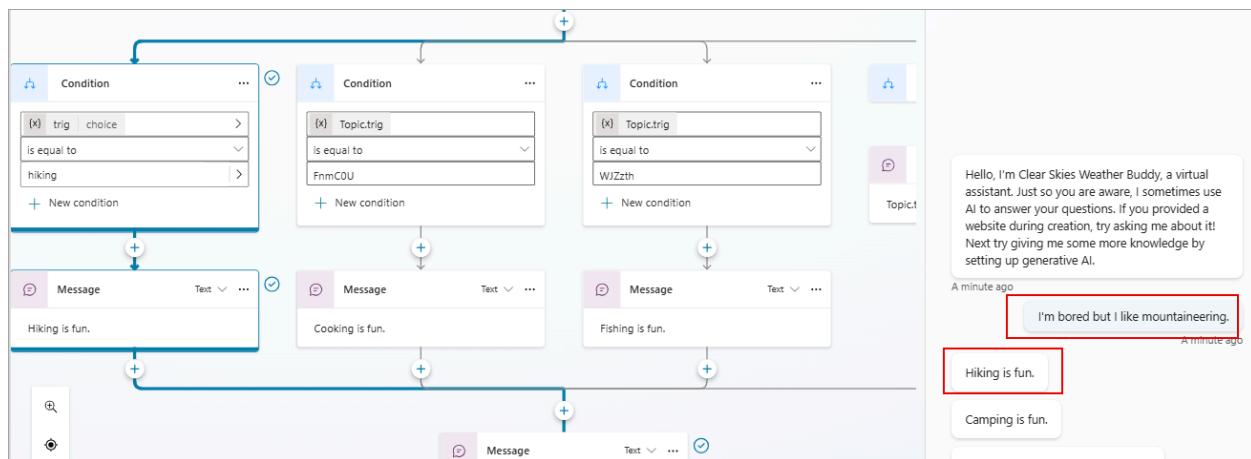
hiking

> 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 will prompt 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 both the trigger phrase that the user wants help with outdoor activities and also provides a second piece of information, "mountaineering", which is an entity-defined synonym for hiking. In this case, the agent fills in both the entity for choosing an outdoor activity, and for the type of activity.

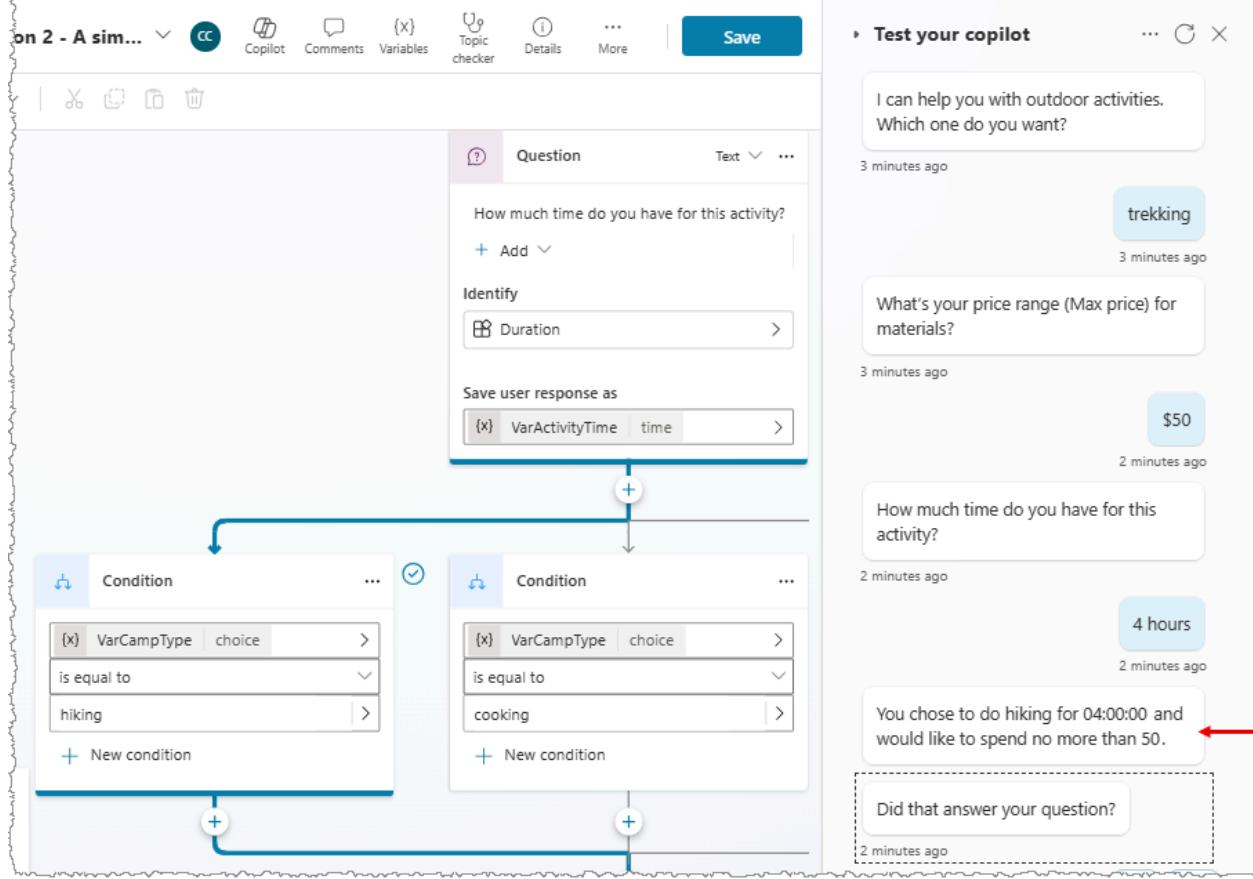


In tracking between topics, you can see the agent takes in this user input, and intelligently skips the question node asking for the type of activity.

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

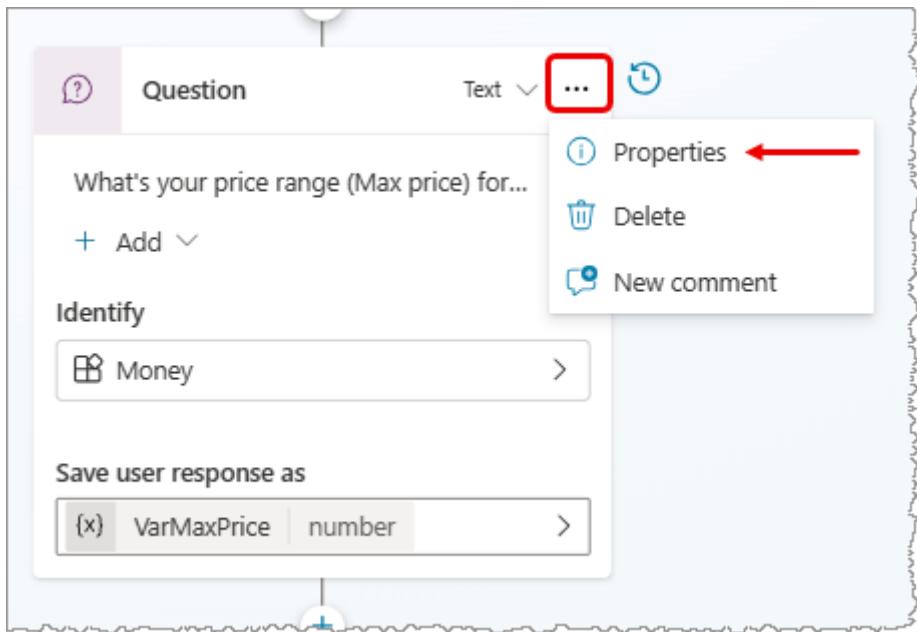
Let's restart the testing again and try another case. In this round, you can add a couple more question nodes asking for things like how much time you have for the activity (using the Duration entity) as well as the price range (using the Money entity).

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.

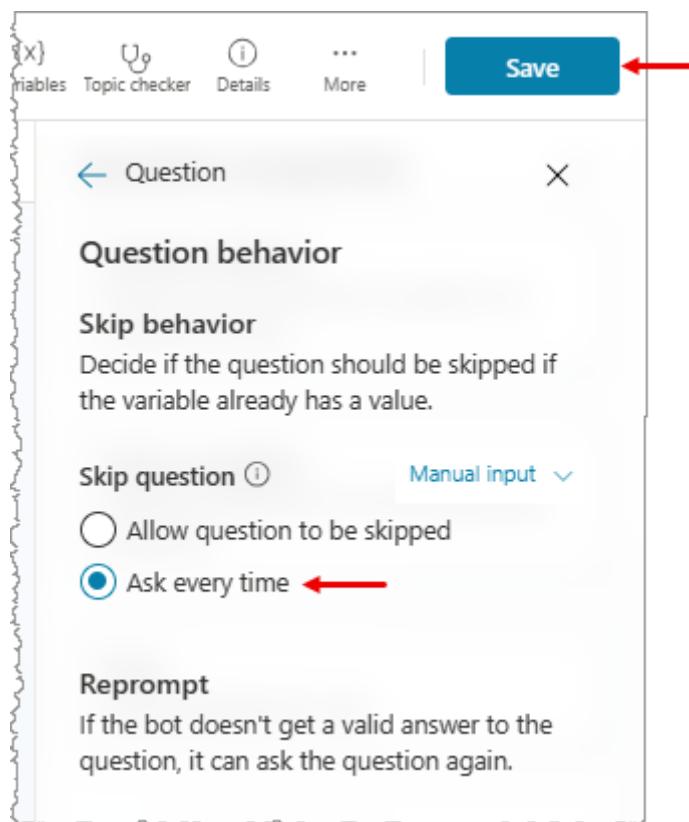


Proactive slot filling can be manually controlled at the node level. If you'd like to always prompt for the question within a specific node, regardless of whether the slot has been filled from previous user responses, you can disable the **Skip question** option for that question node. Perform the following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



Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Create and delete agents

Article • 11/19/2024

Copilot Studio lets you create an agent, using built-in content building blocks containing topics, trigger phrases, and preauthored agent conversations.

ⓘ 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. Go to the [Copilot Studio home page](#), select **Create** in the left navigation, then select on the **Agents** page.

Alternatively, select **+ New agent** on the **Agents** page.

2. Use the chat to describe your agent, using the provided questions for guidance.

Alternatively, select **Skip to configure** and fill the form.

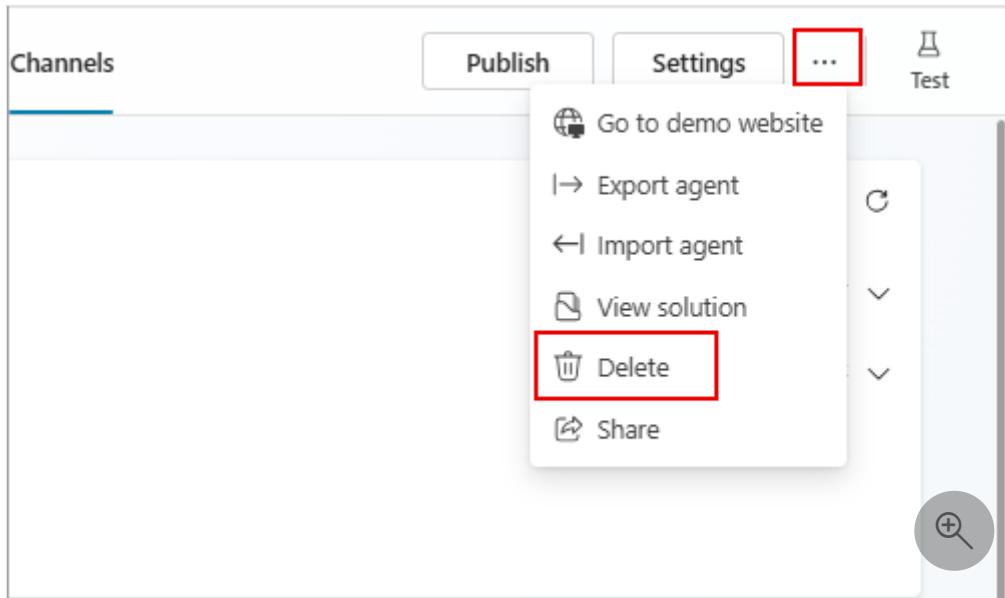
3. Select **Create**.

Delete an agent

Web app

You can delete agents to remove them from your environment.

1. Go to the [Copilot Studio home page](#).
2. On the side navigation pane, select **Agents**.
3. Select the agent you want to delete.
4. On the top menu bar, select the **More options** icon (...), then select **Delete**.



5. Confirm the deletion of the agent by typing the agent's name. Once you confirm, all agent content is deleted after a few minutes.

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](#).

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Test your agent

Article • 11/19/2024

As you design your agent in Copilot Studio, you can use the **Test agent** pane to see how the agent leads a customer through the conversation.

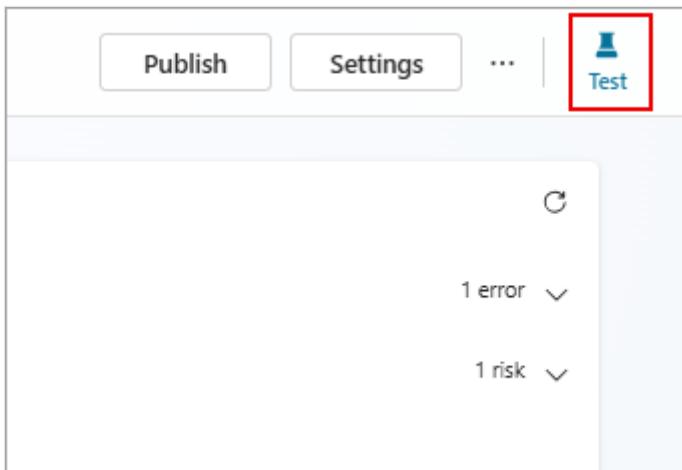
To help you find and fix unexpected behavior, you can enable tracking between topics to take you through the agent conversation step by step, and go to the corresponding node in the authoring canvas.

Web app

Use the **Test your agent** pane to walk through your agent conversations as a user. It's a good way to make sure your topics are working and that conversations flow as you expect.

Preview a conversation

1. If the **Test your agent** pane 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** pane, 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 your agent** pane. This takes you to the topic and the node that sent the response. Nodes that have fired have a colored checkmark and a colored bottom border.

The screenshot shows the Microsoft Bot Framework designer. On the left, a vertical canvas displays a conversation flow. A red box highlights a 'Question' node with the text 'I can help you with outdoor activities...'. Below it is a 'Condition' node. On the right, a 'Test your agent' pane shows a simulated conversation between a user and a bot. The user says 'I'm bored' (2 minutes ago), the bot replies 'I can help you with outdoor activities...' (2 minutes ago), the user says 'trekking' (A minute ago), the bot replies 'Hiking is fun.', 'Camping is fun.', and 'To what state will you be shipping?' (A minute ago). At the bottom of the test pane, there's a note: 'Make sure AI-generated content is accurate and appropriate before using. [See terms](#)'.

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.

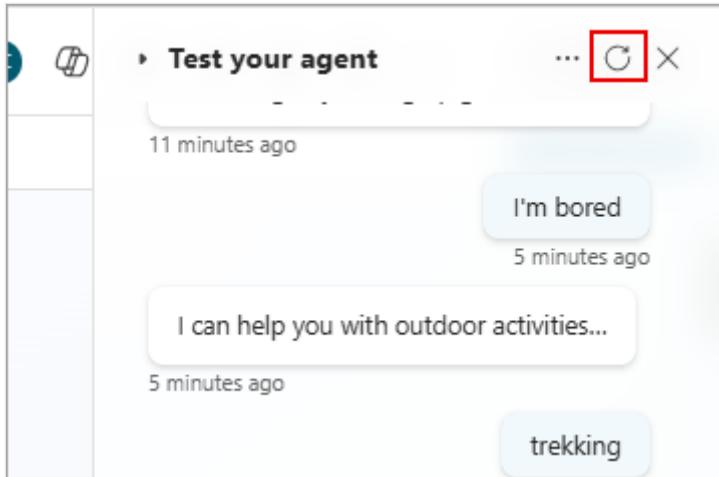
4. If you would like to follow the whole conversation automatically, as it moves from topic to topic, turn on **Tracking** at the top of the **Test your agent** pane.

The screenshot shows the 'Test your agent' pane. A red arrow points to the 'Track between topics' toggle switch, which is turned on (blue). Other options shown include 'Save snapshot', 'Test trigger', 'Manage connections', 'Flag an issue', 'Chat mode' (Text), 'Track between topics' (on), and 'Debug mode' (on). A note at the bottom says 'Next try giving me some more knowledge by setting up generative AI.' and '10 minutes ago'.

5. Continue the conversation until you're satisfied that it flows as intended.

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

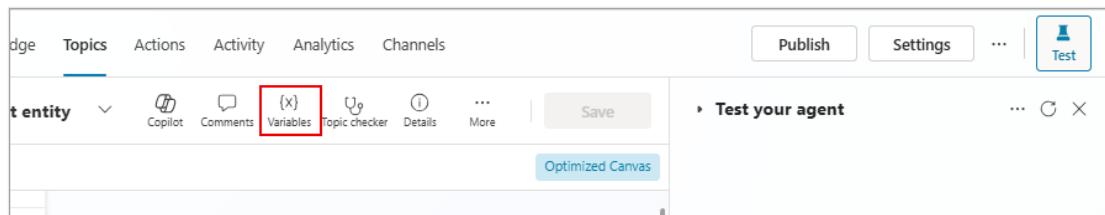
Your conversation is not automatically cleared when you save a topic. If at any point you want to clear the conversation from your test agent and start over, select the **Reset** icon.



Test variable values

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

1. Select **Variables** on the secondary toolbar to open the **Variables** pane.



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.

Variables

X

Browse Test

Topic (2)

CoffeeSize choice
Undefined

CoffeeType choice
Espresso

Global (0)

Environment (0)

3. To inspect variable properties, select the desired variable. The **Variable properties** panel appears.

Variable properties

X

Variable name
CoffeeType

Type
choice

Reference

Question What type of coffee would you like? (e

Type (choice) derived from here

[View all references](#)

Usage

Topic (limited scope)

Receive values from other topics

Return values to original topics

Global (any topic can access)

For information about using the Variables pane to test slot filling, see [Use entities and slot filling in agents](#).

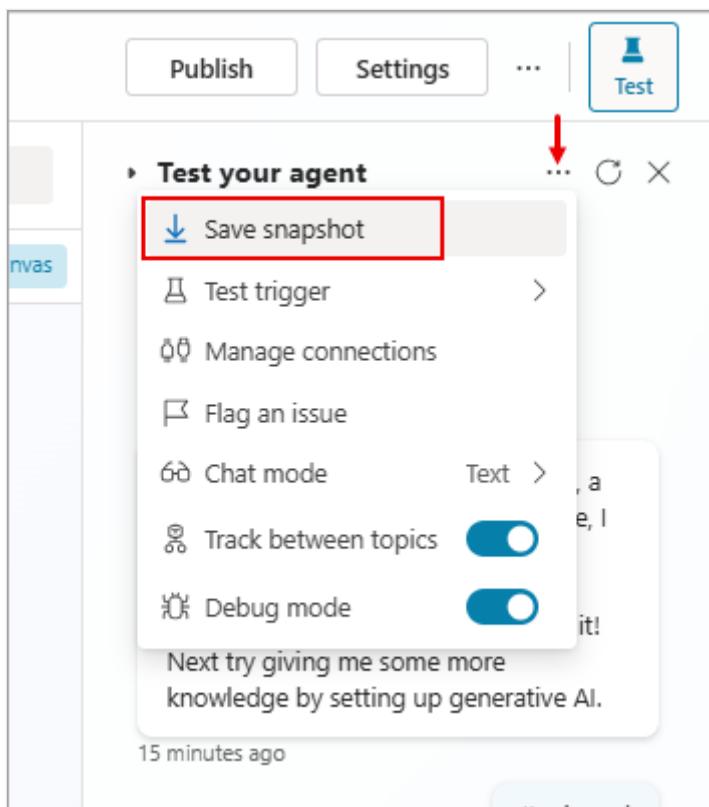
Save conversation snapshots

While you're testing your agent, you can capture the content of the conversation, and diagnostics data, and save it as a file. 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 may include sensitive information.

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](#), to manage the connections used by your test chat, select the More icon (...) at the top of the test pane, then select **Manage connections**.

Note

Help us improve Copilot Studio by reporting issues. Use the Flag icon near the top of the test pane to submit your conversation ID to Microsoft. The ID is a unique identifier that Microsoft uses to troubleshoot issues in a conversation. Other information, such as what is stored in a conversation snapshot file, is not sent when you report an issue. All information collected remains anonymous and will be used to help improve copilot quality.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create a custom agent from a prebuilt agent

Article • 11/19/2024

Copilot Studio provides several prebuilt custom agents to help you get started. Prebuilt agents are agents bundled in with Copilot Studio or available on the [AppSource portal](#) that you can customize to fit your needs. See [Create a custom agent from a prebuilt agent](#).

[+] Expand table

Agent name	Type	Where available
Team Navigator Agent	Custom agent	Create and Home pages of Copilot Studio
Safe Travels Agent	Custom agent	Create and Home pages of Copilot Studio
Store Operations Agent	Custom agent	Create and Home pages of Copilot Studio
Sustainability Insights Agent	Custom agent	Create and Home pages of Copilot Studio
Voice Agent	Custom agent	Create and Home pages of Copilot Studio
Weather Agent	Custom agent	Create and Home pages of Copilot Studio
Wellness Check	Custom agent	Wellness Check on the AppSource portal
Awards and Recognition Agent	Custom agent	Awards and Recognition on the AppSource portal
Inclusivity Agent	Custom agent	Create and Home pages of Copilot Studio
Case Management Agent	Custom agent	Create and Home pages of Copilot Studio
IT Helpdesk Agent	Custom agent	Create and Home pages of Copilot Studio
Leave Management Agent	Custom agent	Leave Management on the AppSource portal
Financial Insights Agent	Custom agent	Create and Home pages of Copilot Studio
Citizen Services Agent	Custom agent	Create and Home pages of Copilot Studio
Self-Help Agent	Microsoft 365 Copilot agent	Create and Home pages of Copilot Studio

Agent name	Type	Where available
Benefits Agent	Microsoft 365 Copilot agent	Create and Home pages of Copilot Studio

Create a custom agent from a prebuilt agent bundled with Copilot Studio

To create an agent from a bundled prebuilt agent:

 **Note**

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

1. From the **Home** or the **Create** page, under **Start with a template**, 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:
 - a. If you need to set up or change the data connection, select the three dots next to the data connection, and then select **Edit**.

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

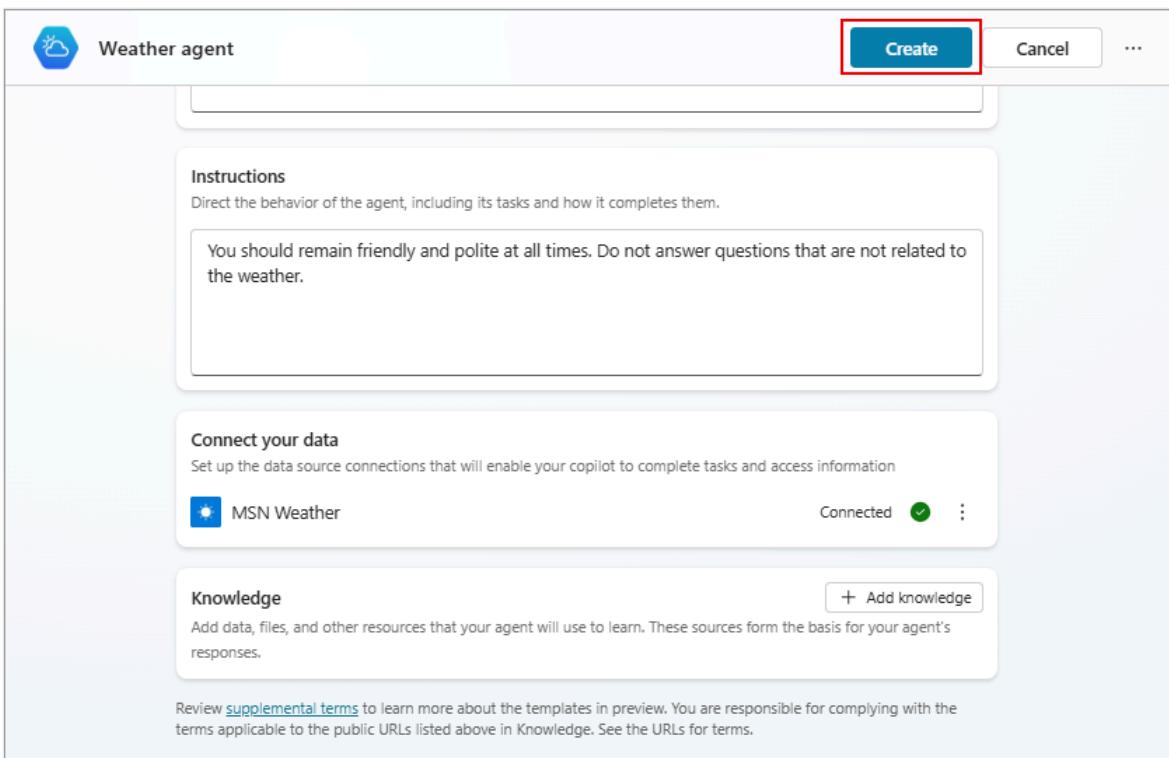
 MSN Weather [Set up connection](#) 

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

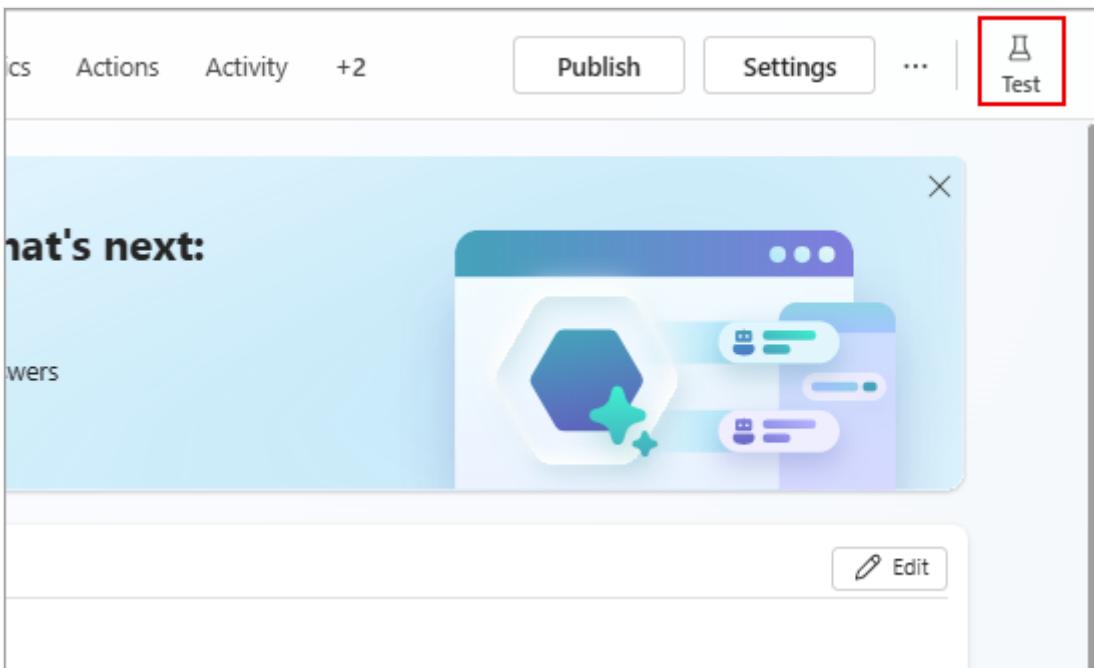
 [Edit](#)

 [Add knowledge](#)  [Delete](#)

- b. 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.



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.



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

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Team Navigator Agent

Article • 11/19/2024

Publisher: Microsoft

Team Navigator Agent is a helpful tool that assists employees in navigating their company's organizational structure within Teams. Team Navigator Agent uses the power of Office 365. Employees can quickly access contact information and reporting lines to find more data by searching for a colleague's name or email. Whether collaborating on a project, seeking feedback, or seeking guidance, Team Navigator Agent streamlines the process of finding the right people within the organization.

ⓘ 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 Team Navigator Agent:

[https://learn-video.azurefd.net/vod/player?id=be65eebc-1b86-4548-9232-d790e6cd7219&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-org-navigator ↗](https://learn-video.azurefd.net/vod/player?id=be65eebc-1b86-4548-9232-d790e6cd7219&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-org-navigator)

Prerequisites

- Microsoft Teams account (preferred platform)
- Microsoft 365 Business or Enterprise plan with Teams
- Meant for companies who are using Outlook Online for their employees

Capabilities

- Employees can ask the agent for a coworker's reporting line using their name or email.
- Employees can ask the agent for details on a colleague using their name or email. The agent returns given and family name, job title, email address, a phone number if there's one, and their manager.

- Employees can get clickable hyperlinks of contact information (emails) to start an email.

Use cases

- A project manager got a recommendation from a colleague for a person to fill the open developer role on the team, but only has their name. They can use Team Navigator Agent to search by name and retrieve the email with a hyperlink.
- A new hire wants to learn more about their team members and their roles. They can use Team Navigator Agent to explore the hierarchy of their team and view the profiles of their colleagues. They can also see who reports to whom and how their team fits into the larger organization.
- A hiring manager needs to contact an employee's direct manager to set up a meeting. They can give Team Navigator Agent the employee's name or email to receive back their manager's name from their reporting line or from their employee's details.

Extension opportunities

- See trending documents recently worked on by your colleagues that you have access to.
- Empower your agent with a full suite of [Power Platform connectors](#) and calls to APIs.
- Add an [Office 365 Outlook connector](#) to see schedules and availabilities within the agent interface.
- Get the agent to understand your questions and context on its own by turning on the Generative (preview) AI feature. Enabling this feature allows your agent to use generative AI to identify the most appropriate combination of actions and topics to respond to a user.

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?

 Yes

 No

[Provide product feedback ↗](#)

Safe Travels Agent

Article • 11/19/2024

Publisher: Microsoft

Safe Travels Agent 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/> 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 with the 'Knowledge' tab selected. A red arrow points to the '+ Add knowledge' button. The table below lists a single item: 'US Travel Website' (Type: Public website), last modified 16 seconds ago, and marked as 'Ready'.

Name	Type	Last modified	Status
US Travel Website	Public website	16 seconds...	Ready

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 traveling, 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 ↗](#).

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

Store Operations Agent

Article • 11/19/2024

Publisher: Microsoft

Store Operations Agent improves the efficiency of retail frontline workers by enabling easy access to store procedures and policies. Agents built from Store Operations Agent can optionally integrate to other data sources, enabling capabilities such as looking up product inventory, checking shipping status, and helping initiate a return.

ⓘ 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 Store Operations Agent:

[https://learn-video.azurefd.net/vod/player?id=1b113a42-e1c9-4a3e-a07d-6d07e13a99e3&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-store-ops ↗](https://learn-video.azurefd.net/vod/player?id=1b113a42-e1c9-4a3e-a07d-6d07e13a99e3&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-store-ops)

Prerequisites

- Maker has a Copilot Studio Account
- Maker has a source of information on policies/procedures (internal documents, websites, and so on)
- Optional: Integration to inventory, shipping, and product return systems

Configuration and deployment

This agent is easy to configure and tailor for your organization's needs. For demonstration purposes, the agent comes preconfigured with two Microsoft knowledge sources—the *Microsoft Store & billing* help page and the *Microsoft Store* website—and with instructions to the generative AI to constrain responses to just Microsoft Store. Before deployment, a maker can change these URLs and/or upload documents to point to information on their store policies and procedures—while also updating the instructions to reflect their organization name.

1. Modify instructions and replace Microsoft with your organization's name.

The screenshot shows the Microsoft Copilot interface for 'Store Operations - cc'. At the top, there's a navigation bar with 'Store Operations - cc', 'Overview', 'Knowledge', 'Topics', 'Actions', 'Analytics', and 'Channels'. A prominent callout box says 'Your copilot is ready! Here's what's next:' with three steps: 'Add actions so your copilot can do things for you', 'Build topics to focus and guide how your copilot answers', and 'Publish your copilot so others can use it'. To the right is a preview window showing a hexagonal icon with a plus sign and some blurred interface elements. Below this is a 'Details' section with an 'Edit' button. It includes a placeholder icon for 'Name' (a shopping cart), the actual name 'Store Operations - cc', a 'Description' field ('Improve the efficiency of retail frontline workers by enabling easy access to store procedures and policies'), and an 'Instructions' field containing a template for responses. The 'Microsoft' brand name is highlighted with a red box in the instructions text.

2. Replace the agent's knowledge sources with pointers to your organization's information on policies and procedures.

Integration with other data sources

There are three integration stubs built into the agent:

- Order status
- Item inventory
- Initiating a return

Makers can use pre-existing connectors to common enterprise data sources and/or create their own integrations to light up these optional scenarios. Upon doing so, makers can modify the agent to map the data for these or other scenarios.

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:

- Provide retail frontline workers with questions on policies and procedures.

Sample prompts include queries like, "What is our return policy?", "Do we offer a discount to veterans?", and "Help me initiate a return."

- Share specific insights, and initiate actions tied to other systems and processes.

Sample prompts include queries like, "**What is the current inventory for Monitors?**", "**Help me track an order**", and "**Help me initiate a return.**"

- Provide general knowledge for Retail workers.

General knowledge might include queries for common acronyms and terms such as "**What is a SKU?**", "**What is an RMA?**", and others to help newer employees ramp up. If desired, a custom knowledge source could also be added to provide corporate specific terminology (department names, for example).

Example: Initiating a return

› Test your copilot

... ⚙️ ✖

Hello, I'm Store Operations - cc, a virtual assistant. I can answer questions on policies and procedures, and help you look up information such as product inventory. 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 [↗](#).

A minute ago

Help me initiate a return.

A minute ago

Are you trying to return a product?

A minute ago

Yes

A minute ago

TO DO: This is a placeholder for where you get the orders that can be returned from your store system. You should replace this node, in the "Initiate product return" Topic, with a call to a [Power Automate Flow](#) [↗](#) or submit to your own API using the [Http Request node](#) [↗](#)

Select an order to start return.

A minute ago

654321

987654

456789

234567

876543

Ask a question or describe what you need

0/2000



Select an order to start return.

3 minutes ago

987654

Just now

TO DO: This is a placeholder for where you need to fetch order details for order 987654 from your store system. You should replace this node, in the "Initiate product return" Topic, with a call to a [Power Automate Flow](#) or submit to your own API using the [Http Request node](#)

Select items you want to return

- Contoso Speaker
- Contoso Monitor

Submit

[Cancel](#)

Just now

Ask a question or describe what you need

0/2000



Select items you want to return

- Contoso Speaker
- Contoso Monitor

Submit

[Cancel](#)

TO DO: This is a placeholder for where you initiate return for the following products Ids: ["456783298"] in your store system. You should replace this node, in the "Initiate product return" Topic, with a call to a [Power Automate Flow](#) or submit to your own API using the [Http Request node](#)



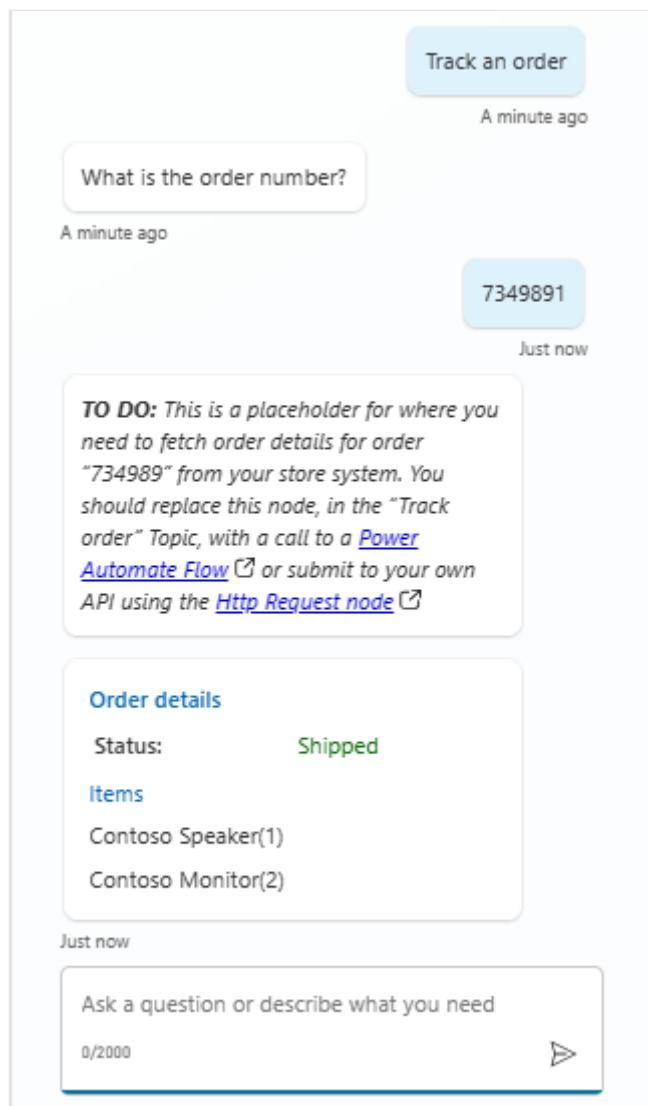
Just now

Ask a question or describe what you need

0/2000



Example: Tracking an order



Example: Starting a return

How do I start a return?

Just now

Are you trying to return a product?

Just now

No

Just now

To start a return, follow these steps:

1. Verify if your product is eligible for return by visiting the 'Order history' page in your Microsoft account.
2. If there is a 'Request a return' button next to the product, it is eligible.
3. Select 'Request a return' and follow the instructions to print out a prepaid shipping label.
4. Ship the product back in like-new condition in its original packaging with all parts and documentation included.
5. Allow a week for the return to be processed, and you will receive a refund to your original payment method within 3-5 business days [\[1\]](#).

1 reference ▾

1 Microsoft Store Refund & Return ... ↗

Just now | Surfaces with Azure OpenAI |

Ask a question or describe what you need

0/2000

This screenshot shows a Microsoft AI-generated support conversation. It begins with a question from the user: "How do I start a return?". The AI responds with "Just now". The user then asks "Are you trying to return a product?", to which the AI replies "No" (also marked as "Just now"). Below this, a large box provides step-by-step instructions for starting a return. The steps involve verifying product eligibility via the Microsoft account's order history, requesting a return, printing a shipping label, packaging the item, and allowing for a refund within 3-5 business days. A note indicates that a week is required for processing. At the bottom of the main content area, there is a reference section with one link to "Microsoft Store Refund & Return ...". Below the main content, there is a footer bar with the text "Ask a question or describe what you need" and a character limit of "0/2000".

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?

Yes

No

[Provide product feedback](#)

Sustainability Insights

Article • 11/27/2024

Publisher: Microsoft

The prebuilt *Sustainability Insights* agent helps you to easily get 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

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 Sustainability Insights:

[https://learn-video.azurefd.net/vod/player?id=e7cd9a72-d476-435f-9ecd-cc0b5cc836a4&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-sustainability-insights ↗](https://learn-video.azurefd.net/vod/player?id=e7cd9a72-d476-435f-9ecd-cc0b5cc836a4&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-sustainability-insights)

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 in the **Explore agents** section, 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 prebuilt agent 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

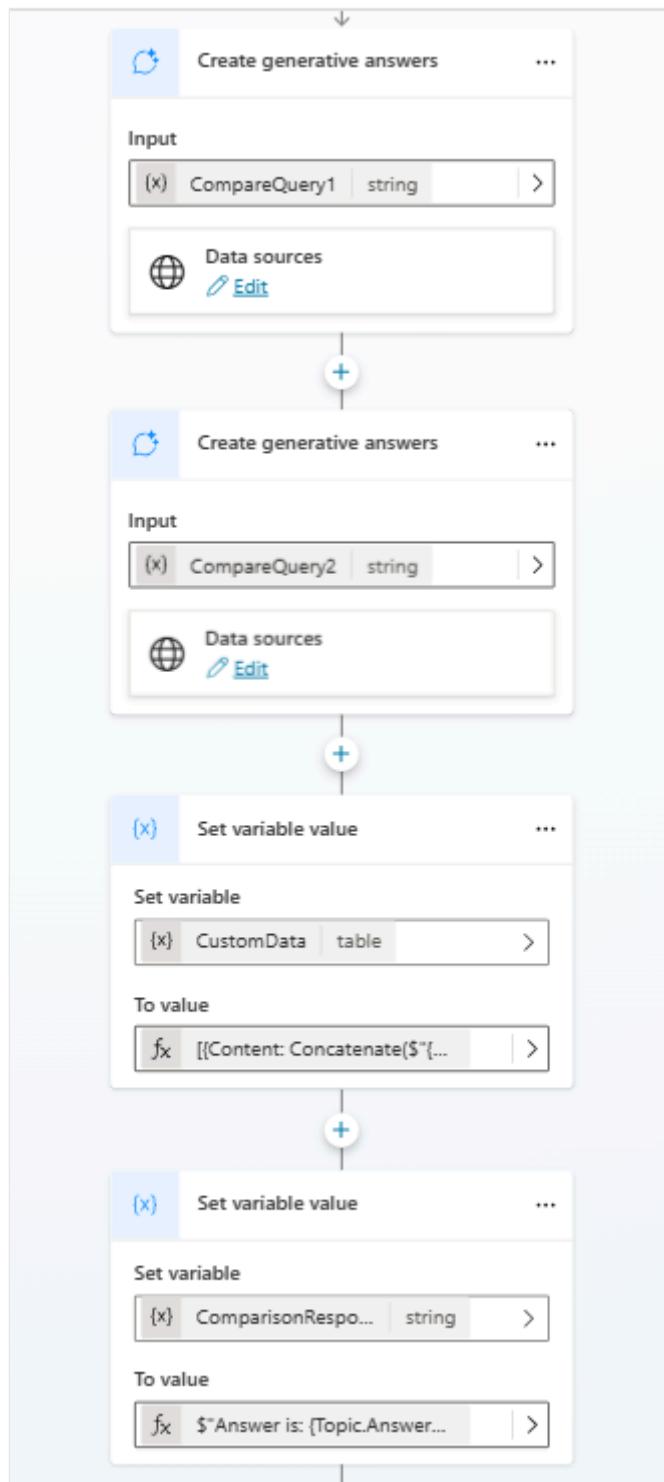
For best results, knowledge sources have overlapping data points and similar information to your organization's source so the agent can perform effective comparisons.

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. That said, it is important to note the ability to compare specific data points is only as good as the data sources provided. If an `OrganizationToCompare` does not list their `Scope 1` emissions for a particular year while `OrganizationName` does, then doing such a comparison will not 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 greenhouse gases that escape into the atmosphere during the production, processing, transportation, storage, and use of fossil fuels and other industrial activities. These emissions are unintended and can contribute to climate change [\[1 ↗\]](#).

1 reference ▾

1 [What Is Net-Zero Emissions? | Mic... ↗](#)

Just now | Surfaced with Azure OpenAI |

Ask a question or describe what you need

0/2000



Example: Inquiring about what Microsoft is doing about sustainability

What is Microsoft doing about sustainability?

Just now

Microsoft is undertaking sustainability efforts by advocating for carbon policies in areas such as carbon reporting, carbon reduction, and carbon removal. They aim to drive tangible progress, promote a flexible approach, and leverage digital technology to support global decarbonization efforts [1 ↗].

1 reference ▾

1 Microsoft carbon policy brief - 2022 ↗

Just now | Surfaces with Azure OpenAI |

Ask a question or describe what you need

0/2000

Example: Inquiring who is Microsoft's chief sustainability officer

Who is the chief sustainability officer at Microsoft?

Just now

The Chief Sustainability Officer at Microsoft is Melanie Nakagawa [1 ↗].

1 reference ▾

1 Water replenishment: Our learnin... ↗

Just now | Surfaces with Azure OpenAI |

Ask a question or describe what you need

0/2000

Extension opportunities

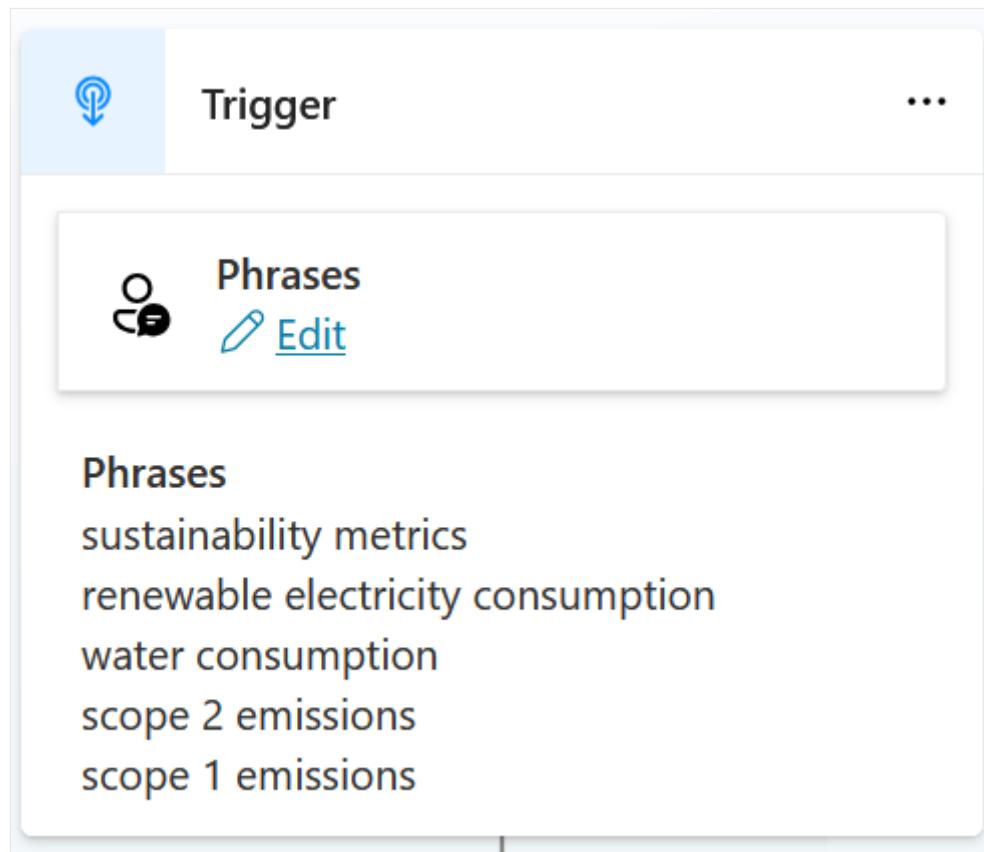
You can configure this agent with more sources of knowledge to answer related queries. You can also integrate the agent with or without systems of record, including using prebuilt connectors provided by the platform.

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 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

This screenshot shows a 'Text' step in a Power Automate flow. The 'Question' field contains the text: "Which sustainability KPI's data are you". Below it, the 'Identify' section shows a button labeled "User's entire response" with a grid icon. The 'Save user response as' section contains a placeholder "(x) Topic.KPIName".

?

Question

Text ...

+ Add (x) fx

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

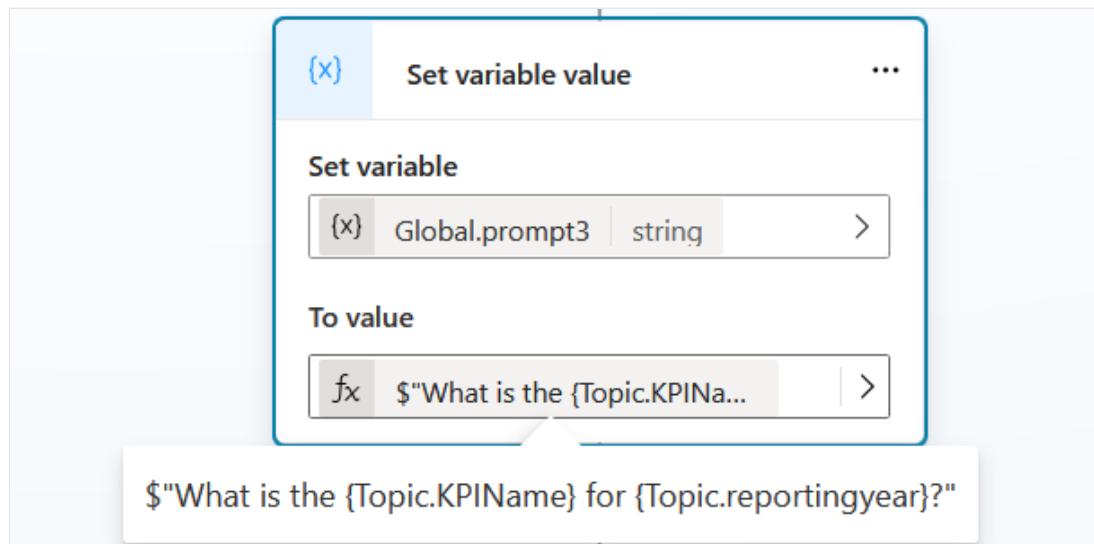
+ New option

Save user response as

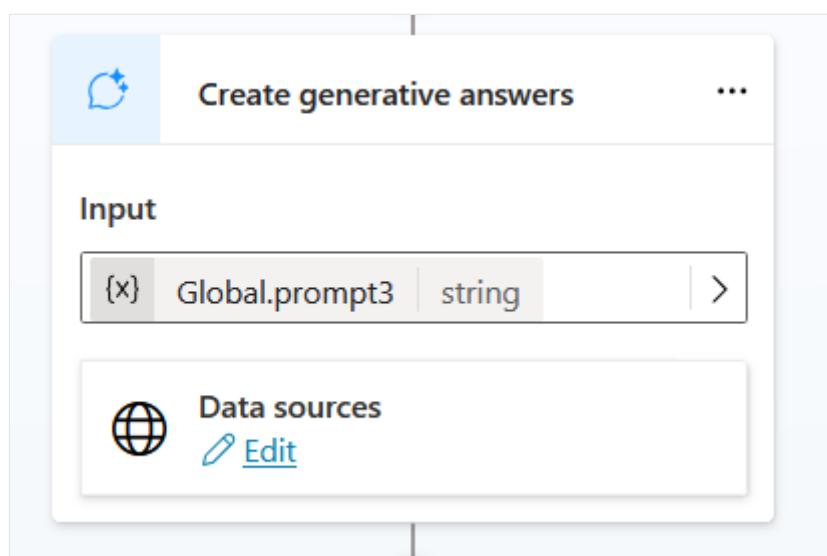
(x) reportingyear choice >

This screenshot shows a 'Multiple choice options' step in a Power Automate flow. The 'Question' field contains the text: "The value for the sustainability KPI will be provided for a specific reporting year. Please select a reporting year from the following options:-". Below it, the 'Identify' section shows a button labeled "Multiple choice options" with a grid icon. The 'Options for user' section lists "2023" and "2022" as options. The 'Save user response as' section contains a placeholder "(x) reportingyear choice".

3. Create a Power Fx formula to extract the key performance indicator (KPI) values for the specified reporting period.



4. Configure the [Create generative answers](#) node, and select the relevant knowledge sources. In the 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](#).

Feedback

Was this page helpful?

Yes

No

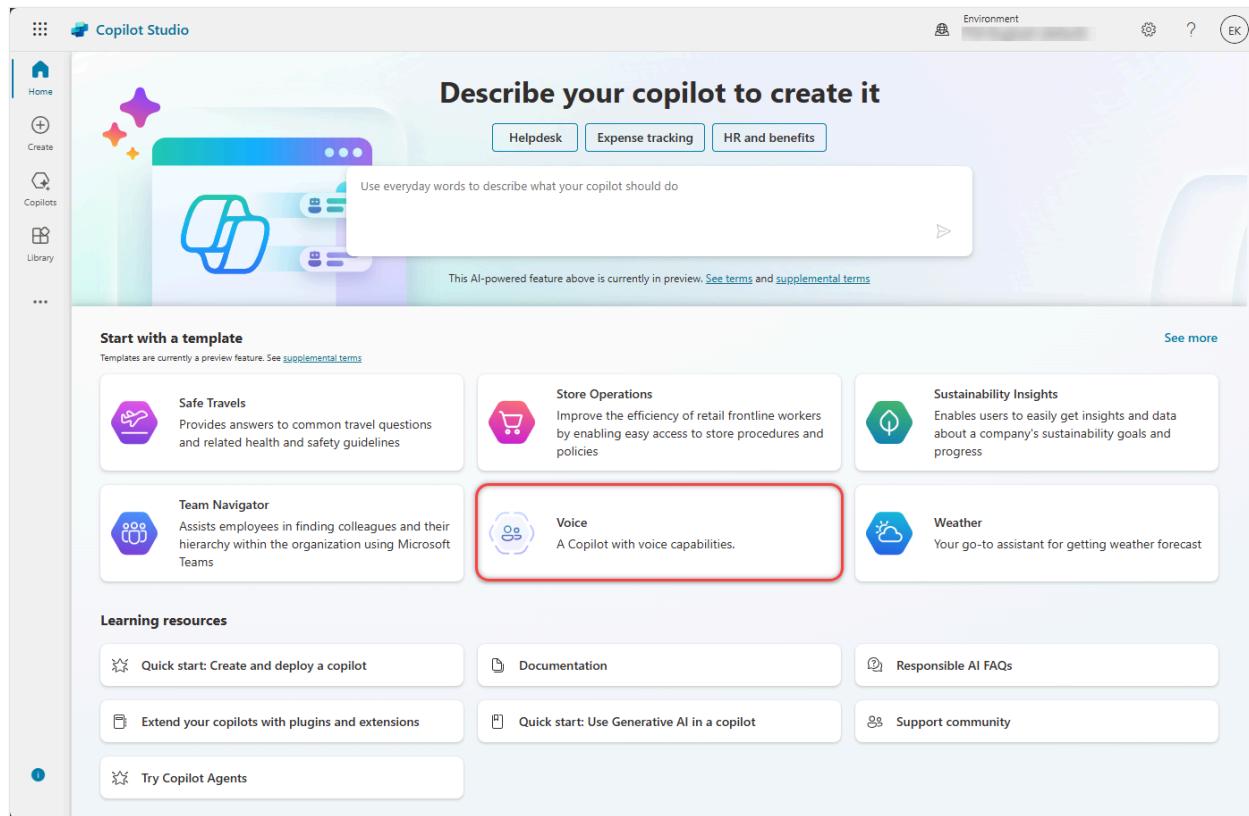
[Provide product feedback](#)

Voice Agent

Article • 11/19/2024

Voice Agent 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 agent natural language and choosing the options (touch-tone) from a menu to resolve the issues faster.

You can [create a new agent in Copilot Studio](#). From the **Home** or **Create** page, under **Start with a template**, select **Voice Agent**.



The primary building blocks of your voice-enabled agent are constructed. Select **Create** to continue with the agent.

To learn about regional settings, see [Regional settings and data locations for organizations using Copilot Studio](#).

Prerequisite

- Copilot Studio Account

Voice-enabled capabilities

When you select Voice Agent, an agent is created with several voice-related features included and enabled by default:

- Telephony channel is enabled by default.
- The Speech & DTMF (Dual Tone Multi Frequency) modality is enabled by default.
- The 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 correctly
- **DTMF-enabled multiple choice options** in the question node
- The **Test agent** 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 ↗](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Weather Agent

Article • 11/19/2024

Publisher: Microsoft

Weather Agent is the 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](#). Users can even 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 Weather Agent:

<https://learn-video.azurefd.net/vod/player?id=9245998b-afe8-4e75-a54c-247b0577dcd7&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 in the area, like temperature and conditions.
- 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](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Wellness Check Agent

Article • 11/19/2024

Publisher: Microsoft

Wellness Check Agent is a helpful tool to assist employees in Teams with advice and inspiration to add more healthy routines to their workday. By default, the agent references data from the Centers for Disease Control (CDC) and National Health Institute (NIH), and can be supplemented with other reference knowledge sources. Employees can ask questions or select prompts and let the agent introduce good habits and ideas to provide balance to each workday.

Employees can add these friendly personal check-in times to their weekly routines by opting in to status notifications, customized to the day and time of their choosing. The check-in supports sending messages on multiple days of the week at a specified time. In addition, the messages that are sent are varied and make recommendations to help keep wellness in check.

Whether looking for a breathing exercise to manage stress, a journaling prompt to gain insights, or just to take a moment for yourself, Wellness Check Agent provides the necessary tools to start feeling better.

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 Wellness Check Agent:

[https://learn-video.azurefd.net/vod/player?id=5c9d7a1c-8e32-422a-bf6c-9673a0c51672&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-wellness ↗](https://learn-video.azurefd.net/vod/player?id=5c9d7a1c-8e32-422a-bf6c-9673a0c51672&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-wellness)

Capabilities

- Users can ask the agent wellness related questions.
- Users can choose wellness advice prompts from agent to explore healthy habits.
- Users can get tips on how to write and achieve wellness goals.

- Users can opt in to receive routine invitations to perform a personal check-in.

Use cases

- An employee feels overwhelmed with their workload and asks the agent for a quick breathing exercise. The agent provides a guided 5-minute breathing session.
- An employee wants to add some healthy habits to their routine, but is unsure where to start. The agent provides multiple starter prompts to inspire the conversation.
- An employee has many meetings during their workday and it's starting to affect their health—physically and mentally. They schedule wellness checks for every Wednesday at noon to ensure they take a break and do some self-care.

Extension opportunities

- Connect custom knowledge sources to make data more relevant and robust.
- Get the agent to understand your questions and context on its own by turning on the generative AI feature. Enabling this feature allows your agent to use generative AI to identify the most appropriate combination of actions and topics to respond to a user.

For more information about custom knowledge sources, see [Knowledge in Copilot Studio](#).

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms](#).

Wellness Check Agent solution components

- One agent: Wellness Check Agent is the agent where users can:
 - Get healthy inspiration and advice from the knowledge sources.
 - Schedule weekly wellness prompts to get reminders.
- Two Power Automate flows:
 - *Wellness-Handle Opt In* allows users to manage their Wellness Check Agent reminder preferences.

- *Wellness-Send Daily Reminder* sends the reminder from Wellness Check Agent based on the user's opt-in settings.
- Two Dataverse tables:
 - *Wellness Opt In* stores a user's preference as to whether or not they want to be notified daily for a wellness check
 - *User* contains more relationships to the *Wellness Opt In* table
- One security role:
 - Wellness Agent User
- Two connection references:
 - Wellness-Dataverse-Connection reference to Dataverse
 - Wellness-Teams-Connection reference to Teams
- Two knowledge sources:
 - [U.S. Centers for Disease Control and Prevention](#) ↗
 - [National Center for Complementary and Integrative Health](#) ↗

Install and set up Wellness Check Agent

To install and set up Wellness Check Agent, follow the steps outlined in this article.

Step 1: Review prerequisites

Review the requirements, identify opportunities, and develop a plan of action that suits your organization's needs.

Confirm licensing

- Microsoft 365 Business or Enterprise plan with Teams
- Copilot Studio (free license that must be assigned to your developers)
- Copilot Studio message capacity
- Power Automate Premium or per flow plan

Set up environments and data policies

It's important to have a cohesive environment and data policy strategy for enterprise templates. To successfully install and manage Wellness Check Agent, confirm these resources and practices are in place.

- Administrators—Be sure admins with the necessary privileges are available to coordinate to assign licensing and create environments. You need:
 - A Microsoft Entra ID global or tenant user administrator to assign Copilot Studio and Power Automate licensing.
 - A Power Platform Administrator to create environments and assign Copilot Studio message capacity.
 - For more information, see [Microsoft Power Platform admin](#).
- Environments—Follow [application lifecycle management \(ALM\)](#) best practices in [creating and managing environments](#). Also consider:
 - A Power Platform environment is set up with a Dataverse database.
 - Environment maker security role privileges, at a minimum, are assigned to the user who installs the solutions in that environment.
 - For more information, see [Environments overview](#).
- Data loss prevention policies—Ensure the environment has access to the connectors used by Wellness Check Agent:
 - [Microsoft Dataverse connector](#).
 - [Microsoft Teams connector](#).
 - For more information, see [Data loss prevention policies](#).
- Allocate environment capacity—Wellness Check Agent requires Copilot Studio message capacity in the production environment where the agent is deployed.
 1. Go to [Power Platform admin center](#).
 2. Expand **Resources** on the left pane and select **Capacity**.
 3. Select **Add-ons**.
 4. Select **Assign to an environment**.
 5. Search for the environment where Wellness Check Agent solution is deployed.
 6. Allocate **Microsoft Copilot Studio messages** to the environment.

For more information, see [Copilot Studio messages and capacity management](#).

Step 2: Create connections

Cloud flows need specific connection references to work correctly. Connection references are included in the solution, but connectors often need to be manually set up.

The installation requires you to set up three new connections. It's best to create connections before importing the solution to make the import smoother. If you create the connections during import, you must jump between browser windows.

To create the connections in advance, take these steps:

1. Go to [Power Apps](#).
2. Go to **Connections** and select **+ New Connection** to create new connections with each of these connectors:
 - Microsoft Dataverse connector
 - Microsoft Teams connector

Step 3: Install solution files

There are two solution installation options available for you to consider:

You can select the **Get it now** button on the template's AppSource offer page, and it automatically takes you to an install window in the Power Platform admin center and installs managed versions of the template.

You can download either a managed or an unmanaged version of the solution files and manually import them into your environment according to your organization's requirements.

Install from AppSource

You can access and install Wellness Check Agent from AppSource, a Microsoft digital storefront. Take these steps to go through the AppSource install process:

1. Go to the [Wellness Check Agent in AppSource](#) and select **Get it now**, which takes you to the **Install Wellness Check Agent** window in the [Power Platform admin center](#).
2. Select the environment that you prepared for the template.
3. Agree to the *Terms and Privacy* statements by selecting the boxes.
4. Select **Install**. Once the installation is complete, the status shows as **Installed**.

Download and manually install solution files

You can also download either managed or unmanaged solution files to manually install into your prepared environment according to your organization's requirements. Skip this step if you previously installed from Microsoft AppSource:

1. Download the following managed or unmanaged solution files:
 - [Managed ↗](#)
 - [Unmanaged ↗](#)
2. Go to [Power Apps ↗](#) and select the environment you prepared for the solution.
3. Select **Solutions** on the left pane.
4. Select **Import solution**.
5. Browse for and select either the managed or unmanaged version of the mpa_Wellness file downloaded in the previous step.
6. Select **Next**.
7. Review the Details and confirm Enable Plugin steps and flows included in the solutions remains checked, and then select **Next**.
8. For each of the connections listed, select the existing connection created in the previous step. If you didn't create the connections in advance, create them now.
9. Select **Import** to start the import process of the Wellness Check Agent solution. Once the import is complete, you receive a notification with a Success message.

For more information, see [Solution concepts](#).

Step 4: Assign security roles

The solution includes one new security role that governs access to data stored in Dataverse:

- **Wellness Agent User**
 - An employee can create and update their Wellness Opt In preferences.
 - Assign this role to regular users who access Wellness Check Agent.
 - Wellness Agent Users can subscribe to receive regular agent reminders with helpful wellbeing tips.

Roles can be assigned from the Power Platform admin center.

1. Go to [Power Platform admin center ↗](#).

2. Select **Environment** on the left pane and then select the environment where the solution was installed.
3. Select **Users > See all**.
4. Select one or more users.
5. Select **Manage security roles**, and then select the *Wellness Agent User* role.
6. Select **Save**.

💡 Tip

For large organizations, create a Microsoft Entra security group containing all targeted users for the Wellness Agent and associate it to a new Microsoft Entra ID Security Dataverse Team created via the Power Platform admin center. Then assign the *Wellness Agent User* role to this Dataverse team.

💡 Tip

If your organization requires access to manage wellness opt-in notification settings on the user's behalf, create a new *Wellness Agent Admin User* security role and grant organizational privilege to the *Wellness Opt In* entity. Then create a simple model-driven Power App to support that administrator if you have access to Power Apps premium licensing.

Step 5: Ensure cloud flows are On

Open the newly installed Wellness solution and verify that the two cloud flows are set to the on status. If they aren't, turn them on.

1. Select **Solutions** on the left pane.
2. Select the Wellness solution from the list.
3. Select **Cloud flows (2)** on the left pane.
4. Select each of the two flows in the list and verify that each one is turned on.
5. Select **Turn on** in the command bar if the flow isn't already set to on.

Step 6: Share the AI Prompt

1. Go to Power Apps in the same environment where you installed the Wellness Check Agent solution previously.

2. Select **AI Hub** from the left navigation.

(!) Note

If AI Hub isn't visible in the left navigation, select **More** to view additional navigation options.

3. On the **Goal Writing Tips** AI Prompt, select the **More** button and then select **Share**.

4. Enter the individual users and security groups who have access to Wellness Check Agent.

5. Select **Share**.

Step 7: Publish to Teams

1. In Copilot Studio, select the same environment where you previously installed the Wellness Check Agent solution.

2. Select **Agents** to review the list of agents in this environment.

3. Open Wellness Check Agent.

4. Select **Publish** to publish the current agent configuration so it can be deployed to channels.

5. Select **Channels** and then select **Microsoft Teams**.

(!) Note

Wellness Check Agent is designed for a Microsoft Teams experience, and the default authentication has been configured to **Authenticate with Microsoft**.

For more information, see [Add user authentication to topics](#).

6. Select **Turn on Teams** to add the Microsoft Teams channel.

7. Select **Edit details**. Review the information and make any desired changes to the icon, descriptions, and developer information for Wellness Check Agent. When finished, select **Save**.

8. Select **Availability options**. Wellness Check Agent can be shared in several ways, including share via link, deploy to the Teams app store, or distribute as a custom app file. Choose your deployment method.
9. If deploying to the Teams app store, you must also decide how broadly to share Wellness Check Agent.
 - a. Select **Show to everyone in my org** to deploy Wellness Check Agent to everyone in your organization, which submits the agent in the Microsoft Teams admin center for approval by Global Administrators and Teams Administrators. After approval, it appears in the **Built for my org** area of the Teams app catalog. For more information, see [Manage your apps in the Microsoft Teams admin center](#).
 - b. Select **Show to my teammates and shared users** to deploy Wellness Check Agent to a more limited audience. Here, you can select both security groups and individual users. When finished with selections, select **Share** to save changes to the agent permissions.

For more information, see [Connect and configure an agent for Microsoft Teams](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Awards and Recognition Agent

Article • 11/19/2024

Publisher: Microsoft

Awards and Recognition Agent is an extension of the *Awards and Recognition Enterprise* template, and is designed to streamline the process of nominating and recognizing your employees for their contributions and achievements.

Employees can use the agent to select open nomination periods, such as the current quarter or calendar year, and browse different award categories tailored to their organization. The agent guides the employee through identifying one or more people and drafting a business case for how they exemplify the award, and captures award ratings assessing the behaviors and accomplishments for that award.

After awards are approved, employees can review a history of who they've nominated for an award and the awards they've received.

ⓘ 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 Awards and Recognition agent:

[https://learn-video.azurefd.net/vod/player?id=d98a28ff-b9da-4e62-9473-818bb8fa2c13&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-awards-recognition ↗](https://learn-video.azurefd.net/vod/player?id=d98a28ff-b9da-4e62-9473-818bb8fa2c13&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-awards-recognition)

Capabilities

- Users can create an award nomination for a single user, group of users, or team.
- Users can view recognitions that they submitted and received.
- Users can view help articles with definitions.

Use Cases

- An employee wants to submit a case to nominate a fellow developer for an award after they found a serious bug during development testing. They use the agent to submit a case for an *Above and Beyond* award set up for that quarter's nomination period.
- A project manager wants to communicate appreciation to the team for their extra hard work over a tough sprint. The project manager uses the agent to submit a case for nomination for a *Team Innovation* award for the associated fiscal year period.
- Two business analysts are essential to the process of completing a deliverable. The developers want to show their appreciation by using the agent to nominate the analysts for a *Best Collaborator* award for that month.
- A manager wants to see the nominations they submitted for their team to ensure the team is regularly motivated to output great work. The manager can ask the agent to display past nominations and review the response.
- An employee wants to track their growth since starting in their new position. They can review a history of nominations with the agent to see previous recognitions they received.

Limitations

AI-generated content can have mistakes, so don't forget to make sure it's accurate and appropriate. Review the [Supplemental Terms](#).

About Awards and Recognition Agent for Power Platform

Awards and Recognition Agent for Power Platform is available both as a Power App and as an agent that can be used to create a custom agent in Copilot Studio.

To learn more about Awards and Recognition Agent and see details about the different Power Platform components included, see the [Awards and Recognition Agent](#) overview article in the Power Platform documentation.

Next steps

To install Awards and Recognition Agent, see the [Install and set up Awards and Recognition Agent for Power Platform](#) article in the Power Platform documentation.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Inclusivity Agent

Article • 11/19/2024

Publisher: Microsoft

Inclusivity Agent is designed to foster a more inclusive and welcoming workplace environment. It provides employees with practical guidance and tools to ensure their interactions and communications are inclusive and respectful.

Employees can enhance their awareness and practice of inclusivity by integrating Inclusivity Agent into daily workflows. When employees focus on inclusive practices, they help to contribute to a more supportive and equitable workplace culture.

Prerequisite

Organization knowledge base containing inclusive guidance similar to:

- [Pronouns.org ↗ Resources on Personal Pronouns](#)
- [CDC Gateway to Health Communication ↗](#)

Links to both sites are provided in the agent.

Organizations can also enable the flag to bring responses from the web. This allows the AI to use its own general knowledge.

Installation

When using this agent, you're equipped with the default sites attached to the agent. If you want to add another data source or replace existing ones, 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 for the Inclusivity agent. The top navigation bar includes Home, Create, Agents, Library, and three dots. The main header has tabs for Overview, Knowledge (which is selected and highlighted with a red box), Topics, Actions, Activity, Analytics, and Channels. On the right, there are Publish, Settings, and Test buttons. The Knowledge tab displays a 'Suggestions (preview)' section with three cards: 'Add knowledge sources' (Review suggestions for references to add to your agent), 'Improve answer rate' (See unanswered questions, and tell your agent how to answer them using the fine tuning tool), and 'Track knowledge use' (Make sure your agent uses knowledge sources in at least 25% of responses). To the right, there are metrics: 'Total knowledge source use 0%', 'Error rate 0%', and 'Total answer rate 0%'. Below this is a search bar with 'All' and 'Public website' filters, and a message indicating it was last refreshed 1 minute ago. A search input field says 'Search knowledge'. A table lists knowledge sources: Pronouns (Public website, last modified 1 minute ago, Ready) and CDC Health-Gender equality guiding ... (Public website, last modified 1 minute ago, Ready).

Name	Type	Last modified	Status
Pronouns	Public website	1 minute ago	Ready
CDC Health-Gender equality guiding ...	Public website	1 minute ago	Ready

Use cases

- Inclusive Meeting Facilitation:** Inclusivity Agent provides a checklist and tips to ensure the meeting is inclusive. Examples include rotating speaking opportunities, using inclusive language, and ensuring accessibility for all participants.
- Content Proofreading for Inclusivity:** The agent proofreads the content, identifying and suggesting changes to any language that might be exclusive or insensitive, helping to create a more inclusive and respectful document.
- Training and Development:** The agent provides resources, examples, and interactive scenarios to help employees understand and practice inclusive behaviors in various workplace situations.

Extension opportunities

Policy and Document Review: The custom agent should be designed to review the documents to ensure they're written in an inclusive manner, suggesting revisions where necessary to align with the organization's commitment to diversity and inclusion.

The agent can be extended to use web resources and videos to showcase the inclusive behavior in action.

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?

 Yes

 No

[Provide product feedback](#)

Case Management Agent

Article • 11/19/2024

Publisher: Microsoft

Case Management Agent helps in the creation and management of cases, enhancing support efficiency for both the requestor and the fulfiller. Easily create new cases by providing information to the agent conversationally--the agent guides you through the process, requesting edits if necessary, to create the case details. After confirming everything looks good, the agent sends out your case and provides details.

ⓘ 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 *Case Management Agent*:

[https://learn-video.azurefd.net/vod/player?id=797b1eff-0c7d-4979-878a-511fdacc4a27&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-case-management ↗](https://learn-video.azurefd.net/vod/player?id=797b1eff-0c7d-4979-878a-511fdacc4a27&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-case-management)

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

ⓘ Note

Make sure the following plugins are installed in *ServiceNow*:

`sn_customerservice` (Customer Service) and `sn_km_api` (Knowledge API).

Integrate with ServiceNow during installation

The screenshot shows the Copilot Studio interface for creating a new agent named "Case Management Copilot".

- Icon:** A placeholder icon with a question mark inside a red circle. A "Change icon" button is available.
- Description:** A text input field containing: "Provide round-the-clock automated support to customers by understanding their issues and creating cases".
- Instructions:** A text input field containing: "This custom agent is your dedicated case manager, here to assist you in new case creation as well as retrieving and displaying details about ones you previously made. This agent will assist users with their case management. If a user does not include a question mark (?) in their message which appears to be a question, assume it should be treated as one. The term case (case) does not mean help (help). Help (help) is unrelated."
- Connect your data:** A section for connecting to ServiceNow. It shows a "ServiceNow" connection entry with a status of "Set up connection" and a warning icon. An "Edit" button is highlighted with a red box and an arrow pointing to it.
- Knowledge:** A section for adding knowledge resources. It includes a "+ Add knowledge" button and a "Delete" button.
- Footer:** A note about supplemental terms and URLs.

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

* Authentication Type ⓘ
Use OAuth2

* Instance Name ⓘ
The instance name used to identify the ServiceNow Site URI

* Client Id ⓘ
The ID used to identify this application with the service provider

* Client secret ⓘ
The shared secret used to authenticate this application with the service provider

Sign in

Confirm Close

Capabilities

Case Management Agent helps you create new cases, once permission is granted, by gathering information from the customer conversationally, and then conveying details back to the customer.

The customer can ask the agent a question and get helpful information with citations returned from the provided documentation from the company. If unsatisfied with the provided information, the agent helps in the creation of a new case.

Use Cases

A customer purchases a new washer but has difficulty with its operation. They open *Case Management Agent* to ask for assistance. The customer describes the problem. The agent guides the customer through troubleshooting steps, such as checking the power supply, water connections, and cycle settings, to resolve the issue.

The customer is unable to resolve the washer issue and needs to escalate the problem to technical support. *Case Management Agent* assists the customer in creating a new

support case by gathering details about the washer issue, such as error codes or specific symptoms. The agent then creates a new case and provides the customer with a summary of the case before submitting it to technical support for further assistance.

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](#)

IT Helpdesk Agent

Article • 11/19/2024

Publisher: Microsoft

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.

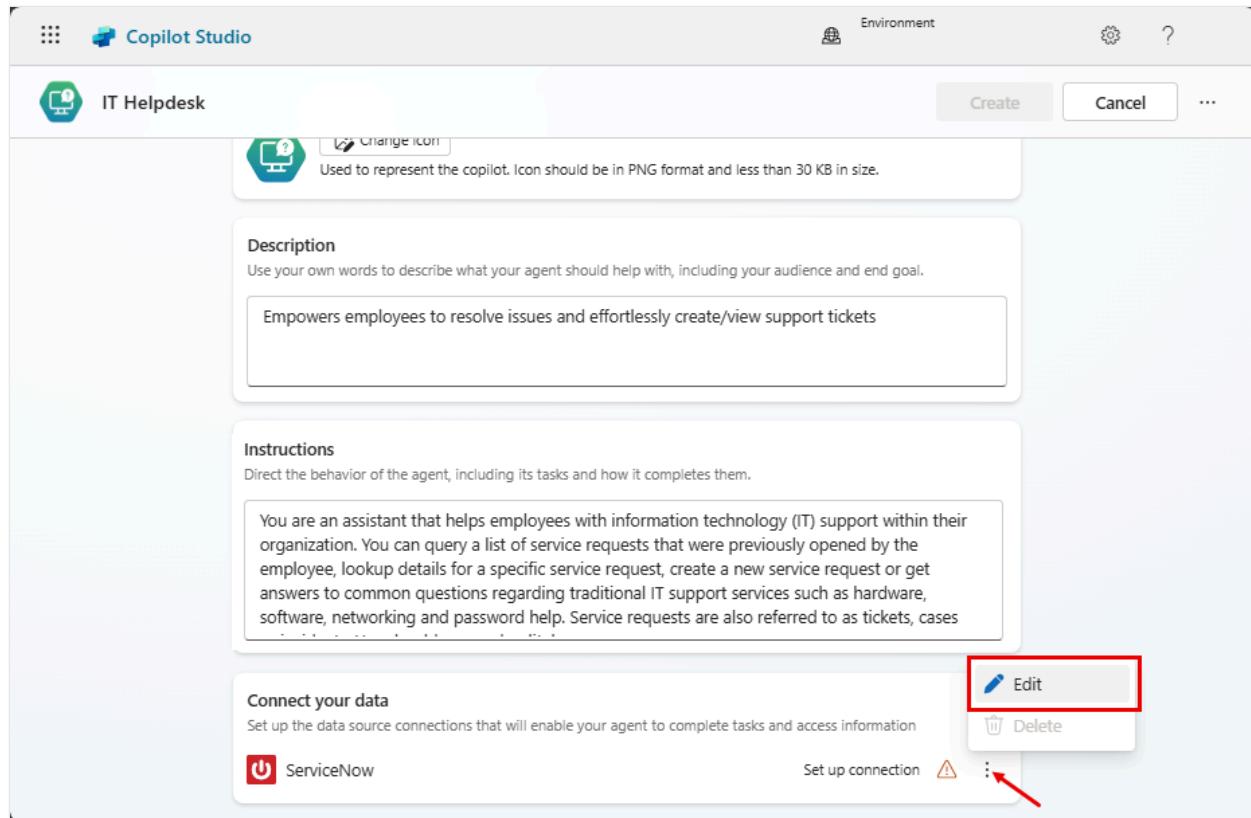
Watch this video to get an overview of how to use *IT Helpdesk Agent*:

<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



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

* Authentication Type ⓘ
Use Oauth2

* Instance Name ⓘ
The instance name used to identify the ServiceNow Site URL

* Client Id ⓘ
The ID used to identify this application with the service provider

* Client secret ⓘ
The shared secret used to authenticate this application with the service provider

Sign in

Confirm Close

Capabilities

Employees can use *IT Helpdesk Agent* 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.

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 topic. 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 Agent* 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

In cases where *IT Helpdesk Agent* is unable to provide a solution, it can escalate the conversation to a human support agent by [configuring an engagement hub](#). The agent can provide the agent 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 Agent 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 ↗](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Financial Insights Agent

Article • 12/04/2024

Publisher: Microsoft

Industry: Financial Services

Overview

Financial Insights Agent 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.

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.
- In order to take advantage of the *financial document analysis skill for agents (preview)* solution for your organization financial documents, we recommend you refer to [financial document analysis skill for agents \(preview\)](#).

 Note

Please 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, Financial Insights Agent includes an out of the box the SEC filings website: <https://www.sec.gov/>. This can be changed to 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**.

The screenshot shows the 'Topics' section of the Microsoft Copilot Studio interface. At the top, there are tabs for 'Overview', 'Knowledge', 'Topics' (which is highlighted with a red box), and 'Actions'. Below the tabs, there's a button '+ Add a topic' and three filter buttons: 'All' (highlighted with a red box), 'Custom (8)', and 'System (6)'. The main area displays a table with columns 'Name' and 'Trigger'. The rows show the following data:

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**.

- Turn on **Search only selected sources**.
- Select the financial news knowledge source you added.
- Select **Save**.

3. Go to the *Check regulatory filings* node and select **Edit**.

- Turn on **Search only selected sources**.
- Select the regulatory filings knowledge source.
- 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](#).

- 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.

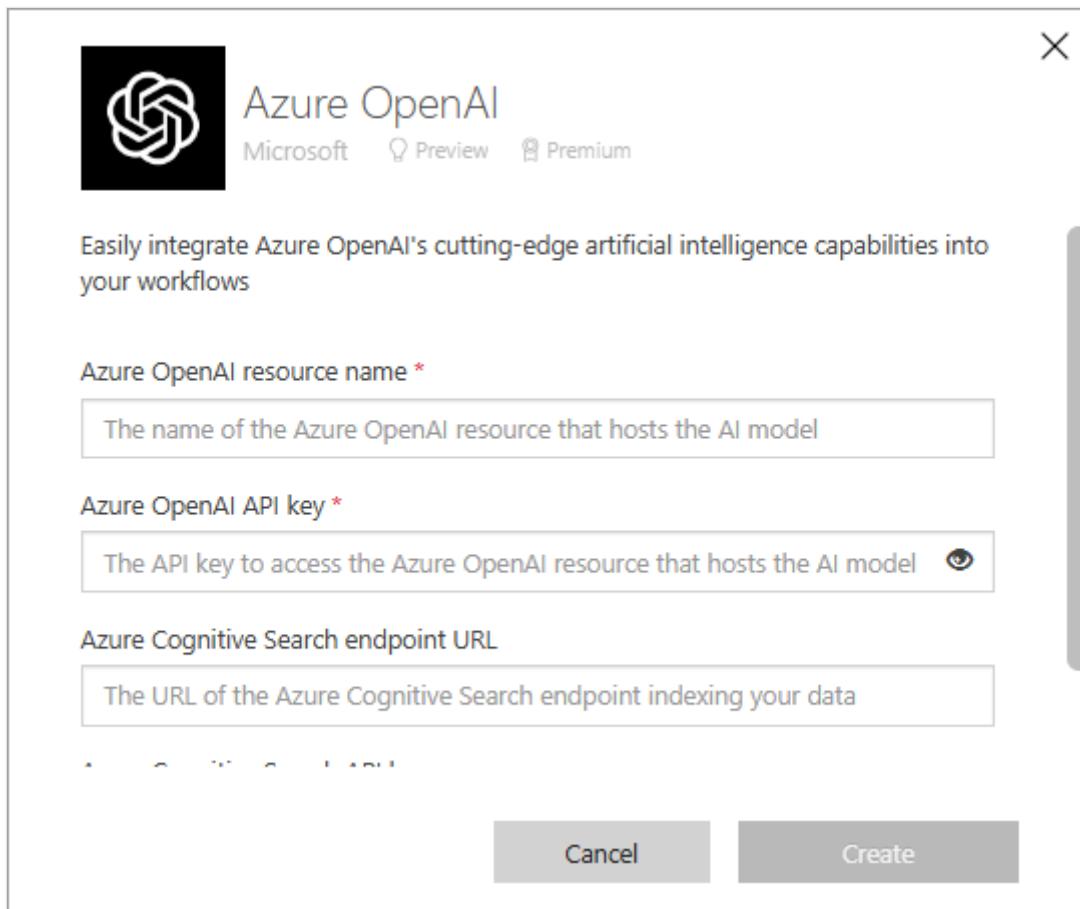
The screenshot shows the Power Apps canvas with a flow step titled "Triggered by agent (preview)". Below it, a "Set variable value" step is shown. A modal window titled "Select a connection" is open, displaying available connections: "Classic data", "Data sources", "SharePoint", and "Azure OpenAI Services on your data". The "Manage connections in Power Apps" button is highlighted with a red box. The "Azure OpenAI Services on your data" section shows an "Add connection" button, which is also highlighted with a red box.

A new browser tab opens, showing **Connections** in Power Apps (might require sign-in).

- Select **Create a connection**.
- In the search field at the top of the page, type **OpenAI**, and then select **Azure OpenAI (preview)**.

The screenshot shows the "Connections > New connection" page in Power Apps. The "Azure OpenAI (preview)" connection is selected and highlighted with a red box. The "Name" field contains "Azure OpenAI (preview)" and the "Type" field is "Standard". The "Microsoft Premium" badge is visible next to the connection name.

c. Complete the connection properties for your Azure OpenAI Deployment.



- 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 the above resource for Endpoint URL.
- v. Select **Create**.

You can now switch back to Copilot Studio and select the Azure OpenAI Connection you added.

The screenshot shows the Power Apps Studio interface. On the left, a modal dialog titled "Select a connection" is open, listing "Azure OpenAI" as a connection option. 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 and a "Manual input" dropdown), and "SharePoint" (with a text input field and a "Manual input" dropdown). Below these sections, there is a section titled "Azure OpenAI Services on your data" with a note about adding a connection from Azure OpenAI as a data source, and a "Add connection" button.

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 resource. It includes sections for Deployment, Api version, Maximum tokens in response, Temperature, Top P, and Stop sequence. Each section has an input field and a 'Manual input' dropdown. The Stop sequence section also includes a note about adding items in bulk.

4. On the **Model Data** Tab, under **Data sources**, select **Add**.

- a. Enter the index name of the Azure AI Search created by your financial document analysis skill for agents, Azure Marketplace deployment.
- b. Enter `simple` for the Query Type.
- c. 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.

Feedback

Was this page helpful?

 Yes

 No

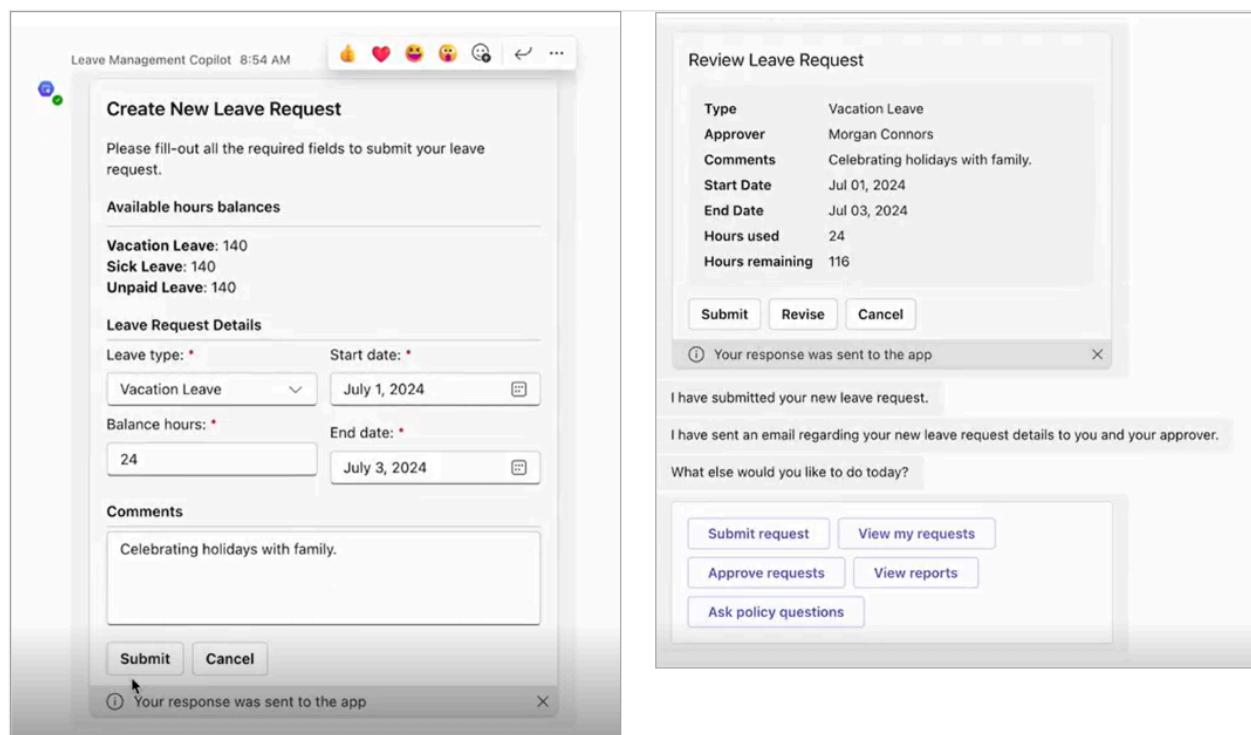
[Provide product feedback ↗](#)

Leave Management Agent

Article • 11/19/2024

Publisher: Microsoft

Leave Management Agent is designed to streamline the process of managing and requesting time off for employees. This agent simplifies the submission of absence requests, provides a summary of leave history, and offers quick access to the company's time-off policies using AI assistance. The agent reduces the administrative burden on HR and managers, ensuring accurate and timely leave management. This agent gives employees a seamless, user-friendly experience. Leave Management significantly improves workflow, enhances employee satisfaction, and promotes effective time-off management within the organization.



ⓘ 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 Leave Management Agent:

[https://learn-video.azurefd.net/vod/player?id=4f7fdf59-b0f8-46c1-adea-c0af83a7a64e&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-leave-management ↗](https://learn-video.azurefd.net/vod/player?id=4f7fdf59-b0f8-46c1-adea-c0af83a7a64e&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-leave-management)

Capabilities

- Initiate the leave request process, view available leave balances, choose absence type and duration, review a summary, and submit the request to a manager.
- Display and revise submitted requests, with statuses, accrued hours, and manager notes.
- Navigate through knowledge bases and get generative answers with citations.

Use cases

- Employees can use the agent to submit requests for vacation, sick leave, or personal days.
- Employees can quickly check their available leave balances for different types of leave without needing to contact HR.
- Employees can view the status of their submitted leave requests, make any necessary revisions, and see manager notes.
- Employees can view a summary of their leave history, including past requests and approvals.
- Employees can ask the agent questions about the company's time-off policies. The agent navigates through connected knowledge bases and provides generative answers with citations.
- Managers can use the agent to review incoming leave requests, see the employee's leave balance, and approve or deny requests.

Extension opportunities

- Add your organization's policy documents or knowledge bases as a knowledge source to enable the agent to answer user questions about your specific leave policies.
- Extend the data model with more leave types to cover other user needs in line with your organization's benefits.
- Connect to your HR systems of record to synchronize leave balances and approved employee leaves.

Assumptions

As the leave criteria and validation differ for each company, we used some assumptions for the agent to keep it generic, however the agent can be fully customized.

- The agent allows employees to submit leaves for past dates.
- Cancellation of approved leaves isn't available.
- Validation on number of leave hours isn't available in the agent.

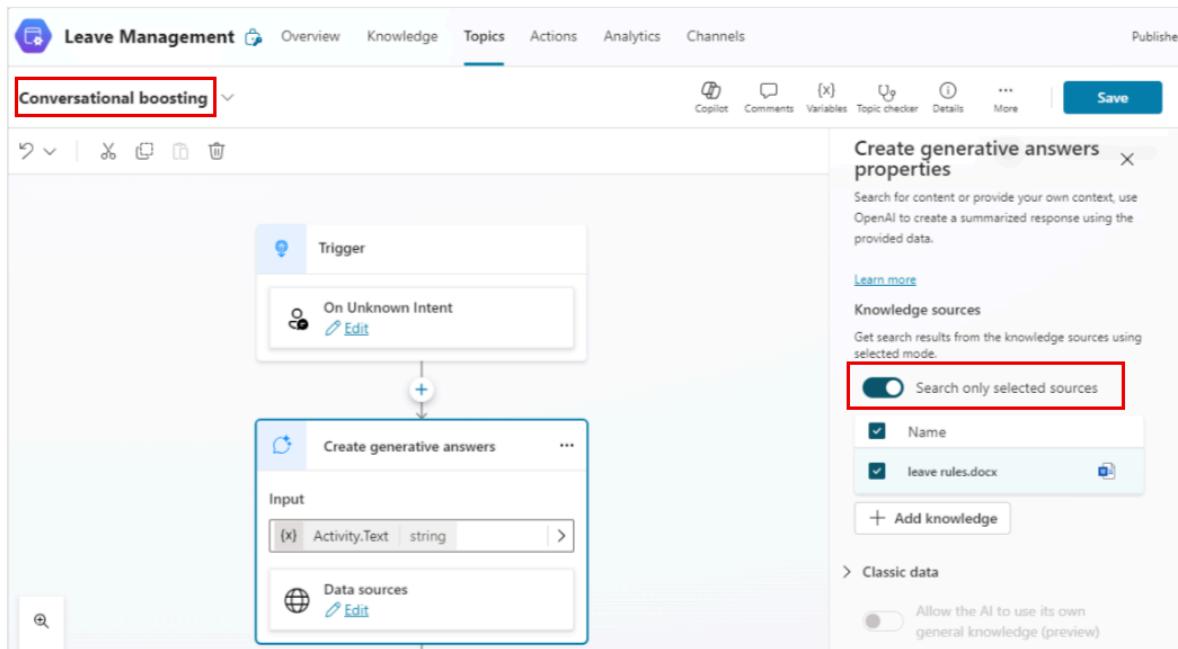
Leave Management solution components

- One agent: Leave Management is the agent where users can:
 - Easily submit and manage time-off requests, ensuring seamless leave-related communication between employees and managers.
 - Access and review leave balances, history, and company policies, providing a comprehensive overview of available leave options.
 - Navigate through company leave policies using AI assistance, obtaining cited information effortlessly.
- One model driven app. Leave Management Administration App is the app where users can:
 - Configure employee leave hour balances.
 - Review and manage employee leave requests.
- One Power Automate flow:
 - Leave Management – Execute Command carries out the user's commands triggered in the agent, including submitting, updating, canceling, and approving leave requests.
- Four connection references:
 - Leave Management - Dataverse - Connection reference to Dataverse
 - Leave Management - Microsoft Teams - Connection reference to Teams
 - Leave Management - Office 365 Outlook - Connection reference to Outlook
 - Leave Management - Office 365 Users - Connection reference to Office 365
- One knowledge source:

How to use documents as knowledge sources:

- Upload a knowledge source file containing instructions on the leave policy and FAQs. Provide detailed description of the purpose of the document.

- Under the system topic *Conversational boosting*, modify the Generative AI data sources to search only the selected sources and choose the uploaded document file.



- Two security roles:
 - Leave Management - Admin
 - Leave Management - User

Install and set up Leave Management Agent

Follow these steps to install and set up Leave Management Agent.

[Step 1: Review prerequisites](#)

[Step 2: Create connections](#)

[Step 3: Install solution files](#)

[Step 4: Assign security roles](#)

[Step 5: Ensure cloud flows are On](#)

[Step 6: Set up the program](#)

[Step 7: Publish to Teams](#)

Step 1: Review prerequisites

Review the requirements, identify opportunities, and develop a plan of action that suits your organization's needs.

Confirm licensing

One of the following Power Apps plans:

- Power Apps Premium
 - Power Apps per app
 - Power Apps per app pay-as-you-go-meter
 - Power Apps use rights included with Dynamics 365 licenses
- Microsoft 365 Business or Enterprise plan with Teams
- Copilot Studio (for your makers and is free)
- Copilot Studio message capacity

Set up environments and data policies

It's important to have a cohesive environment and data policy strategy for enterprise agents. To successfully install and manage Leave Management Agent, confirm these resources and practices are in place.

- **Administrators** - Be sure admins with the necessary privileges are available to coordinate to assign licensing and create environments. You need:
 - A Microsoft Entra ID Global or tenant User Administrator to assign Power Apps, Copilot Studio or Power Automate licensing.
 - A Power Platform Administrator to create environments.

More information: [Microsoft Power Platform admin](#)

- **Environments** - Follow application lifecycle management (ALM) best practices in creating and managing environments. Also consider:
 - A Power Platform environment is set up with a Dataverse database
 - Environment maker security role privileges, at a minimum, are assigned to the user who installs the solutions in that environment.

More information: [Environments overview](#)

- **Data loss prevention policies** - Ensure the environment has access to the connectors used by Leave Management Agent:
 - Microsoft Dataverse connector
 - Microsoft Teams connector
 - Office 365 Outlook connector
 - Office 365 Users connector

More information: [Data loss prevention policies](#)

- **Allocate Environment Capacity** - Leave Management Agent requires Copilot Studio message capacity in the production environment where the agent will be deployed.
 1. Go to the Power Platform admin center.
 2. Expand **Resources** on the left pane and select **Capacity**.
 3. Select **Add-ons**.
 4. Select **Assign to an environment**.
 5. Search for the environment where Leave Management Agent solution will be deployed.
 6. Allocate Copilot Studio messages to the environment.

More information: [Copilot Studio messages and capacity management](#)

Step 2: Create connections

Cloud flows need specific connection references to work correctly. Connection references are included in the solution, but connectors often need to be manually set up.

The installation requires you to set up three new connections. It's best to create connections before importing the solution to make the import smoother. If you create the connections during import, you must jump between browser windows.

To create the connections in advance, take these steps:

1. Go to **Power Apps**.
2. Go to **Connections** and select **+ New Connection** to create new connections with each of these connectors:
 - Microsoft Dataverse connector
 - Microsoft Teams connector
 - Office 365 Outlook connector
 - Office 365 Users connector

Step 3: Install solution files

There are two solution installation options available for you to consider:

- You can select the **Get it now** button on the agent's AppSource offer page and it automatically takes you to an install window in the [Power Platform admin center](#) and will install managed versions of the agent.
- You can download either a managed or an unmanaged version of the solution files and manually import them into your environment according to your organization's requirements.

 **Tip**

Go to the *Templates-for-Power-Platform* GitHub project site to log your questions or issues and get support for Leave Management Agent.

Install from AppSource

You can easily access and install Leave Management Agent from AppSource, a Microsoft digital storefront. Take these steps to go through the AppSource install process:

1. Go to [Leave Management Agent in AppSource](#) and select **Get it now**. This takes you to the **Install Leave Management Template** window in the [Power Platform admin center](#).

Install Leave Management



Name
Leave Management

Description
The Leave Management Copilot is a copilot template designed to streamline the process of managing and requesting time off for employees by simplifying the submission of absence requests, providing a summary of leave history, and offering quick access...
[See more](#)

Publisher
Microsoft Power Platform

Select an environment *

Type or select environment

[Don't see your environment?](#)

1. Select the environment that you prepared for the agent.
2. Agree to the *Terms and Privacy* statements by selecting the boxes.
3. Select **Install**. You're taken to a screen where you can view the installation status.
Once the installation is complete, the status shows as Installed.

Download and manually install solution files

You also can download managed or unmanaged solution files to manually install into your prepared environment according to your organization's requirements:

1. Download the following managed or unmanaged solution files:
 - **Managed:** mpa_LeaveManagement_managed.zip
 - **Unmanaged:** mpa_LeaveManagement_unmanaged.zip
2. Go to Power Apps and select the environment you prepared for the solution.
3. Select **Solutions** on the left pane.
4. Select **Import solution**.
5. Browse for and select either the managed or unmanaged version of the *mpa_LeaveManagement* file downloaded in previous step.
6. Select **Next**.
7. Review Details and confirm Enable Plugin steps and flows included in the solutions remains checked, and then select **Next**.
8. For each of the connections listed, select the existing connection created in the previous step. If you didn't create the connections in advance, create them now.
9. Select **Import** to start the import process of the Leave Management solution. Once the import is complete, you're notified with a Success message.

More information: [Solution concepts](#)

Step 4: Assign security roles

The solution includes two new security roles:

- Leave Management - Admin
 - A program admin can see all leave records.

- Users in this role act as the administrator of the Leave Management program. This security role grants access to the model-driven app where an admin can view all user hours balance and leave request records, and delete and deactivate them.
 - Assign this role to users who manage the Leave Management program in your organization.
- Leave Management - User
 - An employee can create and update their leave requests.
 - Assign this role to regular users who access Leave Management Agent.
 - Employees who are also managers can view and approve leave requests for their direct reports.

Roles can be assigned from the Power Platform admin center.

1. Go to the [Power Platform admin center](#).
2. Select **Environment** on the left pane and then select the environment where the solution was installed.
3. Select **Users > See all**.
4. Select one or more users.
5. Select **Manage security roles**, and then select the appropriate roles.
6. Select **Save**.

 **Tip**

For large organizations, create an Microsoft Entra security group containing all targeted users for Leave Management Agent and associate it to a new [**Microsoft Entra ID Security Dataverse Team**](#) created via the [**Power Platform admin center**](#). Then assign the Leave Management - User role to this Dataverse team.

Step 5: Ensure cloud flows are On

Open the newly installed Leave Management solution and verify that the four cloud flows are set to the on status. If they aren't, turn them on.

1. Select **Solutions** on the left pane.
2. Select the Leave Management solution from the list.
3. Select **Cloud flows (1)** on the left pane.
4. Select each of the four flows in the list to verify that each one is turned on.
5. Select **Turn on** in the command bar if the flow isn't already set to on.

Step 6: Set up the program

Once the agent admin configurations are complete, the next step is to set up the leave balance details for employees before you're ready to share the agent with people in your organization. Your program managers have access to the administration app when you complete the previous step to assign security roles.

The Leave Management program manager needs to configure employee leave balance hours for the employees who use the agent. Take these steps:

1. Go to [Power Apps](#) and select the environment you installed the solution in.
2. Select **Apps** on the left pane, then select **Play on the Leave Management Administration App** from the list.
3. Use the left pane navigation to browse the different Leave Management program data.
4. Create Hours Balance records for employees as needed with the number of vacation, sick and unpaid leave hours for each.

Step 7: Publish to Teams

1. Go to [Copilot Studio](#) in the same environment where you installed Leave Management Agent solution previously.
2. Select **Agents** to review the list of agents in this environment.
3. Open **Leave Management Agent**.
4. Select **Publish** to publish the current agent configuration so it can be deployed to channels.
5. Select **Channels > Microsoft Teams**.

 **Note**

Leave Management Agent is designed for a Microsoft Teams experience and the default authentication has been configured to Authenticate with Microsoft. More information: [Add user authentication to topics](#)

6. Select **Turn on Teams** to add the Microsoft Teams channel.
7. Select **Edit details**. Review the information and make any desired changes to the icon, descriptions, and developer information for Leave Management Agent. When finished, select **Save**.
8. Select **Availability options**.

The Leave Management Agent can be shared in several ways:

- Via link
- Deploying to the Teams app store
- Distributing as a custom app file

Choose your deployment method.

9. If deploying to the Teams app store, you must also decide how broadly to share Leave Management Agent.
 - a. Select **Show to everyone in my org** to deploy Leave Management Agent to everyone in your organization. This submits the agent in the Microsoft Teams admin center for approval by Global Administrators and Teams Administrators. After approval, it will appear in the **Built for my org** area of the Teams app catalog.
More information: [Manage your apps in the Microsoft Teams admin center](#)
 - b. Select **Show to my teammates** and shared users to deploy Leave Management Agent to a more limited audience. Here, you can select both security groups and individual users. When finished with selections, select **Share** to save changes to the agent permissions.
More information: [Add an agent to Microsoft Teams](#)

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?

 Yes

 No

[Provide product feedback](#)

Citizen Services Agent

Article • 11/19/2024

Publisher: Microsoft

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

- Prebuilt agents 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 'Knowledge' tab of the Citizen Services Agent in Copilot Studio. At the top, there are navigation links: Overview, Knowledge (which is selected and highlighted in blue), Topics, Actions, Activity, +2, Publish, Settings, ..., and Test. A large 'Add knowledge' button is prominently displayed. Below this, there are three cards: 'Suggestions (preview)', 'Improve answer rate', and 'Track knowledge use'. The 'Suggestions (preview)' card has a 'View suggestions' button. The 'Improve answer rate' card has an 'Open tool' button. The 'Track knowledge use' card has an 'Open Analytics' button. To the right, there are performance metrics: 'Total knowledge source use 0%', 'Error rate 0%', and 'Total answer rate 0%'. Below these, there are filters for 'All' and 'Public website', a refresh indicator ('Last refreshed 9 minutes ago'), and a search bar. A table lists knowledge sources with columns: Name, Type, Last modified, and Status. Two entries are shown: one for a 'Public website' last modified 10 minutes ago and another for a 'Public website' last modified 10 minutes ago, both marked as 'Ready'.

Name	Type	Last modified	Status
[REDACTED]	Public website	10 minutes ago	Ready
[REDACTED]	Public website	10 minutes ago	Ready

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' (with a dropdown arrow), Copilot, Comments, Variables, Topic checker, Details, More, and several icons for saving and sharing. Below the navigation is a toolbar with icons for back, forward, search, and delete.

The main area displays a flow editor. A 'Trigger' node is at the top, followed by a horizontal line with a plus sign indicating where to add more steps. Below the trigger is an 'HTTP Request' node.

Trigger Node:

- Icon:** A blue gear icon.
- Name:** Trigger
- Phrases:** Are there any road closures, Road alerts, Are there any highway closures or maintena, Find alerts for roads, Find alerts for highways
- Edit:** A blue link.

HTTP Request Node:

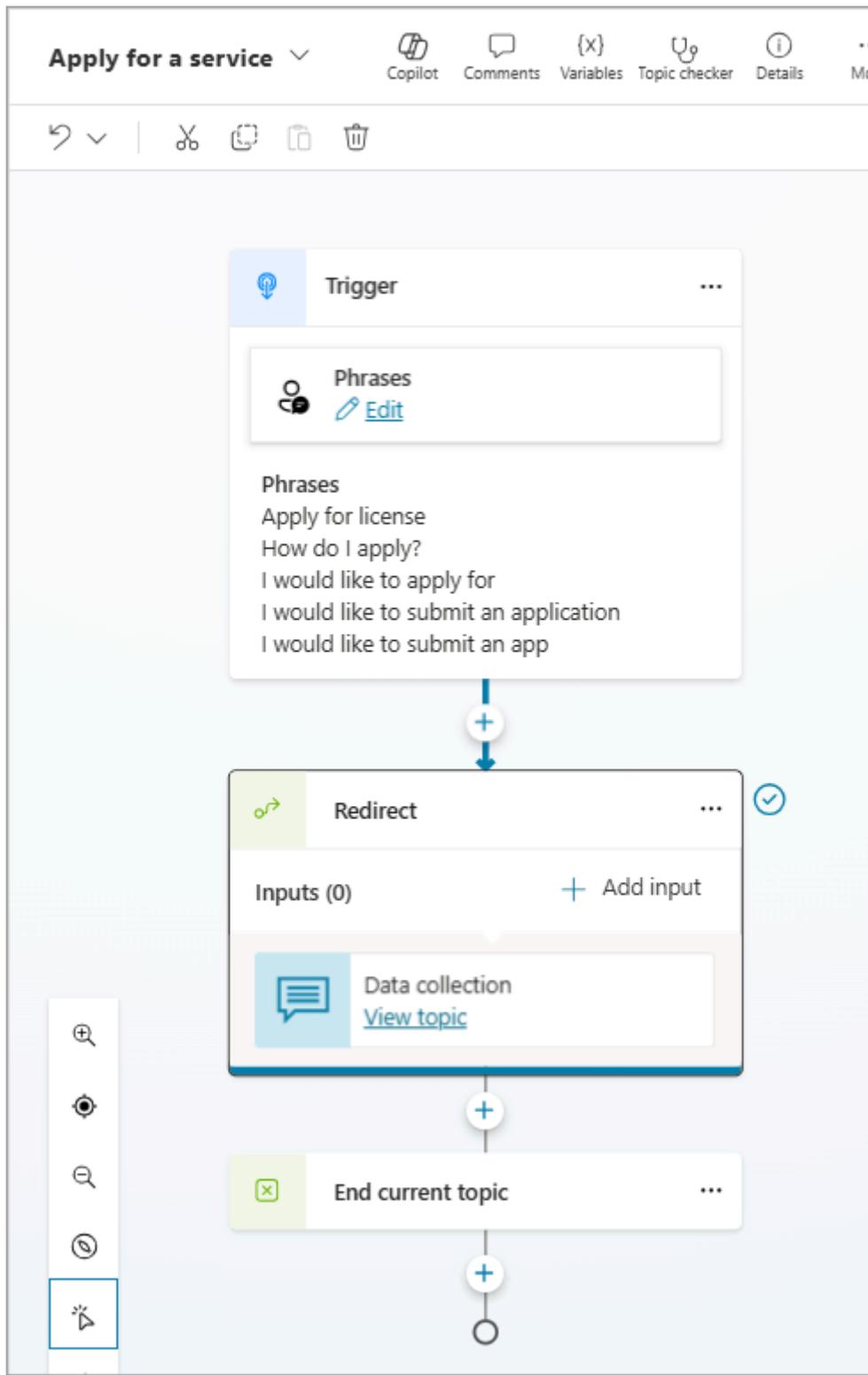
- Icon:** A globe icon.
- Name:** HTTP Request
- URL ***: A red asterisk indicates it's required. The URL field is blacked out.
- Method:** Get
- Headers and body:** An 'Edit' button.
- Response data type:** String

To the left of the nodes is a vertical sidebar with icons for search, filter, refresh, and a play button.

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 Citizen Services Agent can address depends 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 ↗](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Self-Help Agent

Article • 11/19/2024

Publisher: Microsoft

Enhance customer service workflows and optimize operations with *Self-Help Agent*. This solution retrieves active complaints from your customer relationship management (CRM), suggests next steps aligned with company policies, drafts email responses, and updates record statuses.

ⓘ Note

This agent currently integrates exclusively with *ServiceNow*.

Prerequisites

- Microsoft Teams account (preferred platform)
- Microsoft 365 Business or Enterprise plan with Teams & Microsoft 365 Copilot or Microsoft Copilot Studio license
- ServiceNow account
 - The Customer Service Management (CSM) plugin should be installed in the ServiceNow instance.
 - The Knowledge base plugin should be installed.

Integrate with ServiceNow

During initial configuration of Self-Help Agent, in the **Connect your data** section, select the vertical ellipsis in the ServiceNow row, then select **Edit** to configure the ServiceNow connection.

You can connect to your ServiceNow instance by using Basic Authentication under **Authentication Type** with your credentials.

Capabilities

- Retrieve cases from ServiceNow

- Assign cases to self
- Generate case summaries
- Get recommended next steps based on case history, customer communication, and applicable company policies
- Draft response to customer
- Update case status

Use cases

- A customer service representative is overwhelmed with the number of incoming complaints. They can use Self-Help Agent to automatically retrieve active complaints from their CRM and receive recommendations on the next steps based on company policies. This helps the customer service representative resolve issues faster and maintain consistency across responses.
- A customer service team lead is facing a drop in first contact resolution (FCR) rates for complex issues. They can use Self-Help Agent to streamline the resolution process by receiving summaries of relevant case details and company policies. This enables them to diagnose and solve issues during the first interaction, improving customer satisfaction.

Extension opportunities

- Configure your preferred [knowledge source](#) within Copilot Studio.
- Fine-tune the custom instructions to change the personality of the agent.

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?

 Yes

 No

Provide product feedback ↗

Benefits Agent

Article • 11/19/2024

Publisher: Microsoft

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

ⓘ Note

Pre-built agents, such as Benefits Agent, 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 ↗](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Orchestrate agent behavior with generative AI (preview)

Article • 11/19/2024

You can configure an agent to use either of the following types of orchestration. With classic orchestration, an agent responds to users by triggering the topic whose trigger phrases match most closely the user's query. With [generative orchestration enabled](#) your agent can choose the best [actions](#), [knowledge](#), and [topics](#) to answer user queries, or respond to event triggers.

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](#).

i Important

Enabling generative orchestration can impact how billing is calculated. Learn more about [billing for generative mode](#).

The following table compares agent behavior between classic orchestration and generative orchestration.

[] Expand table

Behavior	Classic orchestration	Generative orchestration
Topics	Topics are selected based on matching a user query with trigger phrases	Topics are selected based on the description of their purpose
Actions	Actions can only be called explicitly from within a topic	The agent can choose to call actions based on their name and description

Behavior	Classic orchestration	Generative orchestration
Knowledge	Knowledge can be used as a fallback when no topics match a user's query (or called explicitly from within a topic)	The agent can choose to proactively search knowledge to answer a user's query.
Use of multiple topics, actions, knowledge sources	Agent tries to select a single topic to respond to the user, falling back to knowledge if configured	The agent can use a combination of topics, actions, and knowledge
Asking users for input	You must use question nodes in topics to author messages prompting the user for any required information	The agent can automatically generate questions to prompt users for any missing information required to fill inputs for topics and actions
Responding to a user	You must use message nodes in topics to author messages responding to the user (or call an action from a topic)	The agent automatically generates a response, using the available information from topics, actions, and knowledge that it has called

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 mode 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, actions, and knowledge sources

When a user sends a message, your agent selects one or more actions, topics, or knowledge sources to prepare its response. Multiple factors determine the selection.

The most important factor is the description of the topics, actions, and knowledge sources. Other factors include the name of a topic, action, 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 actions 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 actions or topics, or choose from its knowledge sources, to handle user queries (including multi-intent queries), or to autonomously respond to events. If multiple actions 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, actions, 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

When an agent is configured with generative orchestration, because the agent can use information provided by knowledge, actions, and topics to generate a response for the user, 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 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 mode in Copilot Studio, you can [open the activity map](#) to follow how your agent is responding.

Turn on generative orchestration for an agent

1. Open your agent.
2. Select **Settings**.
3. Select the **Generative AI** tab.
4. Select the **Generative** option.
5. (Optional) Select your preferred level of strictness for [content moderation](#).
6. Select **Save**.

Important

Using generative AI for orchestrating actions with both user and event triggers is a preview feature. Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that customers can get early access and provide feedback.

Tip

It's a good practice to inform your users that some of the conversation (for example, questions generated when running an action) might be generated by AI. For example, you could add an extra message in the [Start conversation 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 topics, actions, and knowledge sources. Good descriptions ensure the agent selects the right topics, actions, and knowledge sources to respond to users.

For actions, 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 actions, see [Use actions with custom agents \(preview\)](#).

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

When you turn on generative 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.

Best practices

Here are some best practices for naming topics, actions, and knowledge sources, and drafting clear, concise, and relevant descriptions for them.

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 action provides weather information" instead of "Weather information is provided by this action."

Use bulleted and numbered lists to clearly separate a series of items, actions, or considerations.

Relevance

Use keywords that relate to the action or topic's functionality and the user's intent. For example, if an action 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 action's or topics's functionality. The summary should be limited to one or two sentences, explain what the action 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 an action "Weather," name it "Weather Forecast" or "Weather Report."

Use specific language to prevent ambiguity between similar topics and actions.

For example, if your agent already has an action 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 action can answer.

Name: Answer Question

Description: This action 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 an action 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 plugin action 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.
- 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: "Get the forecast for the current day in the specified location." and an "Inputs (2)" section with "Location" set to "Seattle". An "Outputs (39)" section lists "Moon Phase", "Moon Phase Code", and "Moonrise Time" among others. A red box highlights the "Missing required information" note.
- Test your agent**: Top right button.
- Test Session Log:**
 - Message from AI: "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." (A minute ago)
 - User message: "What's the weather like in Seattle?" (Just now)
 - AI response: "What measurement system would you like to use for the forecast? You can choose between Imperial and Metric units." (Just now)
 - User message: "Ask a question or describe what you need" (0/2000)
 - AI note: "Make sure AI-generated content is accurate and appropriate before using. [See terms](#)" (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.
- Session started**: Top center button.
- Topic Selection:**
 - Lesson 1 - A simple topic**: Selected.
 - Lesson 2 - A simple topic with a condition and variable**: Available.
- Test your agent**: Top right button.
- Test Session Log:**
 - User message: "I need to get store hours and find my nearest store." (3 minutes ago)
 - AI response: "I'm happy to help with store hours." (2 minutes ago)
 - AI details: "The Redmond hours are:
 - Mon-Fri: 9am to 6pm
 - Sat: 10am to 4pm
 - Sun: Closed
 The Seattle hours are:
 - Mon-Fri: 9am to 6pm
 - Sat-Sun: 10am to 4pm
 - User message: "Did that answer your question?" (2 minutes ago)

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 action, but prepopulates the location with "Kirkland," based on the recent conversation context.

The screenshot shows the AI Platform interface. On the left, there's an 'Activity map' with a 'Session started' node and a 'Lesson 2 - A simple topic with a condition and variable Topic' node. Below it is a 'Get forecast for today Connector' configuration window. The 'Description' field says 'Get the forecast for the current day in the specified location.' The 'Inputs (2)' section shows '(x) Location | String = Kirkland' and '(x) Units | String ='. The 'Outputs (39)' section lists three fields: '(x) Moon Phase | String =', '(x) Moon Phase Code | String =', and '(x) Moonrise Time | DateTime ='. A red box highlights the 'Missing required information' message next to the 'Units' input. On the right, a 'Test your agent' panel shows a conversation: 'what's the weather like there?' (A minute ago), 'Sorry I wasn't able to help better. Would you like to try again?' (A minute ago), 'Yes' (A minute ago), 'Go ahead. I'm listening.' (Just now), 'what's the weather like there?' (Just now), 'What measurement system would you like to use for the forecast? You can choose between Imperial and Metric units.' (Just now), and an input field 'Ask a question or describe what you need' with '0/2000' characters.

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 no longer uses the [Conversational boosting](#) system topic. Therefore any modifications you might have made to this system topic to customize how your agent searches knowledge won't be used. This limitation also applies to classic data sources configured in generative answers nodes, including custom data sources. Learn more about [how knowledge works in generative mode](#).

Custom entity support for topic and action input parameters

Actions 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 configured to 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.

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Review agent activity (preview)

Article • 11/19/2024

Creating an agent is an iterative process. 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.

Viewing activity maps is a preview feature. Preview features aren't meant for production use and might have restricted functionality. These features are available before an official release so that customers can get early access and provide feedback.

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

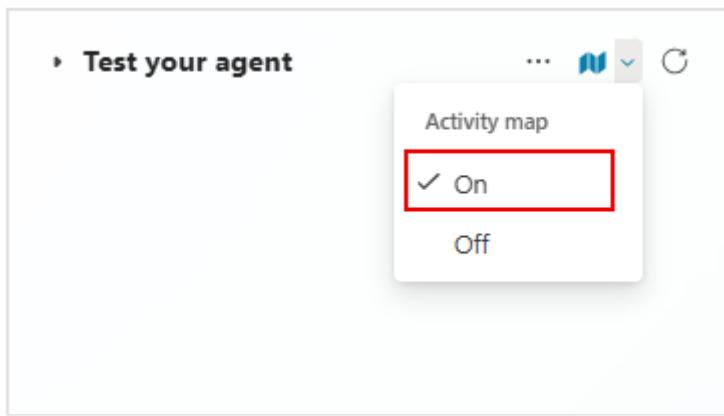
ⓘ Note

This feature is only available for agents with [generative orchestration](#) enabled.

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. 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 input that still needs to be collected, along with annotations from the agent about its decision-making.

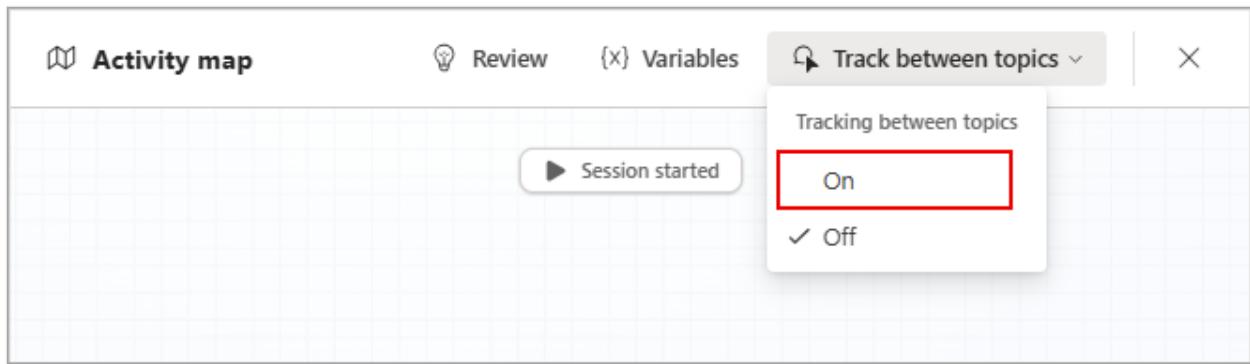
The screenshot shows the Weather Agent interface in Copilot Studio. On the left, the 'Activity map' panel displays a 'Get current weather' connector node. This node has two inputs: 'Units' (String) set to 'I' and 'Location' (String) set to 'Toronto'. It also has 18 outputs, including 'Temperature' (Number) = 49 and 'Temperature Units' (String) = °F. A tooltip indicates a 'Session started'. On the right, a test chat window shows a user message 'What's the current weather?' and an AI-generated response 'The current weather in Toronto, ON is mostly cloudy with a temperature of 49.0°F and a wind speed of 16.0 mph.'. An annotation 'Session in progress' is located at the bottom of the activity map, and another annotation 'Generated by AI. Be sure to check for accuracy.' is located at the bottom of the test chat window.

Testing event triggers

When [testing 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.



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 your agent made during a session, find where the agent's behavior doesn't match your goals, and find error details.

Go to the **Activity** page to review:

- The list of sessions and their details
- The step-by-step activity that happened within a session

ⓘ Important

The **Activity** page records interactions with the agent author or using the agent author's credentials only.

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 trigger*.
- Date: When the first interaction of the session began.
- Summary: An AI-generated summary of what happened during the interaction. You can use the summary for a quick understanding of the session.
- Status: The status of the session.
- Channel: The channel over which the interaction took place. Sessions that happened in the agent test pane have the Copilot Studio icon.

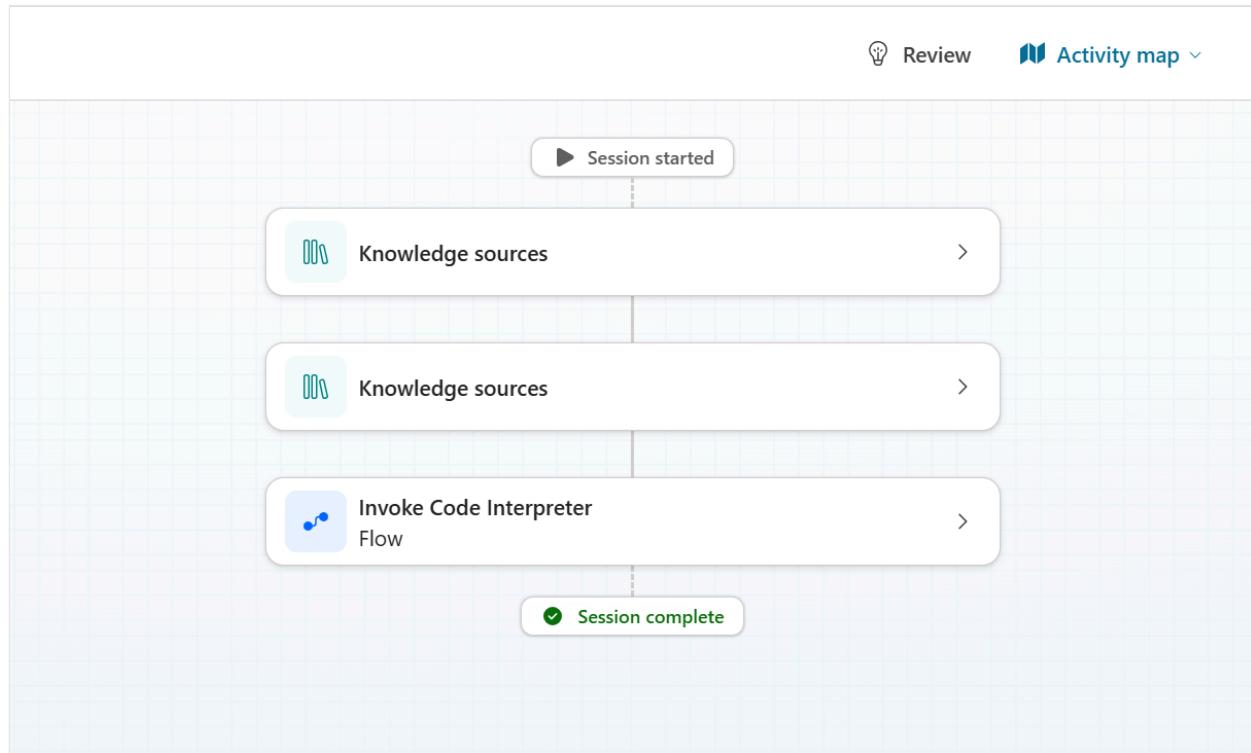
View session activity

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 .

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.



Expand any activity to see more details. Depending on the activity, you can see the different inputs (events) and outputs (reactions) for that point of the interaction.



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

In the activity details of this **Add a to-do** action, you can see that a required input value is missing. To solve the problem, you can modify event triggers, knowledge sources, instructions, or the action itself, and then test your agent again.

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)	Outputs (13)
{x} Title String = Draft a proposal for papercli	{x} Content String =
{x} To-do List String = Tasks	{x} Content Type String = text
	{x} Modified Date... DateTime =
	> See more (10)

Session complete

While viewing the activity map, you can open the **Review** panel to obtain AI-powered insights for understanding and improving your agent's activity. The review panel explains why the agent initiated a certain action and provides error details for unsuccessful actions. You can use this information to see if your agent is correctly and successfully carrying out its instructions.

Select **Edit** to view and change a [topic](#) or [action](#) associated with the selected activity.

View conversation history as a transcript

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.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Knowledge sources overview

Article • 11/19/2024

Within Copilot Studio, knowledge sources act in concert 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: Unlimited Classic mode: Four public URLs (for example, microsoft.com)	None
Documents	Internal	Searches documents uploaded to Dataverse, returns results from the document contents	Generative mode: Unlimited Classic mode: Limited by the Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	Generative mode: Unlimited 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 Copilot connectors	Internal	Connects to Copilot 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 Open AI, or Custom Data. Instead, from the generative answers node properties, use the **Classic data** option for [Bing Custom Search](#), [Azure Open AI](#), or [Custom Data](#) sources.

- For websites, you need to confirm which website(s) your organization owns that Bing will search through Copilot Studio.

Allow the agent to use general knowledge

The **Allow the AI to use its own general knowledge** setting on the [Overview](#) page configures your agent to use generative AI. Generative AI includes general knowledge, which refers to the foundational knowledge that the generative AI is trained on. When this setting is enabled, it allows you to use this general knowledge in your agent's answers. If you prefer that your agent is grounded with your specific knowledge sources, disable this setting.

Source authentication

If you're using SharePoint, Dataverse, or enterprise data using Microsoft Graph 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.

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
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:

- Copilot Studio 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, Copilot Studio 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 such knowledge sources, you must embed them inside a [generative answers node](#) in a topic.

Content moderation

The content moderation settings allow your agent to provide more answers. However, the increase in answers might affect the relevancy of the given answers. 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 be inaccurate. The highest level generates fewer answers, but they're likely to be more relevant. The default moderation level is **High**.

2. Select **Save**.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Add knowledge to an agent

Article • 11/19/2024

Knowledge sources can be added when you first create your agent, or after your agent has been created. 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 [AI orchestration](#).

- [Public website](#)
- [File upload](#)
- [SharePoint](#)
- [Dataverse](#)

 Note

[Microsoft Graph connector knowledge sources](#) obtain their knowledge name and description from the Microsoft Graph connector name and description. These are provided by the tenant admin, and are configured in Microsoft 365.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Add a public website as a knowledge source

Article • 11/19/2024

Public websites as knowledge sources at the agent level use publicly available information to improve the reach of your agent.

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 [AI 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](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Add SharePoint as a knowledge source

Article • 11/26/2024

SharePoint as a knowledge source for [generative answers](#) works by pairing your agent with a SharePoint URL, such as `contoso.sharepoint.com/sites/policies`. When an agent 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.

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.

To add SharePoint 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 **SharePoint**.
4. Provide the SharePoint URL. Separate multiple URLs with manual line breaks (use **Shift + Enter**).
5. Add a name (by default, the file's name is used).
6. Add a description. The description should be as detailed as possible, especially if generative AI is enabled, as the description aids [AI orchestration](#).
7. Select **Add** to finish adding the knowledge source.

Known issues

- We recommend omitting `https://` from the URL. Recognized SharePoint URLs are from the `sharepoint.com` domain.
- [Modern pages](#) are supported. However, modern pages containing SPFx components aren't supported.

- SharePoint sites with an accordion navigation menu or a custom CSS aren't used to generate answers.
 - Classic ASPX pages on SharePoint aren't used to generate answers.
 - Due to memory limitations, generative answers can only use SharePoint files that are under 3 MB. If a file is larger than 3 MB, consider splitting it into multiple smaller files. For more information, see [File size support](#).
 - 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](#).
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Add a Dataverse knowledge source

Article • 12/12/2024

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](#).

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 [AI 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: "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.
Custom field	opportunity revenue	"Opportunity revenue" refers to the "Custom Revenue" column in the "Opportunity" table.

Scenario	Glossary term	Sample description
Complex rules or filter	overdue task	"Overdue task" refers to 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.

Feedback

Was this page helpful?



[Provide product feedback !\[\]\(7c1cf7c693e5dfb4a0f582605b5b8acc_img.jpg\)](#)

Upload files as a knowledge source

Article • 11/19/2024

You can upload your own 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 at 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](#).

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 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 as detailed as possible, especially if generative AI is enabled, as the description aids [AI orchestration](#).
6. Select **Add** to finish adding the knowledge source.

Supported document types

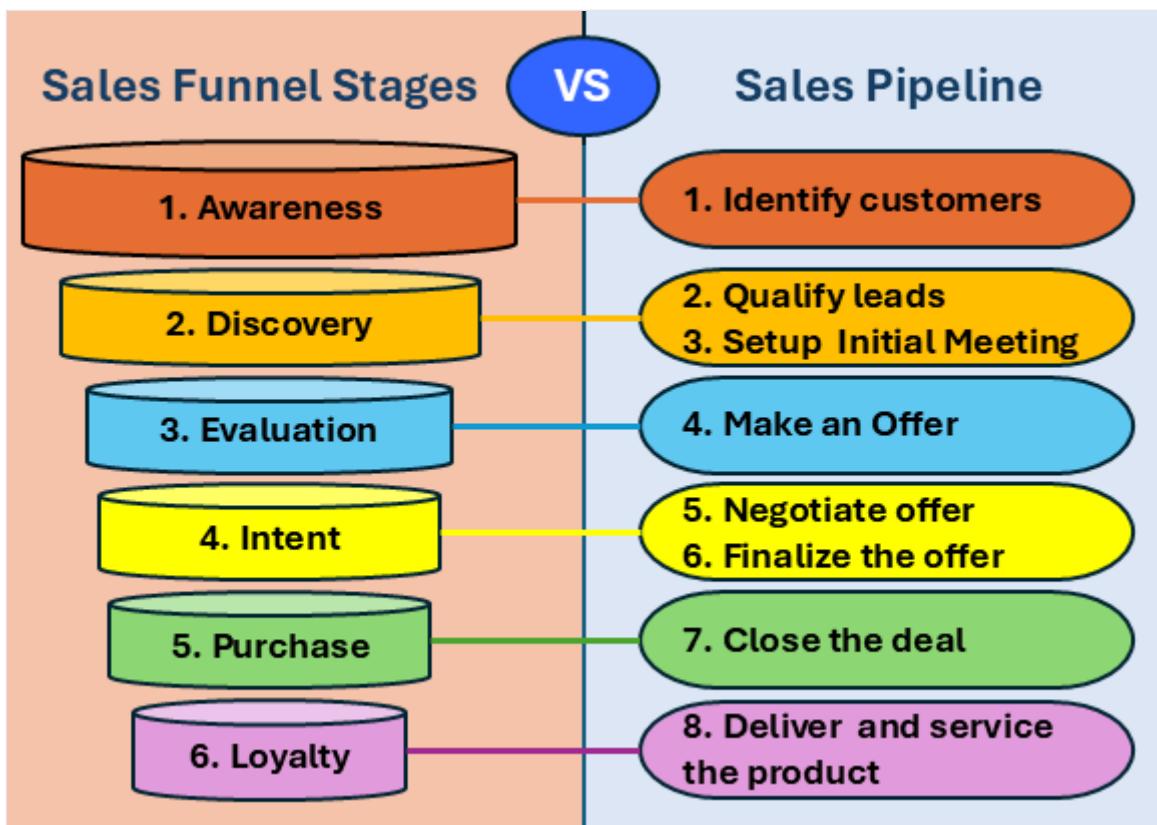
- Word (doc, docx)
- Excel (xls, xlsx)
- PowerPoint (ppt, pptx)

- 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 (preview)

Annotated images that are embedded in PDF files are supported, as well. 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

- Annotated images in PDF files are only supported in newly uploaded files. If you had previously uploaded an annotated PDF file, you need to delete the existing file, and upload the file again.
- 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 only supported in Switzerland and the United States.
- Icons aren't supported.
- Files greater than 512 MB aren't supported.
- Files that contain encrypted content, are password-protected, or contain confidential tags, aren't supported.
- The maximum number of files that can be included as knowledge in an agent is 500 files.

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Add Enterprise data as a knowledge source

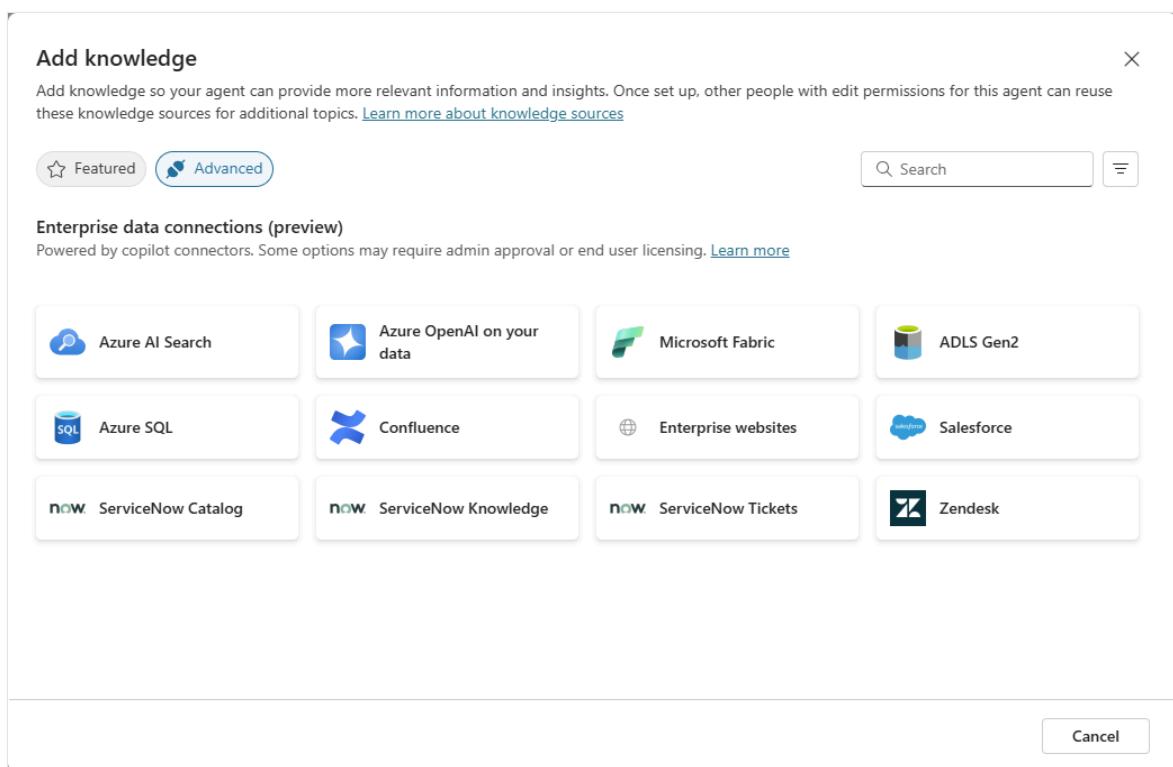
Article • 12/13/2024

With knowledge in Copilot Studio, makers can ground their agent responses with enterprise data available through [Copilot connectors](#).

Use Copilot connectors as a knowledge source for your agent

With Copilot Studio, you can select Copilot connectors previously configured by your tenant administrator, and add them as knowledge sources to your agent.

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 **Advanced**.



4. Select the connector available within your tenant.
5. Select **Add** to complete the connection.

Azure AI Search

Azure AI Search provides a powerful search engine that can search through a large collection of documents. Copilot Studio supports adding an Azure AI Search connector to use 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, the following Azure articles provide information on the setup and configuration needed to use the Azure AI Search connector 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](#)

Note

Currently, you must [create vectorized indexes using integrated vectorization](#).

Prepare your data and choose an embedded model, then use **Import and vectorize data** from Azure AI Search to create vector indexes. This 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.

Add an Azure AI Search connector

After selecting **Azure AI Search** as the connector, follow these steps to complete the connection.

1. Enter a name and description for the connector. The name must be unique.
2. Select **Create** and enter the **Authentication type**, **Azure AI Search Endpoint URL**, and **Azure AI Search Admin Key**.
3. Select **Create** again to complete the connection to the connector. A green check mark appears to confirm the connection to the connector.
4. Select **Next**.

5. Enter the Azure AI Search vector index to be used for the connector. Only one vector index can be added.

6. 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](#). During testing, you can verify that proper references were called by reviewing the files and citations cited by the agent.

Real-time knowledge with connectors

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 loss prevention](#) policies.

Supported real-time connectors

The following real-time connectors are supported:

- Salesforce (API version 51.0 or higher)
- ServiceNow Knowledge
- ZenDesk
- Azure SQL Server

 Note

On-premises Azure SQL Server instances are not supported as an enterprise data source.

Add a real-time connector

After selecting the real-time connector, follow these steps to complete the connection.

1. Select the existing connection, then select **Add**.
2. Select **Sign in**.
3. Select the target location for the connector, and then provide your credentials for the real-time connector.
4. Select **Next**.
5. Select the tables that you want to use as your knowledge source.
6. Add a name and description for the knowledge source. The default name reflects the selected tables.
7. Add [synonyms and glossary definitions](#).

 **Note**

Currently, only ServiceNow Knowledge and ZenDesk connectors support adding synonyms and glossary definitions.

8. 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](#).

Supported enterprise data sources using Microsoft Graph connectors

The following connectors using enterprise data sources are supported in Copilot Studio:

- Power Platform connector actions
- Azure AI Search

- Azure DevOps Wiki
- Azure SQL Server
- Salesforce
- ServiceNow Knowledge
- ZenDesk
- ADLS Gen2
- MediaWiki
- File share
- SharePoint Server
- SAP
- Ado Git
- CSV
- Jira
- Oracle SQL database
- Microsoft SQL server(New)
- ServiceNow Catalog
- Confluence
- ServiceNow Tickets
- Custom Connector

 **Note**

- For the majority of [Microsoft Graph connectors](#), customers can use the same data sources to ground agent responses that are used to augment Microsoft Search. To be displayed in the enterprise data sources, they first must be configured by your tenant admin. However, [Salesforce](#), [ServiceNow Knowledge](#), and [ZenDesk](#) are preconfigured and available automatically.
- You can request your tenant admin to set up a Microsoft Graph connector. For more information, see [Set up Microsoft Graph connectors in the Microsoft 365 admin center](#). Alternatively, if a Microsoft Graph connector isn't available, you might need to connect to the source system and retrieve data. Then select **Create action** to use actions instead.
- For more information about Microsoft Graph connectors and Microsoft Search, see [Microsoft Graph connectors overview for Microsoft Search](#). This article also lists the currently supported Graph connectors.
- Agents that use Microsoft Graph connectors as knowledge sources must be configured with the correct authentication settings when publishing to

channels. The `ExternalItem.Read.All` scope must be provided as part of the manual authentication setting. For more information, see [Configure user authentication in Copilot Studio](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Edit, view, or delete knowledge sources

Article • 11/19/2024

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 Graph 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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Test your agent's knowledge sources

Article • 11/19/2024

Testing an agent with knowledge sources automatically grounds that agent with those knowledge sources.

After adding knowledge sources, 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 is only applicable to file upload and Dataverse knowledge sources.
Not allowed	This knowledge source is blocked by DLP 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 your agent**.
2. In the **Test agent** pane, 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've finished testing that specific knowledge source, reactivate the other knowledge sources that you think should be relevant in that generative answers node.

Forming questions

- Your agent has difficulty answering questions that require calculations, comparisons, or form submissions. Your agent 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.
- You might need to disable the sample topics, lessons 1-3, that automatically come with a new agent. They're used before any URLs are accessed.

Select the  next to a sample topic on your [Topics](#) page and use the **Enabled** toggle to enable or disable the topic.

For more information, see [Use lesson topics](#).

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:

1. Ensure that your knowledge sources are enabled in one or more of your generative answers nodes.
2. Ensure that the number of knowledge sources in your generative answers node don't exceed the maximum limits, which can be found in [Supported knowledge sources](#).
3. Ensure that [Content moderation](#) for your generative answers node is set to the correct level.
4. 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.

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Use lesson topics

Article • 11/19/2024

Copilot Studio 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. These examples help you learn how to create content for your agents.

Prerequisites

- An existing agent, or if you need to create one, see [Quickstart: Create and deploy an agent](#).
- If you're using a bot that has **classic** added to the name, you can [upgrade your bot](#) to see the latest versions of the sample topics. All bots made in the Teams app are classic bots.

Web app

Use sample topics to understand how topics work

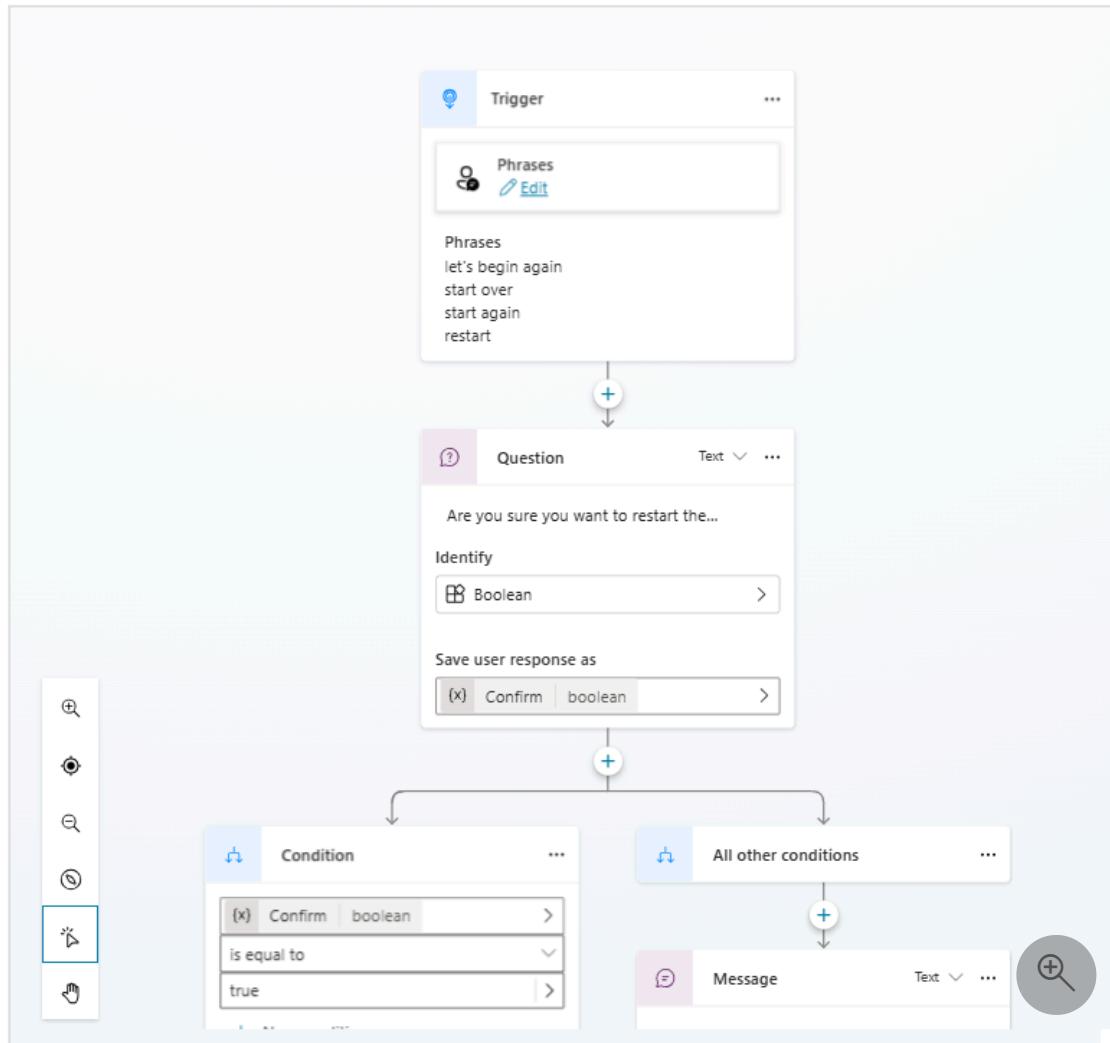
1. Open your agent, or [create one](#), and go to the [Overview](#) page.
2. Review the agent's details for name and description. The agent derives topic phrases from the description (if any).
3. Select **Topics** at the top of the page to view the agent's conversation flow.

The screenshot shows the Copilot Studio interface with the 'Topics' tab selected for the 'Safe Travels agent'. A search bar at the top right contains the placeholder 'Search custom topics'. Below it, there are two tabs: 'Custom (5)' (which is highlighted) and 'System (9)'. A large button labeled '+ Add a topic' is positioned above the list of topics. The table below lists five topics:

Name	Trigger	Description	Editing	Last modified	Errors	Enabled
Goodbye	Phrases	This topic tr...		Christopher Chew...		<input checked="" type="checkbox"/> On
Greeting	Phrases	This topic is...		Christopher Chew...		<input checked="" type="checkbox"/> On
Start Over	Phrases			Christopher Chew...		<input checked="" type="checkbox"/> On
Thank you	Phrases	This topic tr...		Christopher Chew...		<input checked="" type="checkbox"/> On
What Can I ...	Phrases			Christopher Chew...		<input checked="" type="checkbox"/> On

4. Select each topic from the list to open it. In this case, trigger phrases were added manually to the Trigger node.

The flow includes the agent's comments, expected user responses, decision points, and entity references used in each topic. This sample flow is for Lesson 3.



Available sample topics

Each sample lesson topic is designed to teach you how to create basic and advanced agent conversations. We recommend that you start with **Lesson 1** and work through each topic in order.

Lesson 1 - A simple topic

This sample template is a topic that responds to a question about store hours.

Lesson 2 - A simple topic with a condition and a variable

This sample shows you how to create a question that asks customers which store they're interested in and keeps the response in a variable. The condition uses the variable to determine which store's hours to return.

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

This sample asks customers to enter a shipping state. The agent uses the prebuilt **State entity** to recognize the US state in the user's response, and stores it in the **State** variable.

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

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use system topics

Article • 11/19/2024

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 system 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.

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 your custom topics 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.

① 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 an error occurred.

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 agent pane, a detailed error message is included to help the agent author diagnose the issue. Learn how to [troubleshoot errors](#).

- Triggers when an error occurs during the conversation.

Reset Conversation

- Resets the conversation by clearing variable values and forcing the agent to use the latest published agent content.
- 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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create and edit topics

Article • 11/19/2024

In Copilot Studio, a topic defines how an agent conversation progresses.

To author topics, you can [customize provided templates](#), create topics from scratch, or [describe what you want and have AI create the topic for you](#).

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 *conversation 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.

Often, a topic has a set of *trigger phrases*—phrases, keywords, and questions that a customer is likely to use that are related to a specific issue. Copilot Studio agents use natural language understanding, your customer's message, and the trigger phrases for all your topics to find the best topic to trigger. The message doesn't need to exactly match a topic's trigger phrases to trigger the topic. For example, a topic about store hours might include the trigger phrase `check store hours`. If a customer enters "see store opening hours," this phrase triggers your store hours topic.

Prerequisites

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

Natural language understanding

An agent uses natural language understanding (NLU) to parse what a customer typed and find the best trigger phrase or node.

For example, a user might type "Open hours." The agent matches that to the **Store hours** topic, begins a conversation that asks which store the customer is interested in, and then displays the hours the store is open.

The **Test agent** pane shows how the agent conversation progresses at every step. You can use the **Test agent** pane to fine-tune a topic without having to exit the Copilot Studio portal.

For agent and topic limits, see [Quotas, limits, app registration, certificates, and configuration values](#).

Topics in Copilot Studio

Web app

Topic types

An agent can include 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

Option	Description
Send a message	Send the customer a message.
Ask a question	Ask the customer a question.
Ask with adaptive card	Create a JSON snippet to exchange with other apps.
Author using conditions	Branch your conversation based on a condition.

Option	Description
Variable management	Set a value, parse a value, or clear all variables.
Manage topics	Redirect, transfer, or end the topic or conversation.
Call a flow	Call a flow like Power Automate or Excel Online, or use a connector or a plugin.
Advanced	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. Open your agent from the list on the **agents** page. For better visibility, close the **Test your agent** panel for now.
2. On the top menu bar, select **Topics**.
3. Select **Add a topic**, and select **From blank**.

A **Trigger** node appears on an otherwise blank topic authoring canvas.

4. Select the **More** icon (...) of the **Trigger** node, and then select **Properties**. The **On Recognized Intent properties** panel appears.
5. In the **On Recognized Intent properties** panel, select the **Phrases** area. The **Phrases** secondary panel appears.
6. 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**.

You can include punctuation in a trigger phrase, but it's best to use short phrases rather than long sentences.

7. Select **Details** on the toolbar to open the **Topic details** panel.

8. Add your agent topic details:

- 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.

9. Select **Save** on the top menu bar to save your topic.

Important

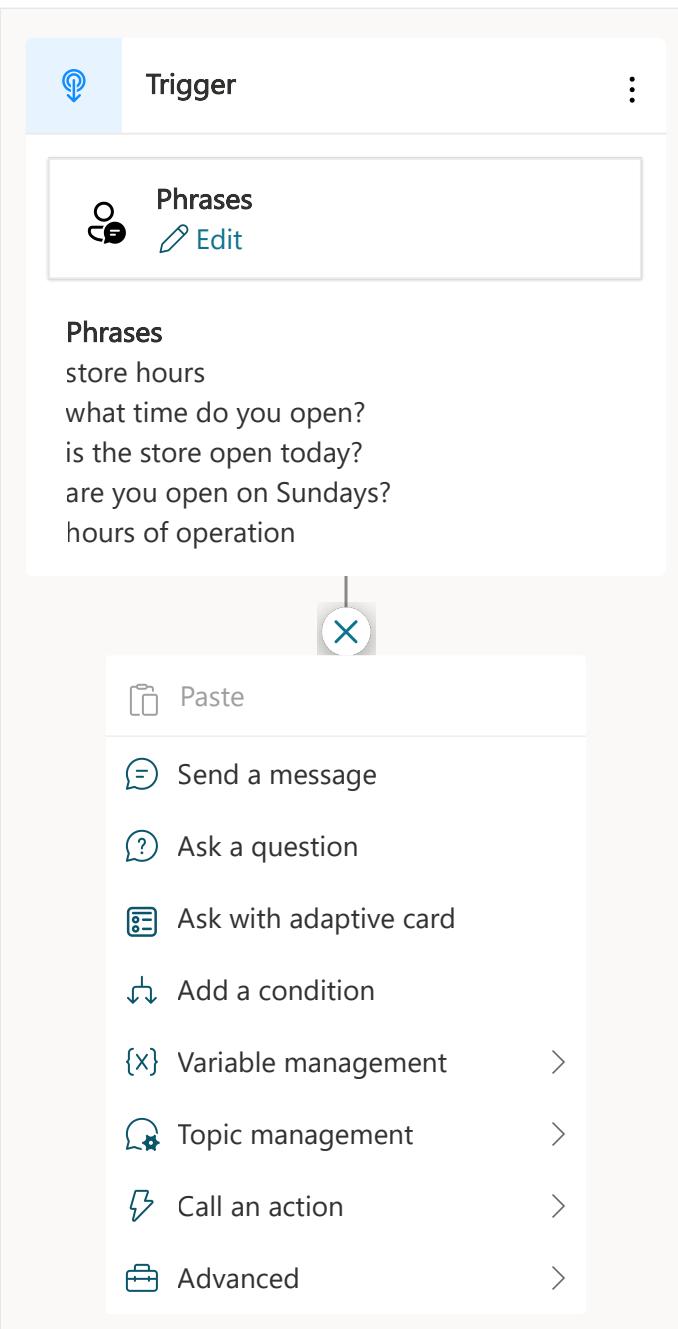
Avoid using periods (.) in your topic names. It isn't possible to export a solution that contains an agent with periods in the name of any of its topics.

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.

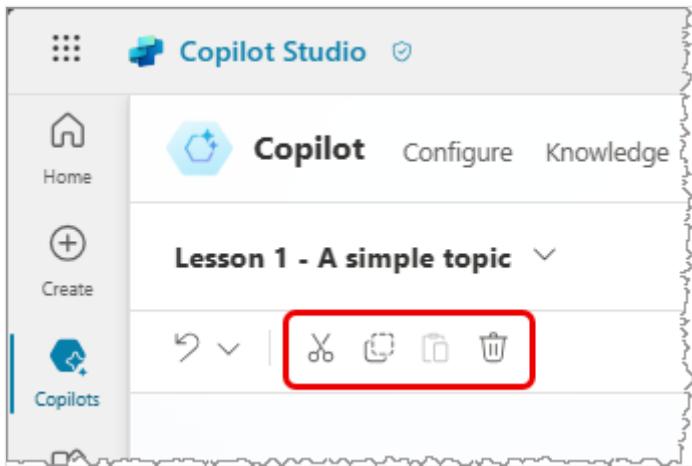


Delete a node

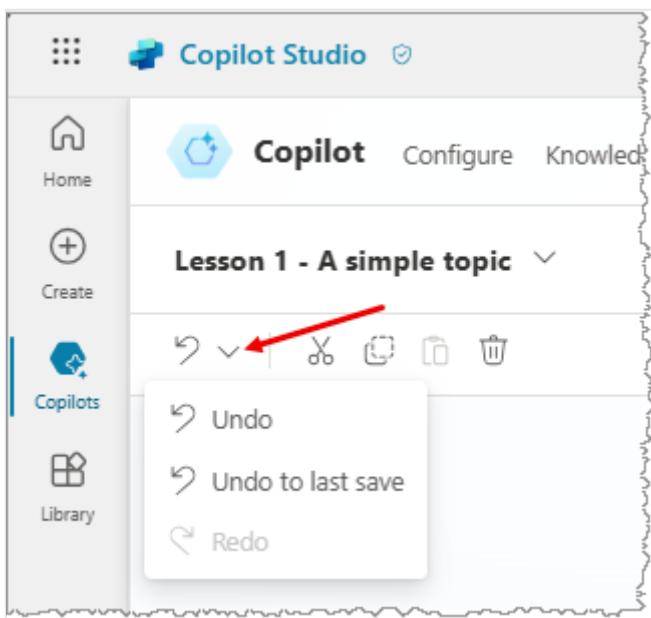
Select the **More** icon (...) of the node you want to delete, and then select **Delete**.

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 an edit. Open the **Undo** menu to revert all actions back to the last save or 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

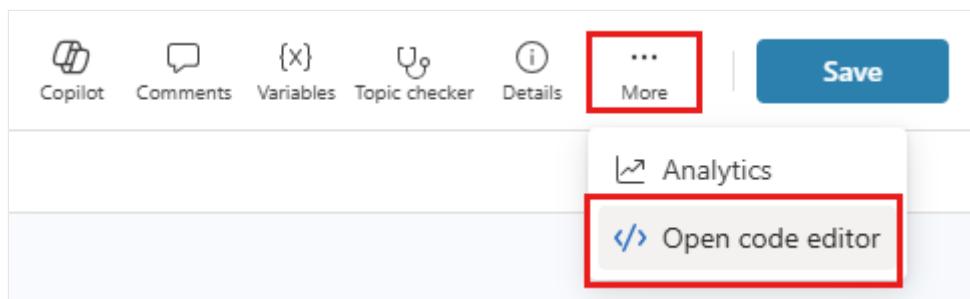
The code editor shows the topic in [YAML](#), a markup language that's easy to read and understand. Use the code editor to copy and paste topics from other bots—even ones created by other authors.

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 **+ Create > 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.

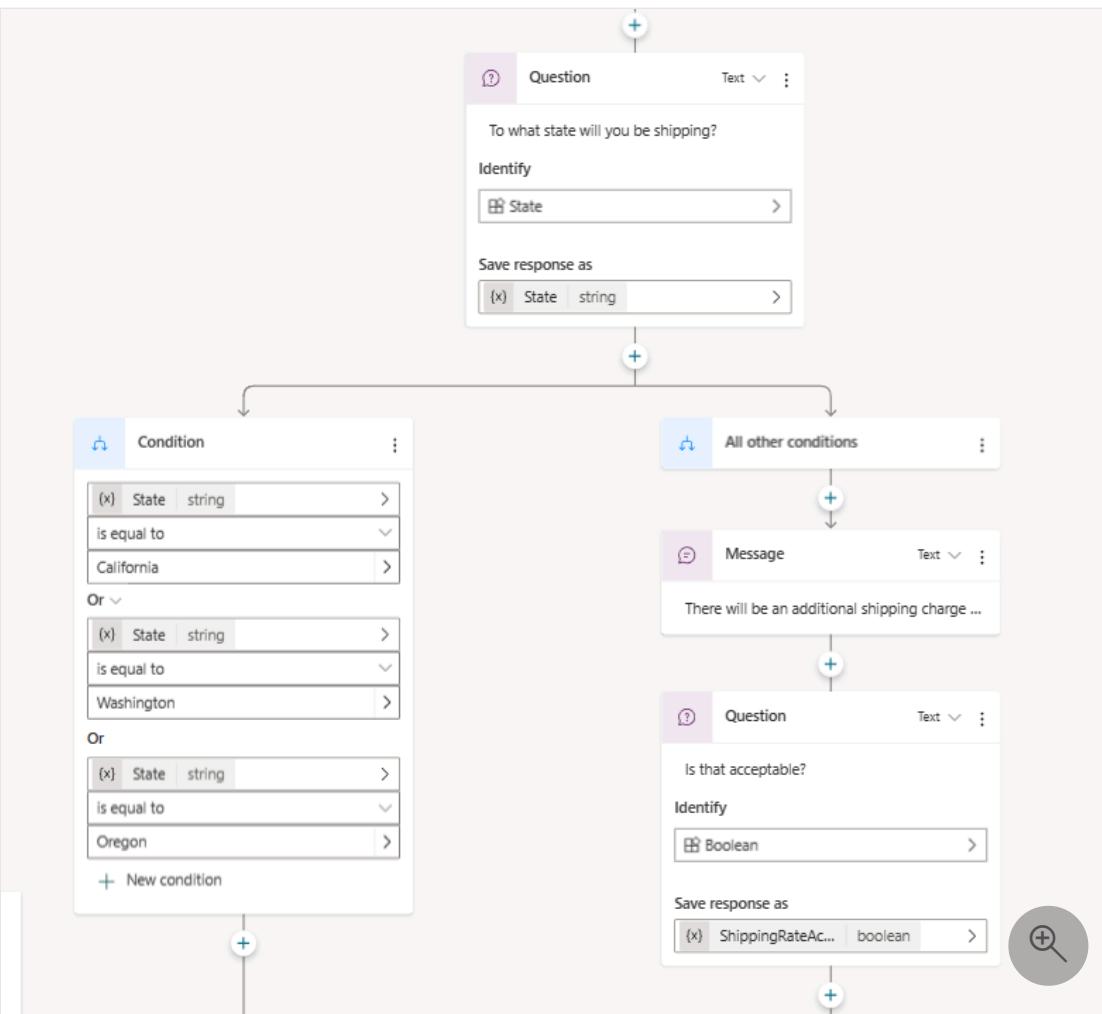
    - kind: Question
      id: 6lyBi8
      alwaysPrompt: false
      variable: init:Topic.ShippingRateAccepted
      prompt: Is that acceptable?
      entity: BooleanPrebuiltEntity

    - 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**. The **Question** node now has many conditions to the question about shipping.



Test and publish your agent

[Test your agent](#) when you make changes to your topics, to make sure everything works as expected.

After you design and test your agent, [publish it to the web, mobile or native apps, or Microsoft Bot Framework channels](#).

Feedback

Was this page helpful?

[Yes](#)

[No](#)

[Provide product feedback ↗](#)

Create and edit topics with Copilot

Article • 10/30/2024

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 copilot authoring overview](#) shows how this capability works. You can also use copilot 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 copilot.
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 copilot 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 copilot, 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*

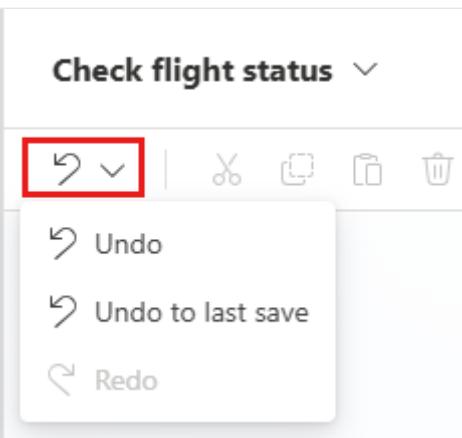
💡 Tip

When describing a task for your copilot, there's no need to mention design elements. For example, if your description includes *ask the user their full name*, it will automatically generate (or update) a question node—you don't have to direct the AI with *ask the user their full name in the question node*.

4. Select **Update**.

You can continue to [edit topics](#) 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.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Create and reply to comments in agent topics

Article • 11/19/2024

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 table in Dataverse in the default solution.

Replying to an existing 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.

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, and 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 topic-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

Select the menu icon at the top of an individual comment, next to the your name, to edit or delete a comment.

You can only edit or delete comments that you created.

- Editing the comment lets you change that comment.
- Deleting the comment removes it completely from that thread. You can't restore deleted comments.

Additional notes

- Comments aren't exported when you [export an agent with a solution](#).
- You can't @ mention other agent makers in a comment.
- New comments are displayed for [other agent 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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Using topic triggers

Article • 10/11/2024

By default, topics use the **Phrases** trigger, which activates when one or more of their 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

- You might want to be familiar with creating and editing topics before understanding topic triggers. For more information, see [Create and edit topics](#).

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](#).

Trigger types

Triggers allow you to intercept and handle 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** type of **Activity**. Agents can receive activities of other types with new changes in the chat.

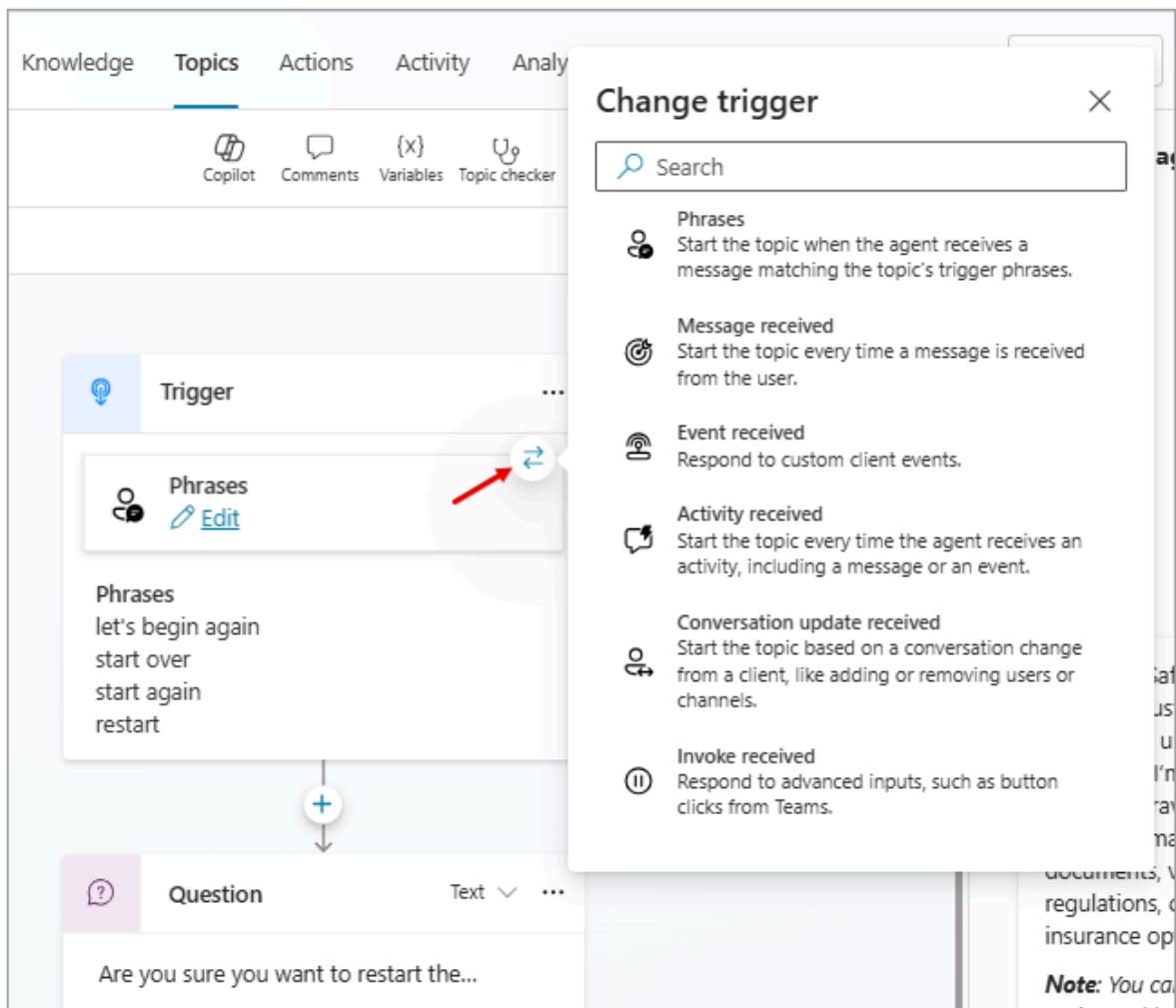
For example, Microsoft Teams sends hidden activities of type **invoke**, which signifies a user action, such as interacting with a Message Extension.

 Expand table

Type	Description
Phrases	When one or more of the trigger phrases match with the incoming message from a user
Activity received	When an Activity of any type is received
Message received	When an Activity of type message is received The most common type of Activity. Received when a user types or says something to the agent.
Event received	When an Activity of type event is received
Conversation update received	When an Activity of type conversationUpdate is received For example, Teams sends an activity of this type when a user joins a conversation.
Invoke received	When an Activity of type invoke is received Most commonly received from the Teams channel. For example, when the user interacts with a Message or Search extension in the Teams app.
Inactivity	When a user hasn't interacted with the agent over time. The time period can be configured.

Changing the trigger for a topic

1. Open your agent in Copilot Studio and go to the **Topics** page.
2. Open any topic.
3. Hover over **Phrases** of the **Trigger** node and select the **Change trigger** icon. You see a list of triggers to choose from.



4. Select the type of trigger for your topic.

ⓘ Important

Your current trigger type content, if any, gets deleted when the trigger type is changed. You're asked to confirm before changing the type.

Discard your content?

Current content will be lost by changing your trigger type.

Discard

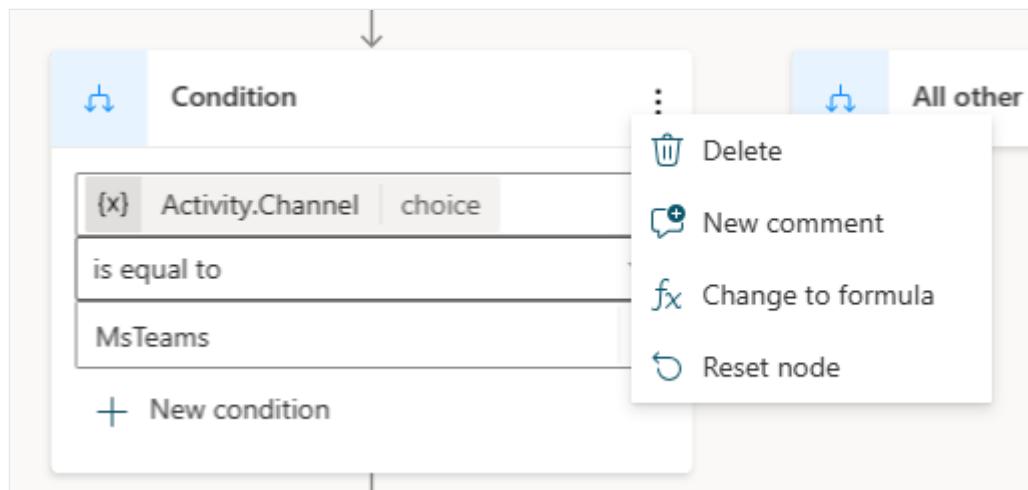
Cancel

5. Select **Edit** on the trigger to open the properties for that trigger type.

Trigger condition

The **Condition** node allows you to specify condition(s) that must be met in order for the trigger to fire. For example, a condition fires only if the channel used by an employee is

Microsoft Teams.



Optionally, you can switch to the full **Power Fx** editor, to author more complex conditions, by selecting the **... Node Menu** and choosing **Change to formula**.

For more information, see [Add conditions to topics](#).

Trigger priority

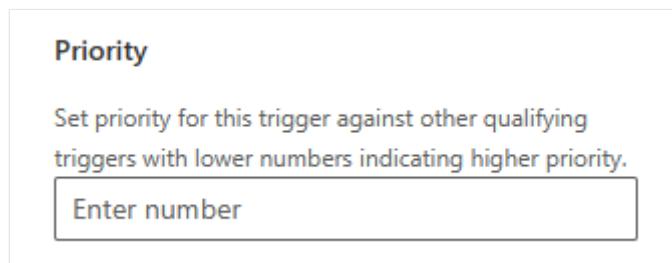
More than one trigger can fire for a single incoming Activity, such as a message. The trigger type determines the order the triggers fire.

Order of execution:

1. Activity Received
2. Message / Event / Conversation Update / Invoke Received
3. Phrases

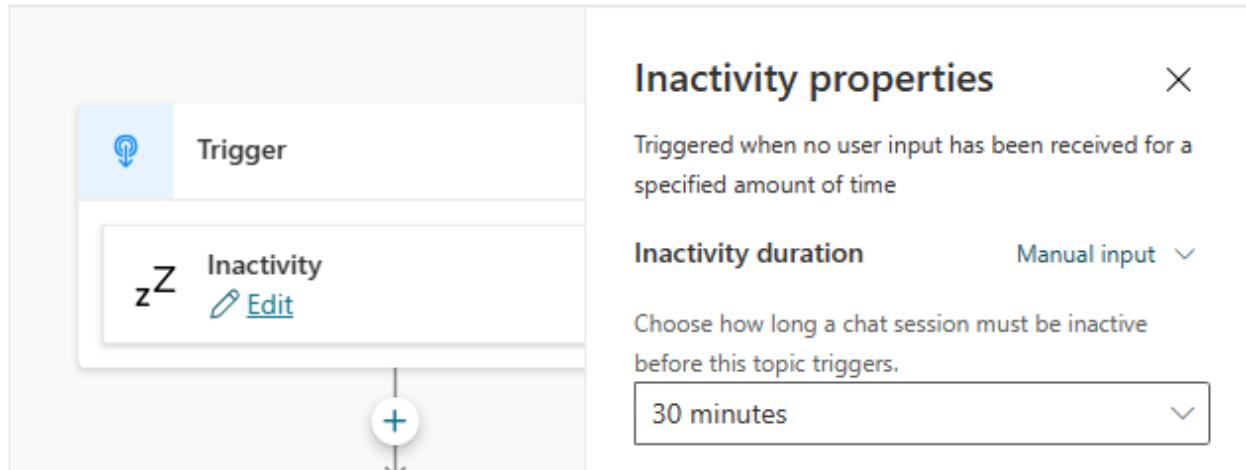
If there's more than one qualifying trigger of the same type, the triggers get executed in the order of creation (oldest first).

You can use the **Priority** property on a node's **Properties** pane to explicitly determine the order of execution.



Other trigger properties

Some trigger types have more properties, beyond **Condition** and **Priority**. For example, the **Inactivity** trigger type lets you set the inactivity duration.



Default trigger types

These properties are empty by default.

However, if you choose an activity or event type, only matching incoming types are fired. For example, if you select the **Hand off** activity type when a conversation needs to escalate to an agent, only incoming messages that trigger **Hand off** fire.

[Expand table](#)

Activity Received	Event Received
Activity types	Event types

For more information, see [Sending event activities](#).

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Send a message

Article • 11/19/2024

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.

💡 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.

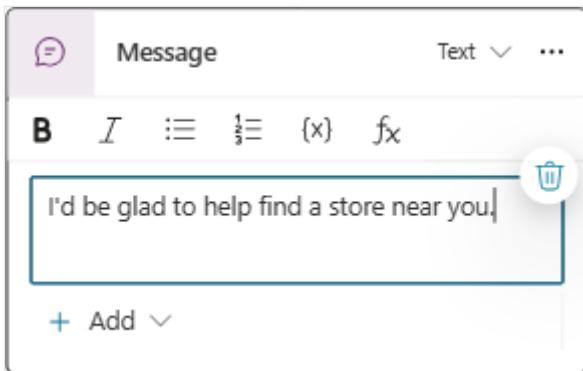
Prerequisites

To work with a message node, you need an agent and a topic created. To understand how topics works, see [Create and edit topics](#).

If you need to create a new agent, see [Quickstart: Create and deploy an agent](#).

Send a text message

1. Open any topic, then select "+" **Add node** and choose **Send a message** to add a new message node.
2. In the text box, enter the message you want the agent to send.
3. Optionally, use the tools in the node's menu bar to add basic formatting, [variables](#), and [message variations](#).

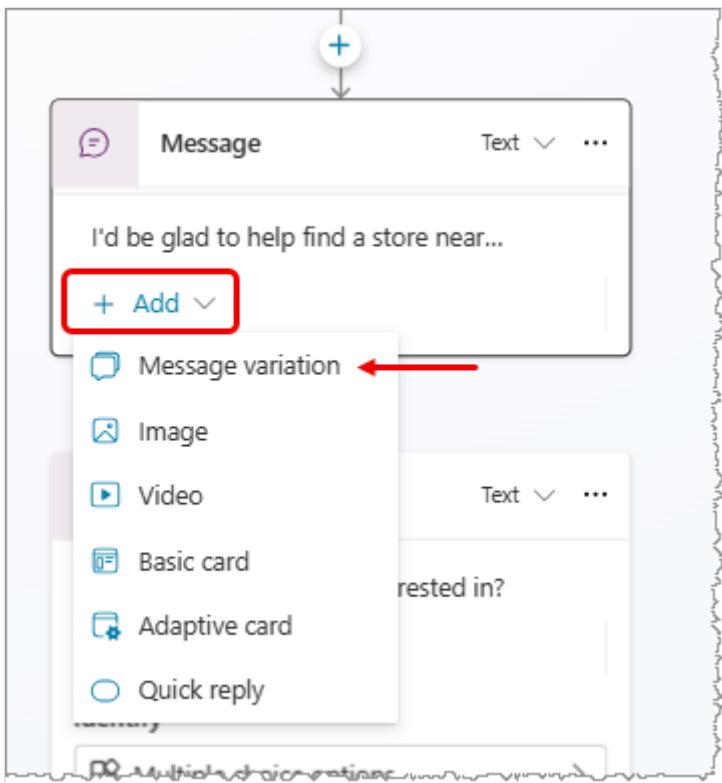


Use message variations

When you add message variations, the agent randomly picks one of them to use each time the node is triggered.

Add a message variation

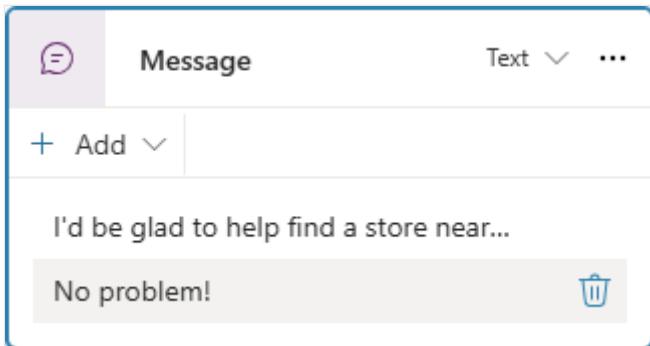
1. In the node's menu bar, select **Add**, and then select **Message variation**.



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 **Remove variation** icon  next to the variation you want to remove.



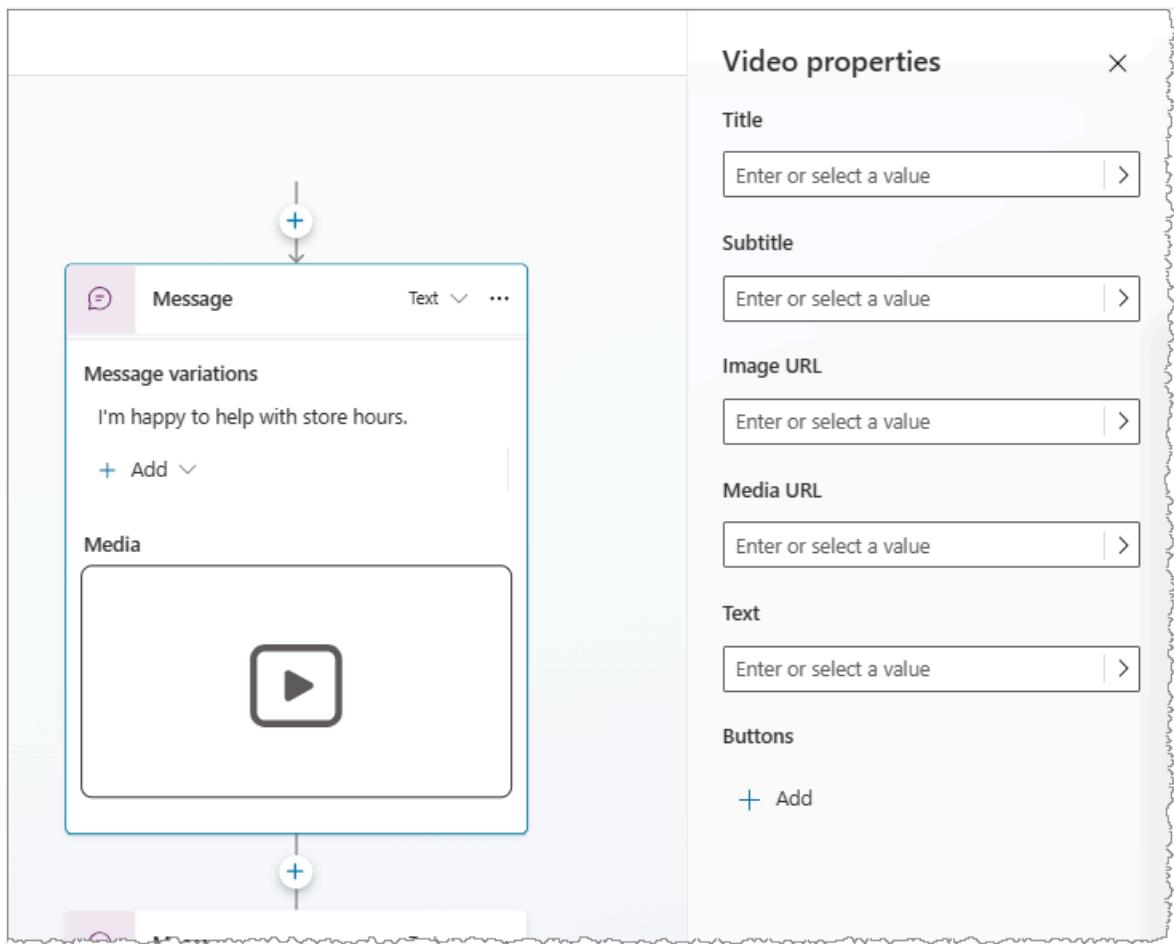
Add an image

To add an image, you need one hosted through a URL.

1. [Add a Message node](#).
2. In the node's menu bar, select **Add** and choose **Image**.
3. Under **Image**, enter the URL of your image.
4. Optionally, enter a **Title** for the image.

Add a video

1. [Add a Message node](#).
2. In the node's menu bar, select **Add**, and then select **Video**.
3. 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.
4. 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 one or more 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. [Add a Message node](#).
2. In the node's menu bar, select **Add**, and then select **Basic card**.
3. In the **Basic Card properties** pane, fill in the properties for the content of your card. You can also add one or more buttons, which have the same effect as [quick replies](#).

Add an Adaptive Card

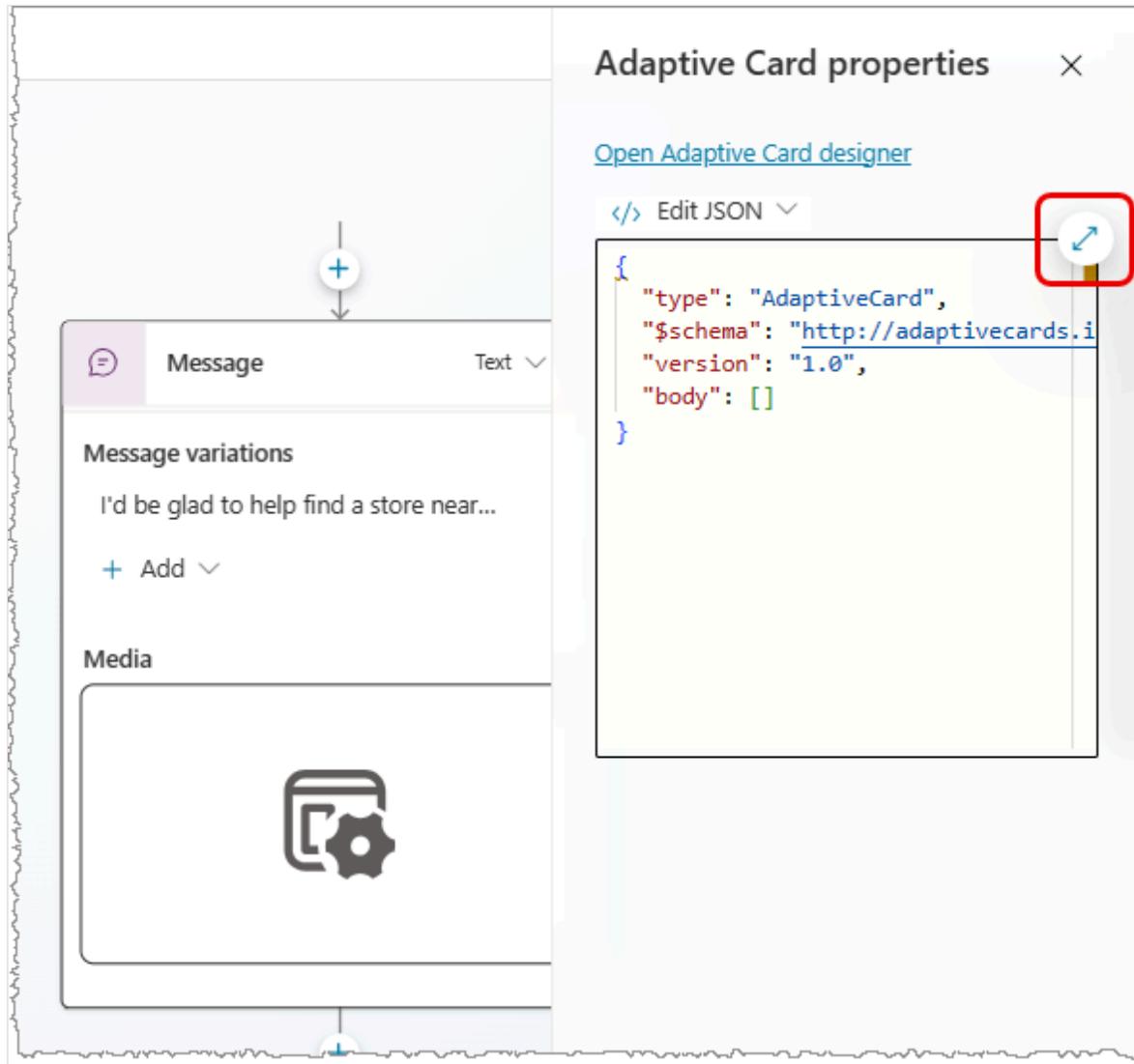
[Adaptive Cards](#) are platform-agnostic cards that you can customize. You can design Adaptive Cards using the [Adaptive Cards Designer](#) or author the JSON directly.

i Important

Copilot Studio supports AdaptiveCard schema version 1.3 or lower.

1. Add a Message node.
2. In the node's menu bar, select **Add**, and then select **Adaptive card**.
3. Under **Edit JSON** of the **Adaptive Card properties** pane, enter the JSON for your card.

To open a larger view of the JSON editor, select the **Expand** icon.



Adaptive card example

If a message node informs users of your store's hours, you might create an adaptive card showing the hours with an image of the store.

Message

Text

+ Add Text

B *I* Text Text Text Text Text

Message variations

I'm happy to help with store hours.

Days of week Open hours

Days of week	Open hours
Monday	9 AM - 5 PM
Tuesday	9 AM - 5 PM
Wednesday	9 AM - 5 PM
Thursday	9 AM - 5 PM
Friday	9 AM - 5 PM
Saturday	Closed
Sunday	Closed

Media



Adaptive Card properties

[Open Adaptive Card designer](#)

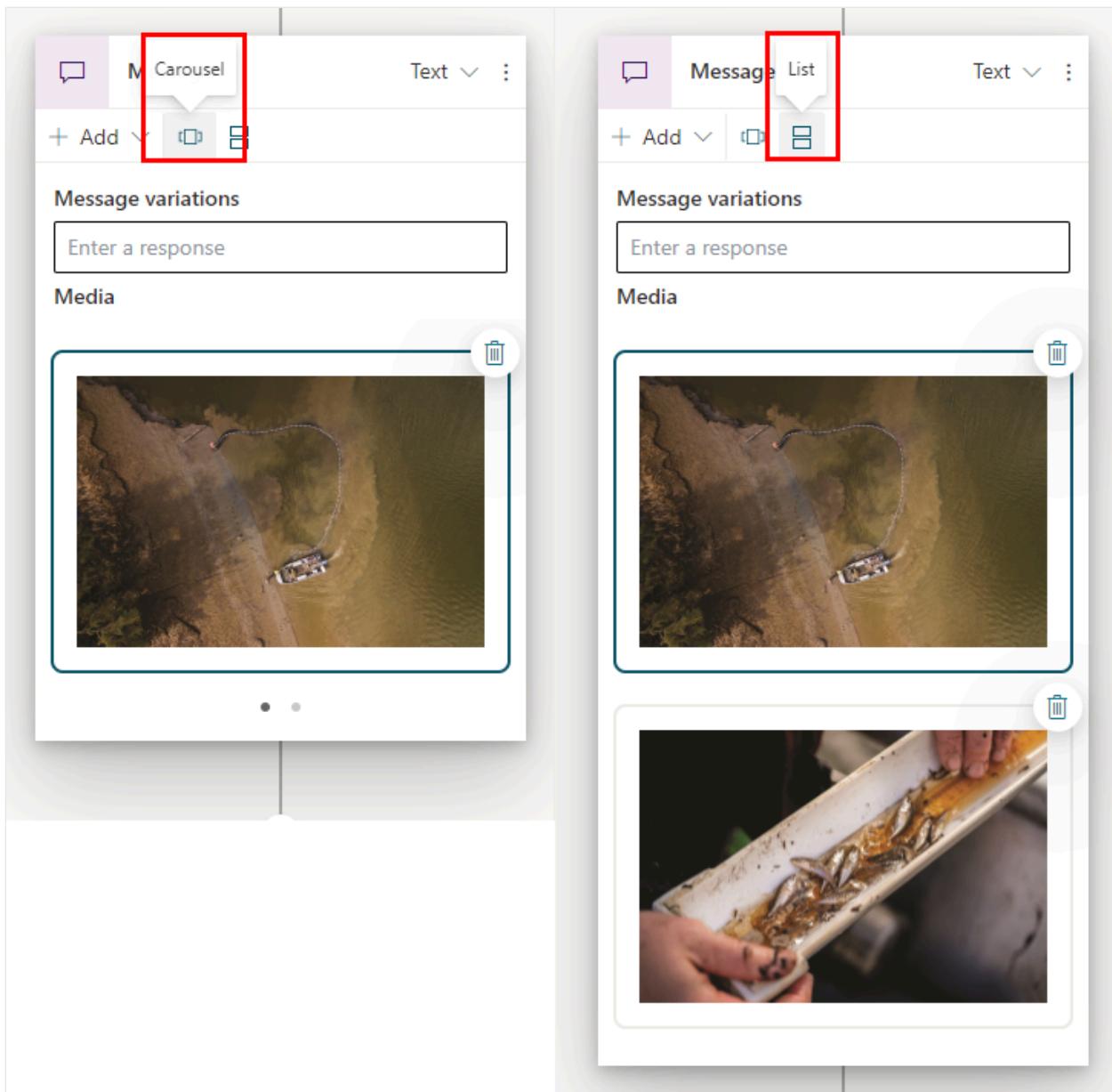
[Edit JSON](#)

```
{ "type": "AdaptiveCard", "$schema": "http://adaptivecards.io/schemas/1.0.json", "version": "1.0", "body": [ { "type": "Image", "url": "https://onmsft.com/wp-content/uploads/2017/09/Windows-10-1024x576.jpg" } ] }
```

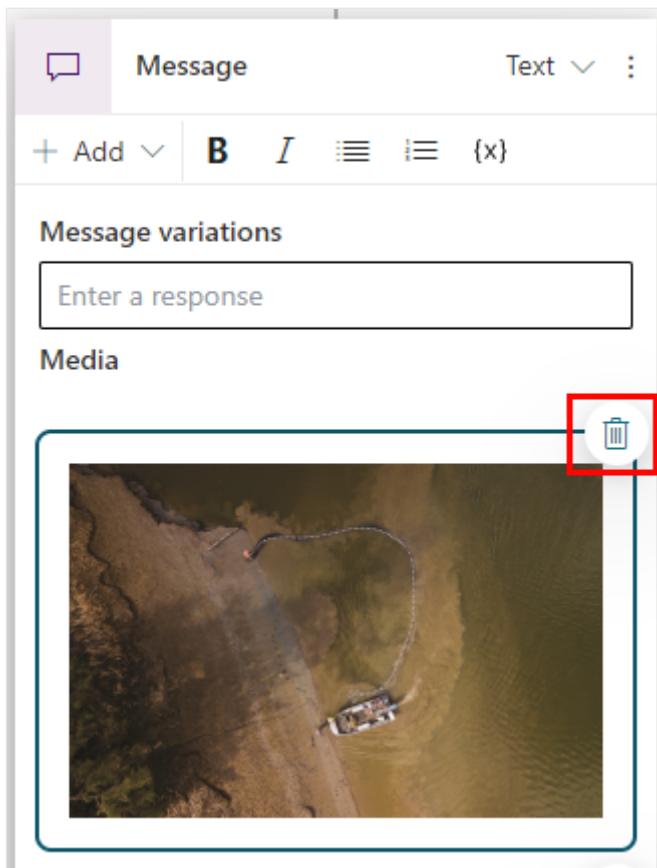
View multiple cards in the same node and remove a card

If you add two or more cards to a node, two display options appear in the node menu of a selected card:

- **Carousel** displays one card at a time.
- **List** displays all cards in a vertical list.



To remove a card from a node, select the **Remove** icon (trash can) at the upper-right corner of the item.



Use quick replies

ⓘ Important

Not all channels support quick replies, which disable them for the user.

Some channels limit the number of quick replies shown at once.

Add quick replies to a message suggests responses or actions for the user. Actions taken by the user get sent back to the agent.

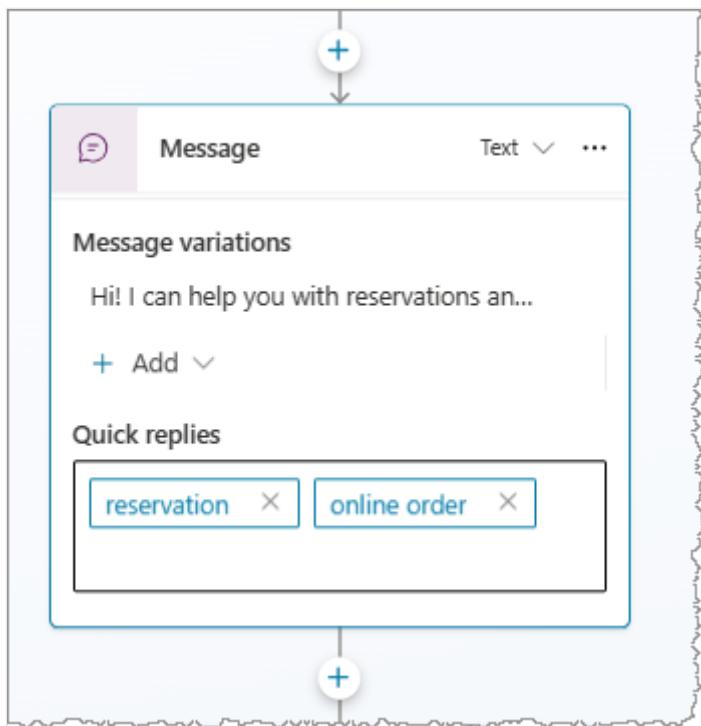
If the user chooses a quick reply text option, it gets shown in the chat history. The quick reply buttons are removed from the chat history when the agent or user sends another activity.

Users can choose to use a quick reply or ignore it by typing 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. Add a Message node.

2. In the node's menu bar, select **Add** and choose **Quick reply**.
3. In the **Quick replies** box on the **Message** node, enter the text for one or more quick replies and press Enter.



Change quick reply type

You can change the properties of quick replies to use other types of actions.

1. Select the **Quick replies** box or an individual quick reply to open the **Quick Replies properties** pane.

The screenshot shows the Microsoft Teams message builder interface. On the left, a preview window displays a message card with a 'Message' section containing the text 'Hi! I can help you with reservations an...' and a 'Quick replies' section with two options: 'reservation' and 'online order'. On the right, the 'Quick replies properties' pane is open for two separate configurations:

- Top Configuration:** Type: Send a message; Text: reservation; Title: Enter or select a value.
- Bottom Configuration:** Type: Send a message; Text: online order; Title: Enter or select a value.

Both configurations have an 'Advanced options' section and a '+ Add' button at the bottom.

2. Select a value for **Type**:

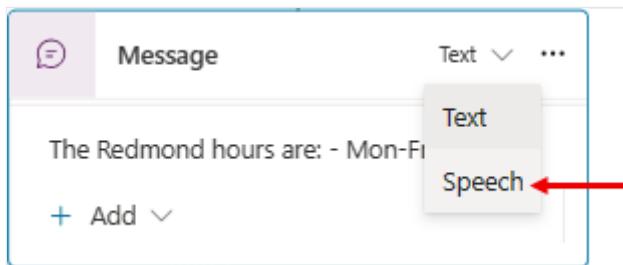
- **Send a message** (default): Send a message to the agent and show it in the chat history.
- **Open URL**: Open a URL. The URL must start with `https://`.
- **Make a call**: Call a phone number. Your entry must be in the format `tel:1234567890`.
- **Send a hidden message to the agent**: Send a message to the agent, but hide it from the chat history.

The properties that are shown change based on the type of quick reply you selected. More options can be found under **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 a **Message** node. In the upper-right corner, select **Text**, and then select **Speech**.

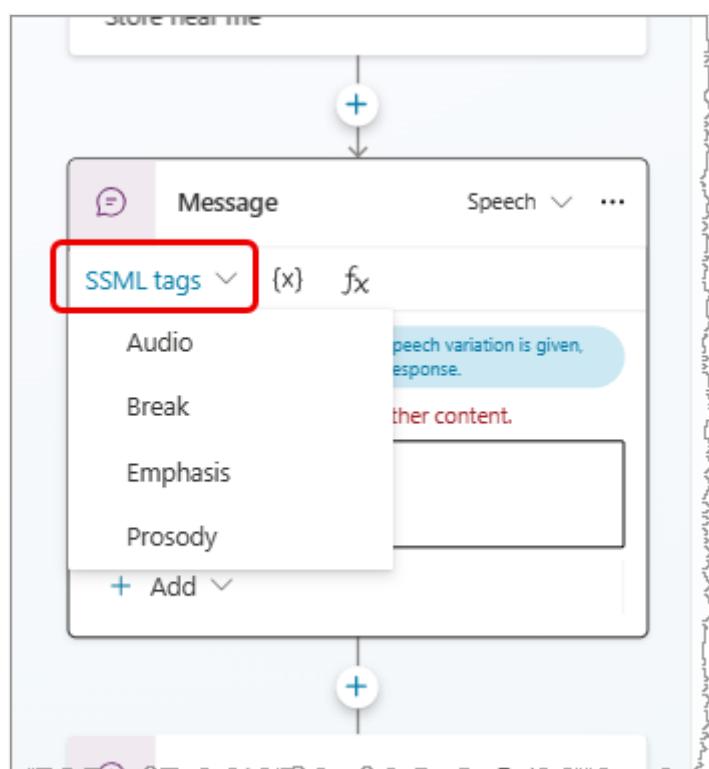


2. Add a message variation to the **Message** card. You see the menu bar changes.
3. Enter what you want the agent to say. Add as many variations for the speech response as needed.
4. Optionally, add SSML to your message variations to control how the agent speaks the message.

Use SSML to customize speech responses

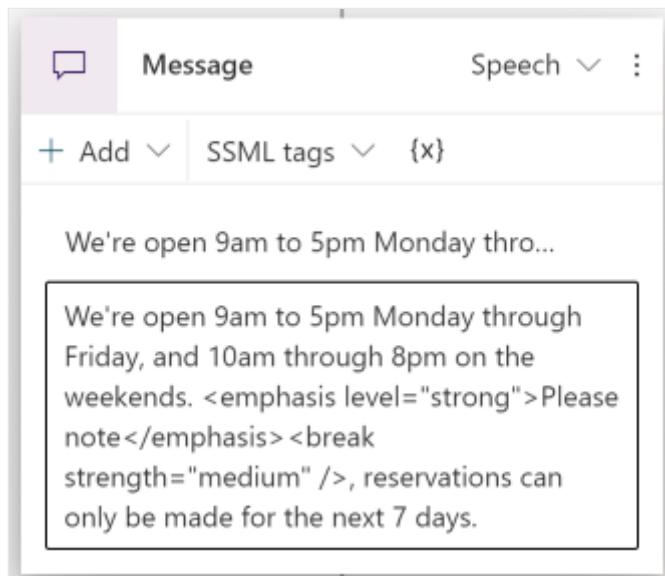
Speech Synthesis Markup Language (SSML) is an XML-based markup language that specifies how text is converted into synthesized speech. SSML is one of the many ways Copilot Studio makes your agent's responses sound like natural speech.

In the **Message** node menu bar, select **SSML tags**, and then select the tags you want to add. You can add multiple tags to a single speech variation.



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.

SSML example



💡 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](#).

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Ask a question

Article • 10/21/2024

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.

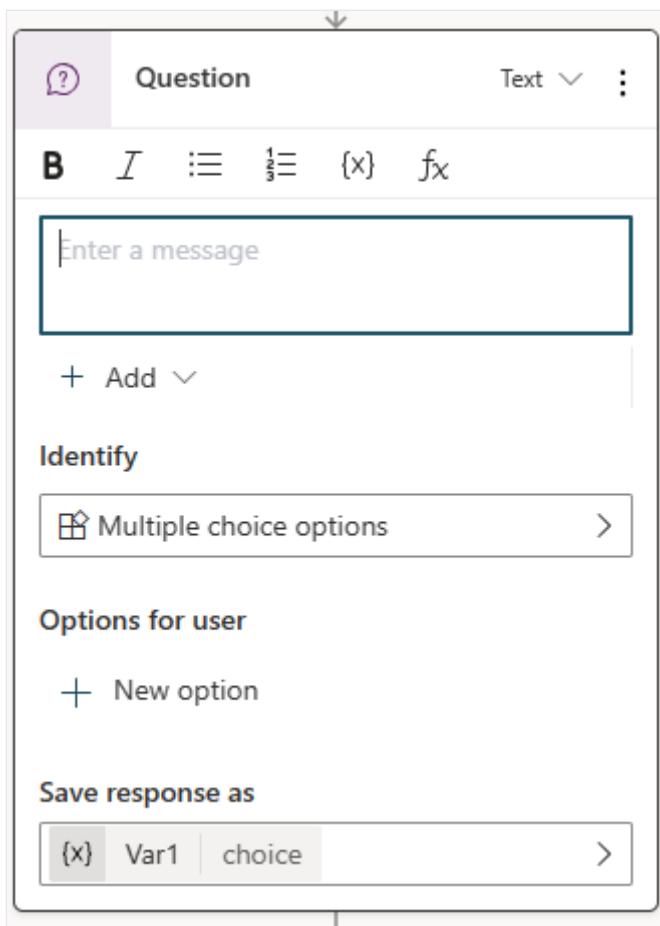
Node names can be up to 500 characters in length.

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 copilot should identify in the user's response. You can select an existing entity or create a new one if needed. [Learn how to use entities in a conversation](#).
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 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 copilot 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 copilot 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 copilot 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 copilot 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 copilot 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. 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 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.
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Ask with Adaptive Cards

Article • 11/19/2024

[Adaptive Cards](#) enable you to 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, as a JSON representation. Because they're platform-agnostic, you can easily tailor Adaptive Cards to your needs.

You design Adaptive Cards using the [Adaptive Cards Designer](#) or author directly using JSON code. The **Adaptive Card** node allows you to send an Adaptive Card containing one or more submit buttons, and optionally, one or more form input fields. Copilot Studio then stores user responses in variables for use later in the conversation.

The node allows you to provide the JSON for the card you want to send to the user or provide a [Power Fx formula](#) to represent the card, allowing you to include dynamic information on the card. [Other properties](#) on the node allow you to control behavior, such as what to do when the user enters an invalid response and if the node can be interrupted.

The **Adaptive Card** node is used 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.

Prerequisites

- [Create and edit topics](#)

Add an Adaptive Card node

1. Select **Add node** (+), then select **Ask with Adaptive Card**.
2. Select the More icon (...) of the node, and then select **Properties**.
The properties pane of the node appears.
3. In the **Edit JSON** section, enter the JSON for your card.

To open a larger view of the JSON editor, select the **Expand** icon.



```
{  
  "type": "AdaptiveCard",  
  "$schema": "http://adaptivecard.io/schemas/adaptive-card.json",  
  "version": "1.0",  
  "body": []  
}
```

Edit schema

💡 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 to the agent user, you should add your card as an attachment to a **Message** node.

4. After adding the JSON code for your card to the editor, select outside of the editor.
The node is updated with a preview of your card. Copilot Studio automatically creates output variables based on the inputs detected within the card.

Adaptive Card

Tell us about yourself

We just need a few more details to get you booked for the trip of a lifetime!

Don't worry, we'll never share or sell your information.

Your name (Last, First) *

Your email *

Phone Number (xxx xxx xxxx)

Submit

Outputs (4) ^

{x} actionSubmitId | string =

{x} actionSubmitId | string >

{x} myEmail | string =

{x} myEmail | string >

{x} myName | string =

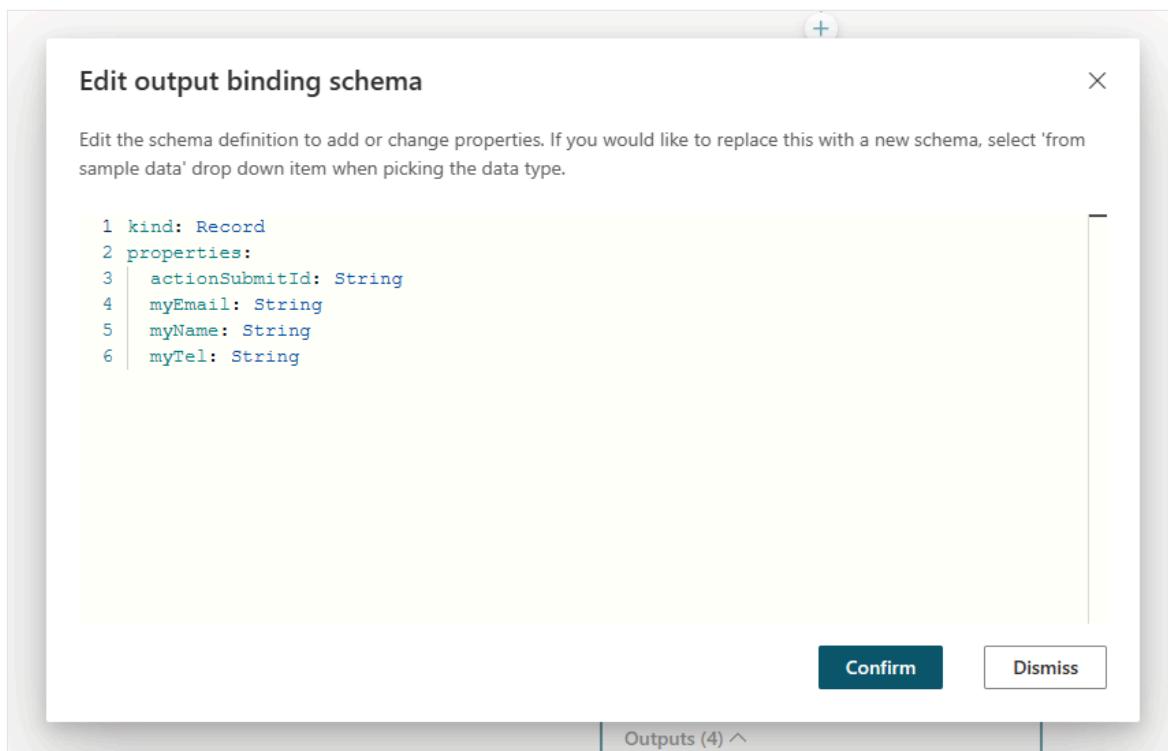
{x} myName | string >

{x} myTel | string =

{x} myTel | string >

Tip

If the output variables generated for your card are incorrect, you can manually update the list of variables and their types by selecting the **Edit Schema** button underneath the JSON editor in the property pane. This example defines four string (Text) variables.



5. Your interactive Adaptive Card is now ready. When a user of your agent uses a submit button on a card, the output variables are populated with their responses.

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 currently awaiting a submission from an Adaptive Card and the user sends a text message instead, this is considered to be an invalid response, *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 checked (default), an incoming message from a user when the agent is awaiting a card submission triggers an interruption and switch to another topic. If a topic switch occurs, the card is sent again to the user once the interrupting topic has completed.

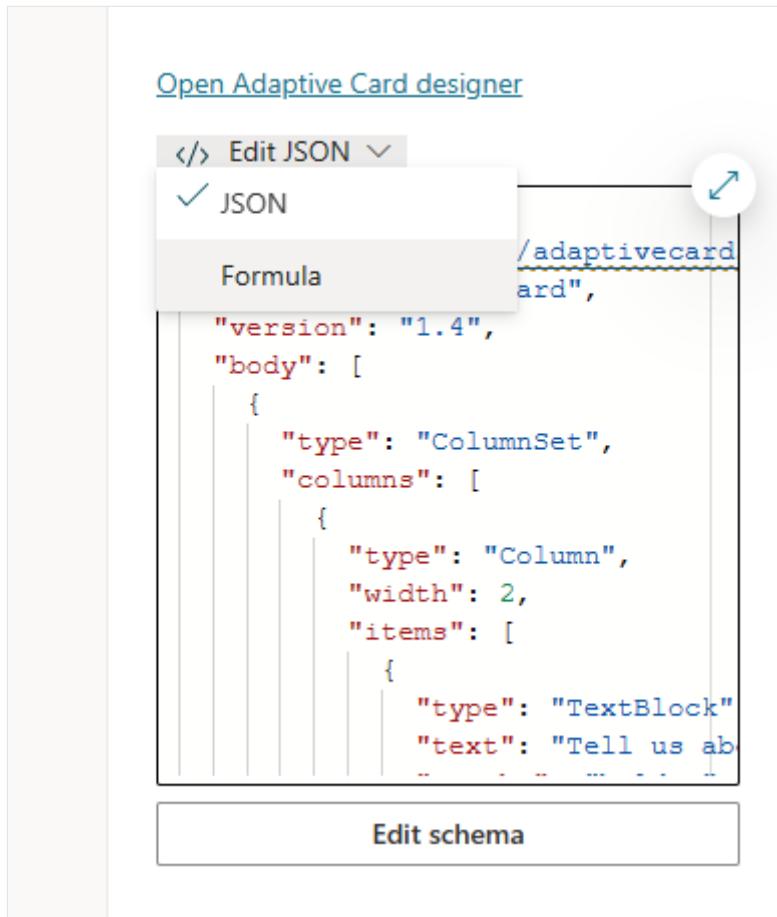
Use Power Fx to make your card dynamic

You can use Power Fx to include dynamic information on your card by referencing one or more variables from your topic or agent within the card.

1. Select the More icon (...) of the node, and then select **Properties**.

The properties pane for the node appears.

2. Select the **Edit JSON** button, then choose **Formula**. Selecting **Formula** automatically converts the JSON in the card into a Power Fx representation.



JSON example

Here's a JSON example of a card and the resulting Power Fx, where two variables **Topic.Title** and **Topic.Subtitle** are used instead of a hard-coded text.

```
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",
          "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"
    }
  ]
}

```

Power Fx

JSON

```
{
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "type": "AdaptiveCard",
  "version": "1.5",
  "body": [
    {
      "type": "TextBlock",
      "text": "Tell us about yourself",
      "weight": "Bolder",
      "size": "Medium",
      "wrap": true,
      "style": "heading"
    },
    {
      "type": "TextBlock",
      "text": "We just need a few more details to get you booked for the trip of a lifetime!",
      "isSubtle": true,
      "wrap": true
    },
    {
      "type": "Input.Text",
      "id": "myName",
      "label": "Your name (Last, First)",
      "isRequired": true,
      "regex": "^[A-Z][a-z]+, [A-Z][a-z]+$",
      "errorMessage": "Please enter your name in the specified format"
    }
  ],
  "actions": [
    {
      "type": "Action.Submit",
      "title": "Submit"
    }
  ]
}
```

```

"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"
    }
  ]
}

```

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

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.

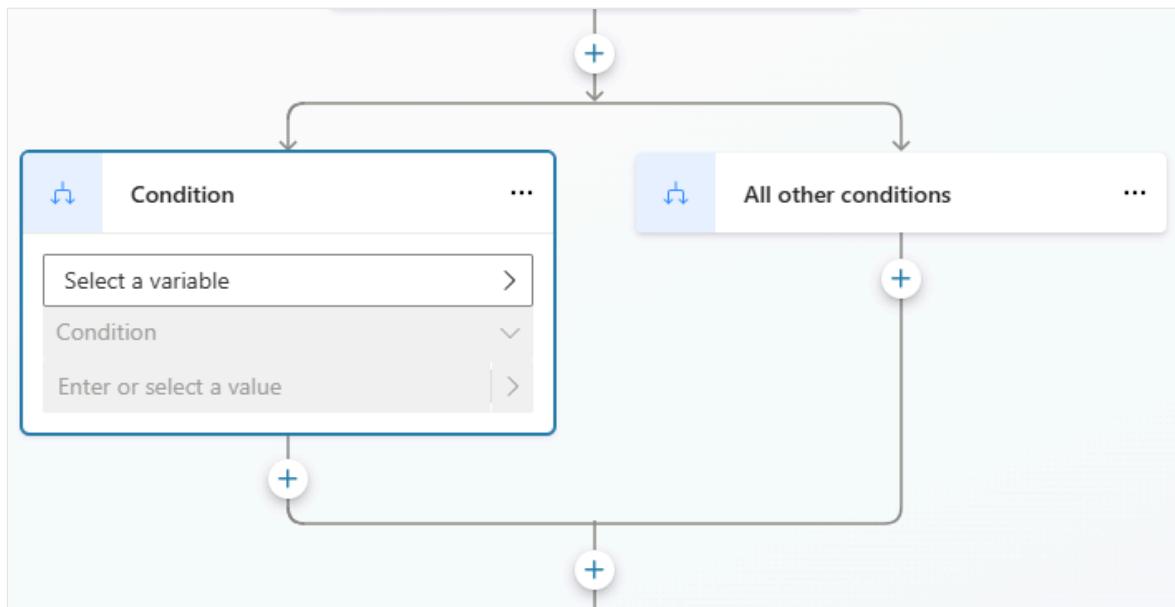
Prerequisites

- [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.

The screenshot shows the 'Condition' configuration screen in Copilot Studio. At the top, there are two parallel condition blocks. The first block has '{x}' selected in the variable dropdown, followed by 'petType' and 'choice'. Below this is a dropdown menu set to 'is equal to', and under it is the value 'cat'. The second block is similar, with 'choice' set to 'dog'. Below these blocks is a section labeled 'Or' with a dropdown menu. At the bottom of the configuration area is a blue plus sign icon followed by the text 'New condition'.

ⓘ 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.

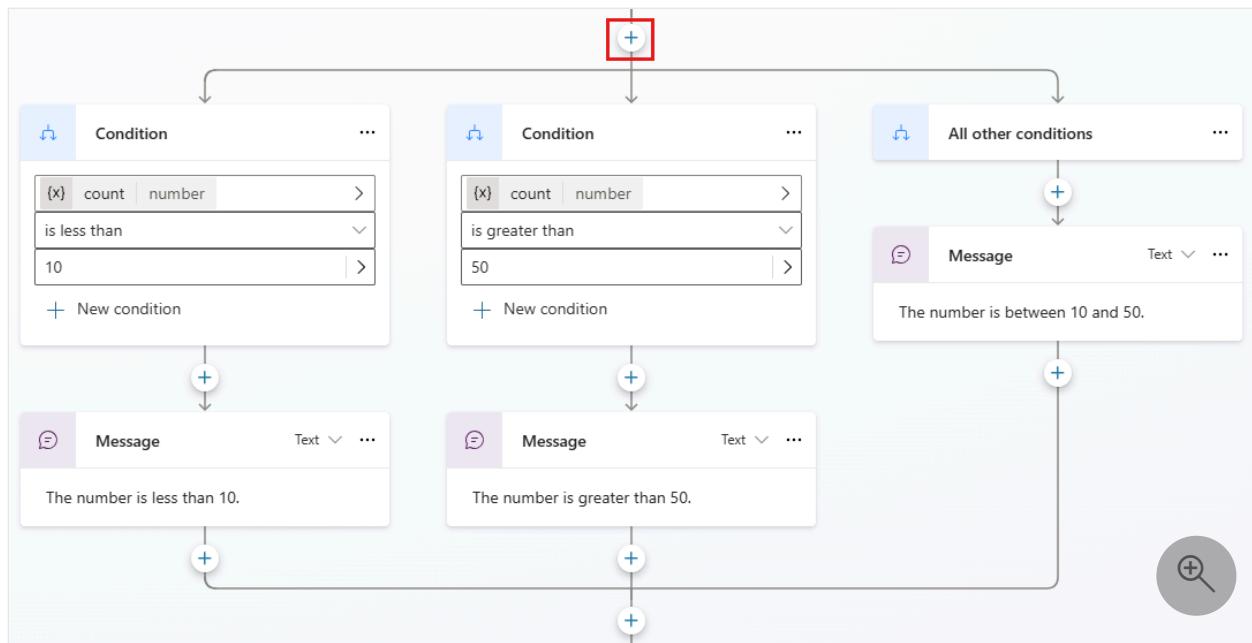
Add more Condition branches

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**.

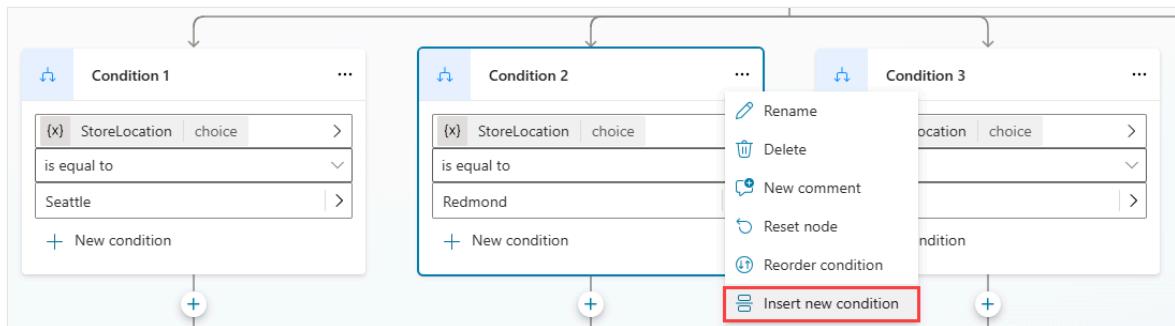


Selecting the **Add node** icon below a condition adds an entire new condition (with a **Condition** node and an **All Other Conditions** node), not a new condition branch.

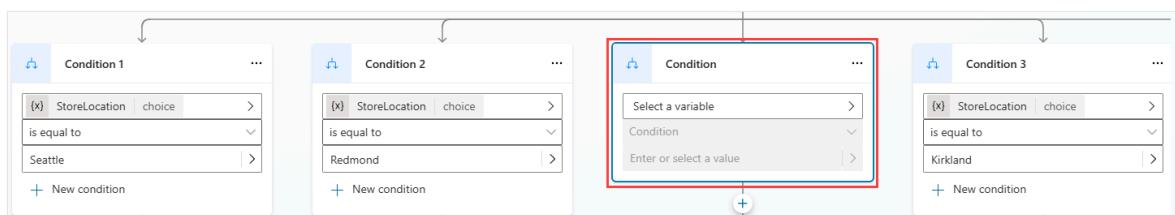
Insert a new condition to a group of conditions

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.



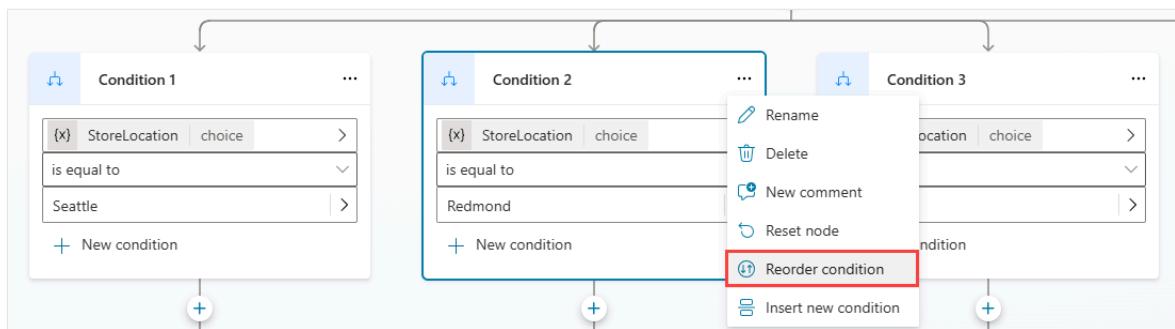
Note

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.

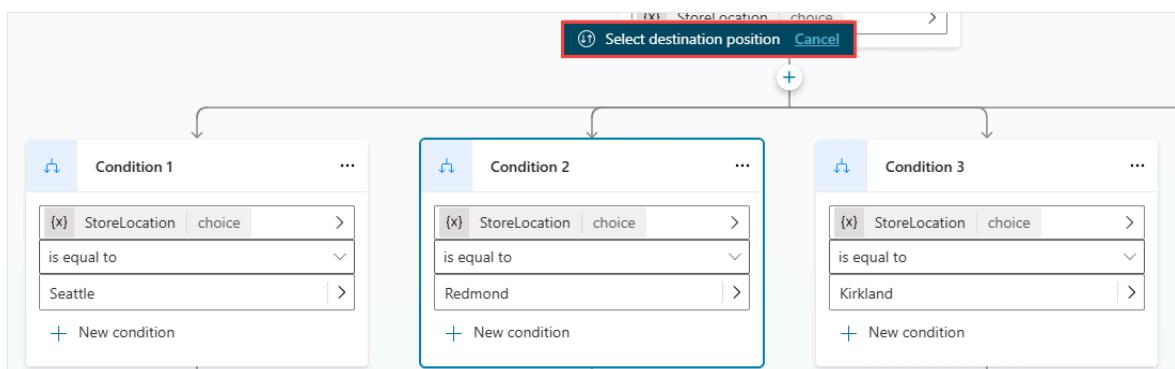
Organize conditions

For more complex condition structures, you can reorder conditions within the same condition group.

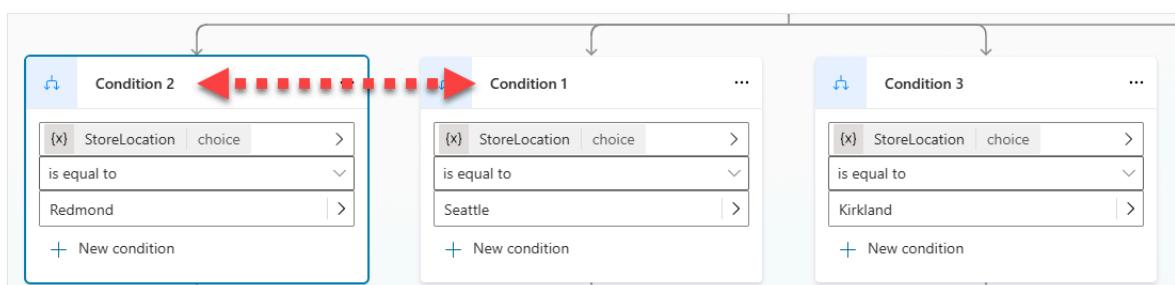
1. Select the **More** icon (...) of the **Condition** node, and then select **Reorder condition**.



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.



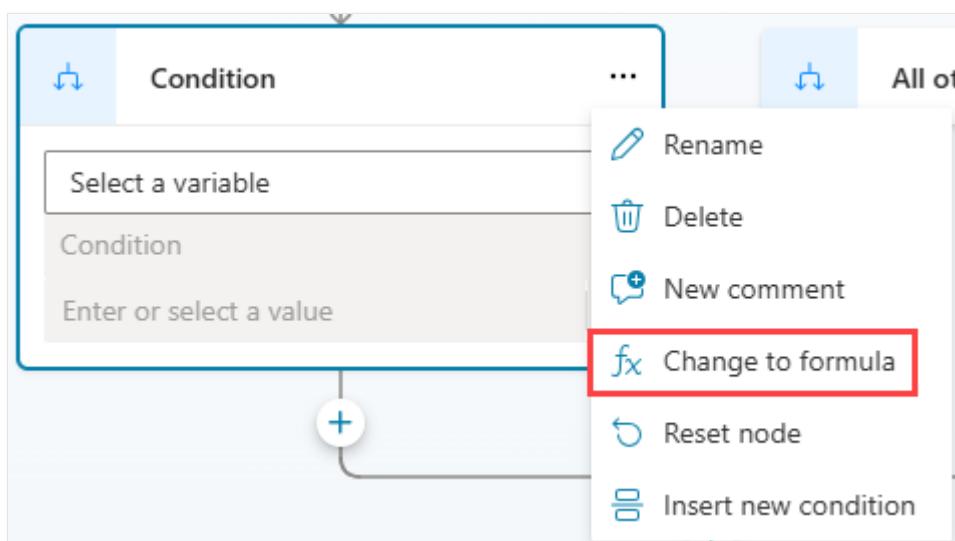
Note

- 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 ↗

Work with variables

Article • 11/19/2024

You can use variables to save 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 `UserName`, 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](#).

Web app

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
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

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.

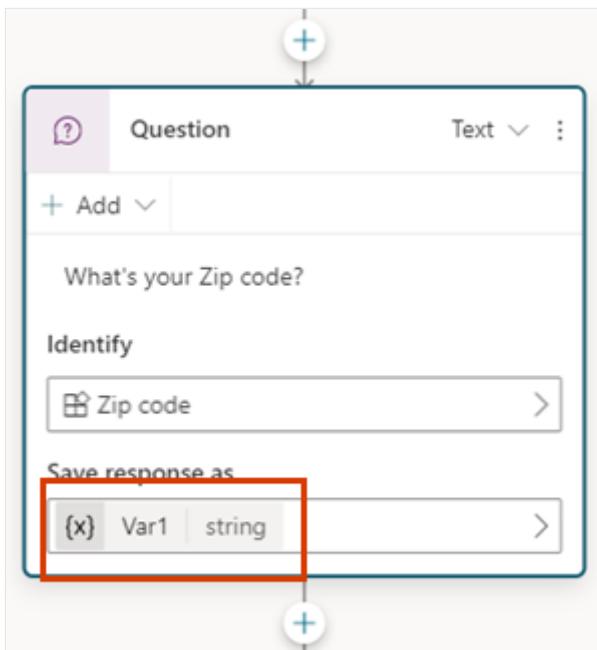
 [Expand table](#)

Entity	Variable Base Type
Multiple-choice options	Choice
User's entire response	String
Age	Number
Boolean	Boolean
City	String
Color	String
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

Entity	Variable Base Type
Weight	Number
Zip code	String
Custom entity	Choice

Create a variable

Any node that prompts you to select a variable as an output, such as a **Question** node, automatically creates an output variable of the appropriate type.



💡 Tip

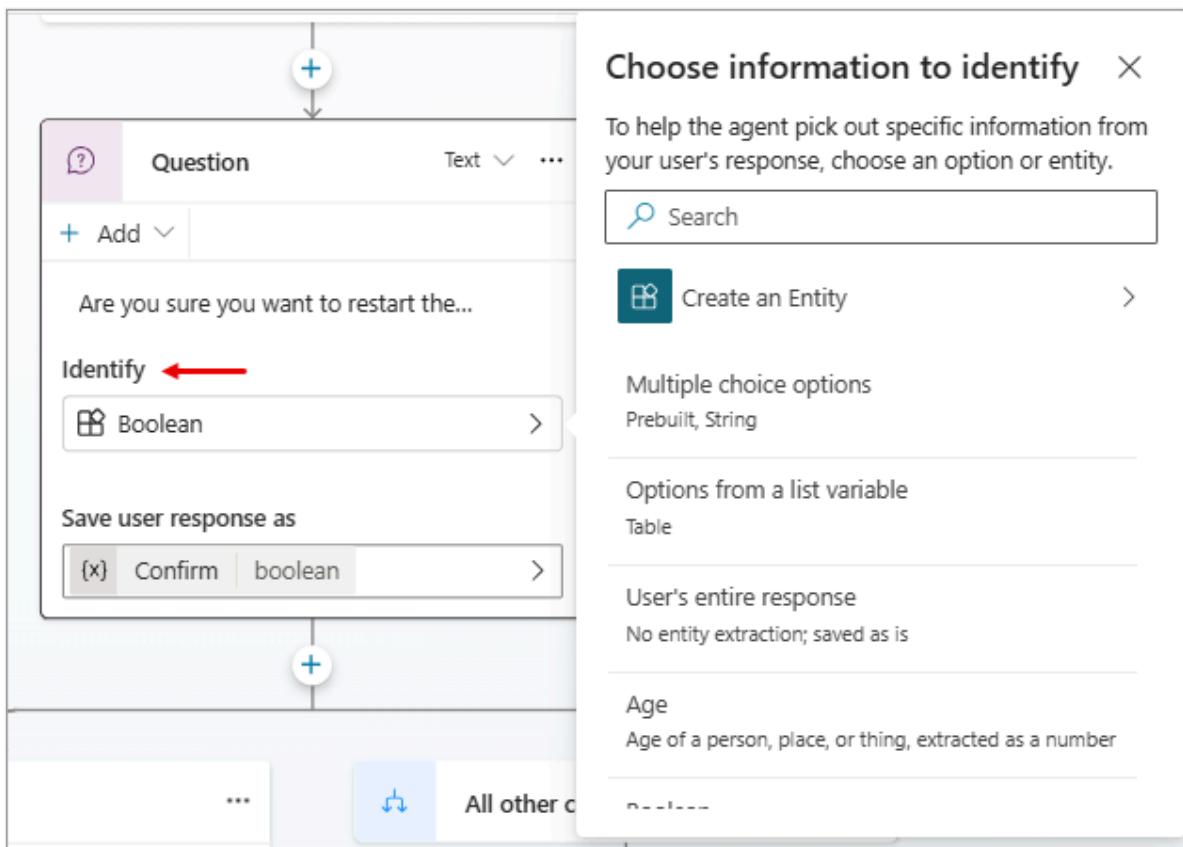
Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

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

Question nodes are created with multiple-choice options by default. To use a different prebuilt or custom entity, select the **Identify** box and choose the type of information the agent should listen for.



Rename a variable

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 else who must maintain your agent.

1. Select the variable to open it in the [Variable properties](#) pane.
2. Under **Variable name**, enter a new name for your variable.

Set a variable

Typically you use a **Question** node to save user input to a variable. There might be situations where you want to set the value yourself, however. In those cases, use a **Set Variable Value** node.

1. Select **Add node (+)** to add a node, and then select **Variable management > Set a variable value**.
2. Select the box under **Set variable**, and then select **Create a new variable**.

The screenshot shows the 'Select a variable' dialog on the left and a flow editor on the right. In the dialog, the 'Custom' tab is selected, showing a search bar and a button to 'Create a new variable'. A red box highlights this button. Below it, several variables are listed: Var1 (Topic.Var1), string; Var2 (Topic.Var2), string; VarActivityTime (Topic.VarActivityTime), time; VarCampType (Topic.VarCampType), choice; and VarMaxPrice (Topic.VarMaxPrice), number. On the right, a flow step titled 'Set variable value' is shown. It has a 'Set variable' section where 'Var1' is selected, and a 'To value' section with a red box around the 'Enter or select a value' input field, which displays the error 'Variable selection is missing'. A red arrow points from the 'Create a new variable' button in the dialog to the 'Select a variable' input field in the flow step.

A new variable is created. Its type is **unknown** until you assign a value to it.

The screenshot shows the 'Set variable value' step in the flow editor. It has a 'Set variable' section where 'Var3' is selected with the type 'unknown'. Below it is a 'To value' section with a red box around the 'Enter or select a value' input field, which displays the error 'Variable selection is missing'.

3. For **To value**, assign a value using one of the following options:

- Type a [literal value](#).
- Select an existing variable of the same type. This action sets your variable to the same value as the variable you select.
- 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.

① Note

While setting a string type variable, if you encounter a mismatch between **Set variable** and **To value**, an error message and **Fix type mismatch** button appears. Select the button to have Copilot Studio automatically generate a [Power Fx formula](#) that resolves the mismatch.

Use literal values

Instead of selecting a variable value, you can enter a literal value into any variable.

The screenshot shows the Copilot Studio interface with two examples of using literal values:

- Set variable value node:** A node titled "Set variable value" with a "Set variable" input set to "productName" (string) and a "To value" input set to "Copilot Studio".
- Redirect node:** A node titled "Redirect" with an "Inputs (1)" input set to "userNmae" (boolean) with the value "John Doe".

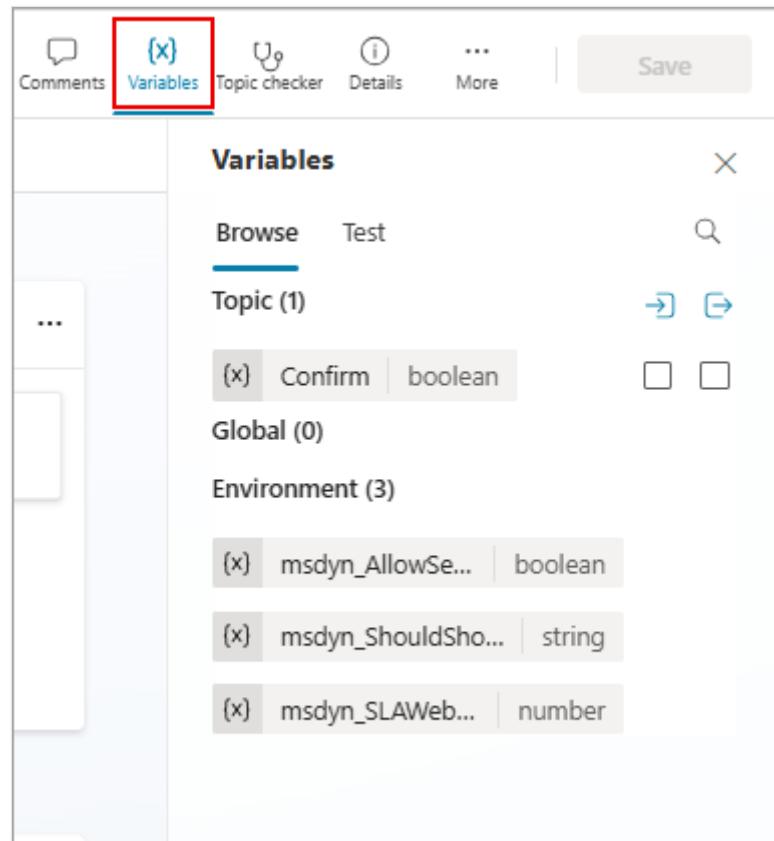
The node attempts to interpret literal values as a string, a number, or a Boolean. For example, `123` is interpreted as a number. If you want it to be interpreted as a string value instead, you can wrap the value in double quotes, like this: `"123"`.

For some scenarios, or where you're using more complex types, use a [Power Fx formula](#) to set a specific type.

Variables pane

The **Variables** pane is where you can view all the variables that are available in the 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](#) pane.

To open the **Variables** pane, in the topic's menu bar, select **Variables**.



Variable properties pane

In the **Variable properties** pane, you can rename a variable, see where it's used, or convert it to a [global variable](#). You can't convert it from a global variable back to a topic variable, however. You can also select whether it can receive values from or pass its value to other topics.

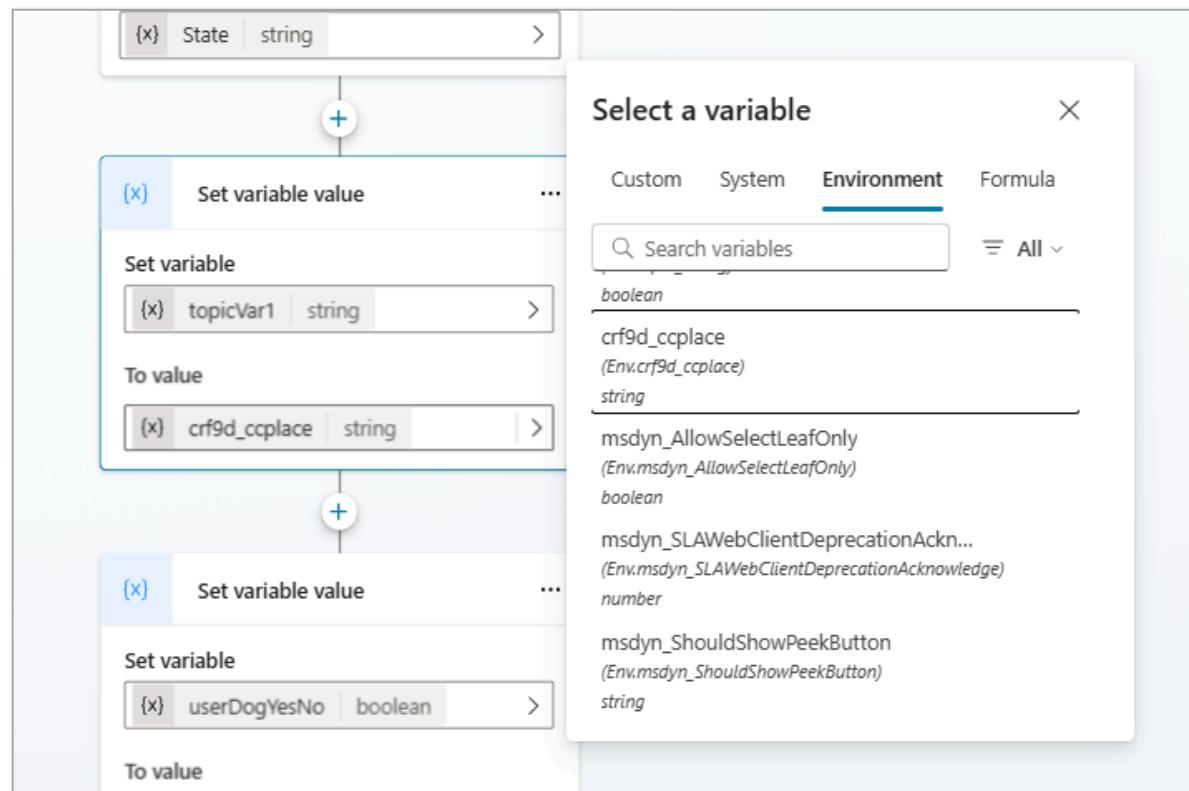
To open the **Variable properties** pane, select a variable in the **Variables** pane. You can also open the **Variable properties** pane by selecting a variable in any node.

The screenshot shows the Power Platform Copilot Studio interface. On the left, a 'Question' component is displayed with the text 'I can help you with outdoor activities....'. Below it, under 'Identify', there is a button labeled 'Camping activities'. Under 'Select options for user', there are three buttons: 'hiking', 'cooking', and 'fishing'. Under 'Save user response as', there is a button labeled '(x) VarCampType choice'. A red arrow points from this button to the 'VarCampType' entry in the 'Variables' panel on the right. The 'Variables' panel has tabs for 'Browse' and 'Test', with 'Topic (6)' selected. It lists variables: productName (string), Var1 (string), Var2 (string), VarActivityTime (time), VarCampType (choice), and VarMaxPrice (number). The 'VarCampType' entry is highlighted with a red border.

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 you can only use environment variables in *read-only* mode. Agent authors cannot modify the environment variable in Copilot Studio. However, administrators can change the value of environment variables in the PowerApps portal.



Environment variable types in Copilot Studio map to PowerApps data types as follows:

[Expand table](#)

Type in Copilot Studio	Type in PowerApps
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

① Note

Environment variable errors are visible in the *Test* chat and when publishing. However, these errors are not shown in the *Topic* list because they are not topic variables.

You cannot author or edit environment variables in Copilot Studio. You must use the PowerApps portal to change or set environment variables. From Copilot Studio, you can use the **Variables** pane to see information about an environment variable. The **Variables** pane also has a link to the native authoring experience.

Use environment variables for Azure Key Vault secrets

An environment variable can reference a secret in the Key Vault. This is a special case of environment variables with unique considerations.

To create a secret environment variable in the PowerApps portal, you must [configure its Key Vault](#).

In addition to the configuration above, you must perform the following actions to authorize Copilot Studio to read this Key Vault:

1. Assign the `Key Vault Secrets User` role to the `Microsoft Virtual Agents 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 comma.
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 to all users who can edit the agent in the environment because such users could edit the agent to return the value in the `SendMessage` node.

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 with their descriptions and types:

- 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)

ⓘ 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 will be 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

Name	Type	Definition
Conversation.Id	string	The unique ID of the current conversation
Conversation.InTestMode	boolean	Boolean flag that represents if the 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. Note: This variable is supported only if the trigger is <code>On Error</code> .
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

Name	Type	Definition
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 return 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. 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> .
User.Language	choice	This variable is used to set the user language locale per conversation.
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> .

Variables for integrated authentication (default)

The following variables are available when you have configured your agent to authenticate with Microsoft. This will use Entra ID authentication and only allow Teams and PowerApps channels. This 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.
User.LastName	string	The last 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 when you have configured your agent 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 when you have configured your agent 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 the agent is authenticated or not.
User.LastName	string	The last 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 voice-enabled agents

The following variables are only available in 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 ASR hypothesis entire result, 0 to 1.
Activity.SpeechRecognition.MinimallyFormattedText	string	Slightly formatted text of the ASR hypothesis result. For example, Five hundred dollars. Words are spelled out, but basic

Name	Type	Definition
		capitalization and punctuation are included.
<code>Activity.UserInputType</code>	choice	The input type from the customer's most recent input. The value can be either text, speech or DTMF.
<code>Conversation.OnlyAllowDTMF</code>	boolean	Boolean flag that represents whether the IVR should be set to DTMF-only mode at runtime.
<code>Conversation.SipUuiHeaderValue</code>	string	The UUI header string used to pass context into IVR on call start.

Pass variables between topics

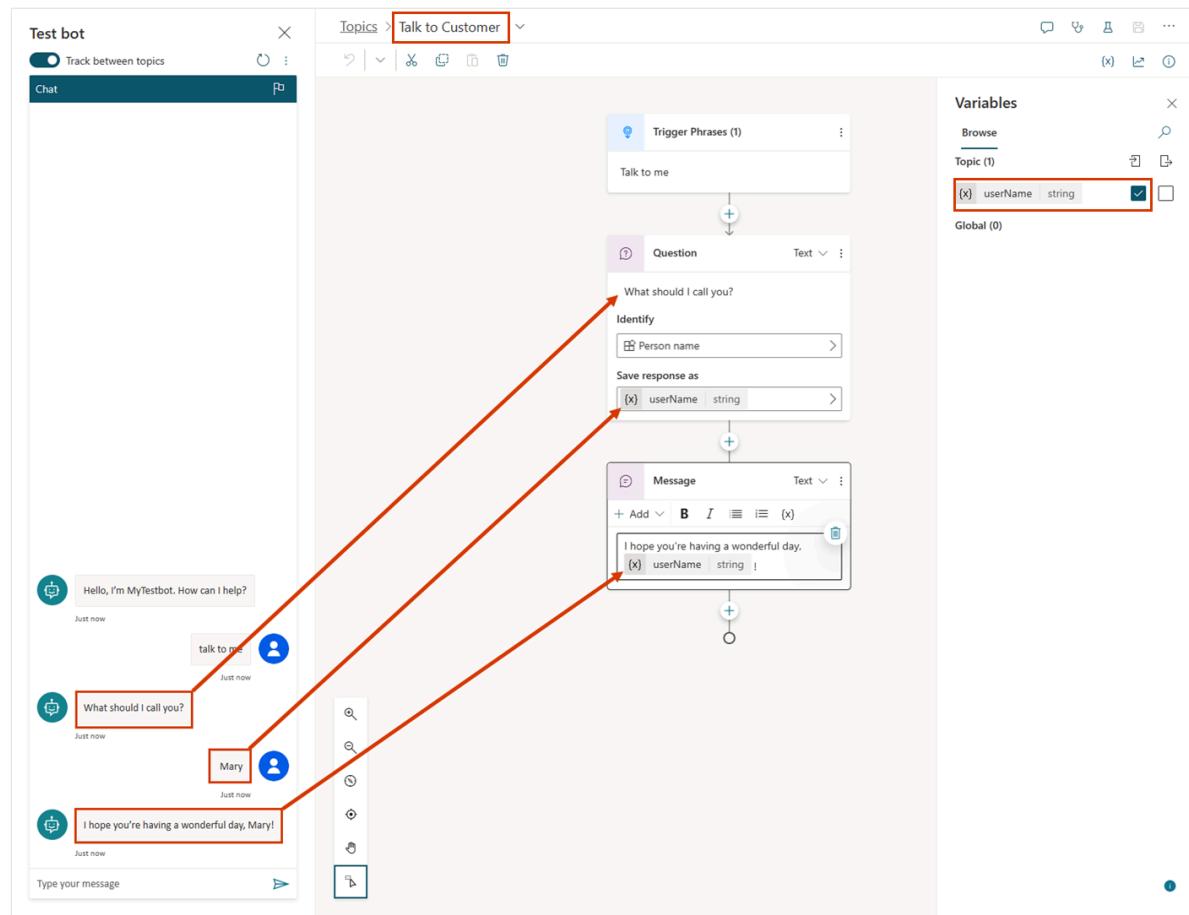
When you redirect one topic to another, you can pass the values of variables between the original topic and the destination topic. Passing variables between topics is especially useful when an earlier topic already collected information that a later topic needs. Your users will appreciate not having to answer the same questions again.

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 has already acquired the value in an earlier topic, there's no reason to ask the question again. In these 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.

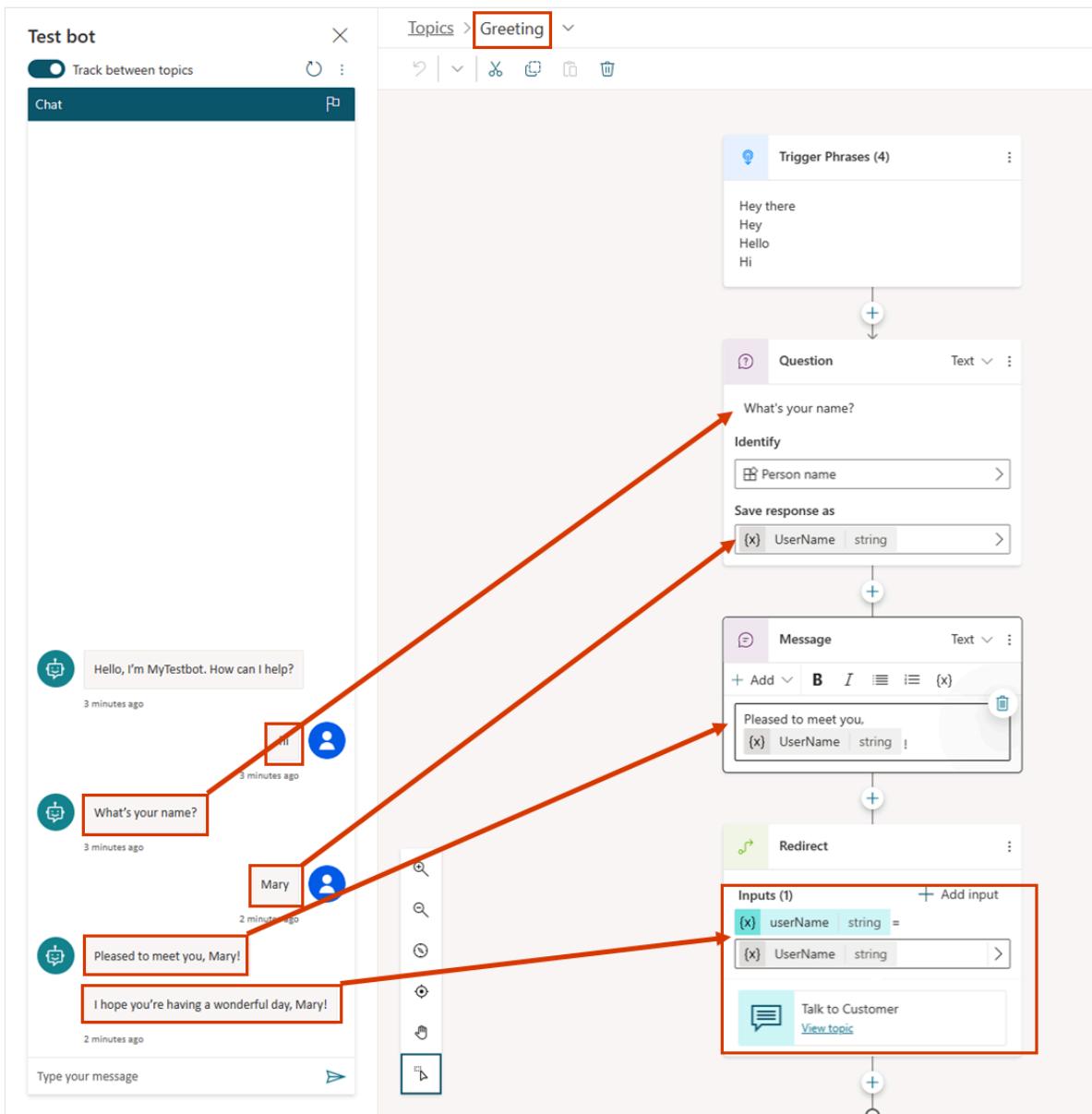
In this example, we'll use two topics, Greeting and Talk to Customer. Both topics ask for the customer's name. However, if the Greeting topic runs first, the Talk to Customer topic skips its question. Instead, it uses the value of the variable that's passed from the Greeting topic.

Here's the flow of the Talk to Customer topic:



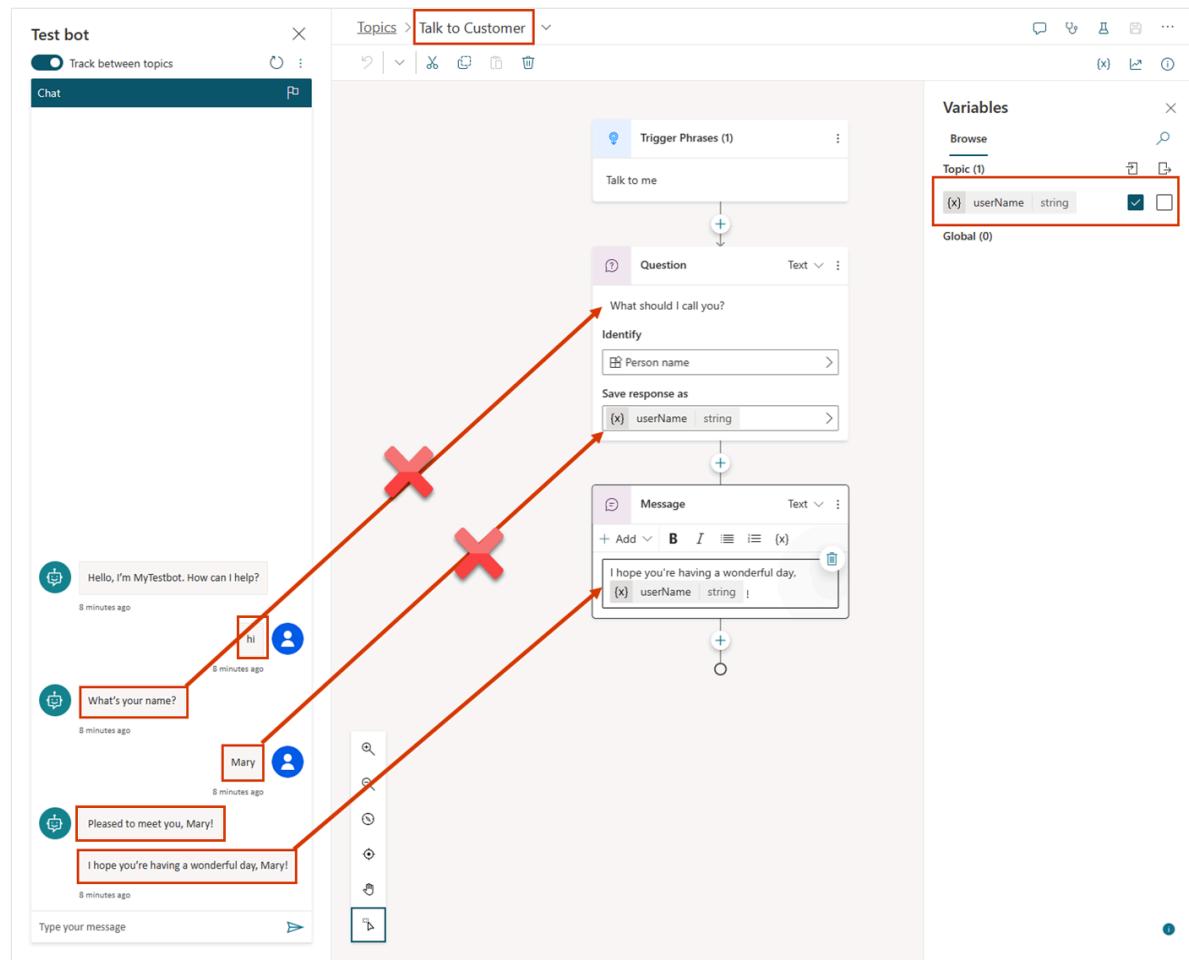
As shown in the Test agent pane, if this topic is triggered first, it asks the user, "What should I call you?" It stores the value in a string variable called `userName`. The `userName` variable is also set to get its value from other topics. The topic concludes with the message, "I hope you're having a wonderful day, {user_name}!"

Here's the flow of the Greeting topic:



As shown in the Test agent pane, if this topic is triggered first, it asks the user, "What's your name?" It stores the value in a string variable called `UserName`. The topic sends the message, "Pleased to meet you, {UserName}!" It then redirects to the Talk to Customer topic, which sends the message, "I hope you're having a wonderful day, {userNName}!" Note, however, that the Talk to Customer topic skipped asking for the user's name again. Instead, it used the value of the `UserName` variable passed from the Greeting topic.

Finally, here's that second conversation again, this time from the perspective of the Talk to Customer topic:

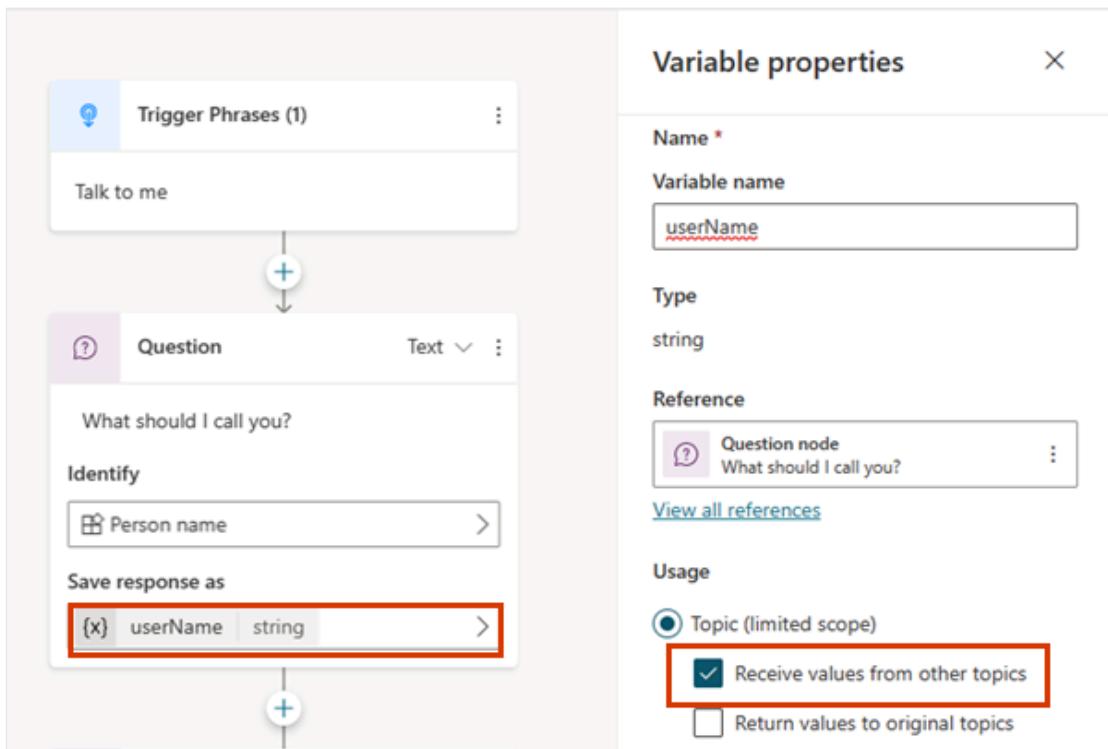


Let's walk through the steps to set up a topic to receive values from other topics. We'll use our current example, but the same steps will work anytime a topic needs to get a value from an earlier topic.

Set up the destination topic

The destination topic is the topic being redirected to, the one that will receive values from other topics. In our example, it's Talk to Customer.

1. Create or go to your destination topic.
2. Add a Question node and enter `What should I call you?` for the message.
3. Under **Identify**, select the prebuilt entity **Person name**.
4. Select the variable to open the **Variable properties** pane. Name it `userName`, and then select **Receive values from other topics**.



5. Add a Message node.

6. In the message box, type `I hope you're having a wonderful day,`.

7. Select the **Insert variable** icon (`{x}`), and then select `userName`.

8. Select the space after the variable and type `!`.

9. Save the topic.

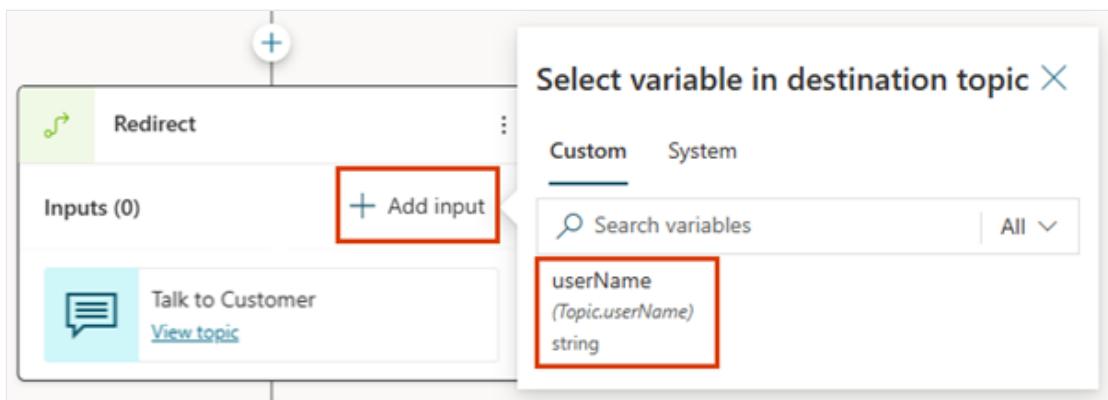
Set up the source topic

The source topic is the topic doing the redirecting, the one that provides the value that will be passed to the destination topic. In our example, it's Greeting.

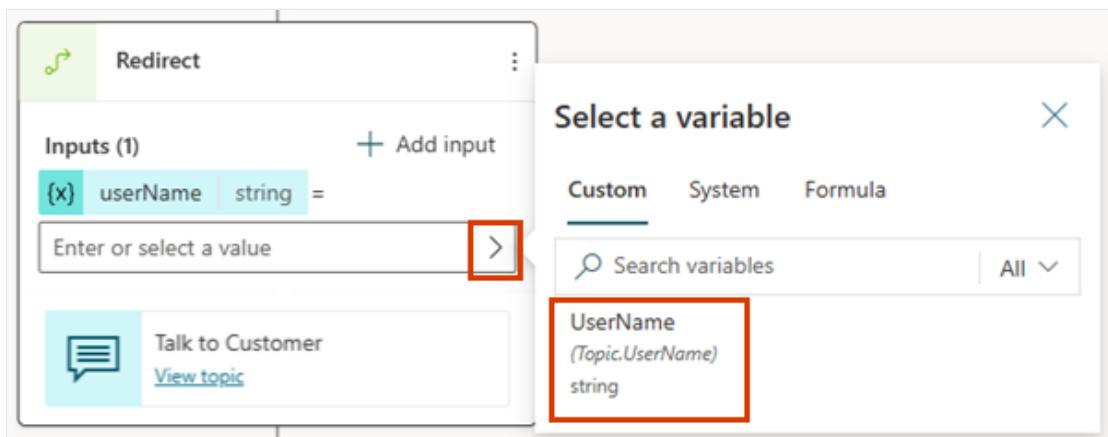
1. Go to the source topic.

2. **Add a Redirect node** and select the destination topic.

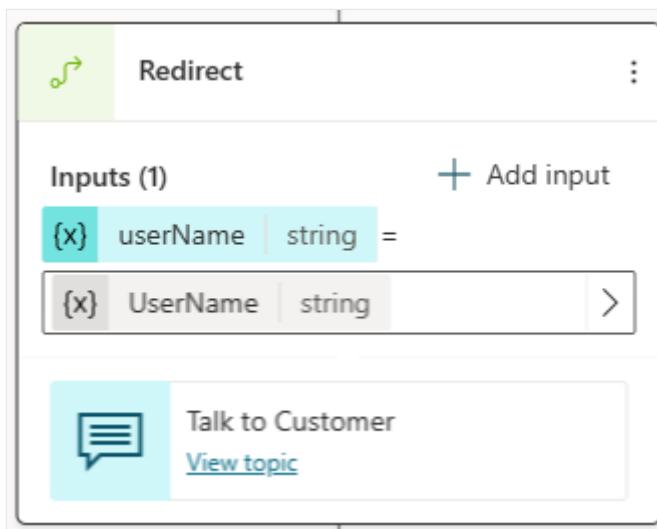
3. Select **+ Add input**, and then select the variable from the destination topic that you want to pass a value to.



4. Select the > icon, and then select the variable whose value you want to pass.



The Redirect node should look like this:



5. Save the topic.

Return values to original topics

When a topic is redirected to and obtains a variable by asking a question or in some other way, the variable can be returned to the original topic. The variable becomes part of the original topic and can be used like any other variable.

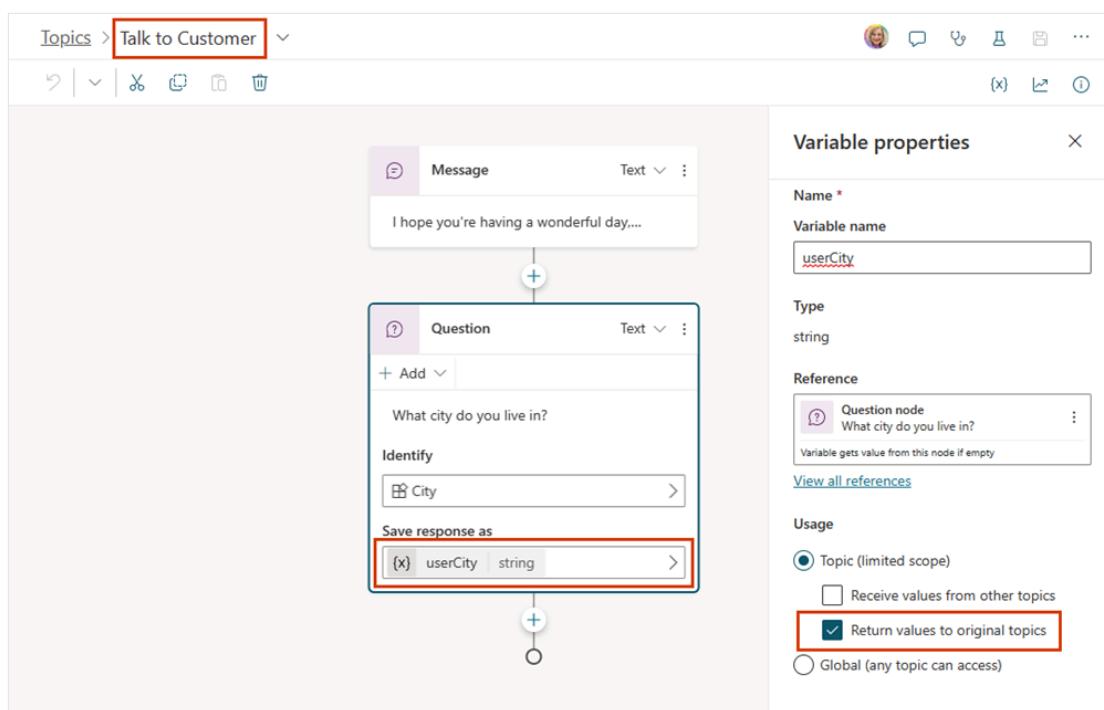
Information the agent obtains is thus available across topics, reducing the need for [global variables](#).

Let's continue with the example from the previous section. We'll 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 being redirected to, the one that provides the value that will be passed back to the original topic. In this example, it's 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** pane. Name it `userCity`, and then select **Return values to original topics**.

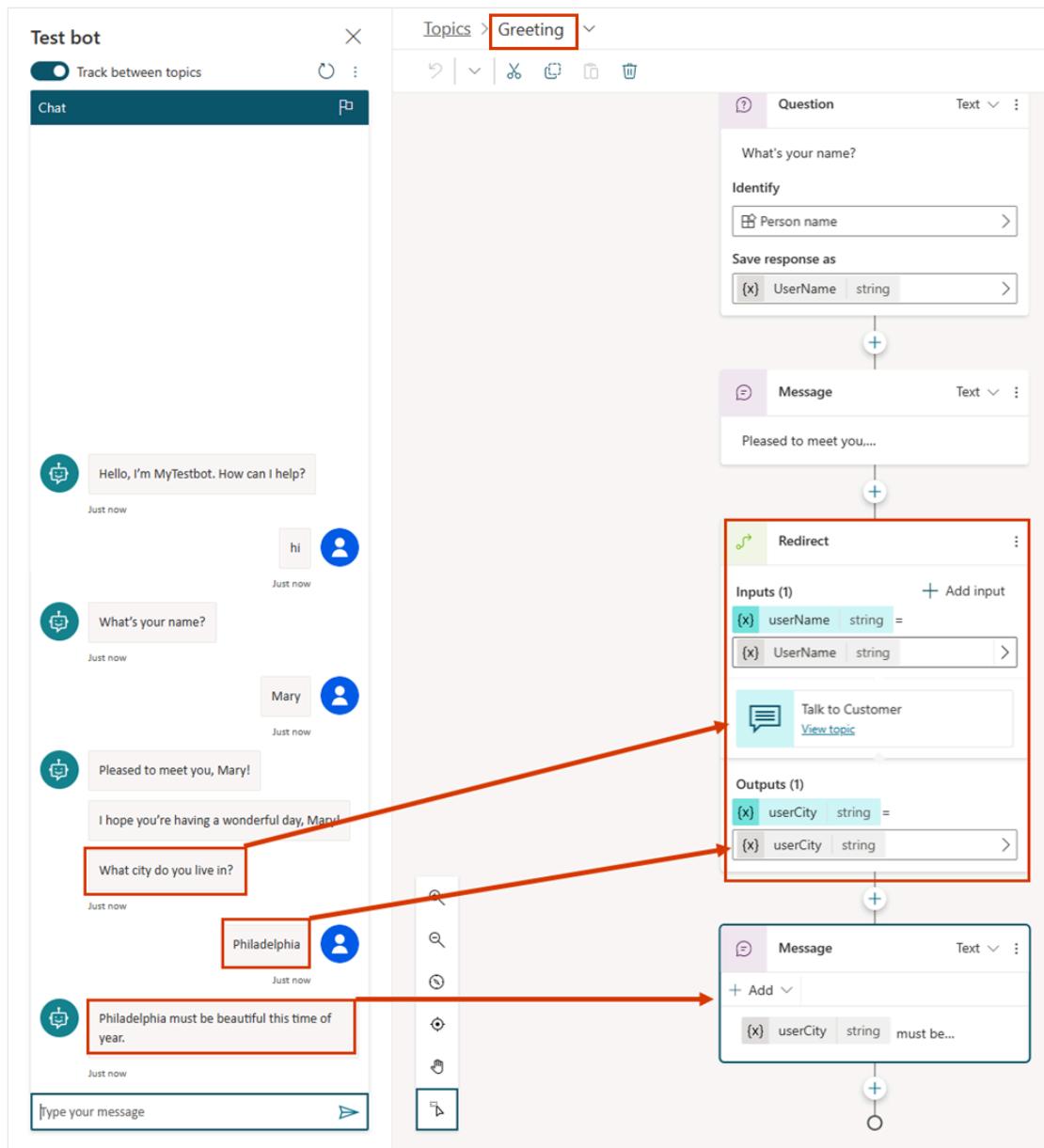


5. Save the topic.

Set up the destination topic for a returned variable

When you're returning a variable to a topic, the destination topic is the topic doing the redirecting, the one that will receive values from other topics. In our example, it's Greeting.

1. Go to the destination topic.
2. The variable you selected in the source topic should appear in the Redirect node as an output variable.



3. Save the topic.

Related content

- Use variables to carry content across topics
- Customize the look and feel of an agent

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Use variables to carry content across topics

Article • 11/19/2024

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 between topics](#). 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.

Web app

Agent variables apply during a single user session. You specify which variables should be treated as agent 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.

1. [Create a variable](#) or [use the Variables pane](#) to open an existing variable.
2. On the **Variable properties** pane, 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`.

Variable properties

Name *

Variable name

Global. UserEmail

Type

string

Reference

Question node
What's your email address?

Type (string) derived from here
Variable gets value from this node if empty

[View all references](#)

Usage

Topic (limited scope)

Global (any topic can access)

Allow to carry between sessions

External sources can set values

3. Save the topic.

A global variable's name must be unique across all topics. If there's a conflict, you'll need to rename the variable before saving your topic.

Use global variables

When you're composing an agent 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.

Question

Text

+ Add {x}

Is this the best email address to reach you?

Identify

Multiple choice options

Select a variable

Custom System

Search variables

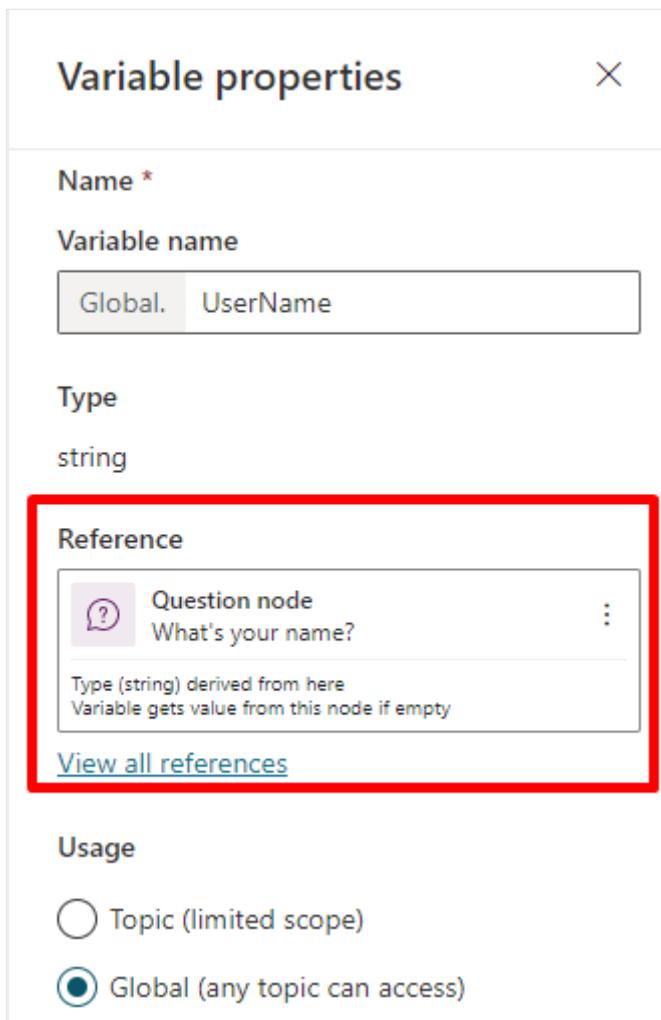
All

UserEmail
(Global userEmail)
string

Find all topics using a global variable

You can find where a global variable is defined and what other topics are using it. This can be useful if you're working on a new agent, or if you have multiple variables and [complex topic branching](#).

1. Select a global variable in the authoring canvas, or open the [Variables](#) pane and select a global variable.
2. On the **Variable properties** pane, in the **Reference** section, select any of the topics where the variable is used to go directly to that topic and node.



Lifecycle of global variables

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. That topic can be triggered either by redirection or when the user types a trigger phrase such as "Start over." In that case, all global variables are reset.

Set a global variable's value from external sources

If you want to make sure the agent starts a conversation with some context, you can initialize a global variable with an external source. Let's say that your site requires users to sign in. Since your agent already knows a user's name, it can greet customers by name before they start typing their first question.

1. Select a global variable.
2. On the **Variable properties** pane, select **External sources can set values**.

Set global variables in an embedded agent

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 provide a simple declaration for the variables. In a production scenario, you might pass in as the query parameter or variable definition another variable that has already stored 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=Renata>.

The parameter name is case-insensitive. `username=Renata` will also work in this example.

Add global variables to a custom canvas

You can also add the variable to a [custom canvas](#).

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 above where `styleOptions` is called (you'll need to 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));
```

Related content

- [Work with variables](#)
- [Customize the look and feel of the agent](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create expressions using Power Fx

Article • 11/19/2024

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 [Power Fx parameters must be separated by commas \(,\)](#).

Prerequisites

- [Create and edit topics](#)
- [Work with variables](#)

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 'Enter formula' interface in Power Automate. A 'Set Variable Value' step is selected. In the formula bar, 'System.Conversation.Id' is typed. The variable 'Var1' is defined as a string type, and its output is also set to 'System.Conversation.Id'. The 'Formula' tab is active.

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 corresponding Power Fx formulas to use with that 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 Trun 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**.

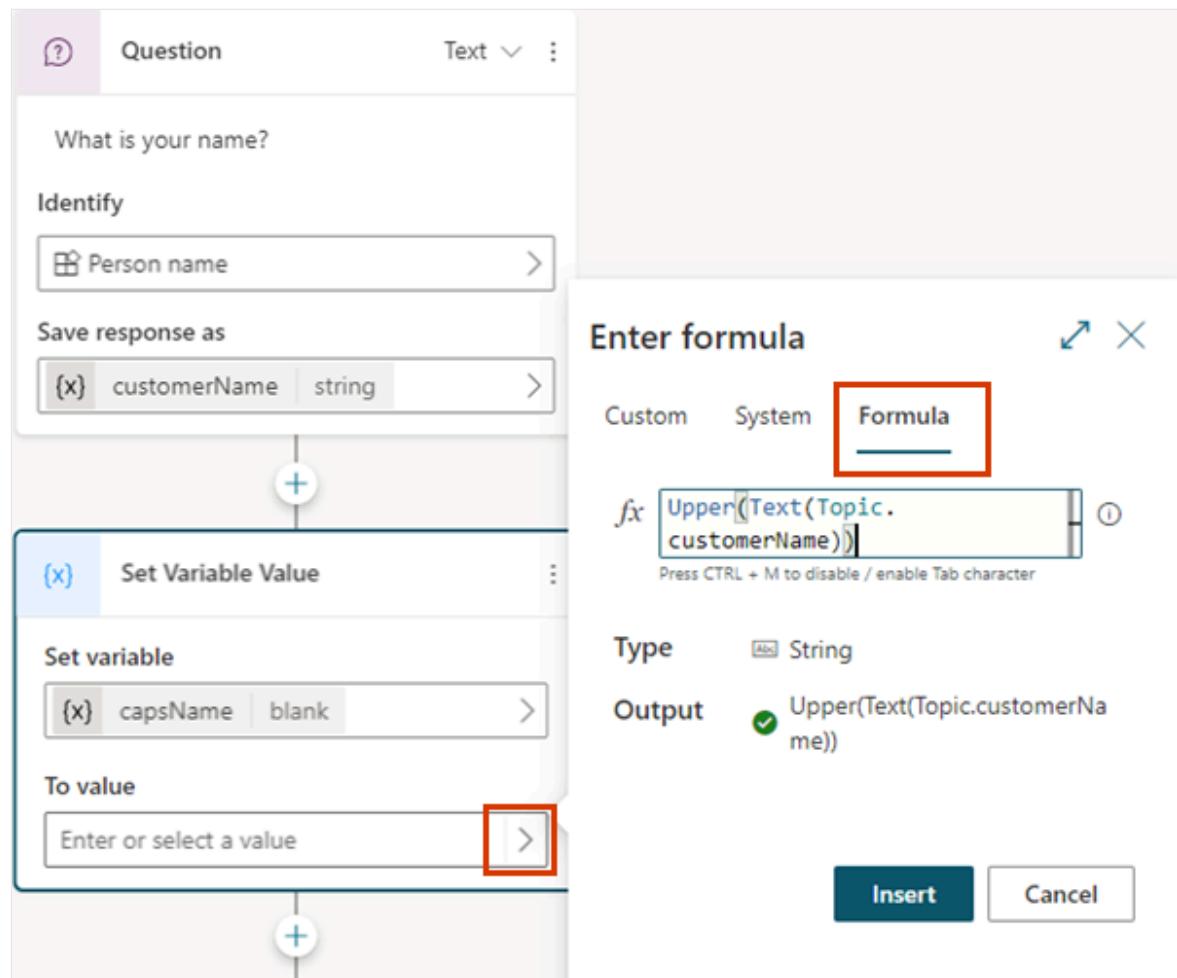
4. Select the box under **Save response as**, and then select the variable `var1` and name it `customerName`.

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`.

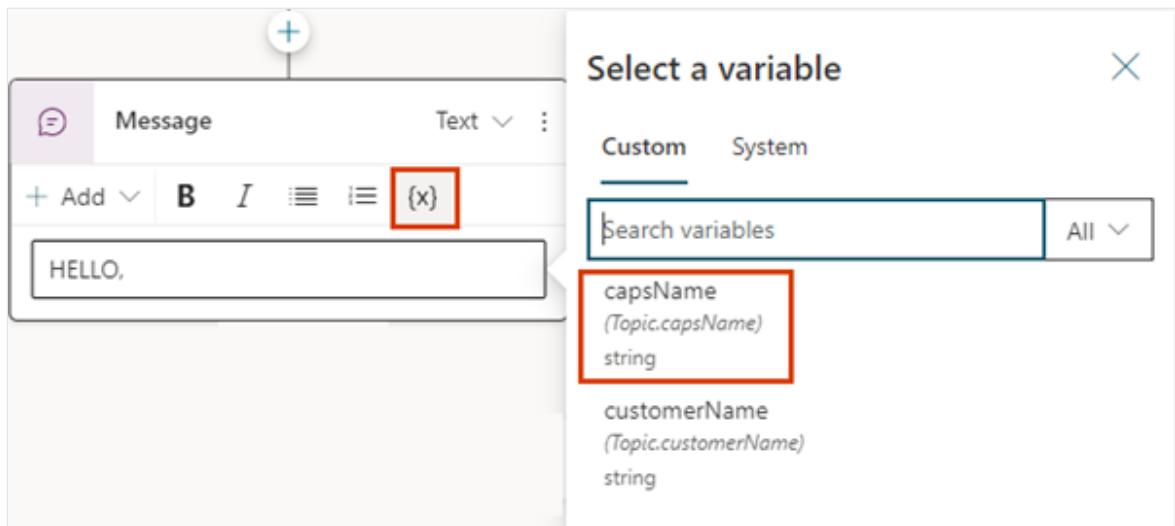
7. In the **To value** box, select the **>** arrow, and then select the **Formula** tab.

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`.

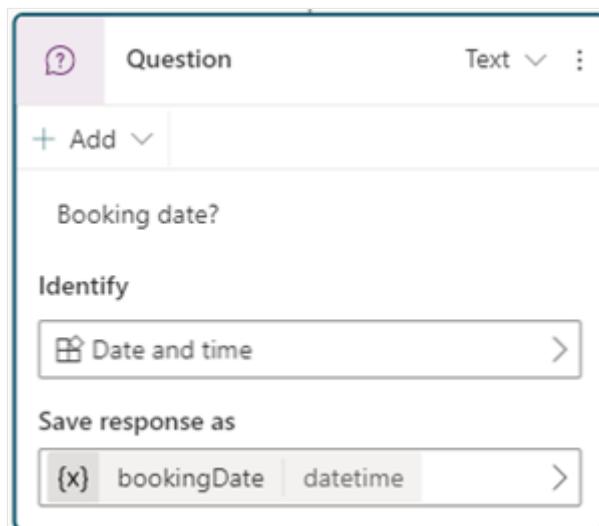


Use a Power Fx formula as a condition

To evaluate more complex expressions, set up **Condition** nodes to use Power Fx formulas.

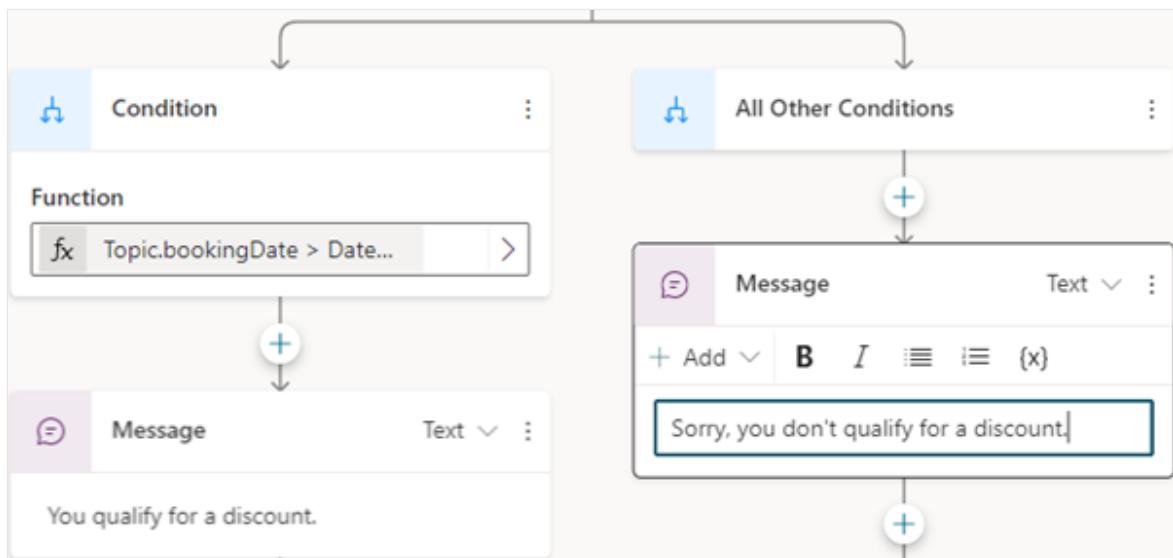
In this example, the agent determines if a booking date qualifies for a discount. To do that, it checks whether the booking date provided by the customer is 14 days or more from the current date.

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 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."



Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

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'. Under 'Create the message to display', there is a rich text editor with a toolbar (B, I, etc.) and a message box containing 'Which location do you want to get the weather for?'. This message box is 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 • 11/19/2024

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.

Web app

Topic errors

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.

View topic errors

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.

Topic status

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**.

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.

Turn a topic on/off

To change the status of a topic, select the **More options** icon (...) next to your topic name and use the **Status** toggle.

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 **More options** icon (...) 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

Article • 11/19/2024

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. [Knowledge sources](#) defined in the generative answers node override 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](#)

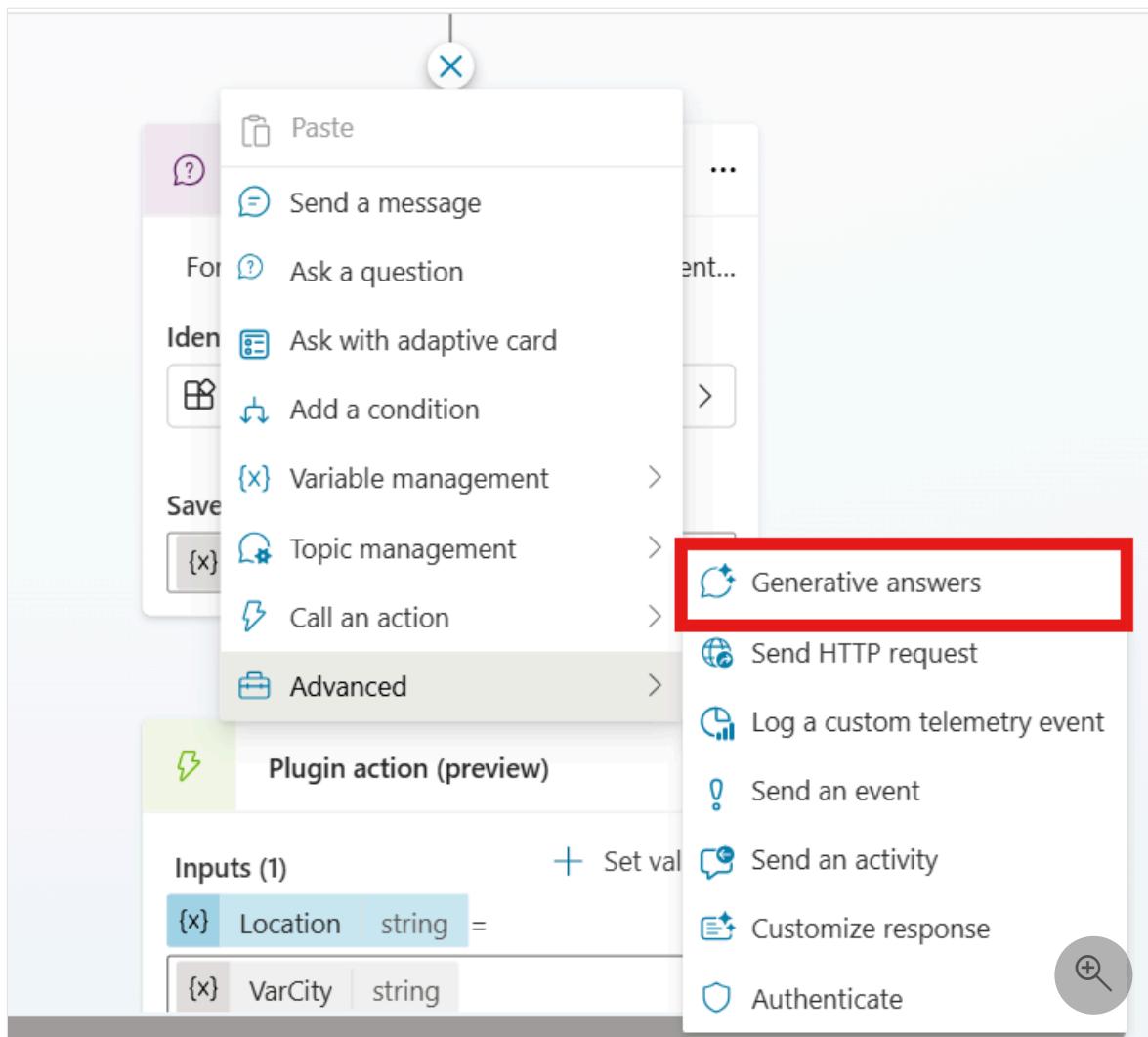
Name	Source	Description	Number of inputs supported in generative answers	Authentication
Public website	External	Searches the query input on Bing, only returns results from provided websites	Generative mode: Unlimited Classic mode: Four public URLs (for example, <i>microsoft.com</i>)	None
Documents	Internal	Searches documents uploaded to Dataverse, returns results from the document contents	Generative mode: Unlimited Classic mode: Limited by the Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	Generative mode: Unlimited 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 Copilot connectors	Internal	Connects to Copilot 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 Open AI, or Custom Data. Instead, from the generative answers node properties, use the **Classic data** option for [Bing Custom Search](#), [Azure Open AI](#), or [Custom Data](#) sources.
- For websites, you need to confirm which website(s) your organization owns that Bing will search through Copilot Studio.

Add a generative answers node

1. Select the **Topics** page and open a topic. Find where you want to use generative answers.
2. Select the plus icon to open the new node menu.
3. Under **Advanced**, select **Generative answers**.



4. A new node called **Create generative answers** is added to your topic. To access the **Properties** pane, select the node's **...** then **Properties**.

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

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 *powerva.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**

Only [modern pages](#) are supported. Content from classic ASPX pages on SharePoint aren't used to generate answers.

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 relevancy of the given answers.

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 be inaccurate. The highest level generates fewer answers, but they're likely to be more relevant. 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](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Search public data or use a Bing Custom Search for generative answers nodes

Article • 11/19/2024

At the topic level, you can search publicly available data by using the Bing search engine API 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.

You can use the default Bing Search configuration or build a tailored search for specific content using [Bing Custom Search](#). Bing Search helps you boost your agent by including many sources without configuring each source individually. To use Bing Custom Search as a source, you need a custom search instance connected to the [Create generative answers node](#).

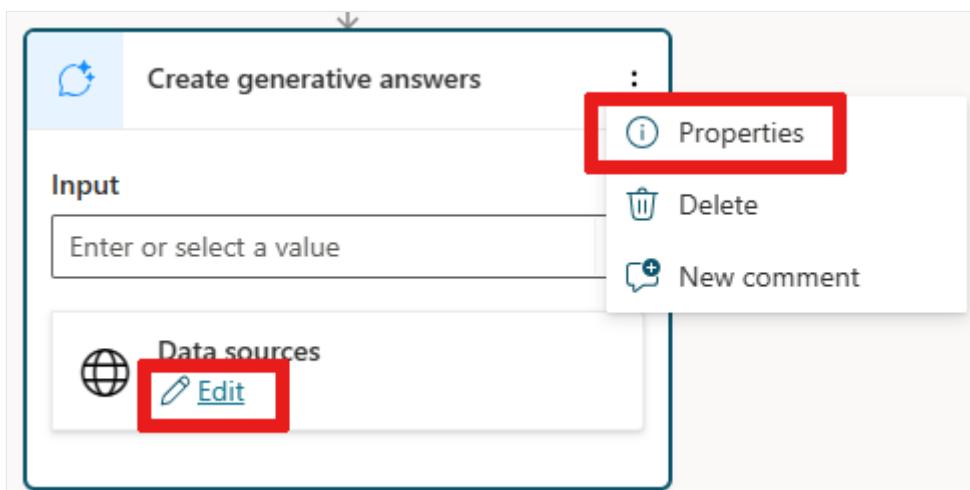
ⓘ Note

To create a Bing Custom Search, see [Quickstart: Create your first Bing Custom Search instance](#).

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.

The screenshot shows the 'Classic data' configuration section. It includes:

- Data sources**: A section for choosing up to 4 public websites and 4 Microsoft internal sites for dynamic answers. It features a 'Search public data' button and two input methods: 'Public websites' and 'Manual input'. A text input field 'Enter text' with a '+' button is provided for manual entry.
- SharePoint**: Another section for adding items in bulk, with a 'Manual input' dropdown and a 'SharePoint' button. It has a similar 'Enter text' input field and '+ button'.
- Azure OpenAI Services on your data**: A section for adding a connection from Azure OpenAI as a data source, featuring a 'Add connection' button and a 'Connection properties' input field.
- Custom data**: A section for entering or selecting a value, with an 'Enter or select a value' input field and a right-pointing arrow button.

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:

- a. From the [Bing Custom Search web portal](#), select the search instance you want to use.
- b. Select **Production** from the top menu, and under the **Endpoints** section, select the **Web API** tab.
- c. Copy the **Custom Configuration ID**.

4. Save changes to your topic.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#)

Connect your data to Azure OpenAI for generative answers (preview)

Article • 11/19/2024

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

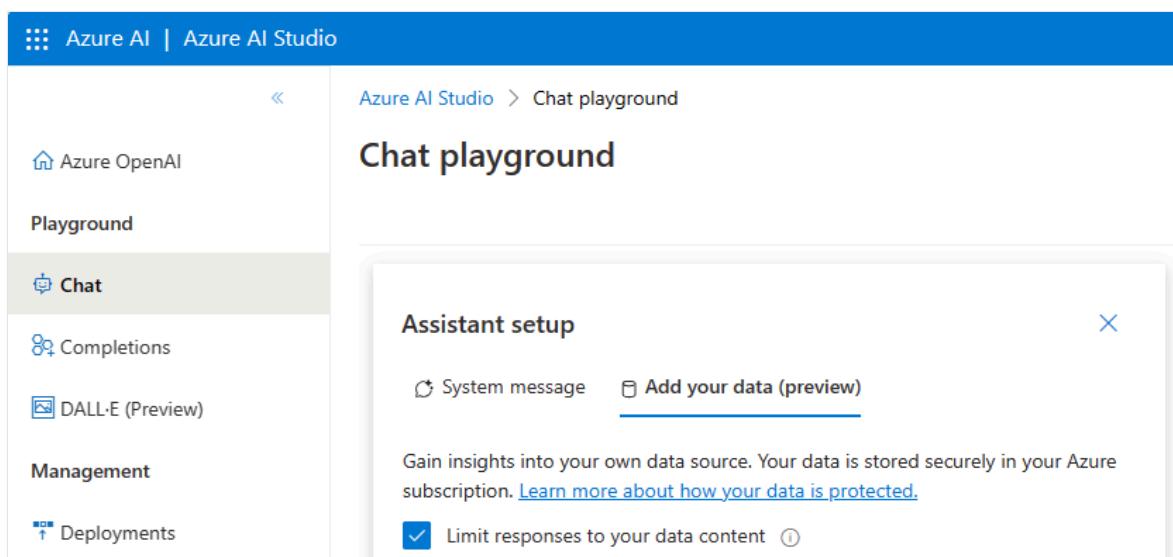
This article contains Microsoft Copilot Studio preview documentation and is subject to change.

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](#).

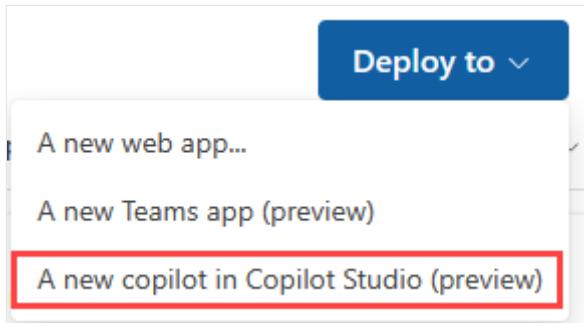
If you're building a production-ready agent, see [Microsoft Copilot Studio Overview](#).

Prerequisites

- Connect your data source using [Azure OpenAI Studio](#).



- Connect a Copilot Studio agent to your data by selecting **Deploy to** and then **A new Microsoft Copilot Studio bot**. The agent is created and automatically connects to your Azure resource in your default environment.



! Note

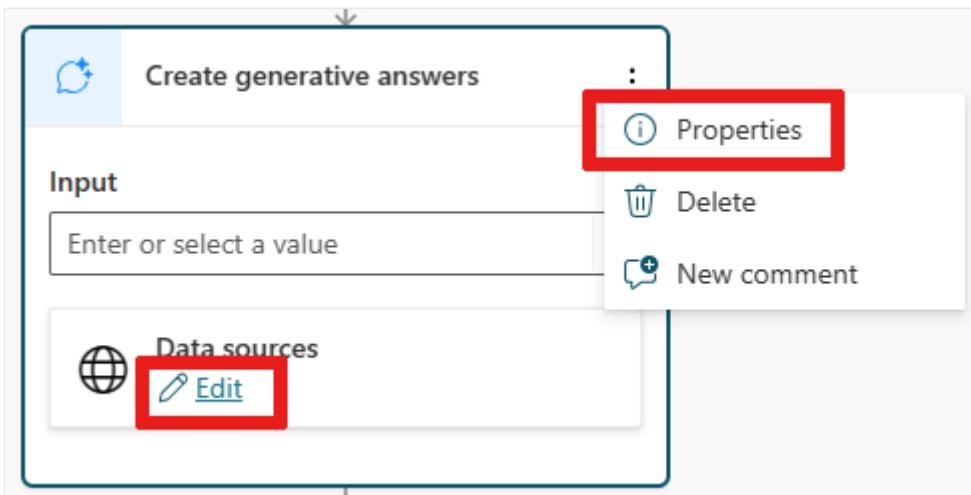
The Conversation Boosting system topic is automatically generated when the agent is created from Azure OpenAI Studio.

Type	Name ↓	Trigger ↑
Conversation Start		On Conversation Start
Conversational boosting		On Unknown Intent
End of Conversation		On Redirect

boosted topic

Use Azure OpenAI in Copilot Studio

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, select the **...** in the **Create generative answers** node, then select **Properties** to open a pane, and finally select **Data source**.

2. Open 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 >

This screenshot shows the 'Classic data' configuration page. It includes sections for 'Data sources' (allowing selection of up to 4 public websites and 4 Microsoft internal sites), 'Search public data' (with a dropdown for 'Search public websites'), and 'Azure OpenAI Services on your data' (which allows adding a connection from Azure OpenAI as a data source). There are also sections for 'Custom data' and 'Connection properties'.

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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use SharePoint content for generative answers

Article • 12/16/2024

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](#).

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](#).

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](#).

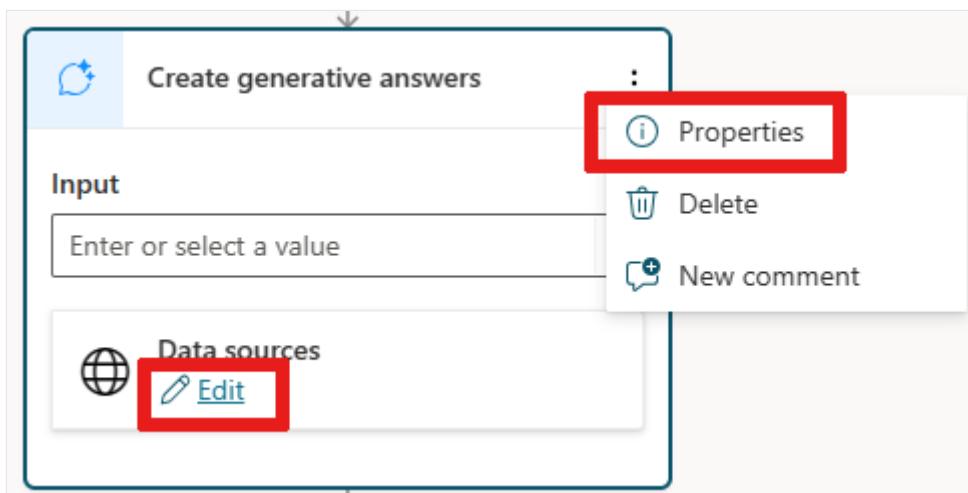
Use SharePoint in a generative answers node

ⓘ 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 ⓘ

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 >

The screenshot shows the 'Data sources' configuration pane. It includes sections for 'Public websites' and 'SharePoint', each with a 'Manual input' dropdown and a text input field with a '+' button. Below each section is a note about adding items in bulk using Shift+Enter. There is also a section for 'Azure OpenAI Services on your data' with a 'Connection properties' button and a 'Custom data' section with a text input field and a 'greater than' button.

3. 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**).

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 [+](#)

Shift+Enter to create line separation

! Note

There are limitations when adding SharePoint as a knowledge source, for more information, see [Known issues](#).

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 *powerva.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.

i 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`, `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`, `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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use a custom data source for generative answers nodes

Article • 11/19/2024

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.

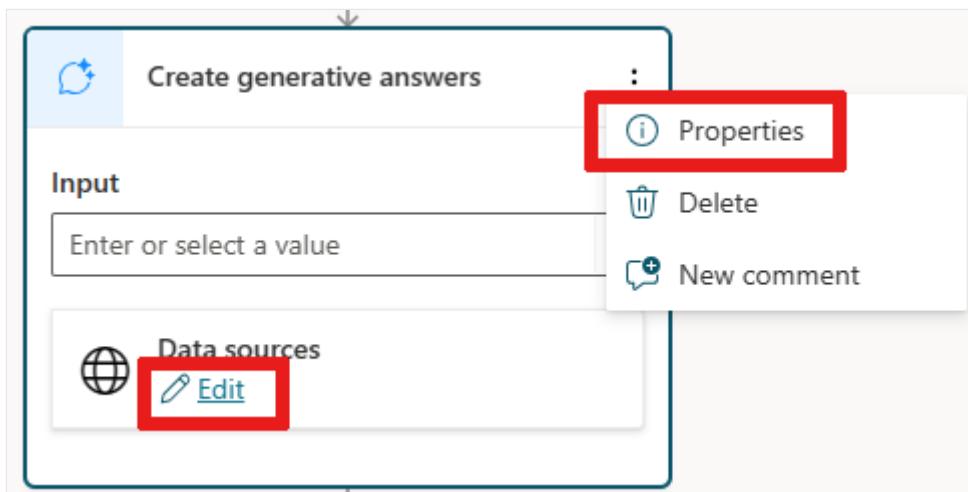
ⓘ Important

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 documents with generative answers nodes

Article • 11/19/2024

You can upload your own documents at the topic level to use with [generative answers nodes](#). The documents are used across your agent, but you can [specify individual nodes that shouldn't use uploaded documents](#).

ⓘ Important

Knowledge sources defined in generative answers nodes take priority over [knowledge sources](#) at the agent level. Agent level sources function as a fallback.

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.

ⓘ Note

Generative answers can process uploaded documents 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 document

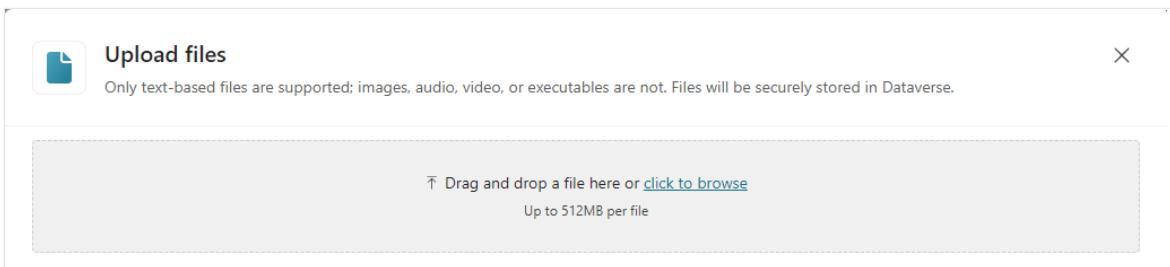
⚠ Warning

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 document:

1. With an agent open, expand **Settings** on the side navigation pane, and select **Generative AI**.
2. Under **Upload a document (preview)**, upload your documents in one of two ways:
 - Drag and drop files and folders onto the field that says **Drag and drop files here or click to browse**.
 - Select **click to browse** to open a file window where you can select the files you want.



ⓘ Important

If you [export agents using solutions](#), ensure that you don't exceed the maximum [solution file size limitations](#) when uploading files.

If you have a single file or combination of files that exceed the solution file size limit during export, you must upload the files directly in the environment where you want to import the solution.

3. Select **Save** at the top of the **Generative AI** page.

After uploading your documents

- Once uploaded, your content in the document might take a few minutes to be ready for use by the agent. When the document is ready, the status of the document changes from **Indexing in progress** to **Ready**.
- Uploaded documents create new copies and don't overwrite existing documents, even if they have the same name.

- The uploaded document becomes part of the agent solution, therefore exporting and importing an agent solution includes the documents.

Downloading or deleting a document

To download or delete an uploaded document, select **More options (...)** and select the download or delete icon.

Disable documents as a data source for a generative answers node

Documents 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 documents 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:

```
yml  
  
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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Make HTTP requests

Article • 11/19/2024

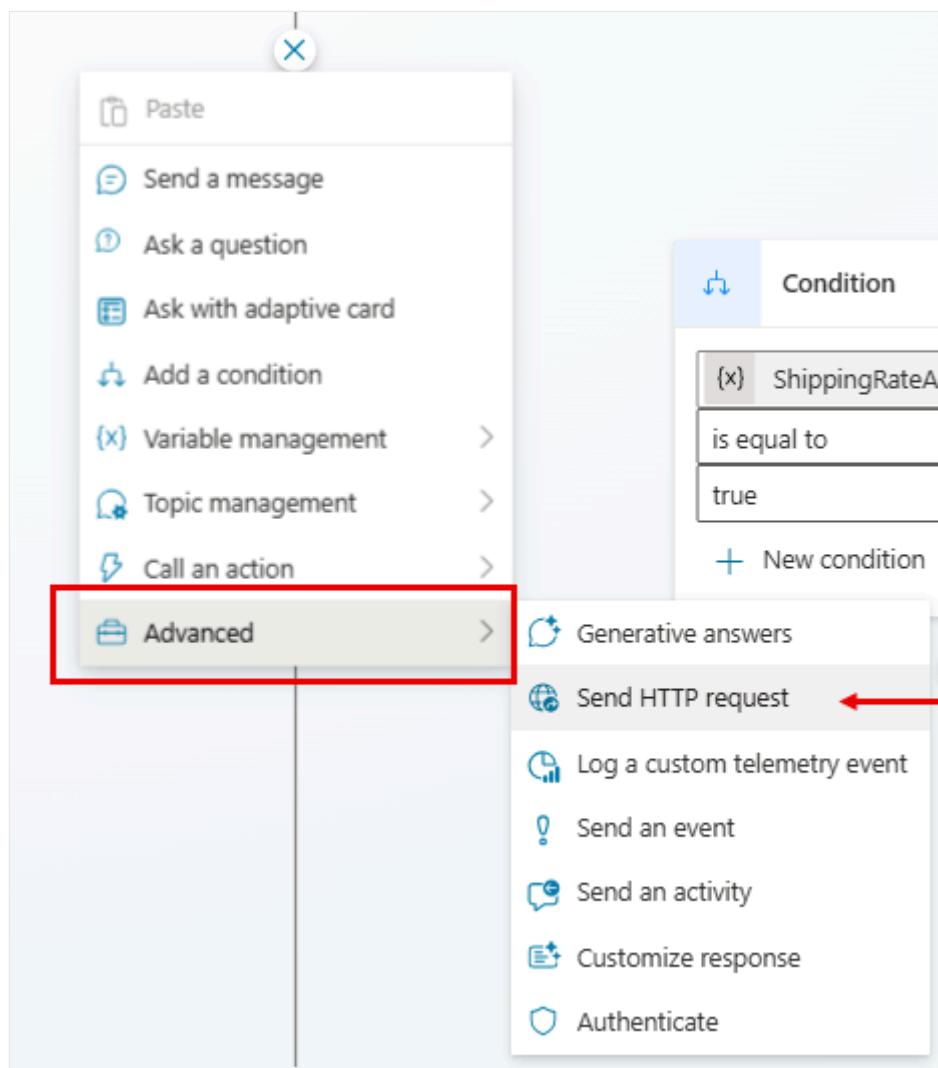
In Copilot Studio, you can extend an agent's capabilities by calling external REST APIs using the **HTTP request** node. This call can be helpful when you need to retrieve data from an external system or manipulate data on that system.

Prerequisites

- [Create and edit topics](#)

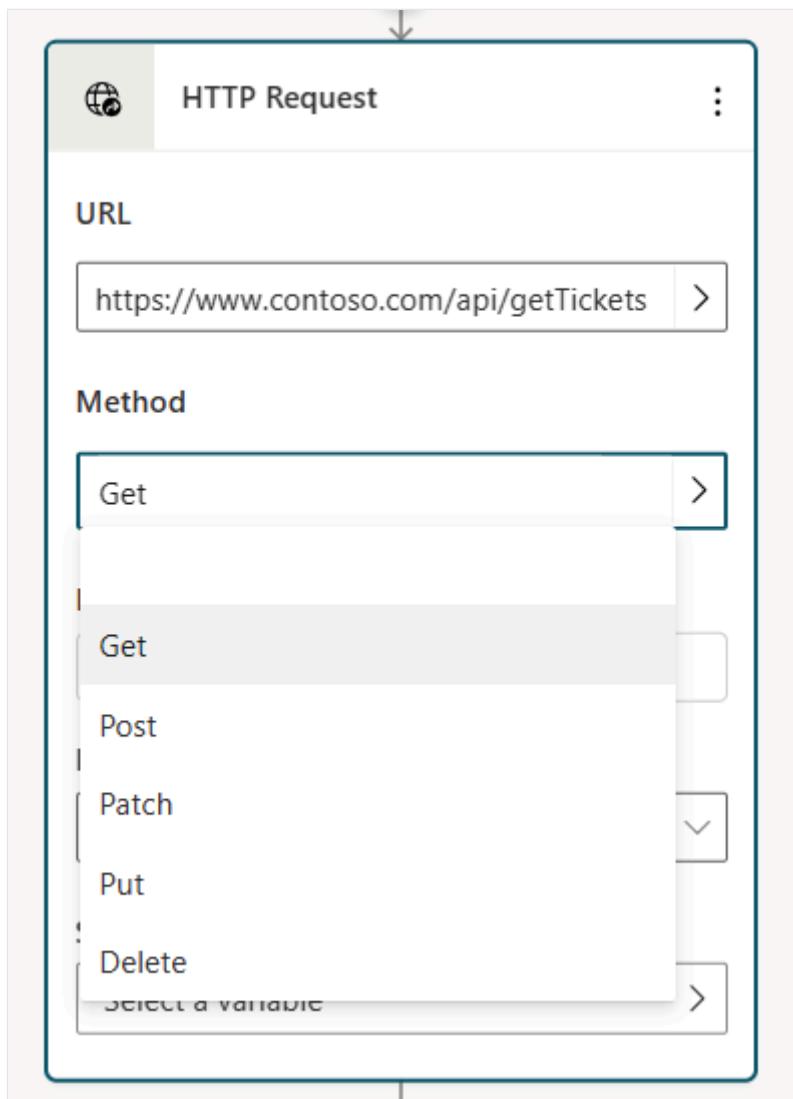
Use the HTTP request node

1. Select **Add node (+)**, open the **Advanced** sub menu, and then select **Send HTTP request**.

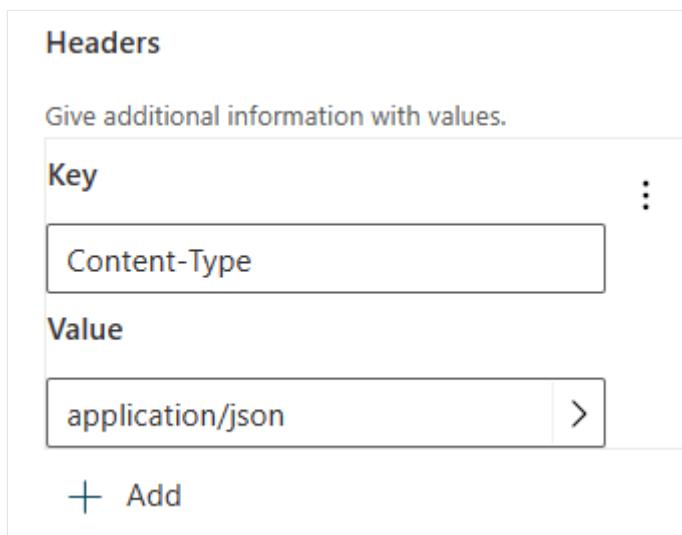


2. In the **URL** box, enter the URL for the API endpoint you want to call.

3. Select the appropriate request **Method** from the dropdown. GET, POST, PATCH, PUT, and DELETE methods are supported.



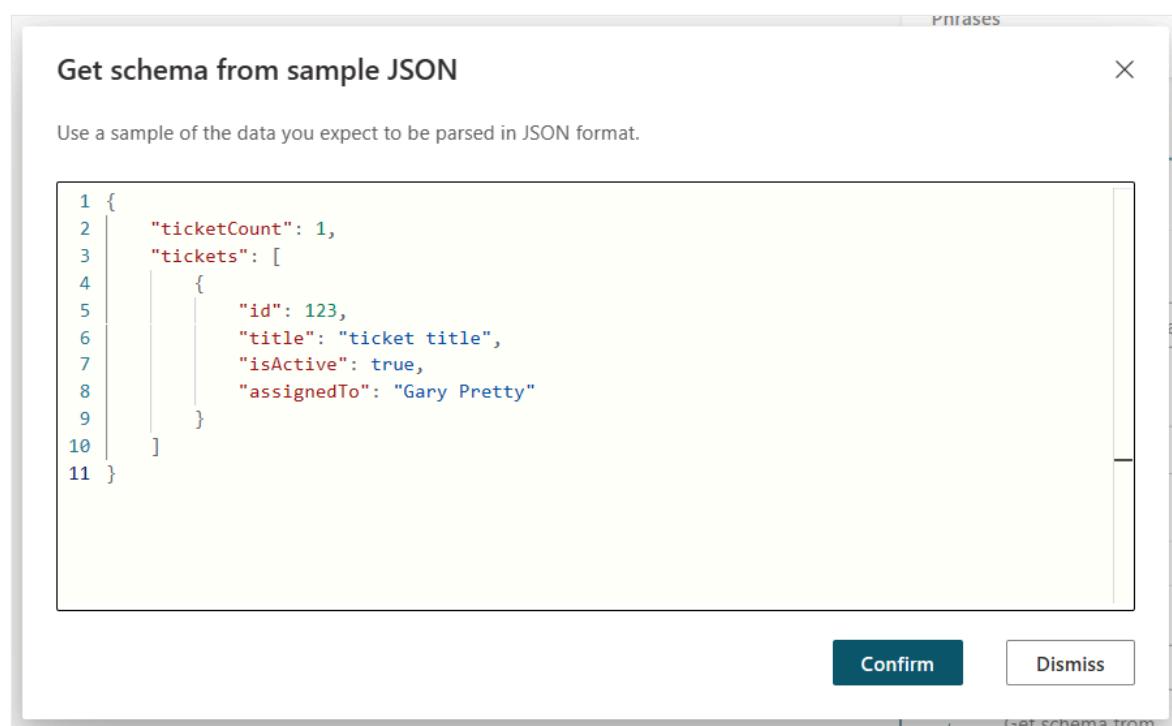
4. Under **Headers and body** on the node, select the **Edit** button, which opens the **HTTP request properties** pane.
5. You can optionally add one or more headers to the HTTP request. Select the **Add** button to add each header's key and value.



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 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** pane.
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.

Body

Decide what to send or send nothing

No content

No content

JSON content

Raw content

- **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.

Body

Decide what to send or send nothing

JSON content

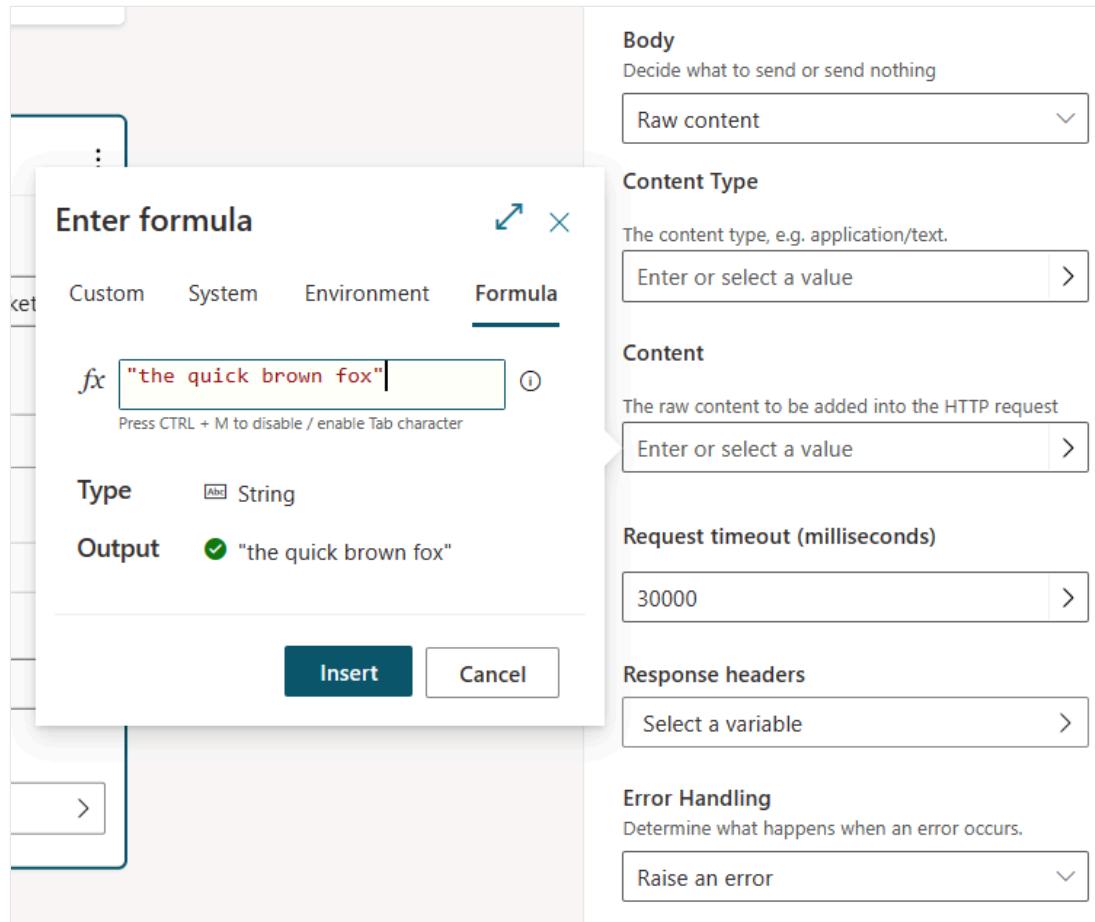
</> Edit JSON ▾

✓ JSON

Formula



- **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 timeout

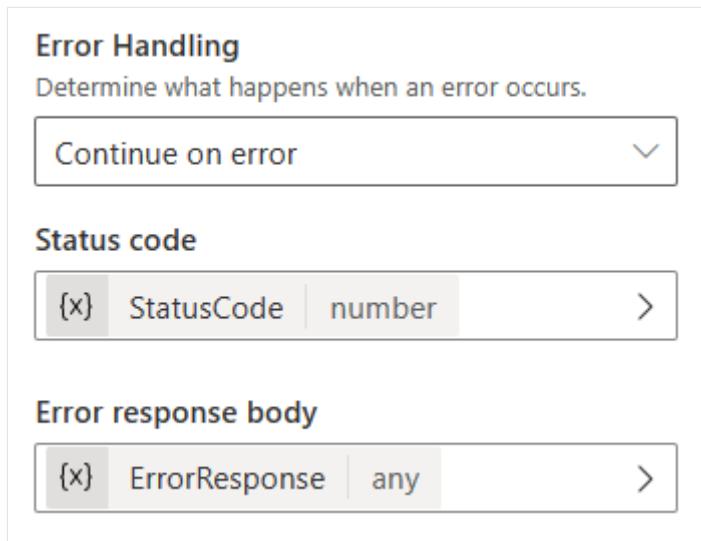
The **HTTP request** node has multiple options for handling situations where the **HTTP** request might fail or return an error.

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 **Error** system topic, displaying an error message.

However, you can configure the error handling to not trigger the **Error** system topic. Instead, it populates the **HTTP** status code and any error response body into variables that you specify, allowing you to interrogate these variables as needed, and for the topic to 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.

1. Under Headers and body on the node, select the **Edit** button, which opens the **HTTP request properties** pane.
2. Choose your desired behavior from the **Error handling** dropdown.
3. If you choose **Continue on error**, configure the **Status code** and **Error response body** variables, either creating new variables or selecting existing variables from the picker.



In this code sample, 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 the [Parse Value node](#) to convert this variable into a Power Fx record.

The property pane also contains a **Request timeout** property, which accepts a value in milliseconds. The default value is 30 seconds, but you can change this value if needed.

Feedback

Was this page helpful?

Yes

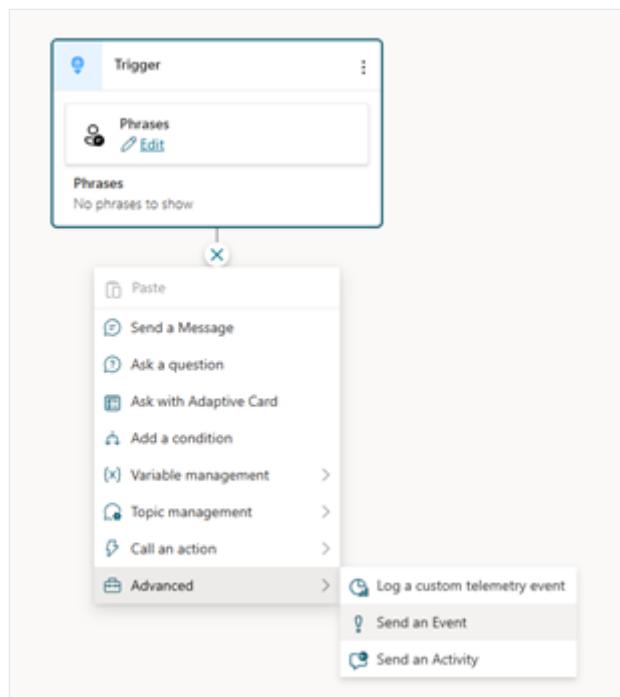
No

[Provide product feedback ↗](#)

Send an event or activity

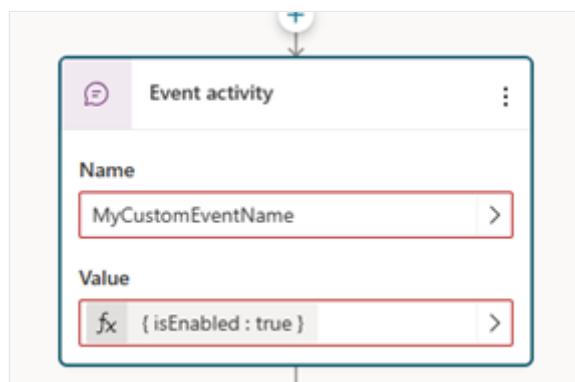
Article • 11/19/2024

Microsoft Copilot Studio provides two nodes that allow you to send non-message activities: Send an Event, and Send an Activity. These node types are discussed in the following sections.



Sending events

The Send an Event 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 on the event. This value can be in any format: a primitive literal value, a variable reference, or a Power Fx formula. The value gets serialized as JSON and added to the outgoing activity.



Uses of this node include:

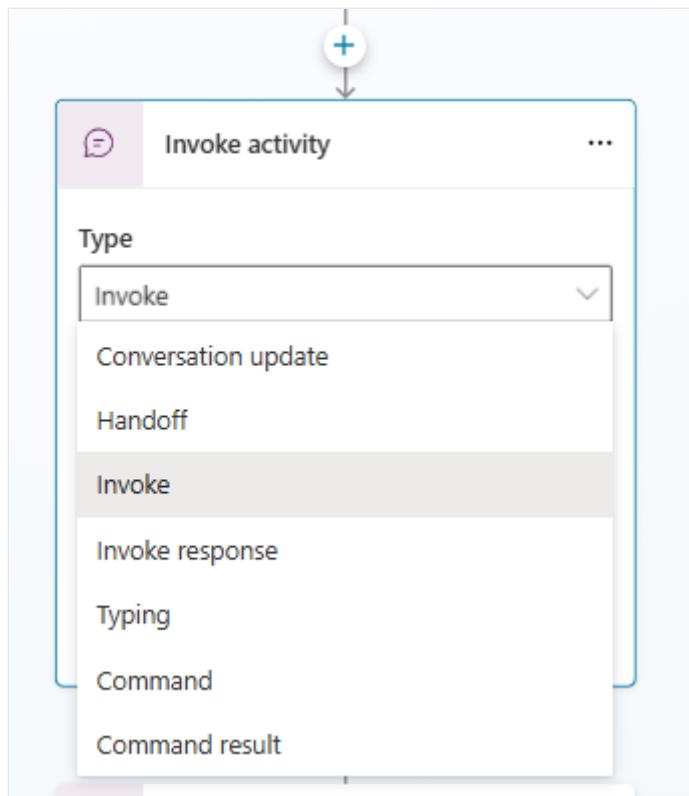
- You can configure a custom web chat control to handle events sent from the agent. For example, you could look for an event coming back from the agent and take an action on the page. The `04.api/c.incoming-activity-event` sample in the [Microsoft Bot Framework Web Chat samples repository on GitHub](#) demonstrates how event handling could work.
- AudioCodes use event activities to let you control the service. For example, starting or stopping call recording. For additional information, see [Recording calls](#).

Sending other activity types

In addition to event activities, you can send activities of other types using the send activity node. Generally, the usage of this node is advanced – hence it being in the advanced subcategory. 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.
- **Delay** – Can be used to provide a delay between messages. For example: sending a message, then sending a delay activity, followed by another message. For the customer, they see the first message, followed by a pause, then the second message. In this case, "Value" is set to the number of milliseconds delay. For details, see [ActivityTypes.Delay Field](#).
- **Invoke/Invoke Response** – Generally used for Microsoft Teams. A user creates a topic with an Invoke trigger to intercept an incoming Invoke from Teams, and use an Invoke Response activity to send an appropriate response back to Teams.
- **Handoff** – Allows a user to send a Handoff activity with explicit control over the value. A handoff is used for some external channels, such as AudioCodes.



Related content

- Use variables to carry content across topics
- Customize the look and feel of an agent

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Event trigger overview (preview)

Article • 11/28/2024

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

Using event triggers is a public preview feature. Using event triggers is free for preview participants, but does require messages that count towards your [usage quotas](#). Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that customers can get early access and provide feedback.

ⓘ 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 loss prevention policies](#), configured in Power Automate by an administrator.

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. The payload 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, without you needing to write 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

 Name
Dataverse agent

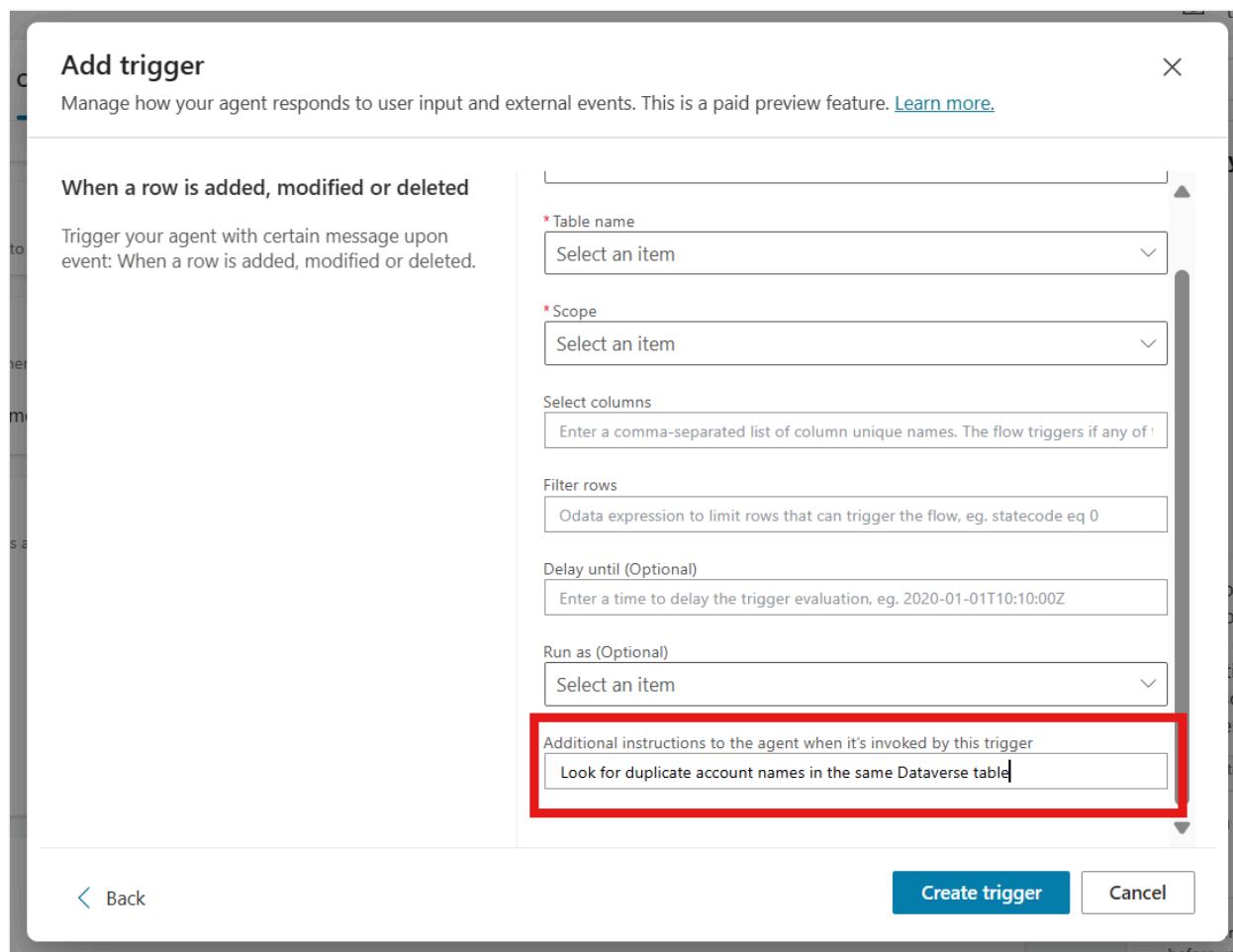
Description
Testing the Dataverse event trigger

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.*



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 have given it determine the action or topic it calls in response. Your instructions allow your agent to select an action or topic 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 mode 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)	Outputs (13)
{x} Title String = Draft proposal for paperclip	{x} Content String =
{x} To-do List String =	{x} Content Type String =
⚠ Missing required information	{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 manage running them reliably.

If your agent still struggles to call the expected action, consider adding a [Power Automate flow](#) that fulfils 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 author's credentials for authentication (that is, the credentials you used to [authorize the connections](#)) for your trigger. This can 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 authentication that doesn't require user input. You can also instruct your agent to not request credentials from users.

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.

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.

Triggers can use maker credentials only

Currently, event triggers can use only the agent author's credentials for authentication. If you publish or share 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.

In order to prevent users from accessing or modifying protected data or systems, carefully consider whether and how data and systems requiring authentication are used by agents with event triggers. Authors should be aware when sharing or publishing agents containing event triggers.

Administrators can also block Copilot Studio users from using event triggers with their agents. For more information, go to [Block event triggers](#).

Limitations using knowledge sources with event triggers

Agents can't reference some knowledge sources in response to an event trigger. Some knowledge sources require an agent to provide authentication to access, but agents can't provide that authentication autonomously.

Avoid these knowledge sources, when referenced in response to an event trigger:

- SharePoint
- Dataverse
- Graph Connectors
- AI Builder Prompts

Reference these knowledge sources in response to an event trigger instead:

- Public websites
- Uploaded files
- Enterprise data, using connectors

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Add an event trigger (preview)

Article • 11/19/2024

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

Using event triggers is a public preview feature. Using event triggers is free for preview participants, but does require messages that count towards your [usage quotas](#). Preview features aren't meant for production use and may have restricted functionality. These features are available before an official release so that customers can get early access and provide feedback.

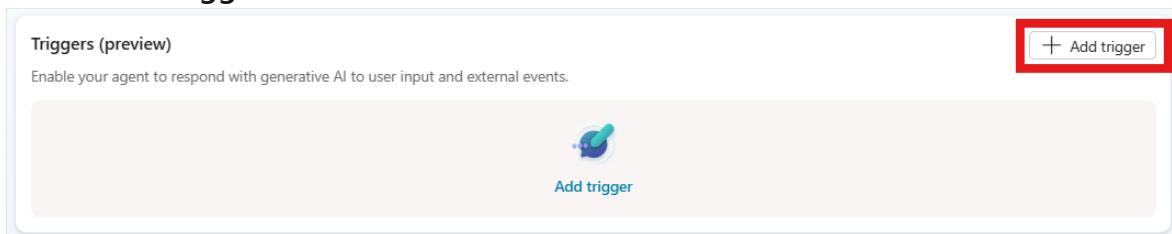
For in depth guidance on how event triggers work and relate to other components of an agent, go to [Event trigger overview](#).

ⓘ 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 use authenticate trigger connectors using the agent maker's account (agent author authentication). You 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 won'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**.

Test your trigger X

 **Recurring Copilot Trigger** Last refreshed now ...

To test your trigger, we'll use an instance from the past when the trigger worked successfully. Which instance should we use?

 Nov 27, 02:59 p.m. ...

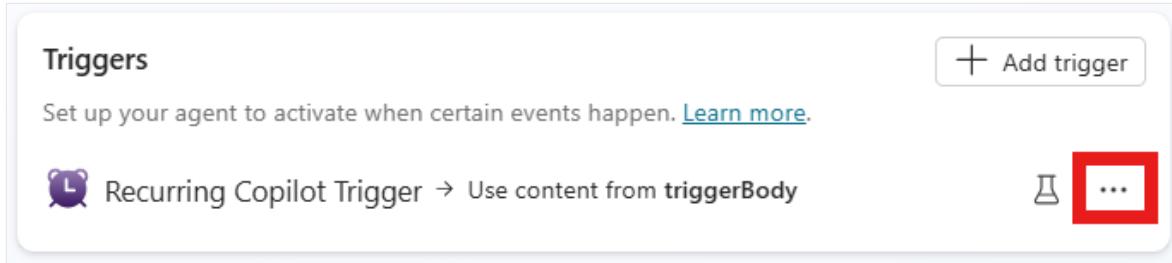
Start testing Cancel

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](#).



The screenshot shows the 'Triggers' section of the Copilot Studio interface. At the top right is a button '+ Add trigger'. Below it is a sub-section titled 'Recurring Copilot Trigger → Use content from triggerBody'. This section includes a purple alarm clock icon, the trigger name, and a red square icon with three dots (...). A small flask icon is also present.

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

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.

Event triggers can only use the agent author's credentials for authentication (that is, the credentials you used to [authorize the connections](#)). This can allow users of an agent without their own authorization the ability to access data and systems they shouldn't. For more information, go to [Troubleshooting and limitations](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use actions with custom agents (preview)

Article • 11/25/2024

You can extend the capabilities of your agent by adding one or more actions. Actions are used by your agent to respond to users automatically, using [generative orchestration](#), or you can call them explicitly from within a topic.

Core action types

If you turn on [generative orchestration](#), your agent can automatically select the most appropriate action or topic, or search across knowledge, to respond to a user.

In classic mode (generative orchestration disabled), an agent can only use topics to respond to the user. However, you can still design your agent to call actions explicitly from within topics.

Actions are based on one of the following core action types:

- Prebuilt connector action
- Custom connector action
- Power Automate cloud flow
- AI Builder prompts (within a topic)
- Bot Framework skill
- REST API connection

Each core action has additional information that describes its purpose, allowing the agent to use generative AI to generate questions. These questions are required to fill the inputs needed to perform the action. Therefore, you don't need to manually author question nodes to gather all inputs needed, such as inputs in a flow. Inputs are handled for you at runtime.

By default, when using generative mode, actions 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 action to always respond immediately, either generating a message or authoring an explicit message.



Tip

When using generated questions from a action, 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.

Add an action

1. Open your agent by choosing **Agents** in the left hand navigation pane and selecting your agent from the list.
2. Select **Actions** to view actions for the agent.
3. Select **Add an action**.
4. In the **Choose an action** wizard, search for an action you want to use and select it.

The default list shown has Power Automate Flows and custom connectors available within your environment. This list includes commonly used prebuilt connectors and Bot Framework Skills that are registered with your agent.

Step 1 of 3: Choose an action
Create an action or browse through our list of actions you want to use to get information from external sources. [Learn more](#) X

Discover an action
Search for flows, skill actions, and commonly used connector actions

Search

11 actions found

[Connectors](#) [Custom Connectors](#) [Flows](#) [Skills](#) [Dataverse](#)

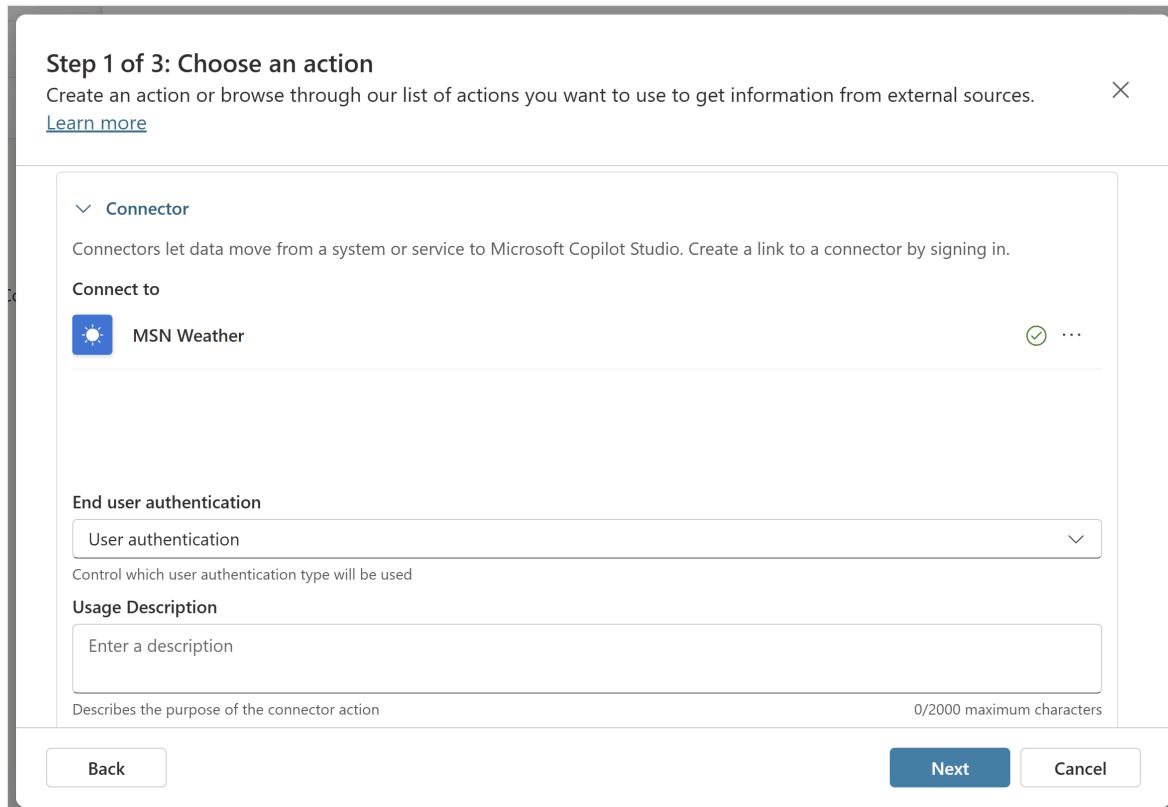
 Untitled Get information about industries, solutions, services and cont...	 Run a flow built with Power Automate for desktop Desktop flows
 Delete a row Excel Online (Business)	 Run script Excel Online (Business)
 Get a row Excel Online (Business)	 Run script from SharePoint library Excel Online (Business)

[Cancel](#)

Important

Some actions created from AI Plugins, such as Dynamic Prompts and those based on Dataverse APIs, require [authentication](#).

Depending on the action, your connection is automatically configured, as shown in the example. Otherwise, follow the wizard's instructions to configure the connection. Once your connection is successfully configured, select **Next**.



5. Review your plugin action configuration, and [make changes](#) if needed.

Select **Next**.

6. Select **Finish** to add the action to your agent.

Test your action

If you enabled [generative orchestration](#), your action is called automatically in response to a relevant user query.

Alternatively, you can [call an action explicitly from within a topic](#).

Your agent selects and runs your action based on the name and description. You can author high quality descriptions, using the activity map to see which actions your agent selected from the [actions](#) page.

You can see a weather action being tested in this screenshot:

Chat



Hello, I'm Virtual Agent. How can I help?

Just now

today's forecast



Just now



What location would you like to get the forecast for? Please provide a valid search query such as City, Region, State, Country, Landmark, Postal Code, latitude and longitude.

Just now

check for Orlando, FL



Just now



Are you sure you want to check the weather for Orlando?

Just now

actually check for Seattle



Just now



Are you sure you want to check the weather for Seattle?

Just now

yes

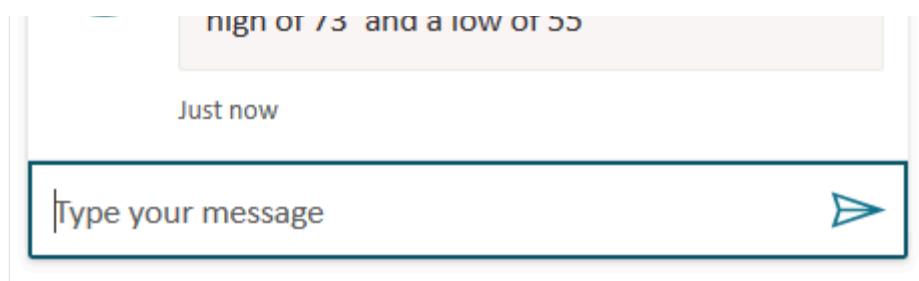


Just now



The weather in Seattle is forecast to have a

High of 63° and a Low of 55°



Make changes to your action configuration

There are many ways to configure your action. On the final step of the action wizard, called **Review and Finish**, you can make changes to your action's configuration by clicking **Edit** next to the appropriate section. From either the **Review inputs and outputs** screen in the wizard, or from the details page for an action that is already part of your agent, you can manage the input and output parameters for the action.

Action details

Here you can configure details about your action, including the **Model display name** and **Model description** for your action. Usually, the name and description are prepopulated for you based on the action you selected.

The **Display name** and **Description** properties for each input help the agent understand the purpose of the input, to generate an appropriate question for the user, and to recognize the correct value in a user's response. For example, the following are the default name and description of the location input parameter for the MSN Weather connector.

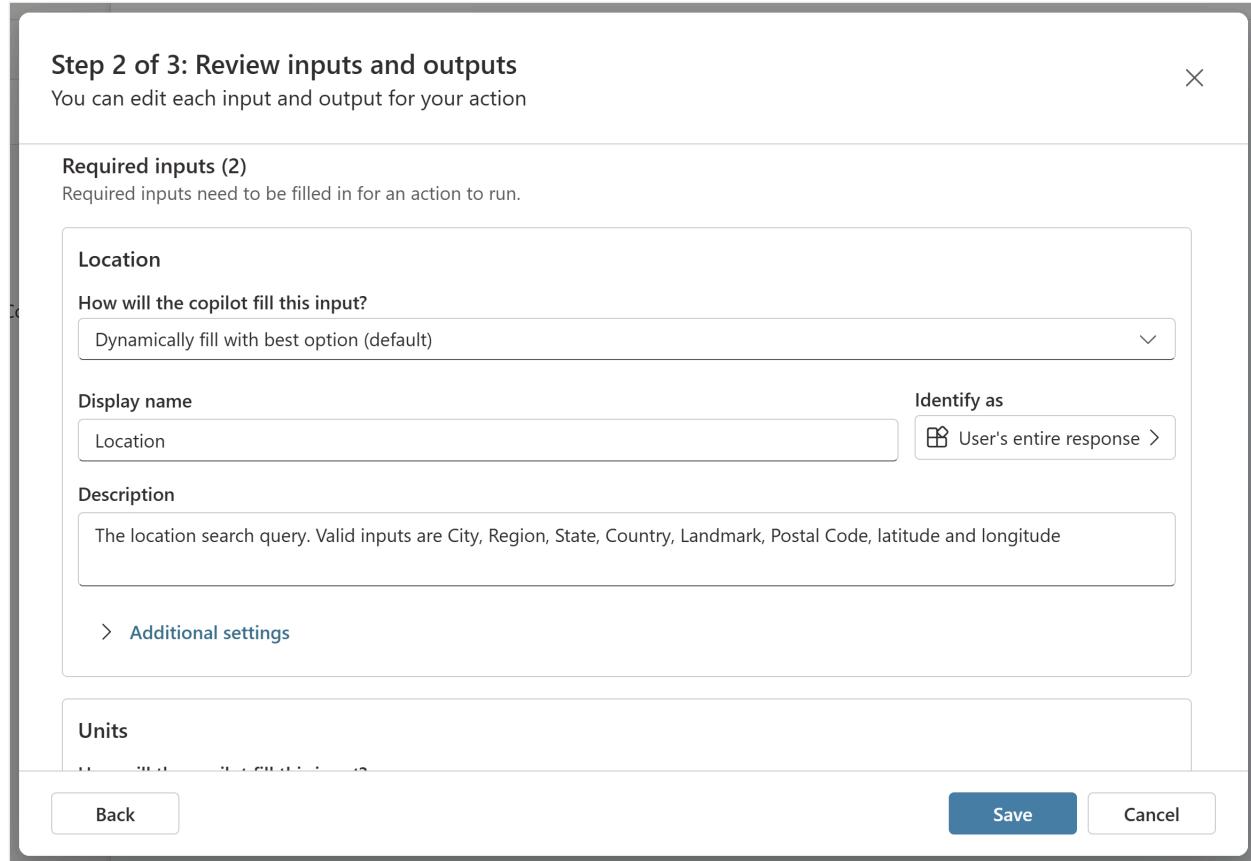
- Name: Location
- Description: The location search query. Valid inputs are City, Region, State, Country, Landmark, Postal Code, latitude and longitude

When the agent prompts the user for a location, it generates a question such as *What is the location you would like to check the weather for? You can enter a city, region, state, country, landmark, or postal code.*

You can change the **Identify as** option to a specific entity type, depending on the input being collected, which can help the agent accurately extract a value from a user's response.

Inputs

By default, each input is set to **Dynamically fill with the best option**. 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. You can override an input with a value instead of asking the user. To set an override, for **How will the agent fill this input?**, select **Set as a value**, and enter the desired value. This value can be a literal value, an existing variable, or a Power Fx formula.

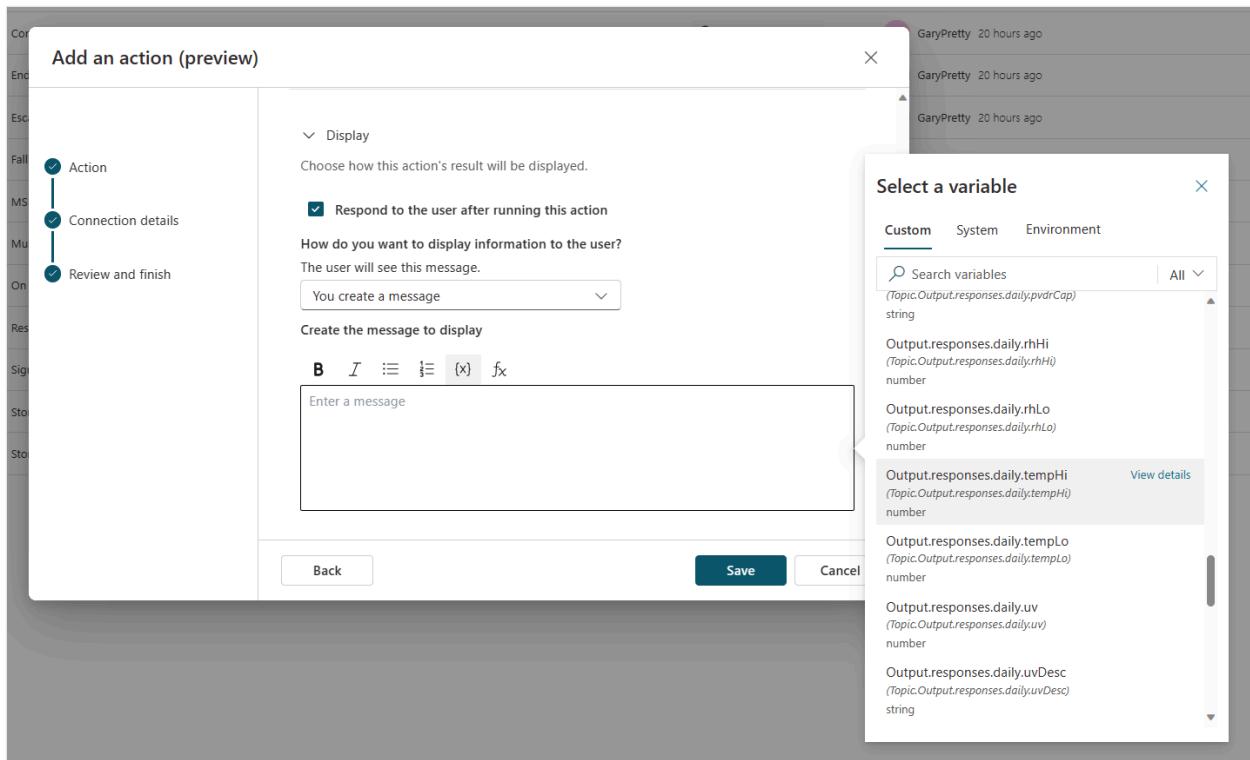


To control your agent's behavior when asking for a value or validating a user's answer, you can configure [additional settings](#) for inputs.

Outputs

By default, your action is configured to automatically generate a contextual response for a user, based on their query and the result of the action. For example, if a user asks, "What is the current temperature in London?", a contextual reply is generated using GPT, such as "The current temperature in London is 9 degrees Celsius."

Alternatively, you can choose to author a specific response for your action to return. As with the confirmation editor, you can insert references to output variables from the action using the variable picker, as shown in this example.



Call an action from within a topic

You can call an action explicitly from within a topic, similar to calling other action types like Power Automate Flows. Depending on your use case, you might use your Flow as part of a wider topic, which uses more nodes. Or, like in the weather example, adding a single action node to a topic might be all you need.

To call an action from within a topic:

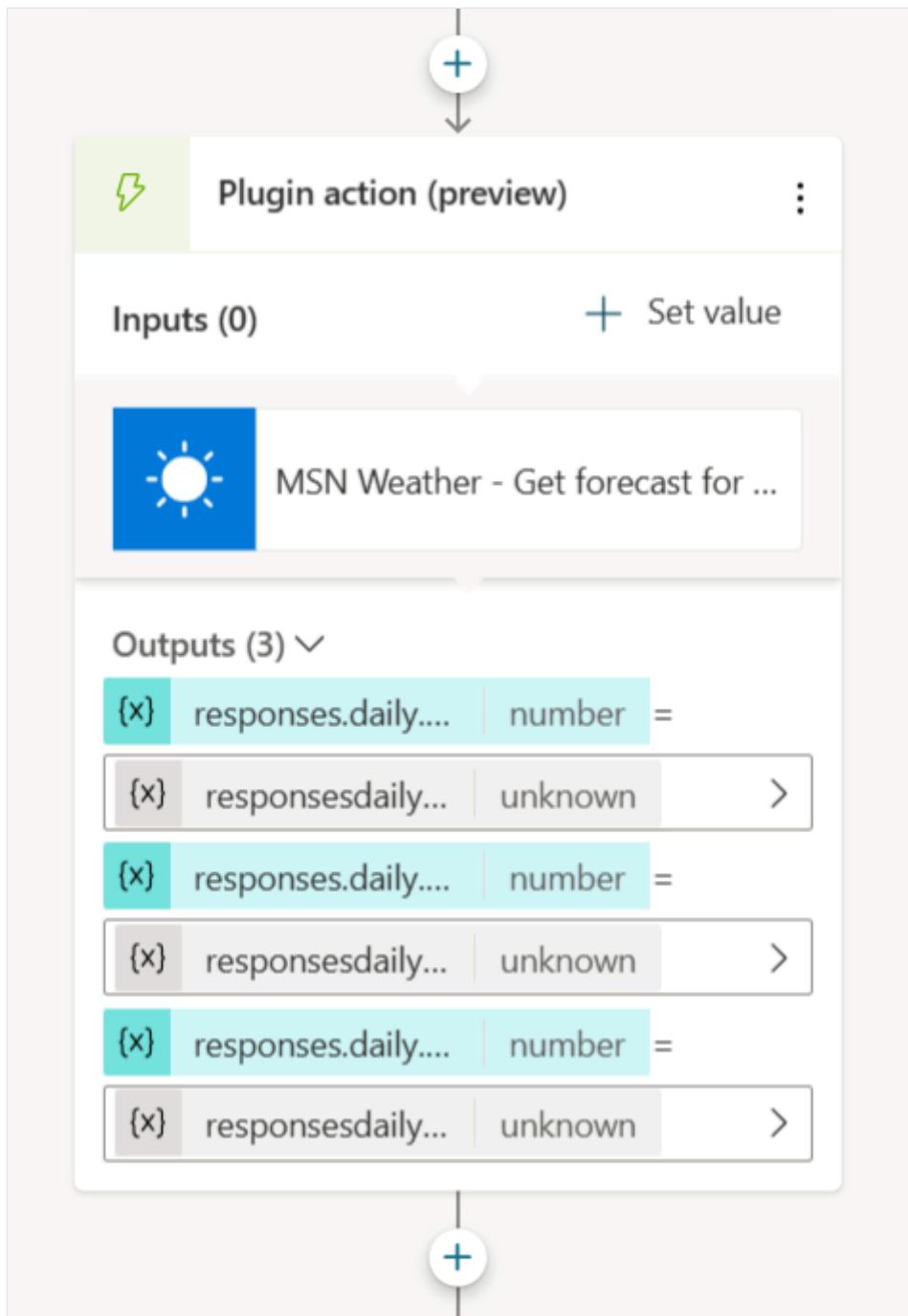
1. In Copilot Studio, go to the [Topics](#) page for the agent you want to edit.
2. Create a new topic, and name it **Get weather**.
3. Add the following **Trigger phrases**:
 - *will it rain*
 - *today's forecast*
 - *get weather*
 - *what's the weather*

The screenshot shows the Microsoft Bot Framework Composer interface. On the left, there is a 'Trigger' node with a list of 'Phrases' including 'will it rain', 'today's forecast', 'get weather', and 'what's the weather'. A context menu is open on the right, titled 'On Recognized Intent'. It includes sections for 'Phrases' (instructions for adding more), 'Add phrases' (with a text input field containing 'will it rain'), and 'Advanced'.

4. Select **Add node (+)** and then select **Call an action**. Switch to the **Plugin actions (preview)** tab and then select the action you created earlier, **Get weather forecast**.

The screenshot shows the 'Select an action' dialog in the Microsoft Bot Framework Composer. The 'Plugin actions (preview)' tab is selected. A search bar is at the top, followed by a list of actions. The 'MSN Weather - Get forecast for today' action is selected, with its description: 'Get the forecast for the current day in the specified location.' visible.

5. Your **Plugin action (preview)** node is now added to your topic.

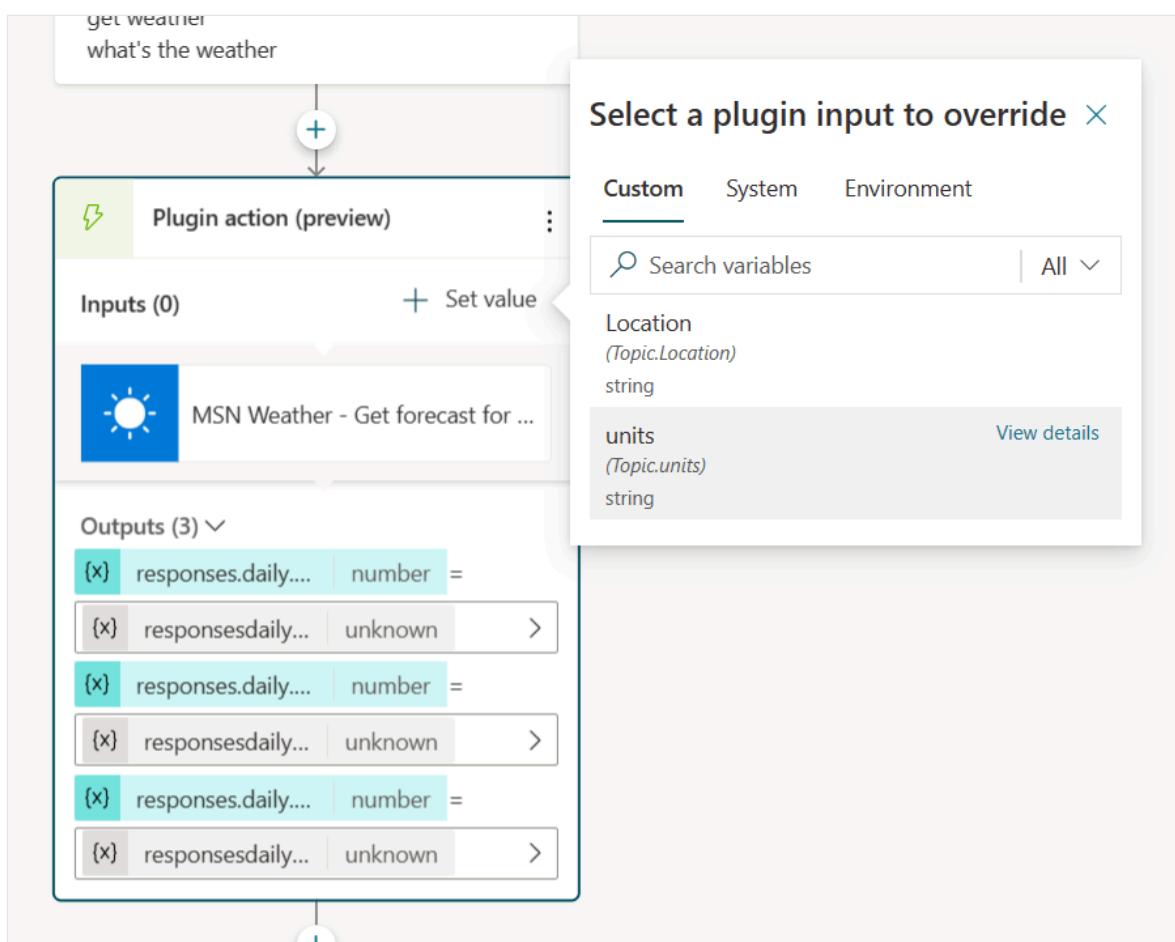


6. Select Save.

Override inputs when calling an action from a topic

By default, when your action runs, your agent generates questions for the user to populate the required inputs. However, you might want to override one or more inputs and provide an explicit value instead of asking the user. You can override in the plugin action configuration within the **Inputs** section. Or, you might override values only when your action is used from a specific topic:

1. Select **Set value** on the node and choose the input you would like to override.



2. Enter the value for the input by typing a value, selecting a variable, or using a formula.

Plugin action (preview)

Inputs (1) + Set value

{x} units string = Metric >

MSN Weather - Get forecast for ...

Outputs (3) ▾

{x} responses.daily.... number = {x} responsesdaily... unknown >

{x} responses.daily.... number = {x} responsesdaily... unknown >

{x} responses.daily.... number = {x} responsesdaily... unknown >

Now that this input is overridden, the agent doesn't ask the user for a value.

Enable authentication for AI actions

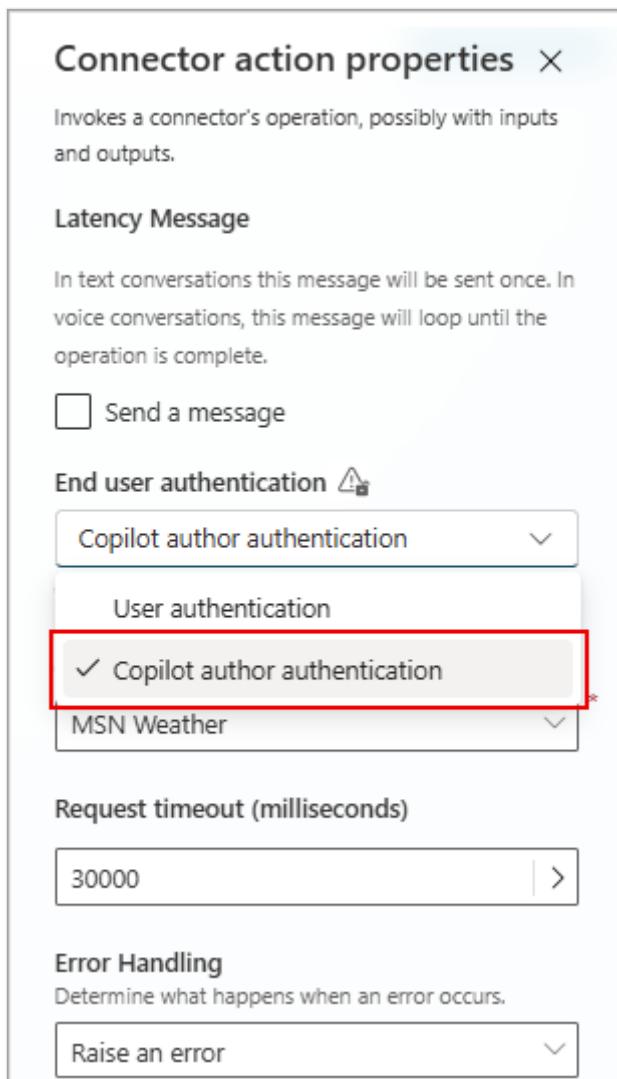
Some actions, when created based on AI actions, require authentication to work correctly, such as Dynamic Prompt or others that call a Dataverse API.

Actions 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:

1. Go to the connector action properties.

2. Under End user authentication, select Agent author authentication.

3. Publish and test the experience in the agent test pane, or in a desired channel.



Use flow actions with user credentials

Flow actions can be [configured to use user credentials](#). Using a flow in a supported channel requires no special configuration in your agent dialog, and users are able to [manage their connections](#) for a given flow in a provided experience.

Authentication for actions based on plugins in your library

When created based on AI plugins, actions require authentication to work correctly. This is most commonly seen with actions based on prompts or others that call Dataverse APIs. Actions based on plugins are always run in the user context and can't be run unless authentication is enabled. Two types of authentication methods are supported:

- **Integrated authentication (Teams and Power Apps):** You can use this authentication method if you want to publish your agent to a Teams channel. This method also works for agents embedded in any Power Platform app that uses tokens built with the PPAPI standard.
- **Manual authentication (third-party agents):** This authentication method involves the setup of a Microsoft Entra ID connection.

To create a manual authentication connection:

1. Set up a [Microsoft Entra ID app](#).
2. Set up [manual authentication](#) for your agent.
3. Allow your app to use delegation to [fetch a token on behalf of the user](#). Be sure to add any necessary scopes.

Authentication X

Require users to sign in

Redirect URL
 Copy

Service provider *

Client ID *

Client secret *

Token exchange URL (required for SSO) [Learn more about SSO](#)

Tenant ID

Scopes ⓘ

Save Close

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

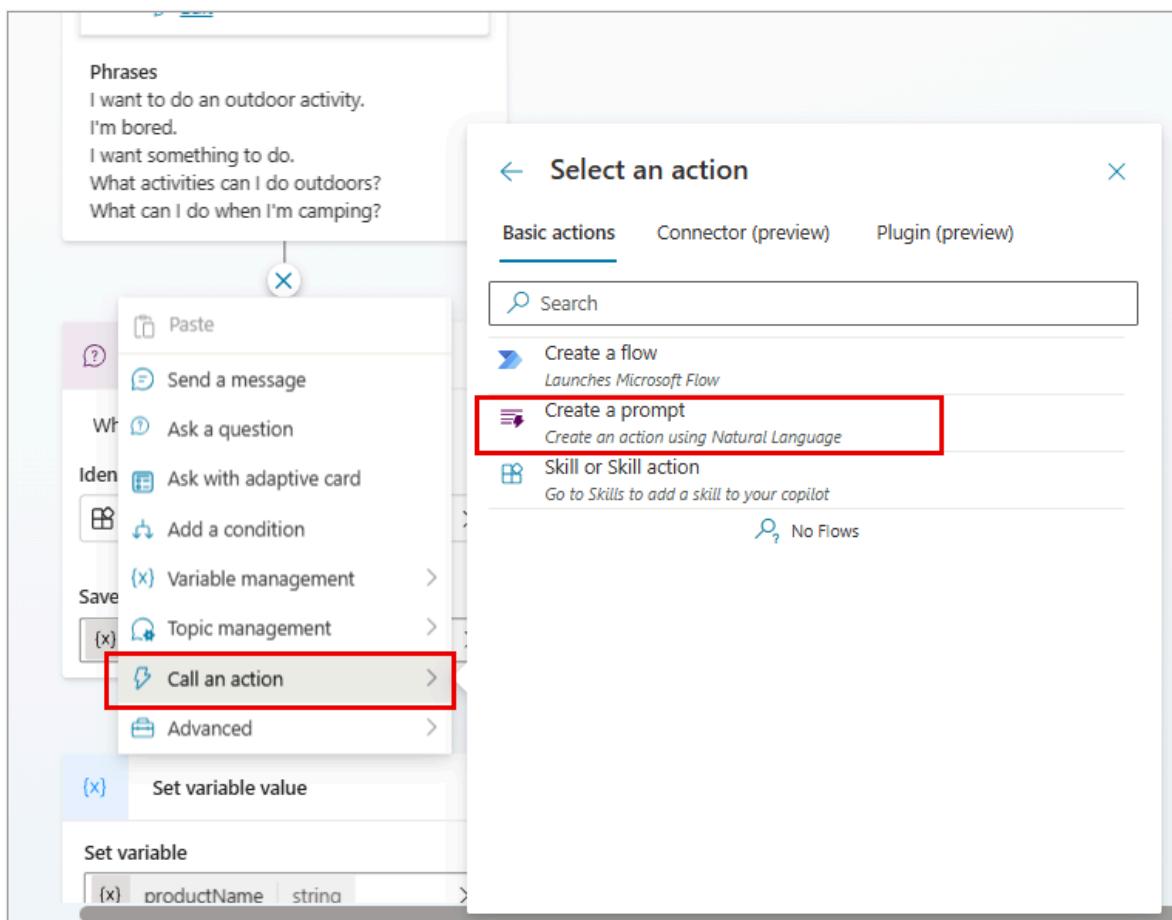
Use prompts to make your agent perform specific tasks

Article • 11/19/2024

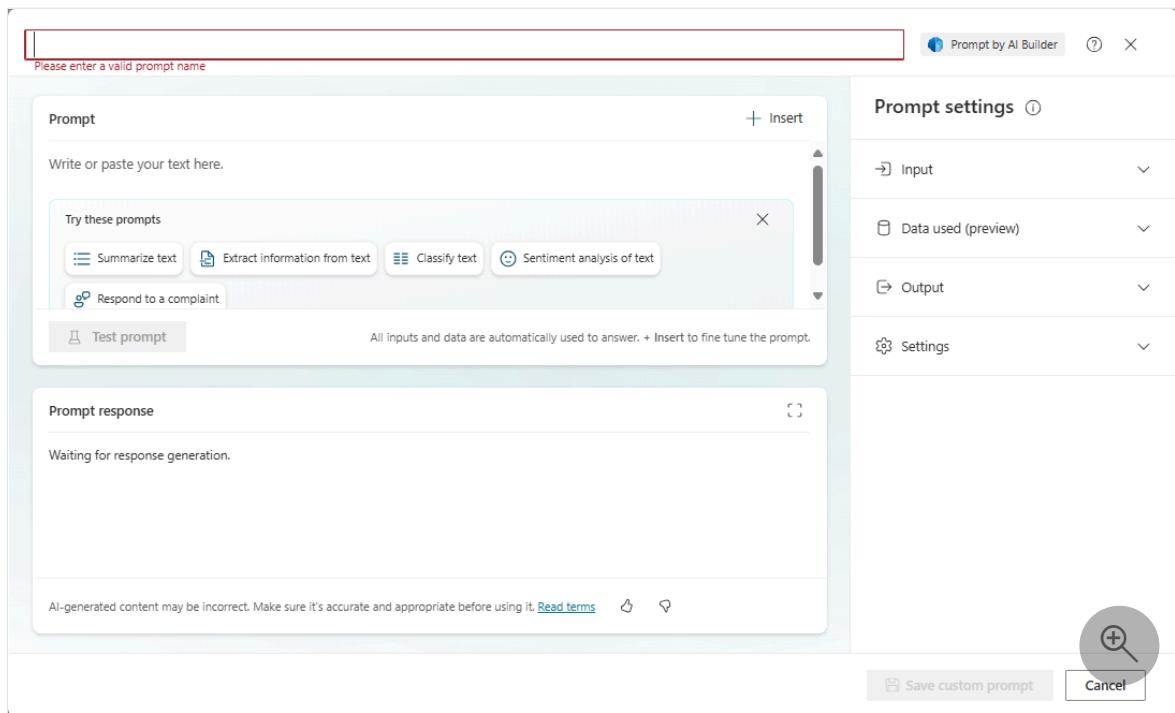
A prompt node integrates the [custom prompt authoring](#) feature of AI Builder to give instructions to the GPT model for your custom agent.

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 Call an action > Create a prompt.



3. Provide the agent with context, instructions, or other information relevant to your use case.



① Note

If Manual Authentication is used, additional scopes need to be configured.

Follow the instructions provided at [Configure user authentication with Microsoft Entra ID](#)

Scopes should include: `https://[OrgURL]/user_impersonation`

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.
- **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

- Prompts run on a GPT-3.5 Turbo model powered by Azure OpenAI Service.
- This capability is limited to specific regions.
- This capability might be subject to usage limits or capacity throttling.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use Power Platform connectors (preview) in Copilot Studio

Article • 11/19/2024

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 a set of prebuilt actions 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

Microsoft Power Platform connectors are essential tools that greatly extend the functionality of Copilot Studio, enabling them to perform a wide range of tasks by connecting with various external services and applications. This integration empowers agent makers to create more dynamic, responsive, and useful agents, tailored to specific business needs and processes.

You can call connectors as [connector actions](#) in your agent, from the **Call an action** node in [conversational topics](#), and through cloud flows as actions or within topics.

Add a connector action

1. Select **Add node (+)** on the authoring canvas.

2. In the node selection window, select **Call an action > Connectors (preview)**, and search for the connector you want to add.
3. Configure the required inputs and outputs and any optional ones you need for your experience.

By default the connection is configured to use user credentials. For more information about the supported authentication modes, see [Configure user authentication for actions](#). To change this behavior, see the following section.

Use connectors with agent author's credentials

Connector actions require a valid set of credentials. By default, connector actions are configured to ask users (users of your agent) to provide their credentials for the associated service, when the action 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 action to your agent as a plugin action, and configure it.
3. Go to the connector action properties.
4. Under **End user authentication**, select **Agent author authentication**.
5. Publish and test the experience in the **Test your agent** pane, or in a desired channel.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use Power Automate flows in Copilot Studio

Article • 11/19/2024

Extend the capabilities of your agent with [cloud flows](#) that you build in [Power Automate](#) using low-code, drag-and-drop tools. You can use flows that already exist in your [Power Apps environment](#) or create a flow from the Copilot Studio [authoring canvas](#).

Important

To use cloud flows in custom agents in Copilot Studio, the flows must:

- Have the **Run a flow from Copilot** trigger.
- Be located in a solution within the same environment as the agent. Existing flows can be [added to a solution](#) to be available when adding actions to your agent.
- [Return a response synchronously](#) with the **Respond to Copilot** action, within the 100-second action limit. To avoid timeouts, optimize the flow logic, queries, and the amount of data returned so that a typical run is below this limit. Actions that need to run longer can be placed after the **Respond to Copilot** action to continue executing for up to the [flow run duration limit](#), which is typically 30 days.

In this section

The examples in this section help you learn to use flows to do more with your agents.

 Expand table

Article	Description
Create a flow	Create a Power Automate flow that provides a weather forecast.
Call a flow as an action	Call a Power Automate flow from an agent topic using an action node.

Article	Description
Use input and output variables to pass information	Pass variables between a Power Automate flow and a Copilot Studio agent.
Return a list of results	Return a list of results from a Power Automate flow to a Copilot Studio agent.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

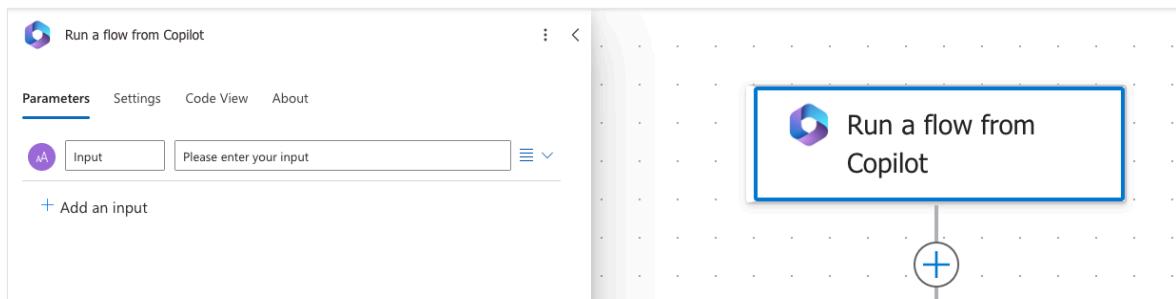
Create a flow

Article • 11/19/2024

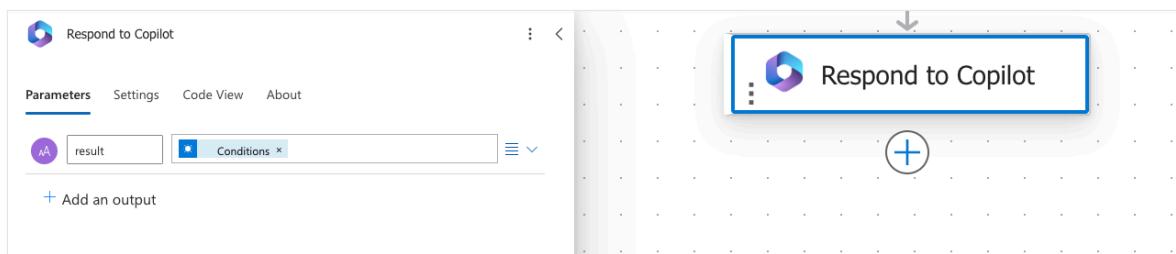
Learn how to use a cloud flow in Copilot Studio by giving an agent the ability to produce a weather forecast on request. In this first example, you'll create a flow that fetches a weather forecast. In the next example, you'll [call the flow as an action from a topic](#) so the agent can respond with the forecast when a customer asks about the weather.

To work with agents, a Power Automate flow requires specific flow trigger and response action.

- Flow trigger: **Run a flow from Copilot**



- Response action: **Respond to Copilot**



When you [create a flow](#) from Copilot Studio, the trigger and the action are added for you. You can [modify an existing flow](#) for use with agents by changing it to use these actions.

ⓘ Important

To use cloud flows in custom agents in Copilot Studio, the flows must:

- Have the **Run a flow from Copilot** trigger.
- Be located in a solution within the same environment as the agent. Existing flows can be [added to a solution](#) to be available when adding actions to your agent.

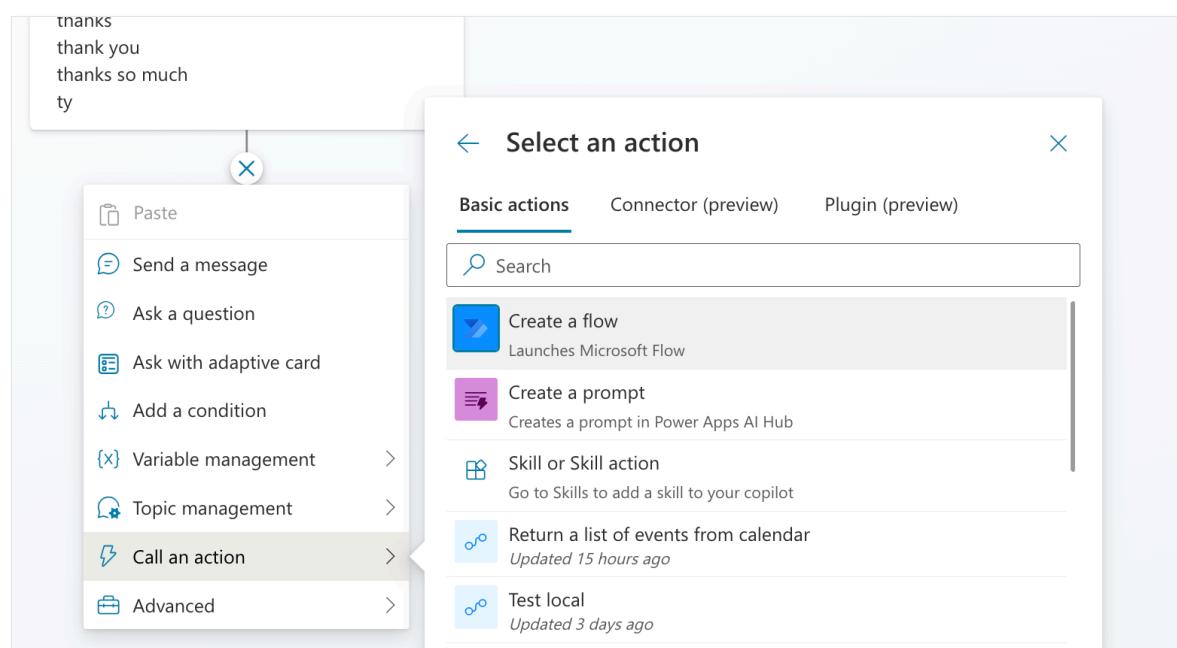
- [Return a response synchronously](#) with the **Respond to Copilot** action, within the 100-second action limit. To avoid timeouts, optimize the flow logic, queries, and the amount of data returned so that a typical run is below this limit. Actions that need to run longer can be placed after the **Respond to Copilot** action to continue executing for up to the [flow run duration limit](#), which is typically 30 days.

Prerequisites

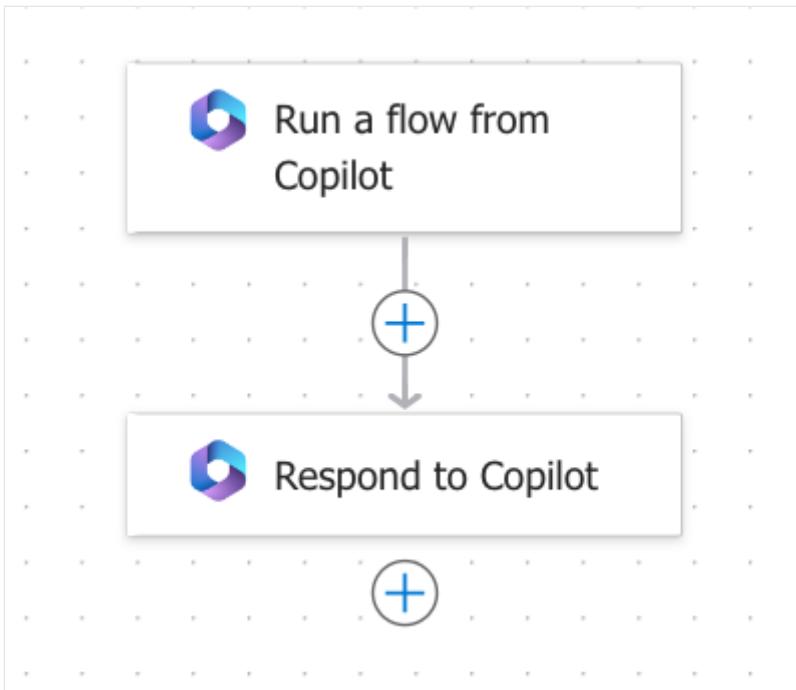
- [Access to Copilot Studio](#) with a license that includes the use of flows within the context of Copilot Studio creations.

Create a flow to 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 **Add node** (+) below an existing node to add a new node.
4. In the node selection window, select **Call an action**, **Basic Action** and then select **Create a flow**.



The [Power Automate portal](#) opens in a new tab, with a starter flow template open for editing. The portal opens in the same environment as the agent and uses the same user credentials.



Tip

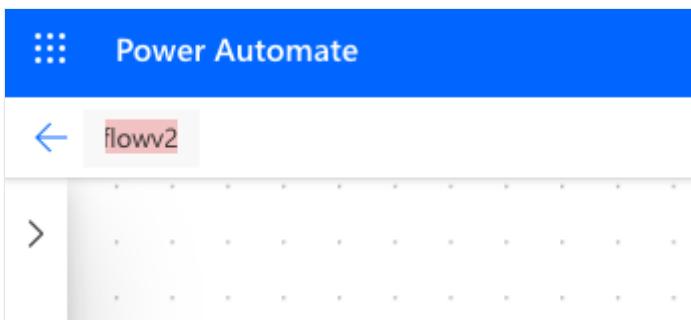
Rename nodes to make them easier to identify. Select the node's name field to update the name directly, or select the **More** icon (...) of the node and select **Rename** from the menu. You can also rename nodes in the [code editor](#).

It's not possible to rename **Trigger** nodes and **Go to step** nodes.

Node names can be up to 500 characters in length.

Now add actions to your flow. In this example, you add an action that fetches the current forecast from MSN Weather.

1. Select the name of the flow and rename it **Get weather forecast**.



2. Add the following input parameters to the **Run a flow from Copilot** trigger:

- City (Text)
- Zipcode (Number)



Run a flow from Copilot



Parameters

Settings

Code View

About



City

Please enter your input



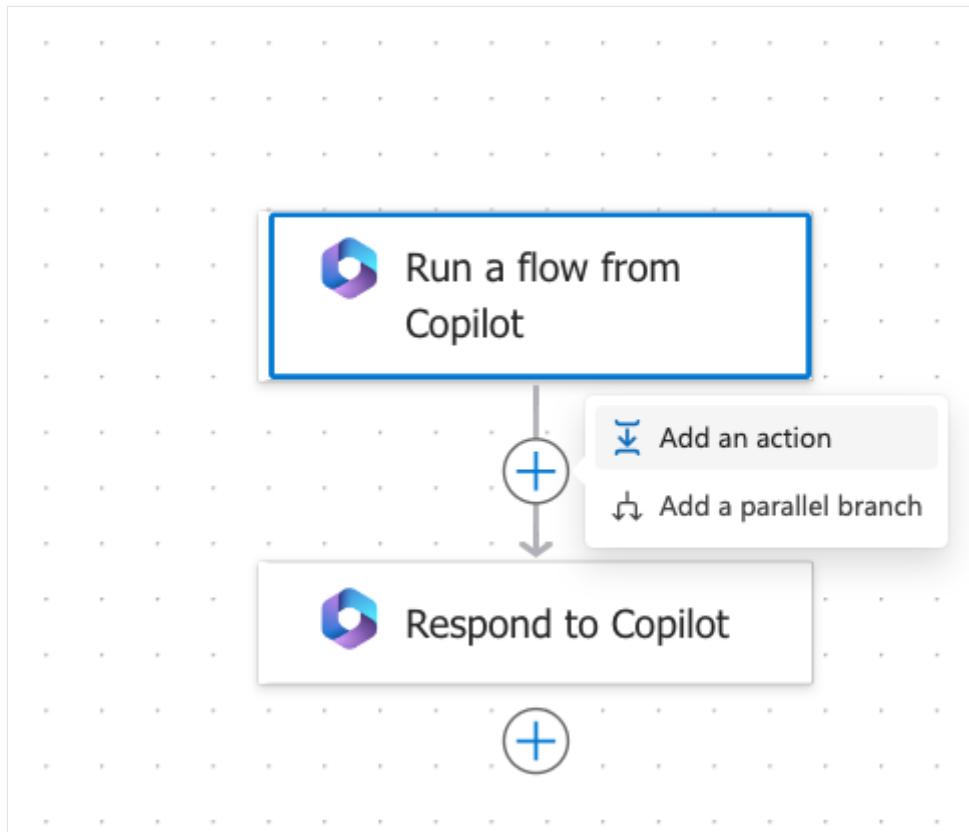
NumZipcode

Please enter a number

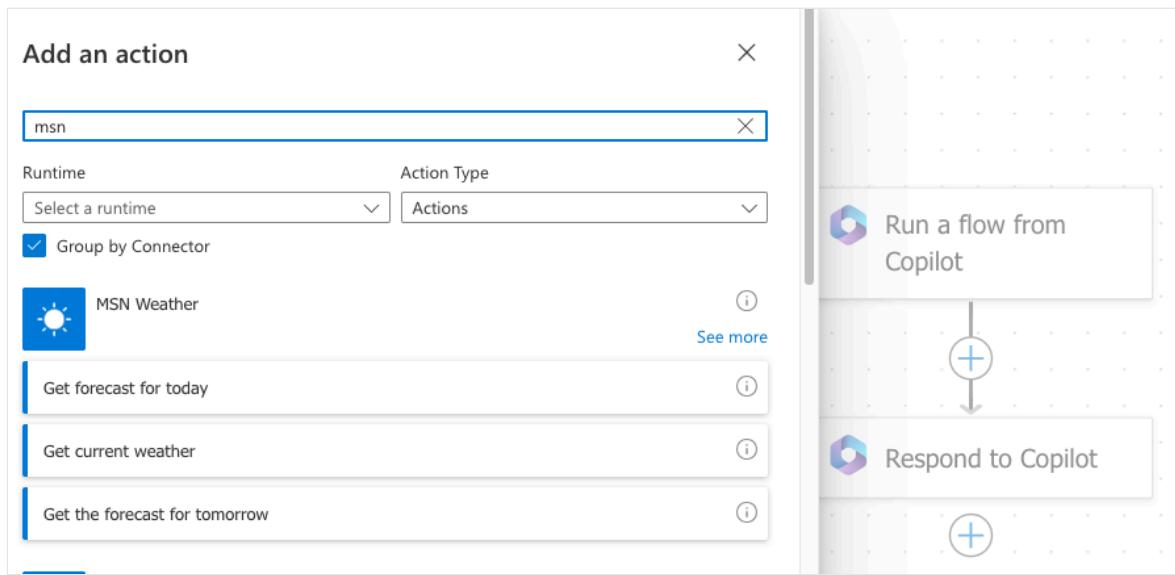


+ Add an input

3. Select **Add an action** to create a new action below the **Run a flow from Copilot** flow trigger.



4. Search for **MSN weather**, and then select the **Get forecast for today** action.



5. In the **Get forecast for today** action, in the **Location** box, select **Add dynamic content**, and then select **City** and **Zipcode**.

Get forecast for today

- [Parameters](#)
- [Settings](#)
- [Code View](#)
- [Testing](#)
- [About](#)

Location *

City NumZipco...

Units *

Imperial

6. In the **Respond to Copilot** action, add the following output parameters and variables:

[Expand table](#)

Output parameter	Type	Variable
day_summary	String	Day Summary
location	String	Location
chance_of_rain	Number	Day Rain Chance



Respond to Copilot

⋮ <

Parameters Settings Code View About

	day_summary	Day ×	
	location	Location ×	
	chance_of_rain	Rain Chance ×	

+ Add an output

7. Select Save.

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. If you are using the flow in a supported channel, this requires no special configuration in your agent dialog, and users will be able to [manage their connections](#) while using the agent.

Cloud flows in environments using [customer-managed keys \(CMK\)](#) cannot 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**.

Modify an existing flow to use with an agent

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 **Run a flow from Copilot** trigger and **Respond to Copilot** response action, and make sure it's in a solution in the same environment as the custom agent.

1. Go to the [Power Automate portal](#) and use the Environment selector at the top right to choose the environment where your flow is stored.

2. Navigate to **My flows** in the left side and search for the flow.

3. Select the flow.

The screenshot shows the Power Automate interface with a search bar at the top containing 'get weather'. Below the search bar, the results are displayed under 'Default Solution > All'. A red box highlights the first result, 'Get weather forecast', which is a Cloud Flow. The table has columns for 'Display name', 'Name', and 'Type'.

Display name ↑ ↓	Name ↓	Type ↓
Get weather forecast	Get weather fore...	Cloud Flow

Add the Copilot trigger and response action

If your flow doesn't already include the Copilot 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 **Run a flow from Copilot**.
3. Select **New step**.
4. Search for the **Copilot connector** and select **Respond to Copilot**. 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

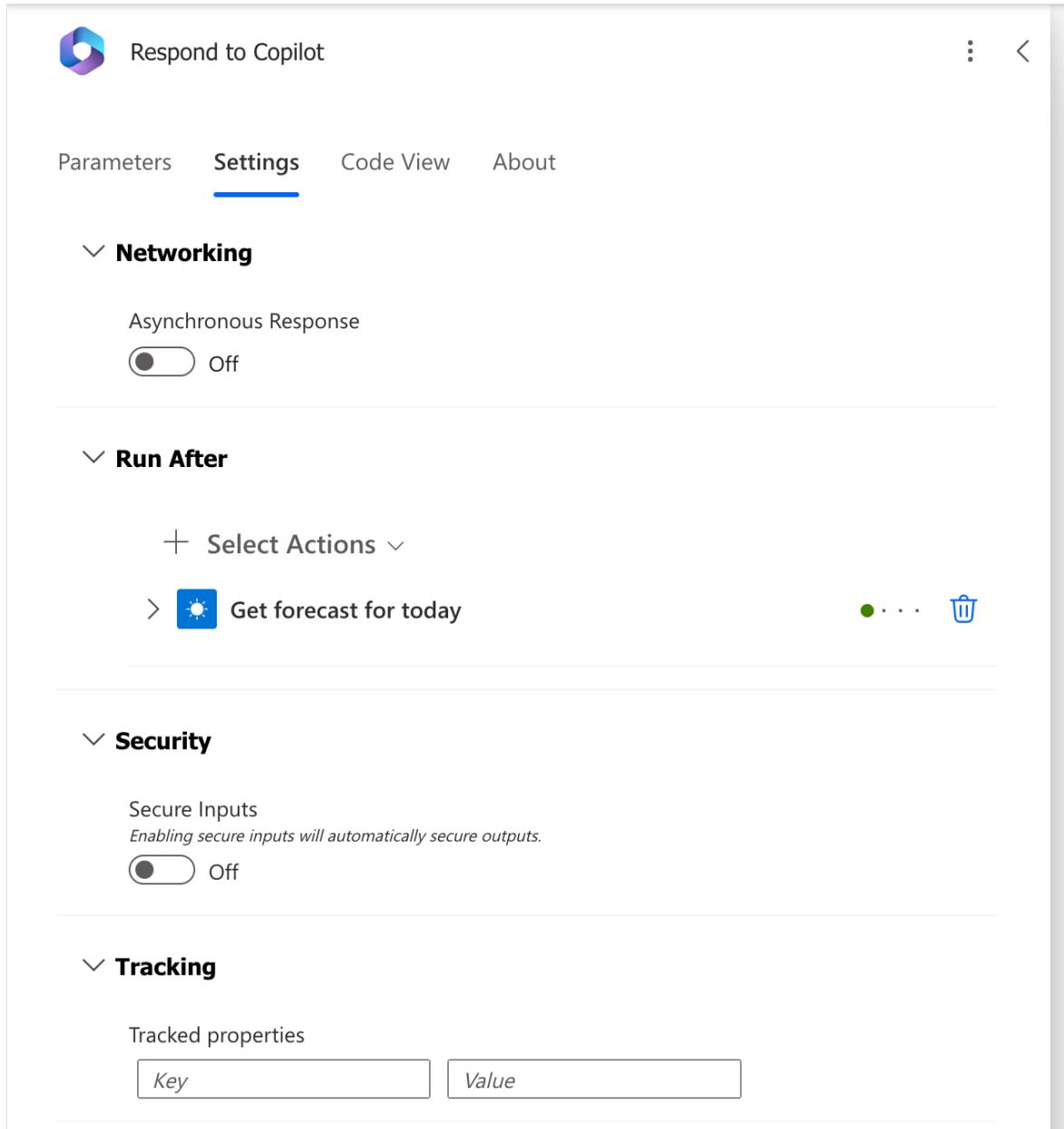
Flows that you want to use in an agent must return values in real time, or [synchronously](#). Flows that run in the background, or asynchronously, may cause an error when your agent tries to run them. Instead of running the flow, the agent will say, "Something unexpected happened. We're looking into 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 is turned off:

1. Open your flow and find the **Respond to Copilot** action(s).

2. Select the **Settings** tab.

3. Set **Asynchronous Response** to **Off**, and then select **Done**.



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 will be 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 the [Power Automate portal](#).

2. Go to **Solutions** and either select an existing solution or create a solution for your flows.

To create a solution, select **New solution**.

The screenshot shows the 'Solutions' tab in the Microsoft Power Apps portal. A red box highlights the '+ New solution' button in the top navigation bar. The main area displays a table of existing solutions with columns for Display name, Name, Created, and Version. Three solutions are listed: 'Contextual Help Base' (msdyn_ContextualH..., 12/6/2019, 1.0.0.10), 'Contextual Help' (msdyn_ContextualH..., 12/6/2019, 1.0.0.10), and 'PowerApps Checker Update' (msdyn_PowerAppsC..., 11/15/2019, 1.1.1.6).

Display name	Name	Created ↓	Version
Contextual Help Base	msdyn_ContextualH...	12/6/2019	1.0.0.10
Contextual Help	msdyn_ContextualH...	12/6/2019	1.0.0.10
PowerApps Checker Update	msdyn_PowerAppsC...	11/15/2019	1.1.1.6

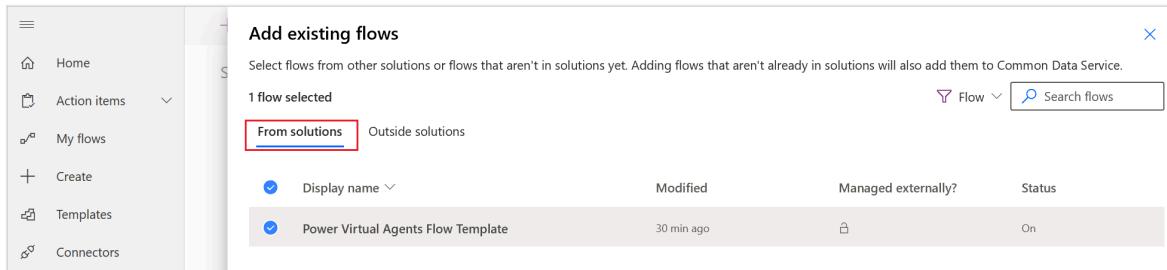
Give your new solution a name, select **CDS Default Publisher** in the **Publisher** field, enter a **Version** number, and then select **Create**.

The screenshot shows the 'New solution' dialog box. It contains fields for 'Display name' (Demo Bot Flows), 'Name' (DemoBotFlows), 'Publisher' (CDS Default Publisher), and 'Version' (1.0). A 'More options' section is partially visible. At the bottom are 'Create' and 'Cancel' buttons.

3. On the **Solutions** tab, go to the solution you want to use. Select **Add existing** to add a flow to it.

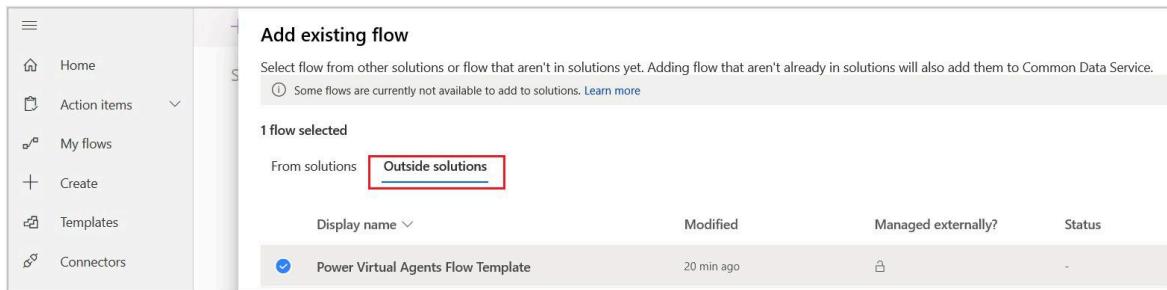
The screenshot shows the details of the 'Demo Bot Flows' solution. The top navigation bar has a red box around the '+ Add existing' button. The main area shows a message 'No components found.' with a note: 'There are no components to show in this view. Create one or more components to get started. [Learn more](#)'. The left sidebar includes links for Home, Action items, My flows, Create, Templates, Connectors, Data, AI Builder, and Solutions, with 'Solutions' being the active tab.

4. On the Add existing flow page, select the **From solutions** tab, select your flow, and then select **Add**.



The screenshot shows the 'Add existing flows' interface. On the left is a sidebar with options: Home, Action items (expanded), My flows (selected), Create, Templates, and Connectors. The main area has a title 'Add existing flows' and a subtitle 'Select flows from other solutions or flows that aren't in solutions yet. Adding flows that aren't already in solutions will also add them to Common Data Service.' Below this is a section titled '1 flow selected' with two tabs: 'From solutions' (selected) and 'Outside solutions'. A table follows, with columns: Display name, Modified, Managed externally?, and Status. One row is shown: 'Power Virtual Agents Flow Template' (modified 30 min ago, managed externally, status On). A search bar at the top right contains 'Flow' and 'Search flows'.

To move a flow from the **My flows** tab to a solution, select **Outside solutions**, and then select **Add** to add your flow to the solution.



The screenshot shows the 'Add existing flow' interface. The sidebar is identical to the previous screenshot. The main area has a title 'Add existing flow' and a subtitle 'Select flow from other solutions or flow that aren't in solutions yet. Adding flow that aren't already in solutions will also add them to Common Data Service.' It includes a note: 'Some flows are currently not available to add to solutions. Learn more'. Below this is a section titled '1 flow selected' with two tabs: 'From solutions' and 'Outside solutions' (selected). A table follows, with columns: Display name, Modified, Managed externally?, and Status. One row is shown: 'Power Virtual Agents Flow Template' (modified 20 min ago, managed externally, status -). A note at the top of the table says 'Display name'.

Related content

- Call a flow as an action
- Use input and output variables to pass information
- Add user authentication to topics

Feedback

Was this page helpful?



Provide product feedback ↗

Call a flow as an action

Article • 11/19/2024

Learn how to use a cloud flow in Copilot Studio by giving an agent the ability to produce a weather forecast on request. In the previous example, you [created a flow](#) that fetches a weather forecast. In this example, you'll use a **Call an action** node in an agent topic to call the flow when a customer asks about the weather.

If you haven't already [created the weather forecast flow](#), do that now.

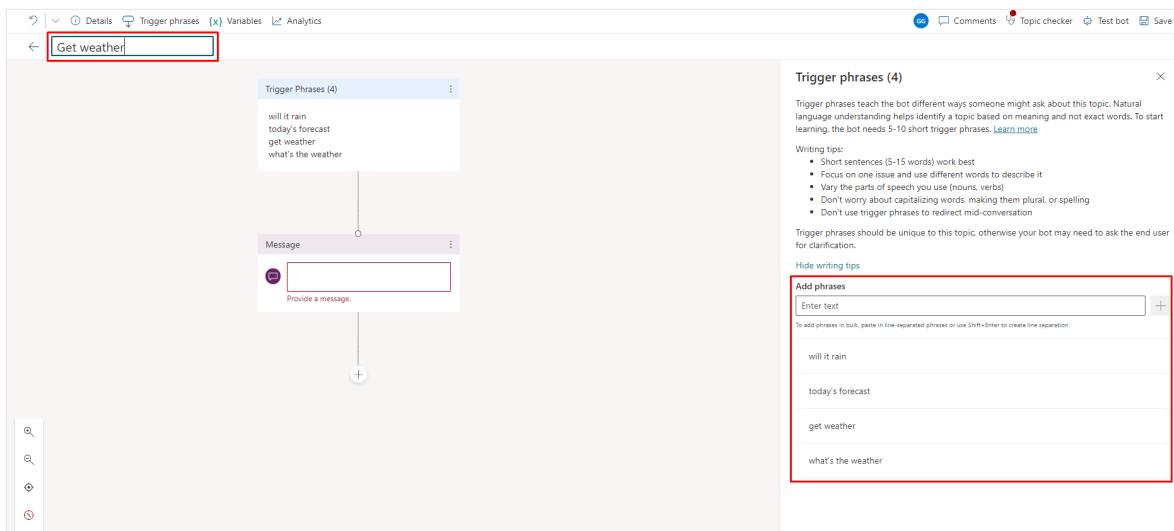
Prerequisites

- [Create a flow](#)

Call a flow from a topic

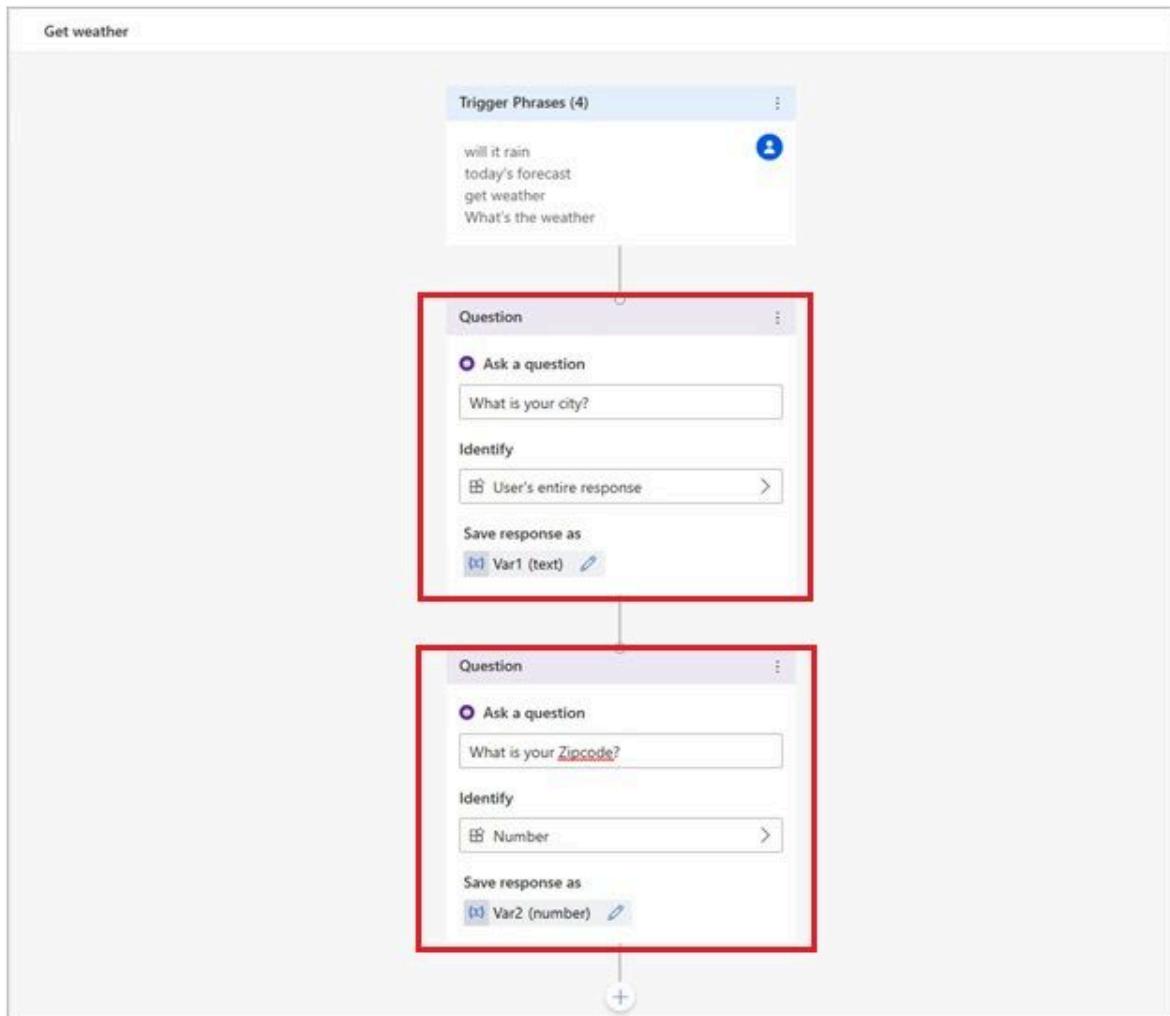
1. In Copilot Studio, go to the [Topics](#) page for the agent you want to edit.
2. Create a new topic, and name it **Get weather**.
3. Add the following **trigger phrases**:

- will it rain
- today's forecast
- get weather
- what's the weather



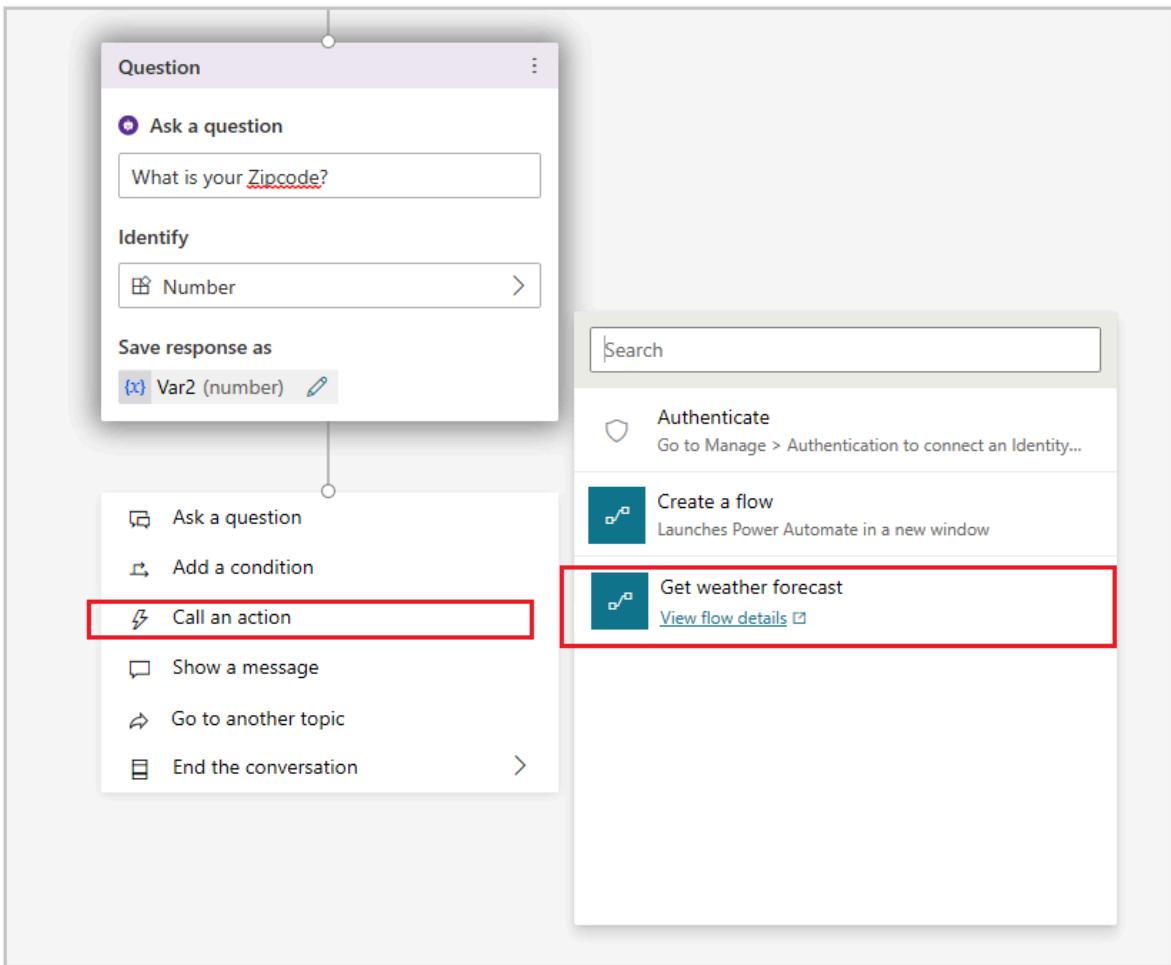
A message node is created.

4. Enter I can help you with that in the message node, and then select **Add node (+)** under it to add a new node.
5. Add two **Ask a question** nodes for the **City** and **Zipcode** inputs the flow needs, as shown in the following example.



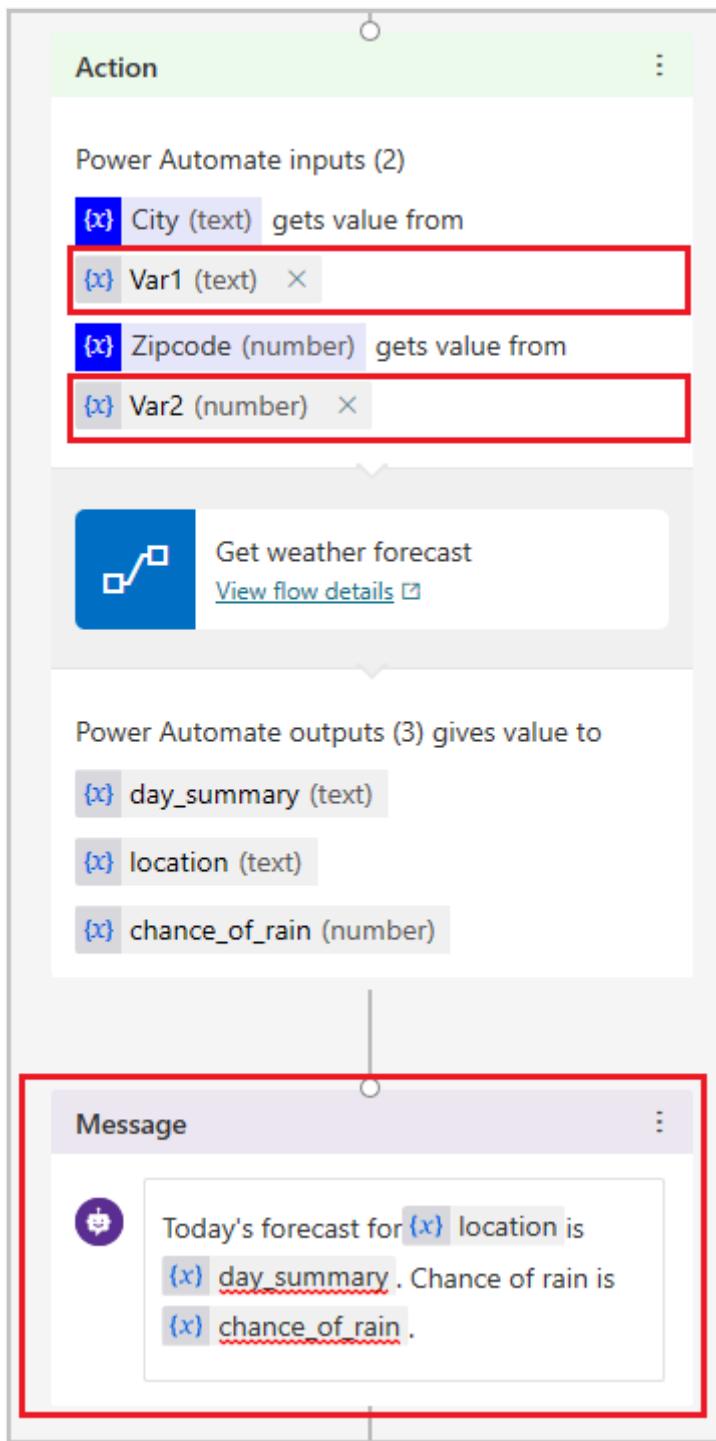
Note the names of the variables associated with each response, **Var1** for **City** and **Var2** for **Zipcode**. You can leave them as-is or select the pencil icon and give them a more meaningful name such as "City" and "Zipcode."

6. Select **Add node (+)** under the **Zipcode** question node.
7. In the node selection window, select **Call an action**, and then select the flow you created earlier, **Get weather forecast**.



8. Assign the flow inputs to the output variables from the question nodes. **City (text)** gets its value from `Var1` and **Zipcode (number)** gets its value from `Var2`. If you changed the names of the variables, enter the new names instead.
9. Under the flow node, add a **Message** node, and then enter a message that uses the flow's outputs. For example:

Today's forecast for `(x)location:{x}day_summary`. **Chance of rain is**
`{x}chance_of_rain%`



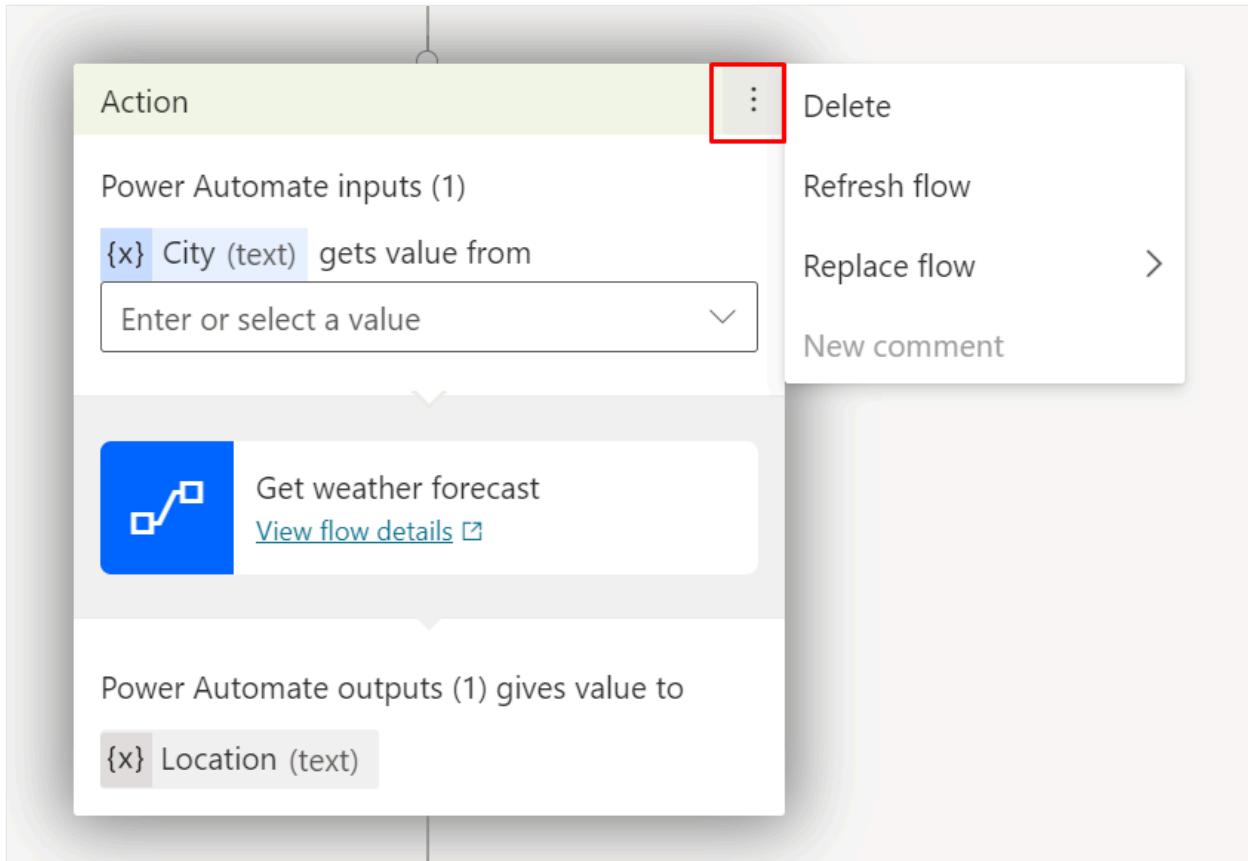
10. Select Save.

Manage a flow used in an agent topic

After you've called a flow from an action node in an agent topic, you can use the node menu to manage the flow. The Copilot Studio authoring canvas gives you three options:

- **Delete:** Select to permanently delete the node, and the flow, from the topic.
- **Refresh flow:** If you've made changes to the flow in the Power Automate portal, select to reload and validate the flow in the topic. You'll need to fix any problems with the flow before you can save the topic.

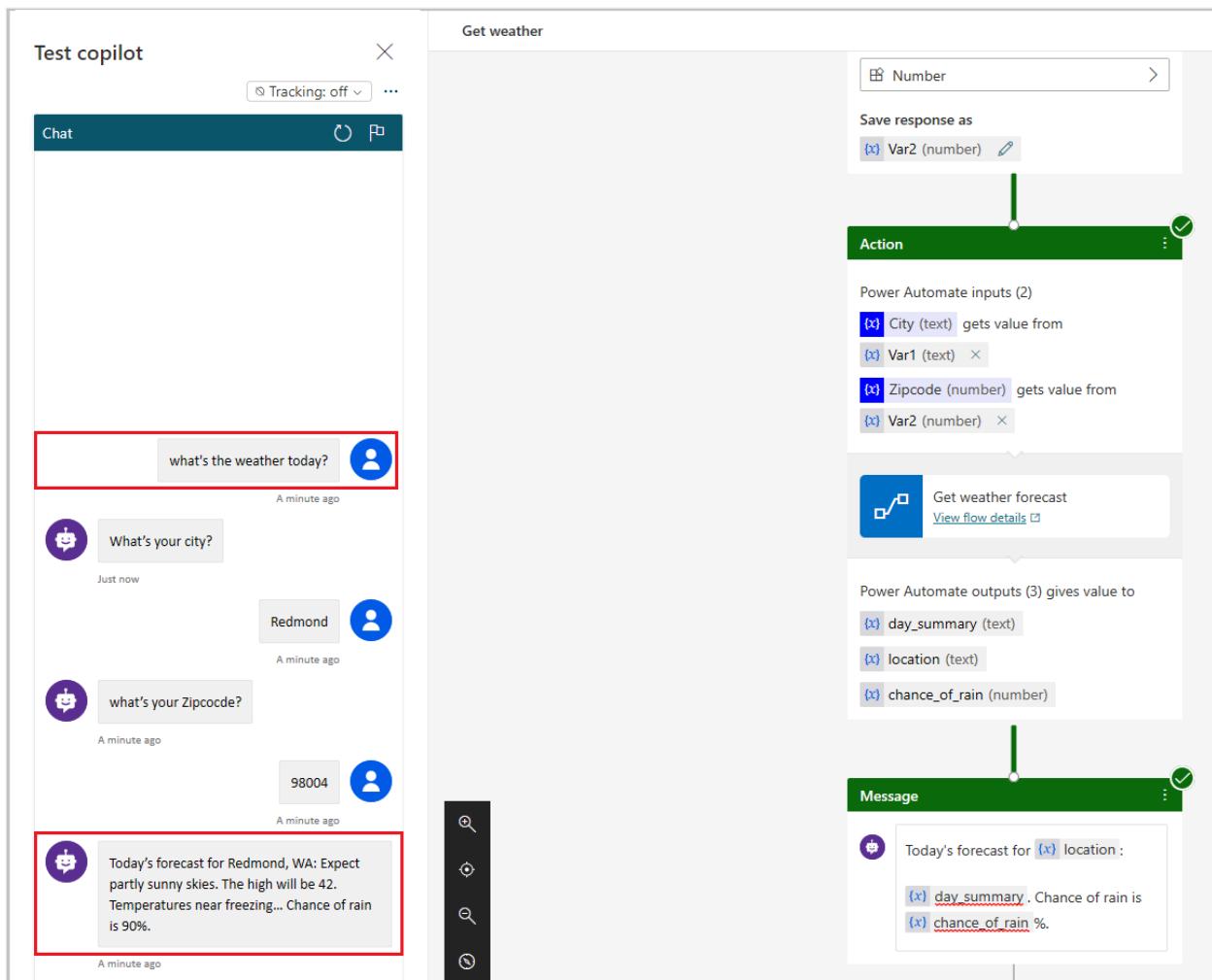
- **Replace flow:** Select to open the action picker, where you can select or create a different flow.



Test your flow and topic

It's a good idea to test the flow and the topic that calls it.

In the **Test agent** pane, 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. Make sure the agent returns today's weather forecast and your message looks the way you expect.



Troubleshoot your agent

Test your agent when you make changes to topics and flows to make sure everything is working. If the agent encounters a problem during a conversation, it responds with an error message.

Find most flow-related issues in the [Flow Checker](#). Any issues on the authoring canvas appears in the [topic checker](#).

Feedback

Was this page helpful?

[Yes](#)

[No](#)

[Provide product feedback ↗](#)

Use input and output variables to pass information

Article • 11/19/2024

Use variables as input and output parameters to pass information between Power Automate 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 the amount of data an agent can send or how many variables an agent can pass.

In this example, you'll 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

Copilots can use the following types of inputs and outputs with Power Automate flows:

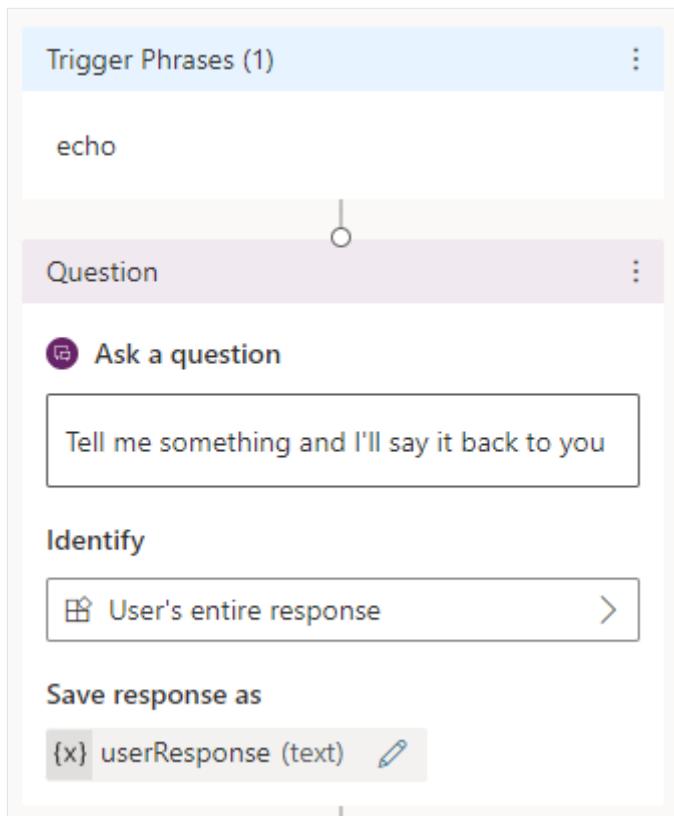
- Number
- String
- Boolean

The following types aren't supported:

- Object
- Date
- Timestamp
- List [String]
- List [Number]
- List [Boolean]
- List [Object]
- List [Date]
- List [Timestamp]

Create a topic and flow

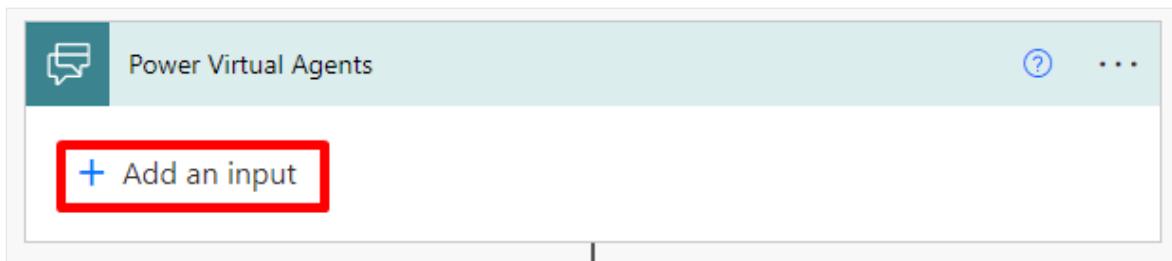
1. Go to the [Topics](#) page for your agent.
2. Create a new topic called **Echo message**.
3. Add the trigger phrase **echo**.
4. Add a **Question** node and enter the message **Tell me something and I'll say it back to you**.
5. For **Identify**, choose **User's entire response**.
6. For **Save response as**, select the pencil icon and rename the variable to **userResponse**.



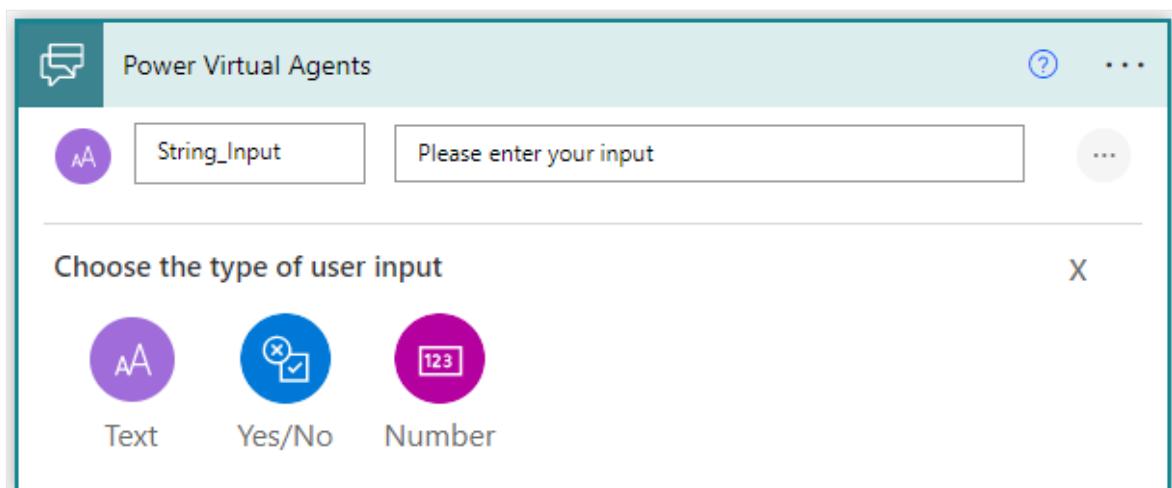
7. On the authoring canvas, select **Add node (+)**. Select **Call an action**, and then select **Create a flow**.
8. In the Power Automate portal, name the flow **Echo parameter**.

Add input parameters to the flow

1. In the Microsoft Copilot Studio flow trigger, select **Add an input**.

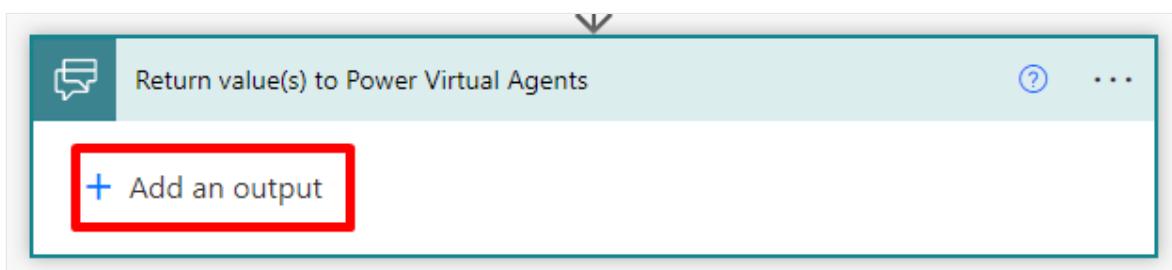


2. Select **Text** and enter the name **String_Input**.

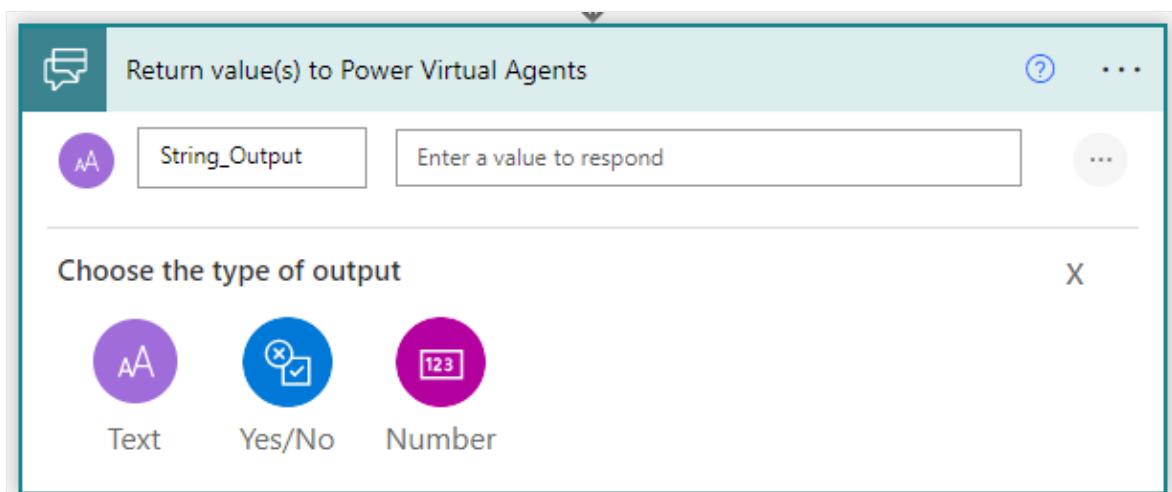


Add output parameters to the flow

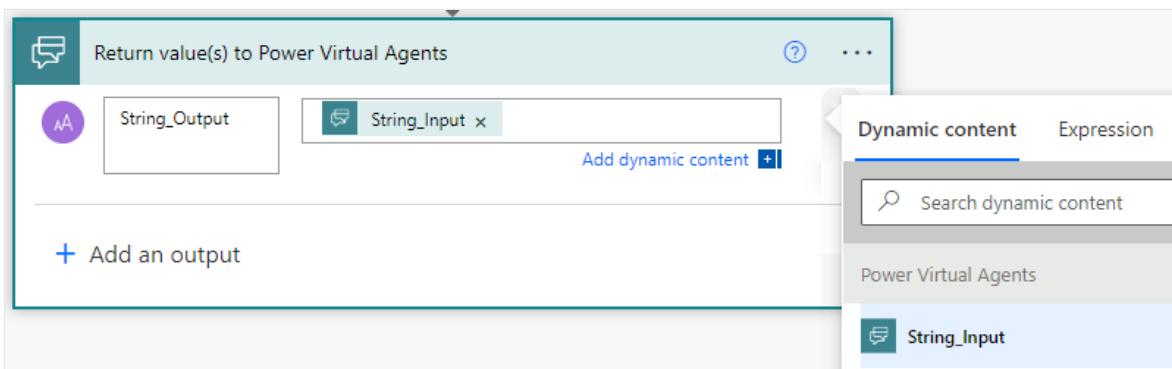
1. In the **Return value(s) to Microsoft Copilot Studio response** action, select **Add an output**.



2. Select **Text** and enter the name **String_Output**.



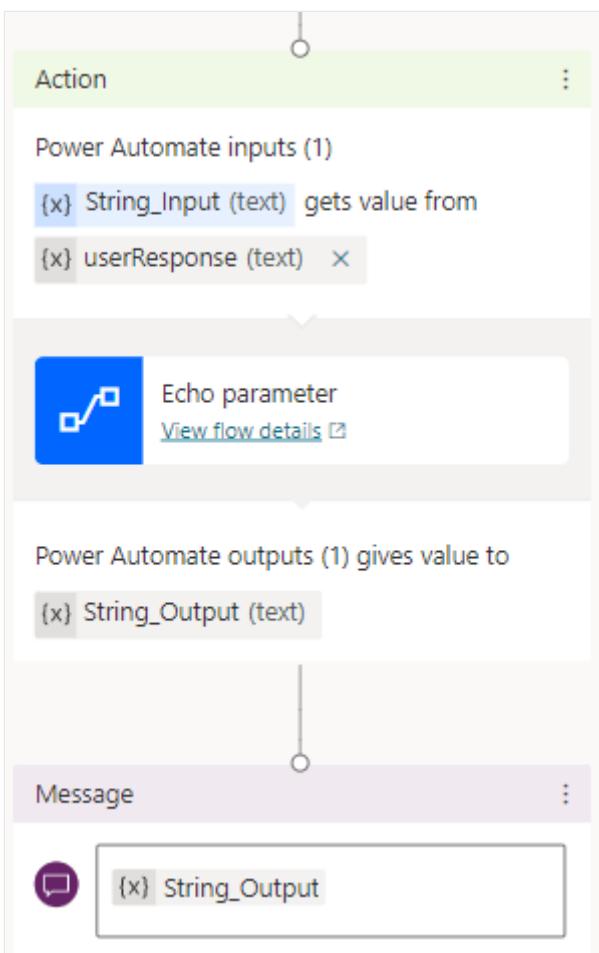
3. For the value of **String_Output**, select the **String_Input** variable.



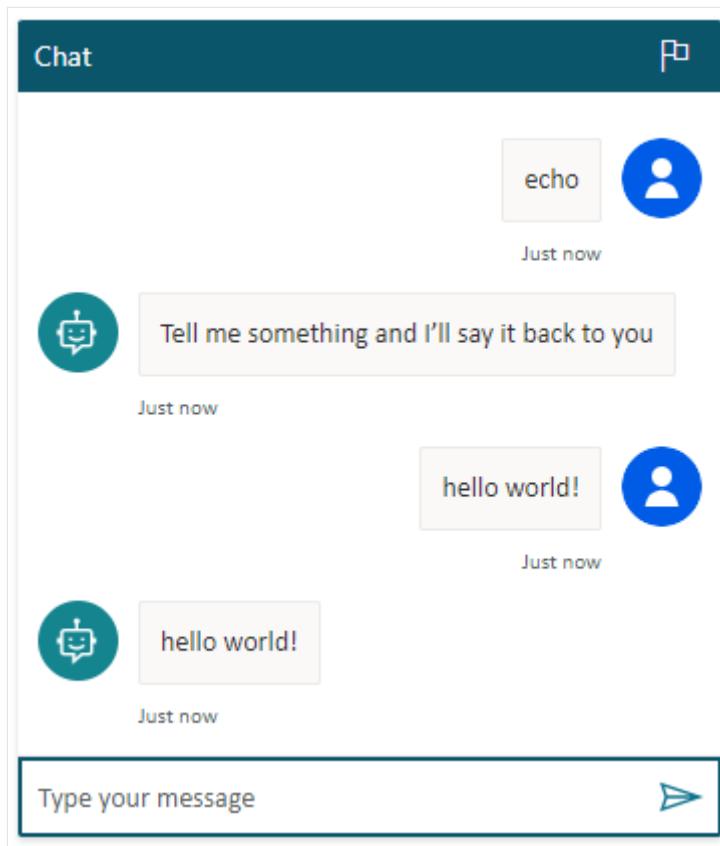
4. Select Save.

Select variables in the agent topic

1. On the Microsoft Copilot Studio authoring canvas, select **Add node (+)**, and then select **Call an action**.
2. Select the flow you created earlier, **Echo parameter**.
3. For **String_Input(text)** gets value from, select **userResponse**.
4. Add a **Message** node. Select **Insert variable**, and then select **String_Output**.

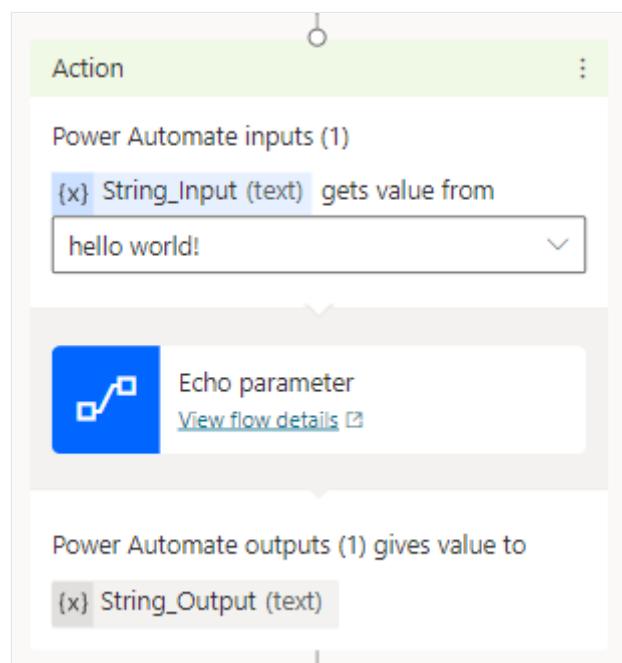


5. Test your topic in the **Test agent pane**.



Optionally use literal values for input parameters

Alternatively, instead of using a variable as an action's input parameter, you can type the value directly into the "gets value from" box.



Related content

- [Work with variables](#)
-

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

Return a list of results

Article • 11/19/2024

It's common to ask an agent to look for data in an external system and return a list of results. To do that, Microsoft Copilot Studio can call a Power Automate flow to:

- Authenticate and connect to an external solution
- Run a query based on the user's inputs
- Format the results
- Return the results to the agent

In this example, you'll use the Dataverse connector in Power Automate to search for accounts. Then you'll return a list of results that includes the account name, city, and account number to Microsoft Copilot Studio.

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](#) that's included automatically in Power Platform environments. However, it doesn't come with account data. Before your agent can search for accounts, you'll need to populate the Account table.

If you already have an Account table with data, skip this step and move on to [Create a topic](#). However, you'll need to use different search terms when you test your agent in later steps.

1. Go to the [Power Apps portal](#).
2. In the side pane, under **Dataverse**, 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](#)

Address 1: City	Account Name	Account Number
Seattle	Contoso Inc	AC0125
San Francisco	Contoso Residences	AC0255
Olympia	Contoso Meal Delivery	AC0035

Create a topic

1. 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 will also work. However, selecting the **Organization** entity takes advantage of the agent's language understanding capabilities to extract the organization name from the response.

The screenshot shows the configuration of a Power Virtual Agents flow. It includes:

- Trigger Phrases (4):** I'm looking for a customer, I'm looking for an account, search account, search customer.
- Message:** Sounds like you're looking for an account. I can look that up for you.
- Question:** Ask a question
- Identify:** Organization
- Save response as:** organization (organization)

7. For **Save response as**, rename the variable to **organization**.

Create a flow

1. On the authoring canvas, select **Add node (+)**. Select **Call an action**, and then select **Create a flow**.
2. In the Power Automate portal, name the flow **Search Account**.
3. Add a **Text** input named **Organization**.

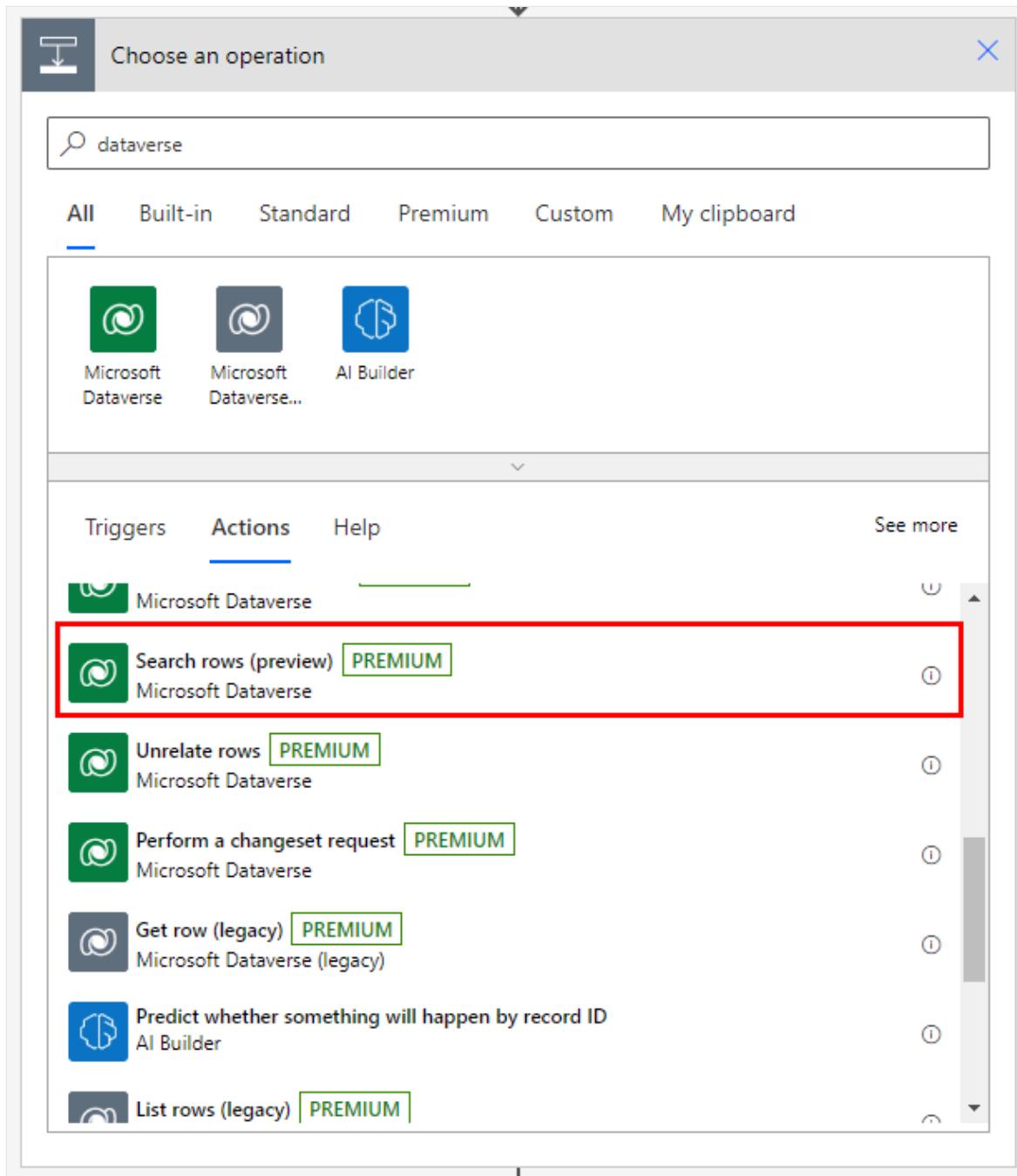
The screenshot shows the configuration of a 'Create a flow' node in Power Automate. It includes:

- When Power Virtual Agents calls a flow (V2):** Organization
- Please enter your input:** Please enter your input
- Choose the type of user input:** Text, Yes/No, Number

4. Select Insert a new step (+) and select Add an action.

5. Select the Microsoft Dataverse connector, and then select the Search rows action.

This action uses fuzzy matching to find relevant accounts in your [Dataverse Account table](#).



6. For Search term, select the Organization variable.

7. Select Show advanced options and set the following items as given:

- Table filter Item: account
- Sort by Item - 1: @search.score desc
- Sort by Item - 2: name asc

*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

Table filter Item - 1

account

+ Add new item

Sort by Item - 1

@search.score desc

Sort by Item - 2

name asc

+ Add new item

Facet query Item - 1

Enter a facet query to narrow the search

+ Add new item

Skip rows Enter the number of search results to be skipped

Return row count Choose an option

Format results

The **Search rows** action returns the **List of rows** variable, which contains JSON data. Before you can use the data, you'll need to analyze it with the **Parse JSON** action.

1. Select **Insert a new step (+)** and select **Add an action**.
2. Select the **Data Operation** connector, and then select the **Parse JSON** action.
3. In the **Content** box, under **Search rows**, select the **List of rows** variable.
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"  
        ]  
    }  
}
```

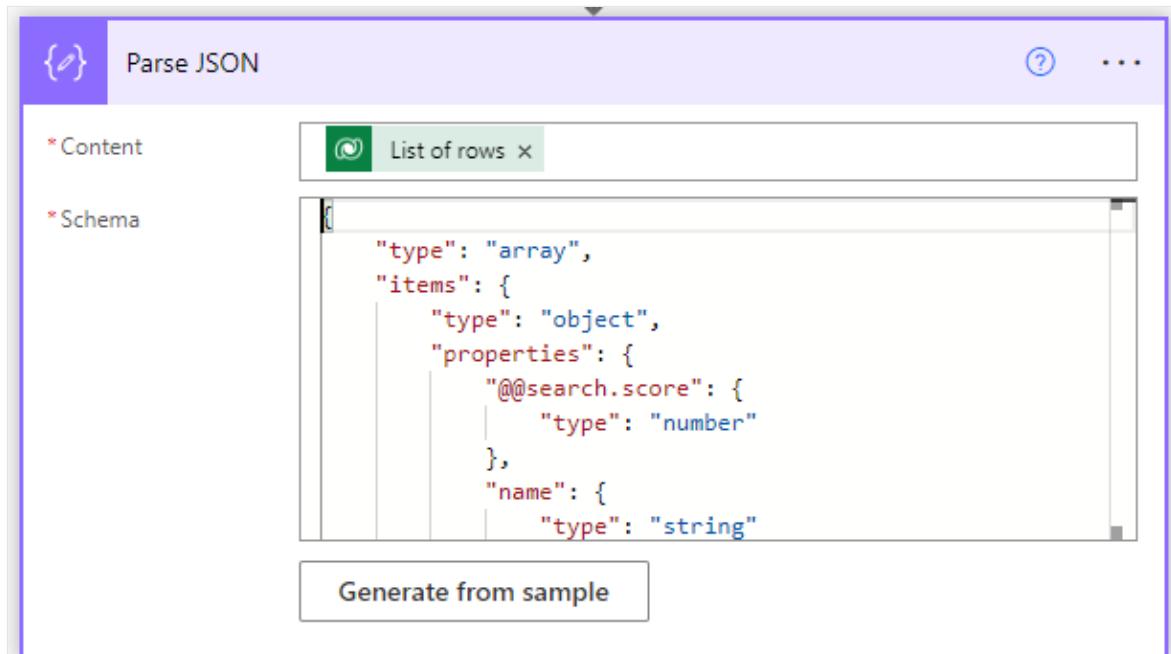
Parse JSON

*Content List of rows

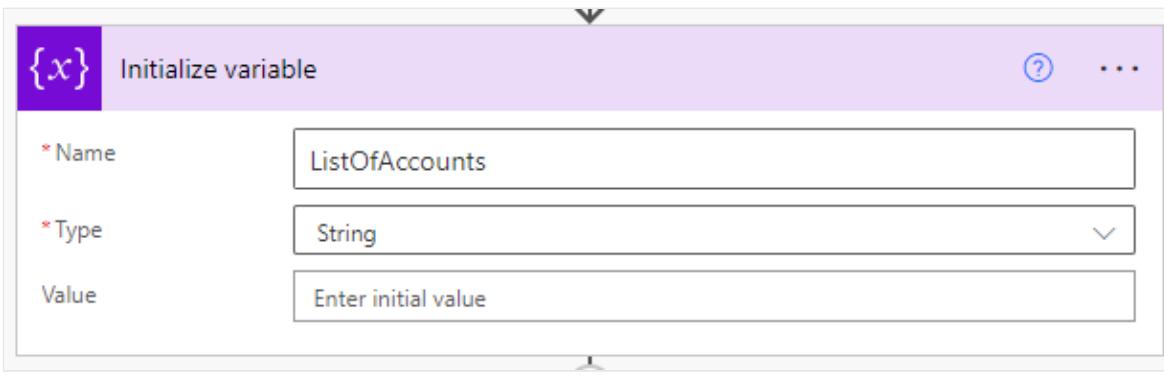
*Schema

```
[  
    {  
        "type": "array",  
        "items": {  
            "type": "object",  
            "properties": {  
                "@@search.score": {  
                    "type": "number"  
                },  
                "name": {  
                    "type": "string"  
                }  
            }  
        }  
    }  
]
```

Generate from sample



5. Select **Insert a new step (+)** and select **Add an action**. Select the **Variable** connector, and then select the **Initialize Variable** action.
6. For **Name**, enter **ListOfAccounts**. For **Type**, select **String**.



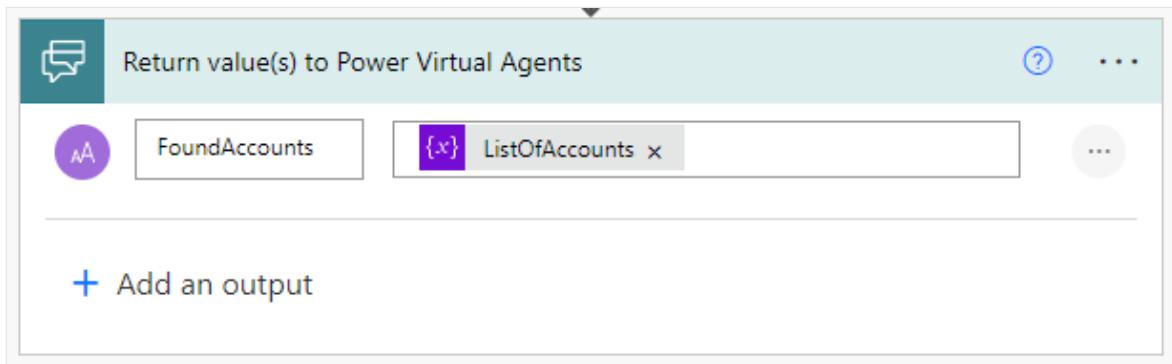
7. Select **Insert a new step (+)** and select **Add an action**. Select the **Control** connector, and then select the **Apply to each** action.
8. Select the **Select an output from previous steps** box to open the **Dynamic content** menu. Search for **body**, and then under **Parse JSON**, select the **Body** variable.
9. Select **Insert a new step (+)** and select **Add an action**. Select the **Variable** connector, and then select the **Append to string variable** action.
10. For **Name**, select **ListOfAccounts**. 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']}
```

11. Add a line break after the snippet to make each result appear on its own line.

The screenshot shows the 'Apply to each' step configuration. The 'Select an output from previous steps' dropdown is set to 'Body'. Below it, an 'Append to string variable' step is configured with 'Name' set to 'ListOfAccounts' and 'Value' set to the formula: '- {e} accountnumber x ; {e} name x - {e} address1_city x'.

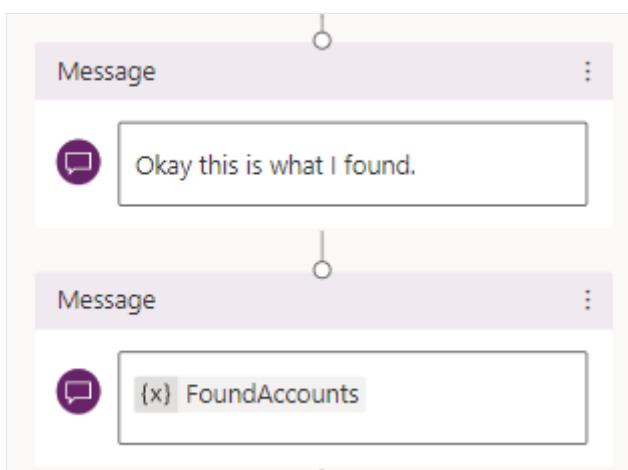
12. In the **Return value(s) to Microsoft Copilot Studio** action, add a **Text output**. For **Name**, enter **FoundAccounts**. For **Value**, select **ListOfAccounts**.



13. Select **Save**.

Call the flow from Microsoft Copilot Studio

1. On the Microsoft Copilot Studio authoring canvas, select **Add node (+)**, and then select **Call an action**.
2. Select the flow you created earlier, **Search Account**.
3. For **Organization gets value from**, select the **organization** variable.
4. Add a **Message** node and enter the message **Okay, this is what I found**.
5. Add a second **Message** node. Select **Insert variable**, and then select **FoundAccounts**.



6. Select **Save**.
7. Test your agent in the **Test agent** pane.

Chat

search accounts

Just now

Sounds like you're looking for an account. I can look that up for you.

Just now

What's the name of the customer you're looking for?

Just now

contoso

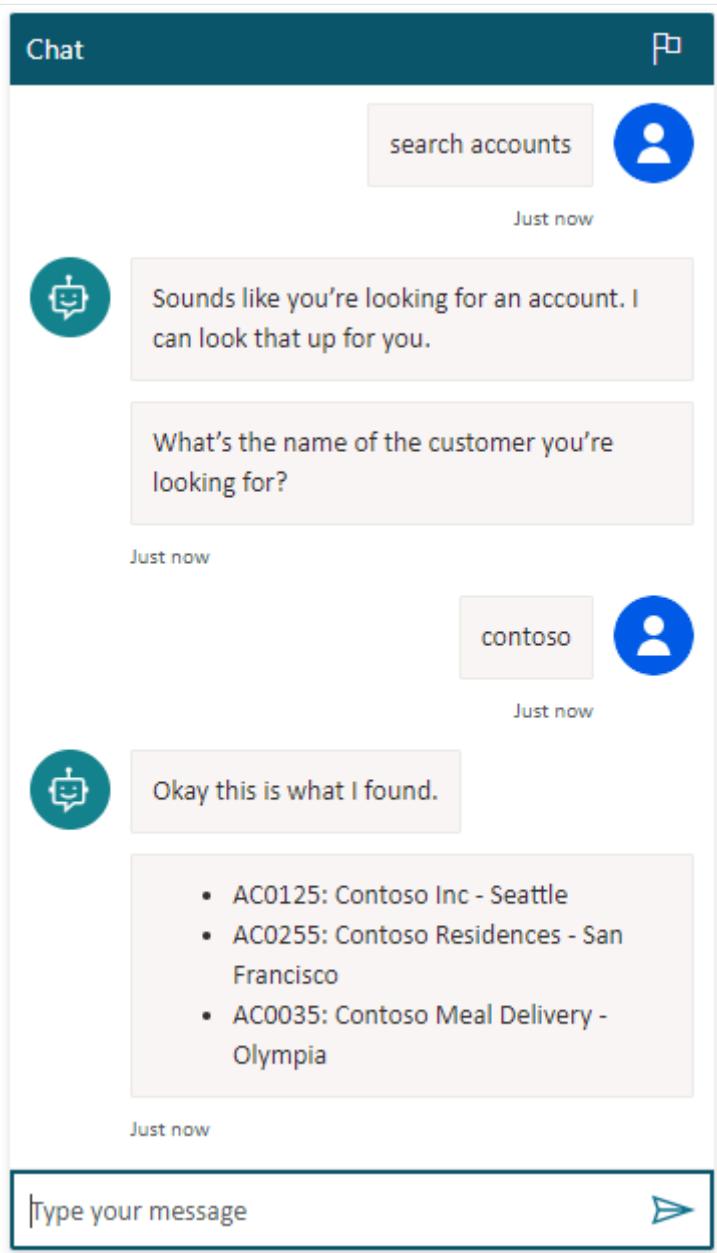
Just now

Okay this is what I found.

- AC0125: Contoso Inc - Seattle
- AC0255: Contoso Residences - San Francisco
- AC0035: Contoso Meal Delivery - Olympia

Just now

Type your message 



This screenshot shows a Microsoft Teams chat window. At the top, there's a teal header bar with the word 'Chat'. Below it is a search bar labeled 'search accounts' and a blue circular profile icon. The main area contains a series of messages. A message from the bot starts with 'Sounds like you're looking for an account. I can look that up for you.' followed by a question 'What's the name of the customer you're looking for?'. The user replies with 'contoso'. The bot then responds with 'Okay this is what I found.' and lists three results: 'AC0125: Contoso Inc - Seattle', 'AC0255: Contoso Residences - San Francisco', and 'AC0035: Contoso Meal Delivery - Olympia'. The entire conversation is timestamped as 'Just now'. At the bottom, there's a text input field with the placeholder 'Type your message' and a blue send arrow icon.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(7366eb85cf62ce28da681a7ca672d910_img.jpg\)](#)

Extend an agent with REST API actions (preview)

Article • 12/05/2024

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.

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 copilot experiences like Copilot for Microsoft 365.

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

API actions must be created from an OpenAPI v2 specification. This is due to the capabilities of the Power Platform in processing the API specifications. If a v3 specification is submitted, it is automatically translated to a v2 specification during the creation process.

Prerequisites

A few prerequisites are needed to extend Microsoft-branded copilot experiences with connector actions.

For all copilot 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

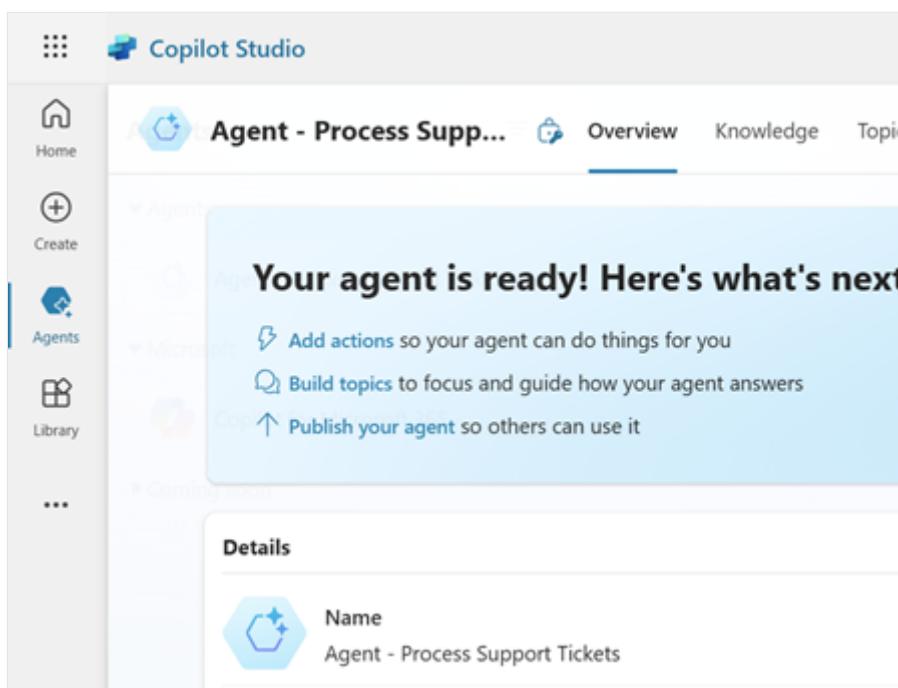
Adding a REST API to your agent involves a few steps. The following sections walk you through the process.

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 **Actions** section, select **Add Action**. You can also go to the **Actions** tab and select **Add Action**.

The screenshot shows the Copilot Studio interface with the 'Agent - Process Support' tab selected. On the left, there's a sidebar with icons for Home, Create, Agents, Library, and three dots. The main area has tabs for Overview, Knowledge, Topics, Actions, Activity, and Analytics. Under the Overview tab, there are sections for 'Provide ticket IDs in responses', 'Knowledge' (with a note to add data, files, and resources), and 'Actions' (with a note to add actions to empower the AI). A 'Create' button is visible on the left.

The action creation wizard is launched with the **Choose an action** page displayed.

Provide API specification, description, and solution

1. At the bottom of the page select **Add an API for a Custom Connector**.

The screenshot shows the 'Step 1 of 3: Choose an action' wizard. It lists several actions in a grid:

Get forecast for today MSN Weather	Get worksheets Excel Online (Business)
List rows present in a table Excel Online (Business)	Run a flow built with Power Automate for desktop Desktop flows
Run script Excel Online (Business)	Run script from SharePoint library Excel Online (Business)
Send an email notification (V3) Mail	Update a row Excel Online (Business)

Below the grid, there's a note: 'Can't find an action you want to add? Try adding a new action.' and three buttons: 'Create a new flow', 'Upload a new skill', and 'Add an API for a custom connector'. A 'Cancel' button is at the bottom right.

2. 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.

Add REST API plugin

X

- Upload specification
- API plugin details
- Authentication
- Select Actions
- Configure action
- Select action parameters
- Review
- Publish

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](#)

Drag and drop files here or click to browse

Max file size up to 5 MB

Next

Save and Close

① Note

The OpenAPI specification must be a JSON file in v2 format. If a v3 specification is submitted, it is 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](#)

File Name

swagger 3.json

Browse API specification

API specification

```
0. {
  1. "openapi": "3.0.0",
  2. "info": {
  3.   "title": "SunnyADO Ticket Manager Service",
```

Next

Save and Close

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.

3. 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' configuration page. At the top, there's a header 'API plugin details' and a sub-instruction 'Provide your plugins details'. Below this, there are two main sections: 'Action name *' and 'Description *'. The 'Action name' field contains 'SunnyADO Ticket Manager Servic'. The 'Description' field contains 'A simple service to manage tickets'. Underneath these fields is a 'Solution' section with a help icon and a note: 'If no solution is selected, one will automatically be created for you.' A dropdown menu labeled 'Select a solution' is shown. At the bottom of the page are three buttons: 'Back', 'Next' (which is highlighted in blue), and 'Save and Close'.

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 action. 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."

4. Enter an improved description under the **Description** field.

5. 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](#).

Solution ⓘ

If no solution is selected, one will automatically be created for you.

Select a solution

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.

ⓘ Note

If you do not 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](#).

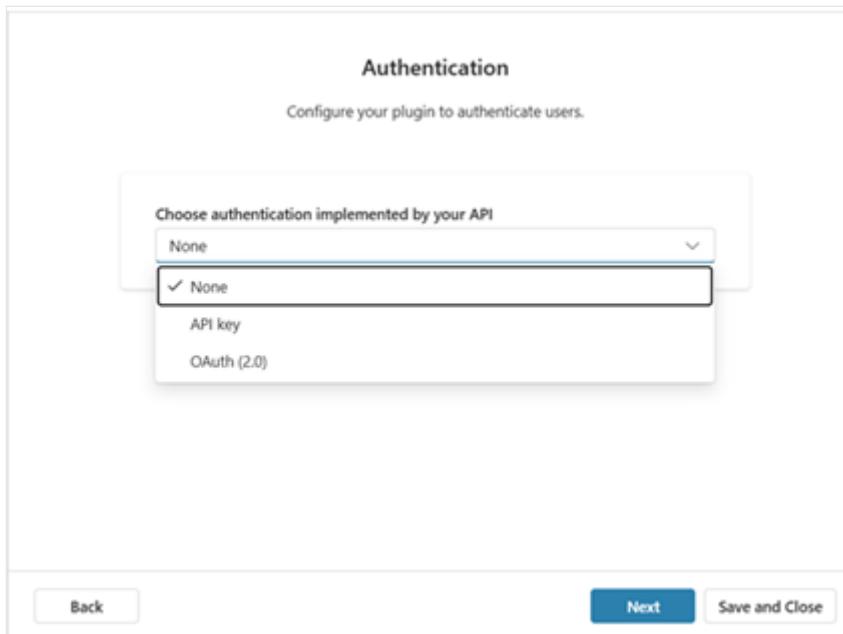
6. With a solution selected, select **Next** to proceed.

Provide authentication details

The **Authentication** page is displayed, to select which type of authentication to use for the API.

ⓘ Note

Currently the available options are None, Auth 2.0 and API. More methods will be available at a later date.



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: 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 will 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 action** page where you can select actions 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 actions for the API

Choose the API-supported actions to enable. Generally, a REST API offers a range of actions 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 an action 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.



List all tickets

Action name *

List all tickets

Action description * ⓘ

Returns a list of tickets with their details. Find, locate, and get tickets or records.

Back

Next

Save and Close

2. Configure the selected action. As with the overall API, you're asked to provide an **Action name** and **Action 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 action 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 action.

3. Once the fields are completed, select **Next**.

The **Review your action's parameters** page is displayed.

Review your action's parameters

Review your input and output values and update descriptions as needed

 List all tickets

Input name

text

Input description *

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

id

Output description *

The unique identifier of the ticket

title

The short summary of the ticket

[Back](#)

[Next](#)

[Save and Close](#)

The contents of this page display 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 action is now configured and appears in the list of **Selected actions** on the [Select and configure your plugin action](#) 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 actions you want your agent to support, select **Next**.

The **Review your action** page is displayed. The page provides the details of the configured REST API action.

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

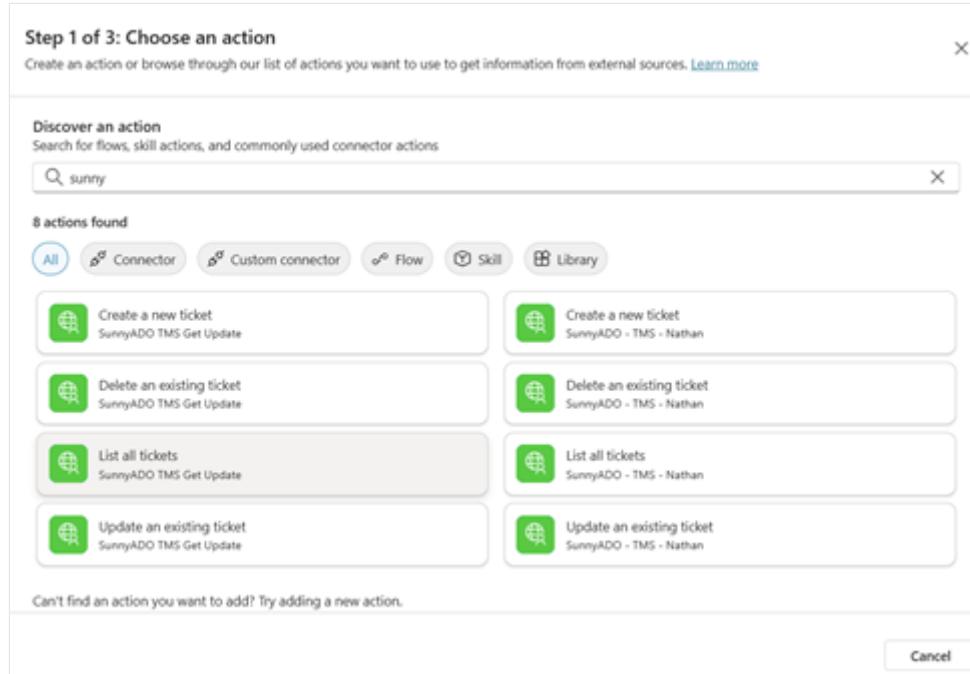
Save and Close

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 action 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 an Action** 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 action is now available for use in your agent.

💡 Tip

To more easily find your action, use the Search bar to locate it.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Use interactive voice response in your agents

Article • 11/19/2024

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

Feature	Tip or limitation
	365. The sequence is for channel reconnection.
Language/Locale	For a full list of supported languages and locales, see Language support . 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"> - 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. - If you use multiple consecutive action nodes for one topic and hit any unexpected results, add a message node between the consecutive action nodes.
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 authentication to <i>No authentication</i> to be able to publish on the Dynamics 365 Customer Service channel.
Customer engagement hub	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: <ul style="list-style-type: none"> - Genesys - Live person - Salesforce - ServiceNow
Generative AI for voice-enabled agents	Creating and editing topics with Copilot isn't supported for voice-enabled agents. Copilot doesn't create messages for Speech & DTMF, and doesn't configure DTMF mappings.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

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?

 Yes

 No

[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. <code><break time="5000ms"/></code> 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.

```
yml
```

```
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

Article • 11/19/2024

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.
- There's no change to the existing agent flow, 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 won't have the *Main Menu (preview)* topic. You must recreate that topic, if needed.

Disable optimization for voice

You can disable **Optimize for voice** in agent authoring if you don't enable the **Telephony** channel. After you disable **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 the disabled DTMF topic is referenced in other topics.

- No change to your agent flow and channel setting, since disabling optimization doesn't turn off the **Telephony** channel.
- Enabling or disabling the *optimize for voice* option doesn't take effect until you publish your agent. If the agent is enabled or disabled accidentally and switches between modalities, you have time to fix it.

ⓘ Important

If your **Telephony** channels are enabled, disabling **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:

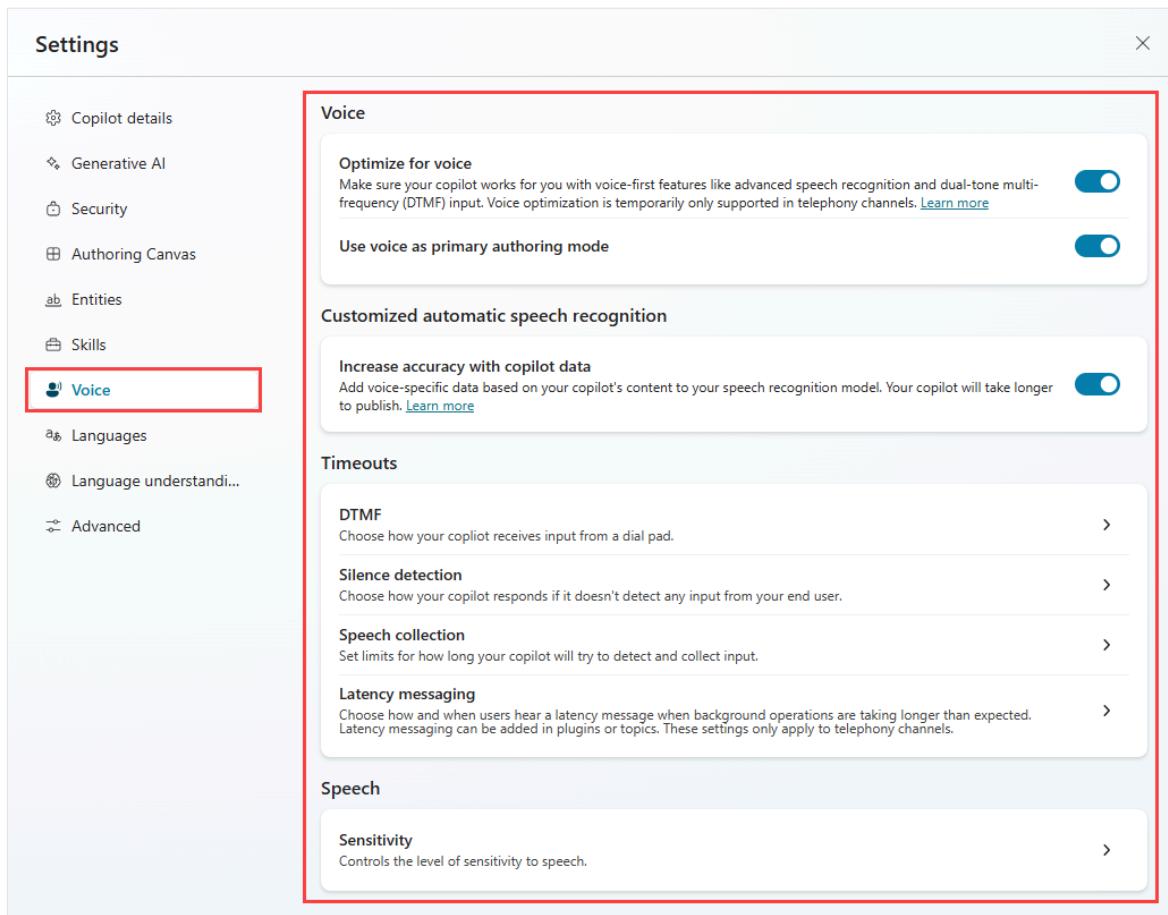
1. With your agent open, select **Settings > Voice**.
2. Select **Increase accuracy with agent data** to enable the agent's default customized automatic speech recognition settings.
3. Select **Save** to commit your changes.
4. Publish your agent to see the new changes.

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**.



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 Multi-digit DTMF input
DTMF	Termination timeout	Maximum duration (milliseconds) to wait for a DTMF termination key. Limit applies when user reaches maximum input length and didn't	2000 ms	Question node with voice properties for

Voice-enabled agent-level section	Setting	Description	Default value	Node-level override
		press termination key. Applies only to multi-digit DTMF input.		Multi-digit DTMF input
		After the limit times out and terminating DTMF key doesn't arrive, agent ends the recognition and returns the result up to that point.		
		If set to "continue without waiting," the agent doesn't wait for termination key. Agent returns immediately after user inputs the max length.		
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	<p>Question node with voice properties for Multi-digit DTMF input</p> <p>System topic (silence detection trigger properties) for Configure silence detection and timeouts</p>
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 milliseconds. Anything above 3000 ms reduces to 3000 milliseconds.</p>	1500 ms	Question node with voice properties
Speech collection	Speech recognition timeout	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	12,000 ms	Question node with voice properties

Voice-enabled agent-level section	Setting	Description	Default value	Node-level override
		reprompts the question. If no response, the voice is beyond <i>Speech recognition timeout</i> .		
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 long-running operation
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 long-running operation
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

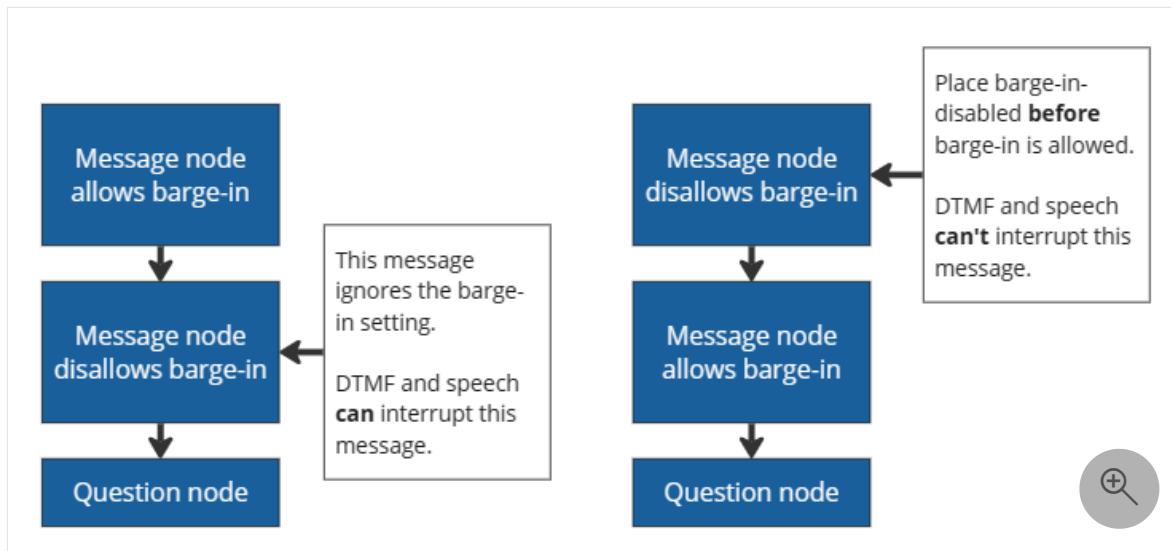
Enabling barge-in allows your agent users to interrupt your agent. This feature can be useful when you don't need the agent user to hear the entire message. For example, callers might already know the menu options, because they heard them in the past. With barge-in, the agent user can enter the option 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

1. With a **Message** or **Question** node selected, set the desired modality to **Speech & DTMF**.
2. Select the **More** icon (...) of the node, and then select **Properties**.

a. 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 to commit your changes.

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 global setting 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.

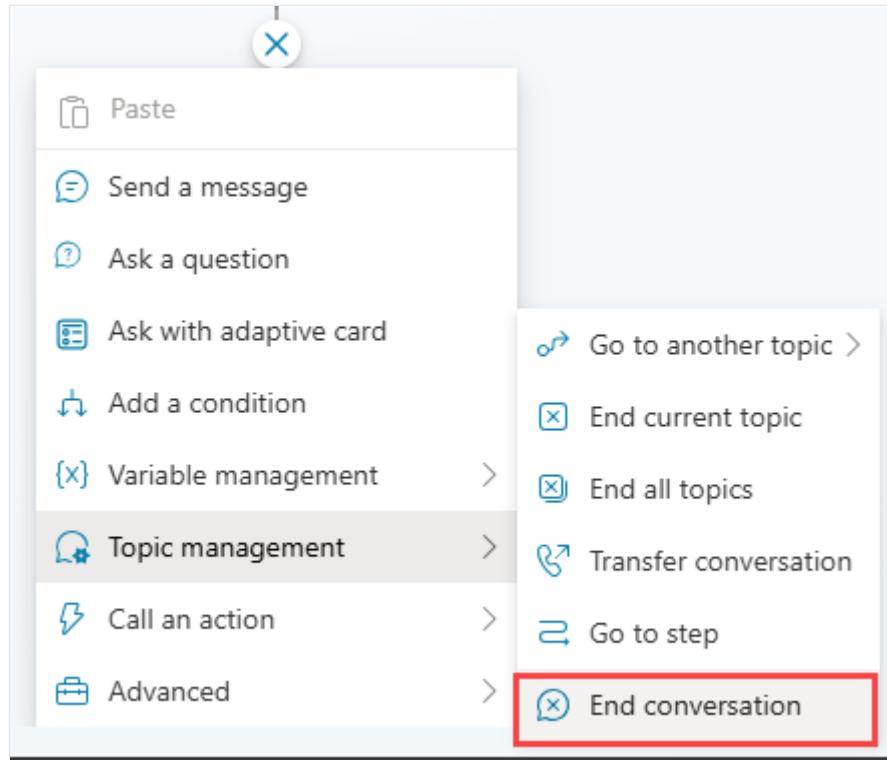
In Copilot Studio, your agent can repeat a message after triggering a Power Automate flow:

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

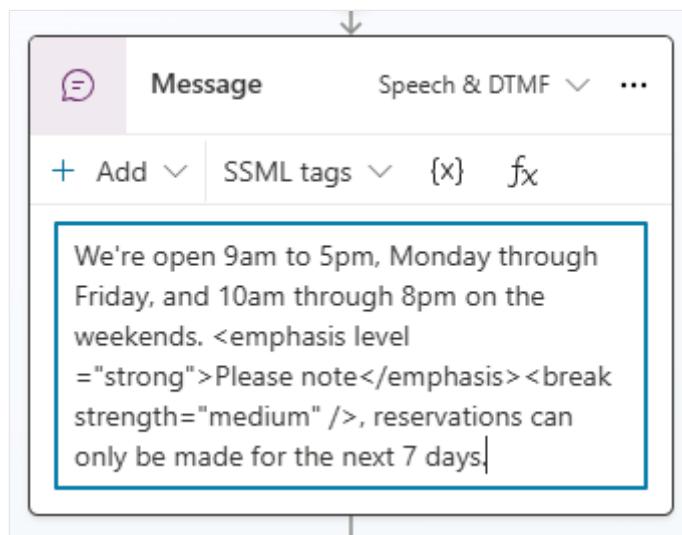
You can use speech synthesis markup language (SSML) to change how the agent sounds when it reads messages out loud. For example, you can change the pitch or frequency of the spoken words, the speed, and the volume.

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><audio src="_URL_to_an_audio_file_"/></code>	Add the URL to an audio file within the tag. The file must be accessible by the agent user.	Add recorded audio
<code><break /></code>	Insert pauses or breaks between words. Insert break options within the tag.	Add a break

SSML tag	Description	Link to speech service documentation
<emphasis> Text you want to modify </emphasis>	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.	Adjust emphasis options
<prosody> Text you want to modify </prosody>	Specify changes to pitch, contour, range, rate, and volume. Add prosody options in the opening tag. Add the closing tag after the text you want to modify.	Adjust prosody options
<lang xml:lang="xx-xx"> Text you want to modify </lang>	Adjust the speaking language within the same message when using a multilingual neural voice.	Adjust speaking languages



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 additional 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
<code>System.Activity.From.Name</code>	String	The agent user's caller ID
<code>System.Activity.Recipient.Name</code>	String	The number used to call or connect to the agent
<code>System.Conversation.SipUuiHeaderValue</code>	String	SIP header value when transferring through a direct routing phone number

Variable	Type	Description
<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 true
<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 natural language understanding model

① 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](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

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.

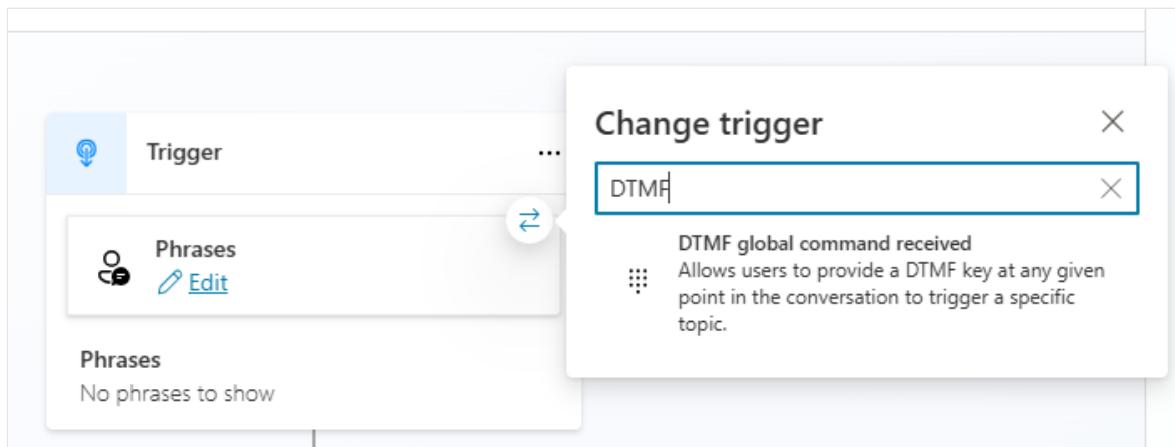
If the key input by the agent's user isn't mapped to the DTMF global triggers, the agent triggers the System topic, *Unknown dialpad press trigger (voice)*. You can customize it for your business logic.

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.

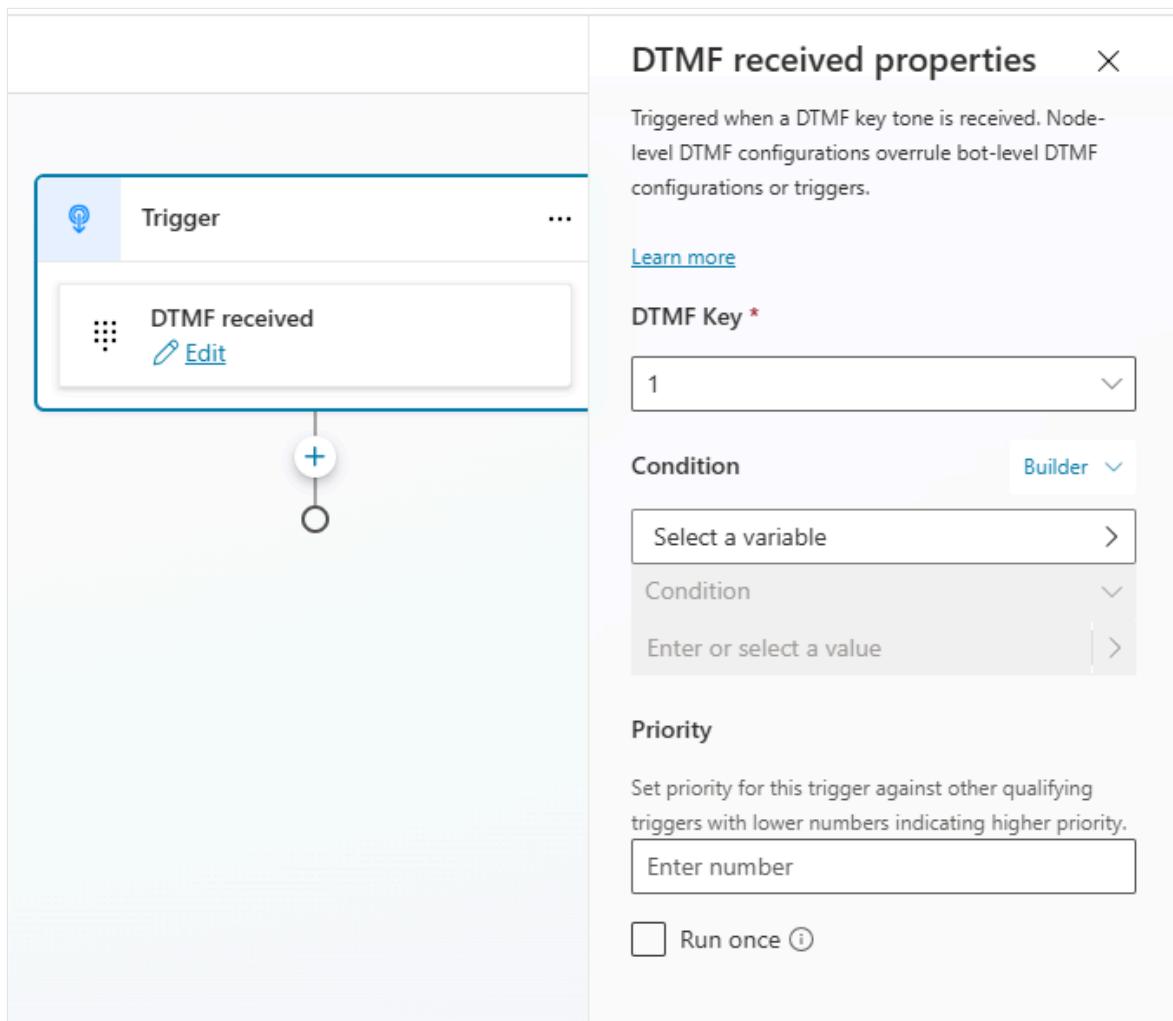
Name	Trigger	Description	Editing
Conversation Start	On Conversation Start	This system topic triggers when the b...	
End of Conversation	On Redirect	This system topic is only triggered by ...	
Escalate	On Talk to Agent	This system topic is triggered when t...	
Fallback	On Unknown Intent	This system topic 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 2 3 4	Can accept multi-digit DTMF input
Age	1 2 3 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 2 3 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 2 3 4	Can accept multi-digit DTMF input
Number	1 2 3 4	Can accept multi-digit DTMF input
Ordinal	1 2 3 4	Can accept multi-digit DTMF input
Organization	✗	DTMF not supported
Percentage	1 2 3 4	Can accept multi-digit DTMF input
Person name	✗	DTMF not supported
Phone number	1 2 3 4	Can accept multi-digit DTMF input

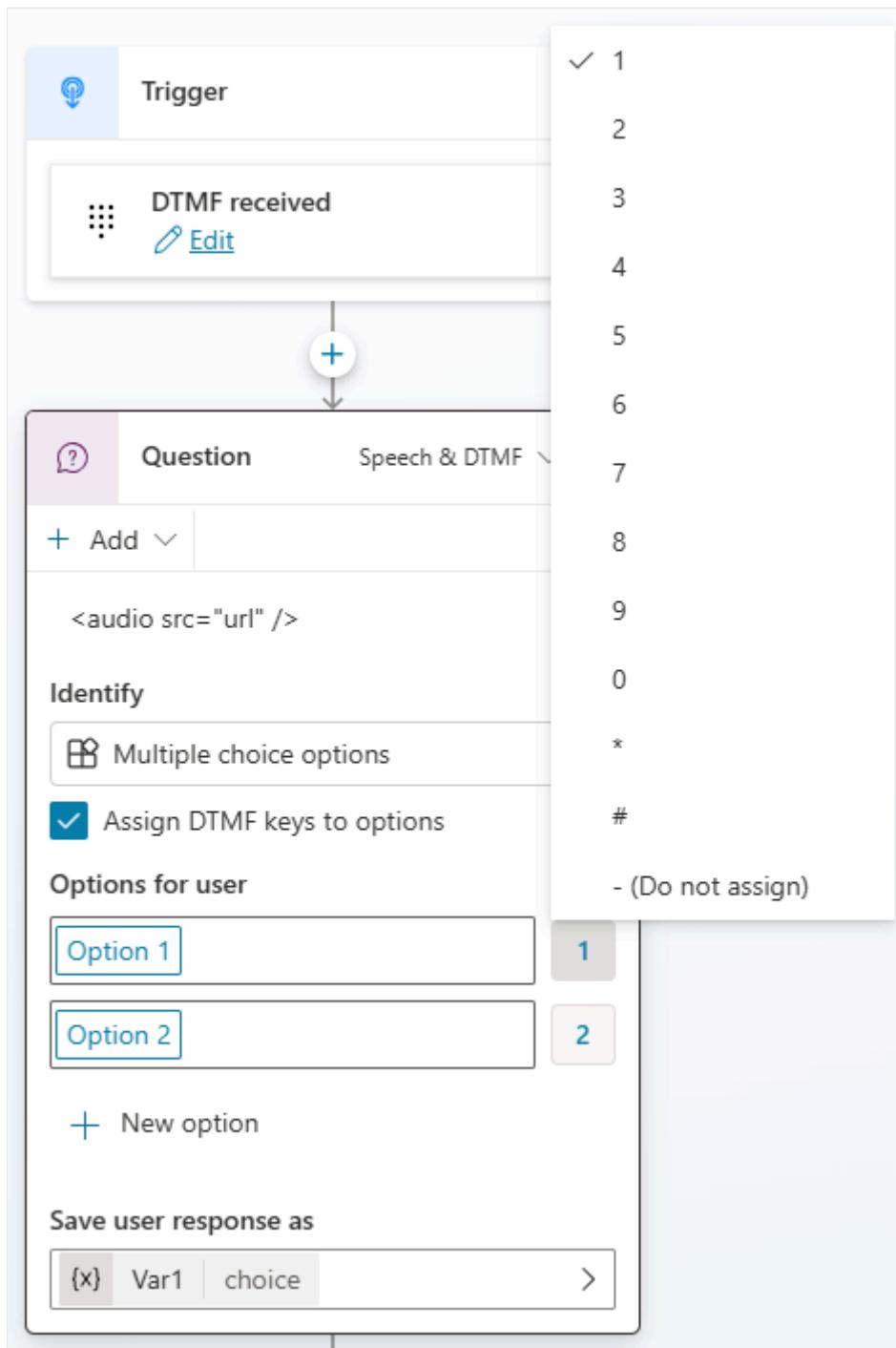
Entity (under Identify)	DTMF support	Details
Point of interest	✗	DTMF not supported
Speed		Can accept multi-digit DTMF input
State	✗	DTMF not supported
Street address	✗	DTMF not supported
Temperature		Can accept multi-digit DTMF input
URL	✗	DTMF not supported
Weight		Can accept multi-digit DTMF input
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.

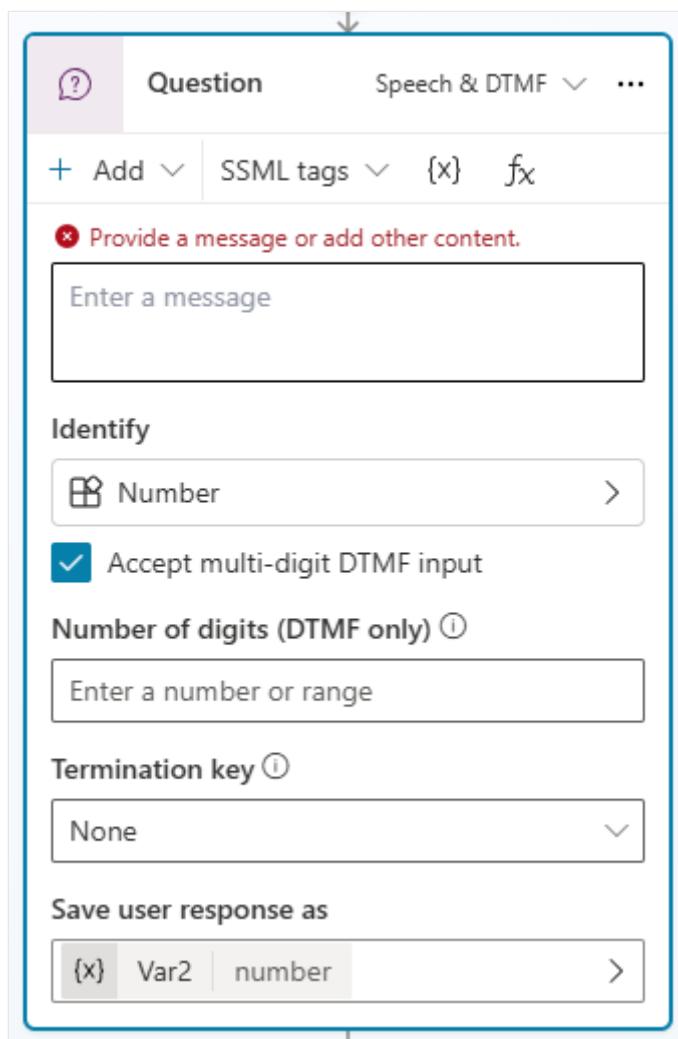
The agent assigns keys in the order that the options are in, from 0 to 9.

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

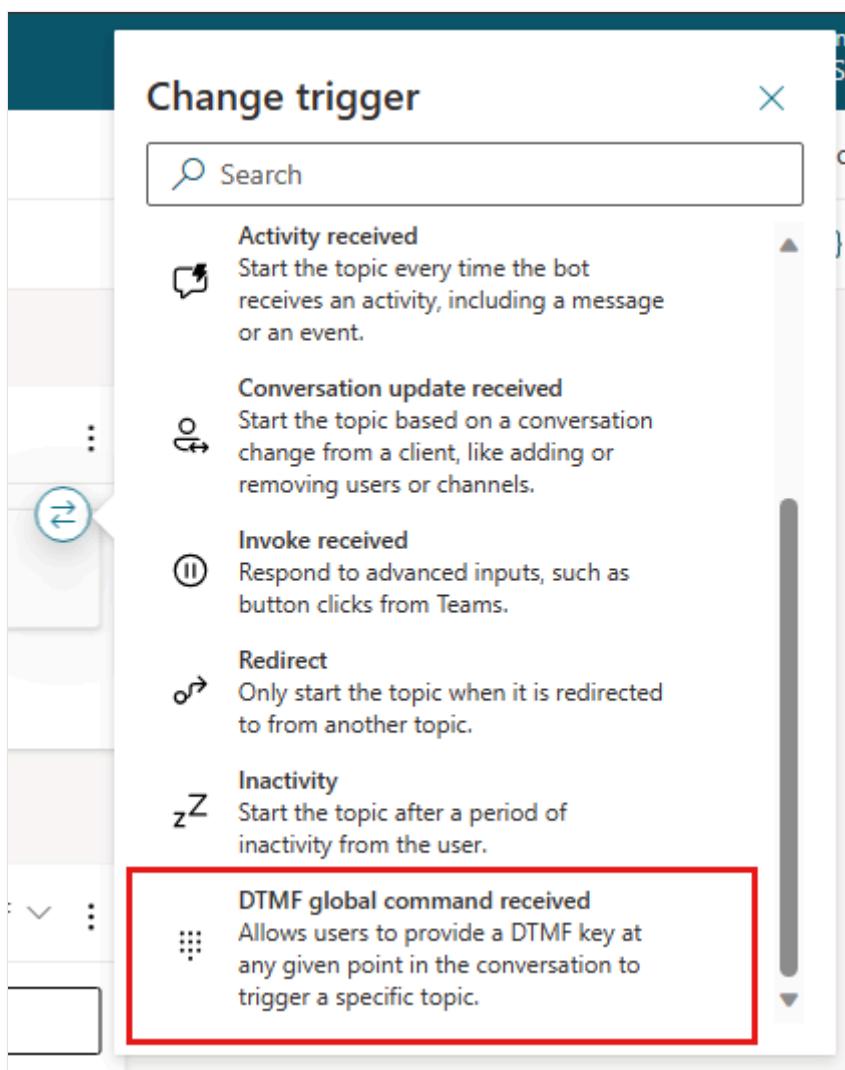
You can configure what key should be pressed for each option that the user can input and how long the agent should wait before parsing the user's input.

ⓘ 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

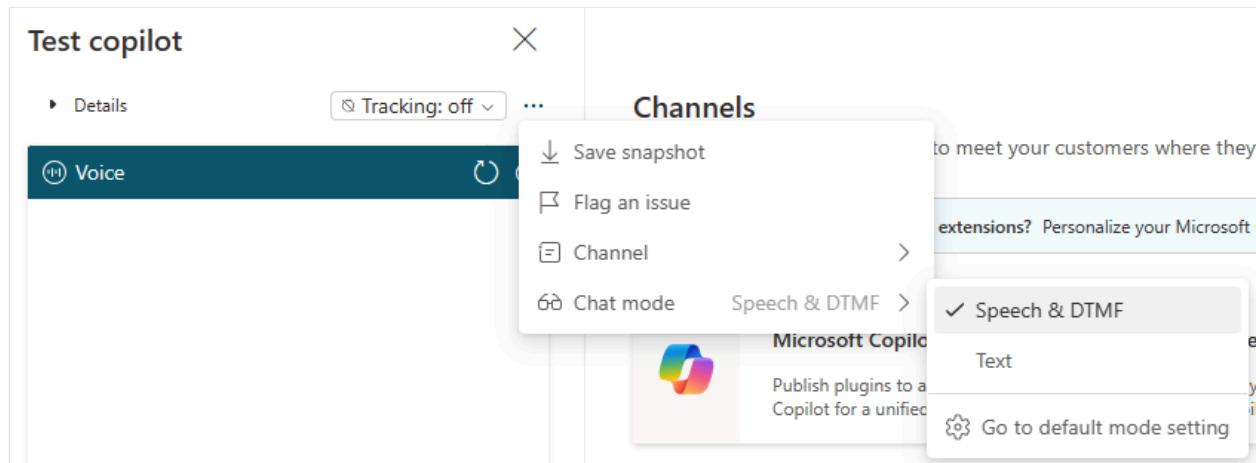
[Provide product feedback ↗](#)

Test your voice-enabled agent

Article • 11/19/2024

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)	- If the customer consents, the call is automatically transferred to the post-call survey agent at the end of the personal digital assistant call. - If set to false, the call ends when the agent hangs up, which requires you to install the OC extension solutions.

- 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*.

The screenshot shows the Twilio Studio interface. A 'Trigger' block (DTMF received) is connected to a 'Question' block. The 'Question' block is set to 'Speech & DTMF'. Under 'Identify', 'Multiple choice options' is selected. There is a checked checkbox for 'Assign DTMF keys to options'. Under 'Options for user', there is a '+ New option' button. Under 'Save user response as', the path '{x} Var2 choice' is shown.

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*.

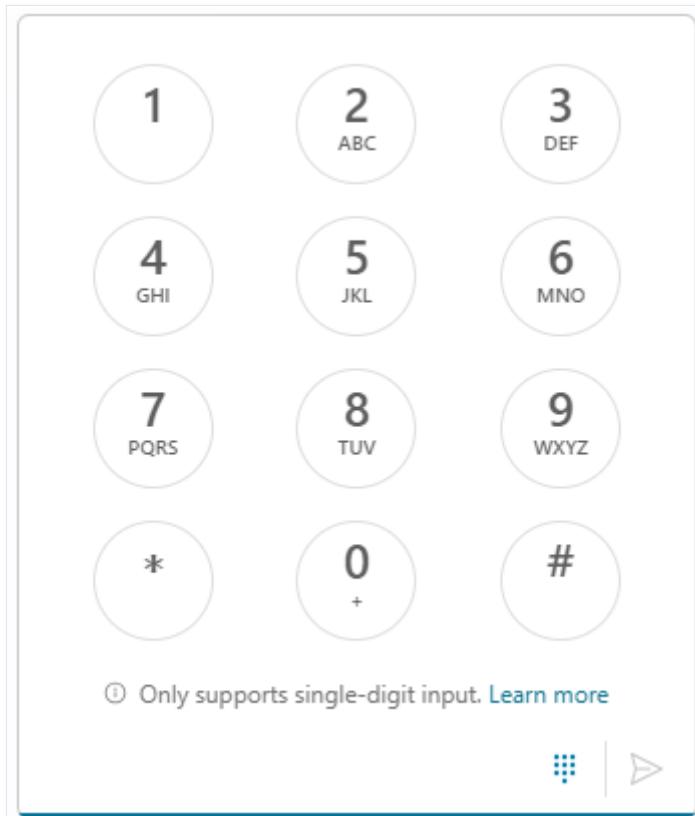
- For messages with barge-in disabled, you see **Barge in is not allowed**.

The screenshot shows a Twilio message preview. It features a speech bubble with the text: "Hello. Thank you for calling Please listen closely to the following options.". Below the message, a status message says: "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.

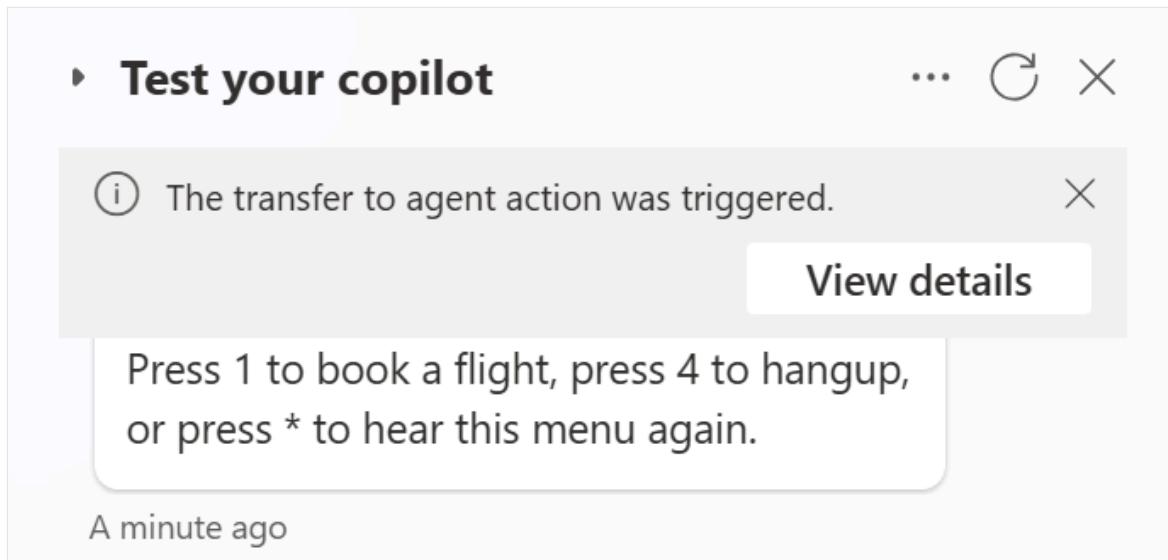
- For Multi DTMF, you can only use the command "/DTMFkey **DTMF keys**." For example, /DTMFkey 123, to simulate batched, multi-digit DTMF.

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](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Use prompt modification to provide custom instructions to your agent

Article • 11/19/2024

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

[+ Add knowledge](#)

> Classic data

Allow the AI to use its own general knowledge (preview)

Content moderation ⓘ

High

>

{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 about what you seek,

Custom Instruction	User question	Agent response
		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>Step 1: Open the email application on your device.</p> <p>Step 2: Select the option to add a new account.</p> <p>Step 3: Enter your email address and password.</p> <p>Step 4: 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.

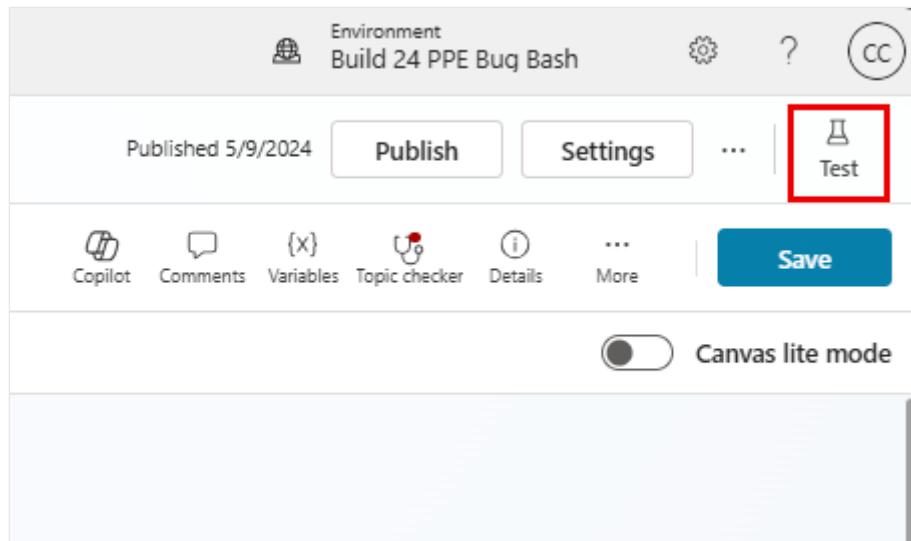
Capabilities

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.

Test your new prompt

Be sure to test your new prompt.



In the **Test your agent** panel, ask the agent questions that take advantage of the new prompt instructions.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Configure and create multilingual agents

Article • 12/05/2024

Multilingual agents are agents that can communicate with customers in different languages while keeping all the content in a single agent. In many cases, they can 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.

For the list of supported languages, see [Language support](#).

Note

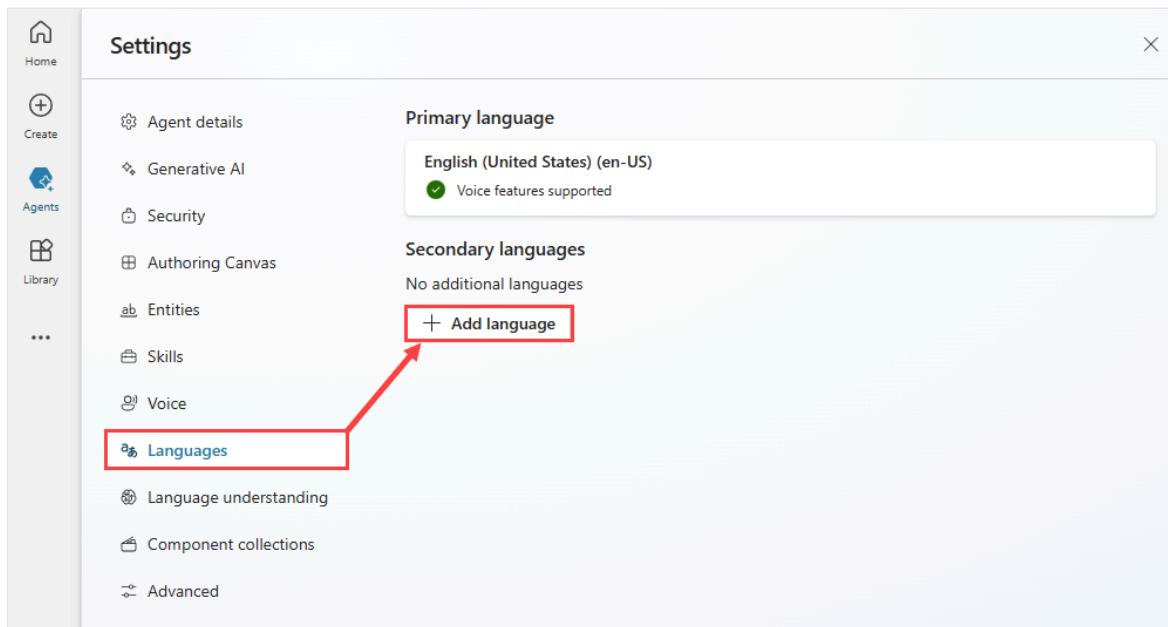
- Copilot Studio classic chatbots don't support multilingual agents. For more information about converting a classic chatbot to an agent, see [Upgrade to Copilot Studio unified authoring](#).
- When published to a channel, if no language is specified at session start, the multilingual agent defaults to the primary language. You can configure the agent to change the language during a conversation, see [Authoring considerations for multilingual agents](#).

Configuring a multilingual agent

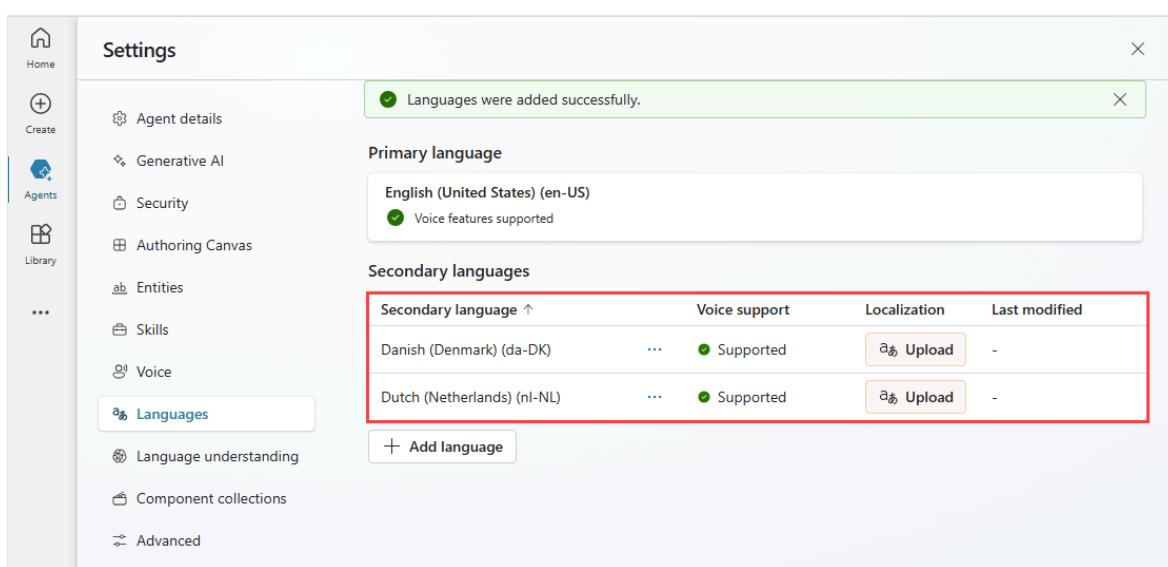
When an agent is initially created, a primary language is defined. For more information on how to select the agent's primary language, see [Language support](#).

To enable a multilingual agent:

1. With an agent open, select **Settings** at the top of the page, and then select **Languages**.
2. Select **Add Languages** on the **Settings** page.



3. Choose the languages you want to add to the agent in the **Add Languages** pane and select **Add Languages** at the bottom.
4. Confirm the list of languages is correct on the **Languages** page.



ⓘ Note

You can add as many secondary languages as desired to the agent, as long as they're in the list of supported languages.

Managing localization in a multilingual agent

ⓘ Note

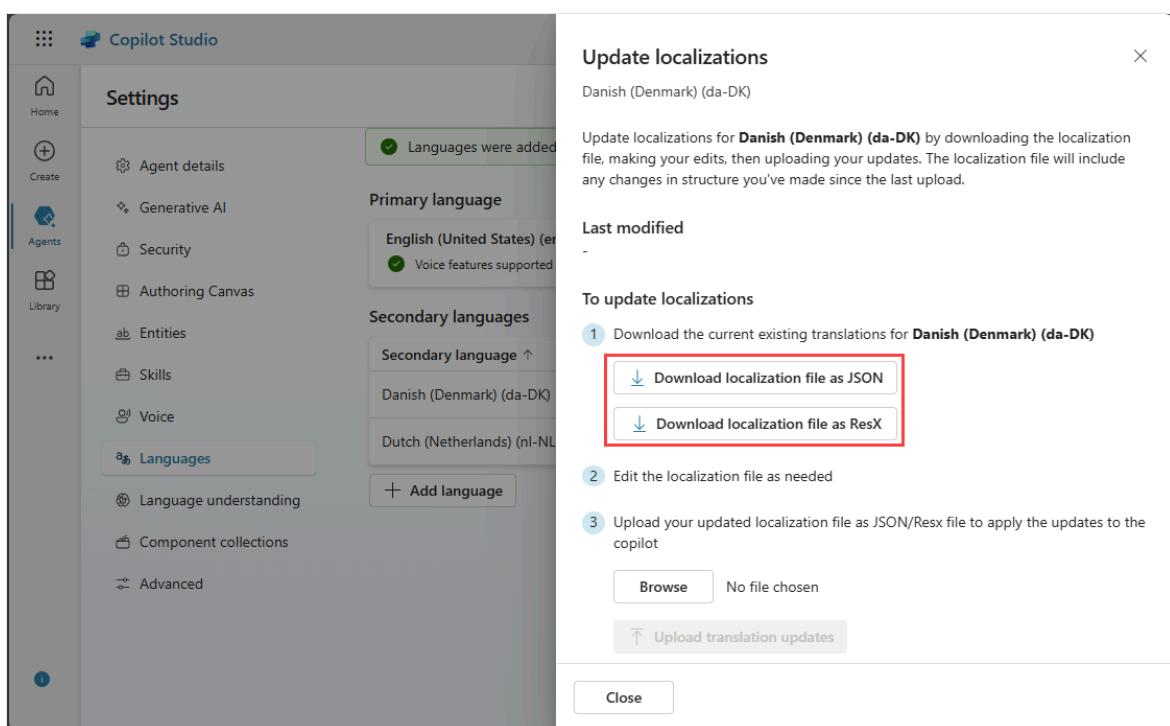
All agent topic and content editing must be done in the agent's primary language.

To edit the agent's secondary language to localize the strings, you must use the steps in this section.

Once the agent strings are localized and uploaded to the agent as described here, you can see the localized content in the authoring canvas by [switching the language in the test agent](#).

To add the localization strings to your multilingual agent:

1. With an agent open, select **Settings** at the top of the page, and then select **Languages**.
2. On the **Languages** page, in the table of added secondary languages, select **Upload** in the **Localization** column for the language you want to update.
3. On the **Update localizations** page for the selected secondary language, 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](#) to download previous versions.

4. Open the downloaded file and replace the primary language strings with the appropriate translated text.
5. Return to the **Update localizations** page, select **Browse**, and upload the translated file.
6. Close the **Update localizations** page.

Updating localized content

When you first download the localization file for a secondary language, the strings are in the primary language. After you download a localization file, use it with your preferred localization process.

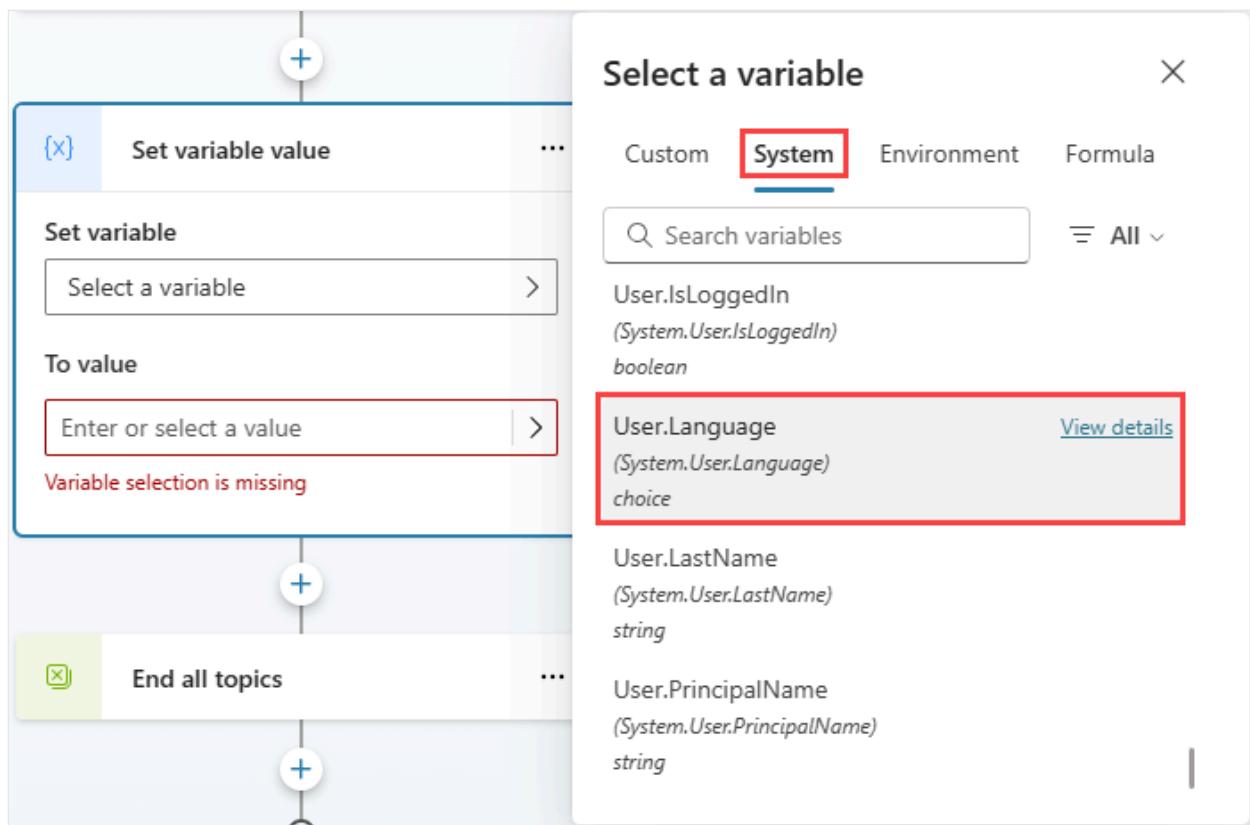
If you make changes to the primary language strings, you need to also update the localized content in the secondary language. This process includes both new content and modified content. Incremental changes aren't automatically translated. You must download the secondary language JSON or ResX file, and 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 secondary language localization file, 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 is using the same ID, you must compare the new file against the previously uploaded file.

Authoring considerations for multilingual agents

When authoring, you can configure the agent to change the current spoken language in the middle of a conversation. That 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.

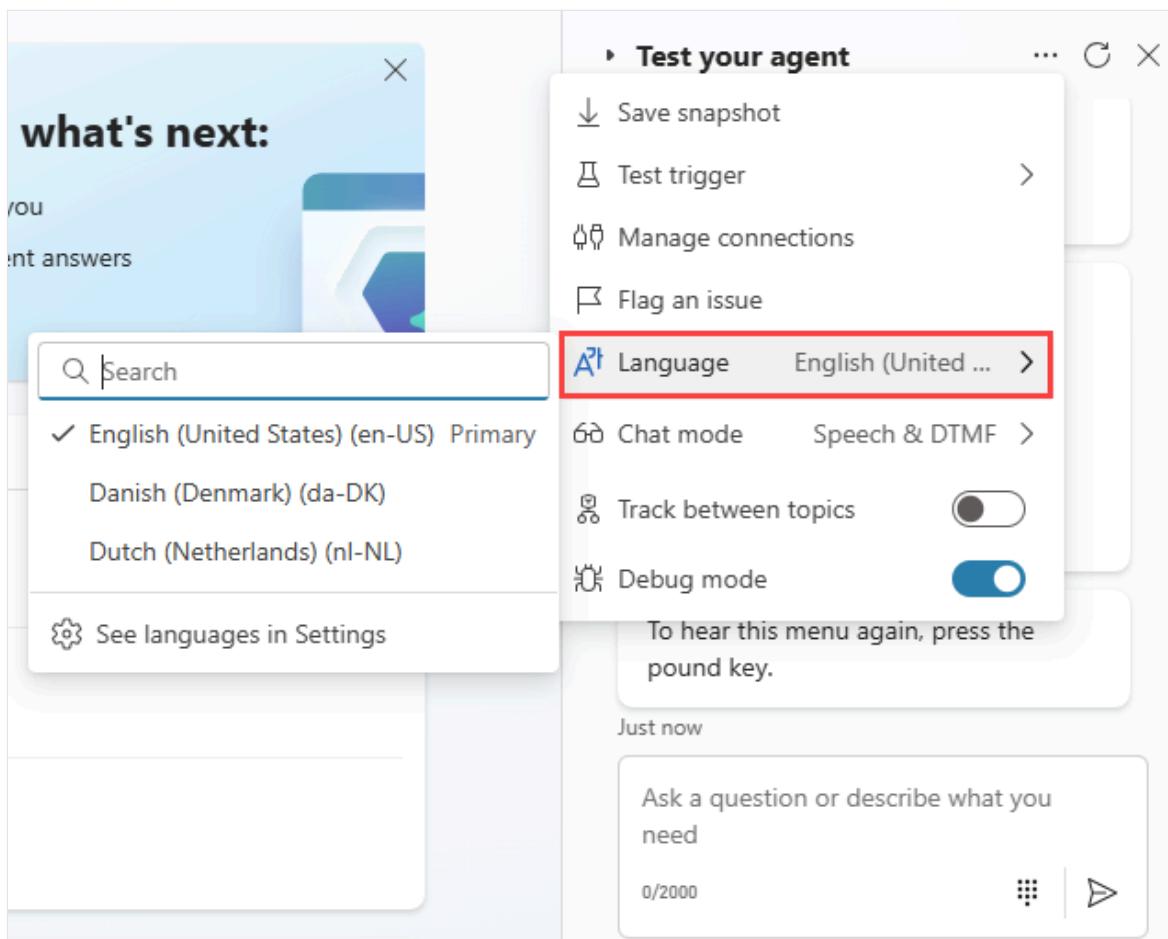
To change the agent's current language, you can set the `User.Language` variable value to one of your agent's secondary languages. This selection changes the language spoken by your agent immediately.



Testing a multilingual agent

To test a multilingual agent:

1. Open the **Test your agent** pane.
2. Open the menu at the top of the test agent, and select one of the secondary languages.



3. The test agent reloads itself, this time using the selected secondary language. The authoring canvas shows the secondary language.
4. To test the agent, enter a message in the secondary language.

You can also set your browser language to the secondary language supported by your agent, and open the [Demo agent website](#). The demo website opens in the secondary language, and the agent chats using the secondary language strings.

Multilingual agent behavior for languages that aren't configured

If the user of an agent has their browser configured for a language the agent wasn't configured for, the agent always falls back to its primary language.

i Important

The agent's primary language can't be changed after agent creation.

Multilingual agent behavior for missing translations

If the agent author makes a change to the agent in the primary language and doesn't [upload the translations](#), the agent shows the untranslated changes in its primary language. Always make sure the translations are up-to-date after making changes to the agent in the primary language.

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 secondary localization file contains either duplicate synonyms or a synonym that matches a `DisplayName` value. This error is typically encountered when a node contains an `Entity.Definition.'closedListItems'` where either of the following scenarios occurred:

- One of the `Synonyms` elements is *not* 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 your secondary language's JSON or ResX file, and identify any instances where this condition might be present.

Related content

- [Language support](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

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

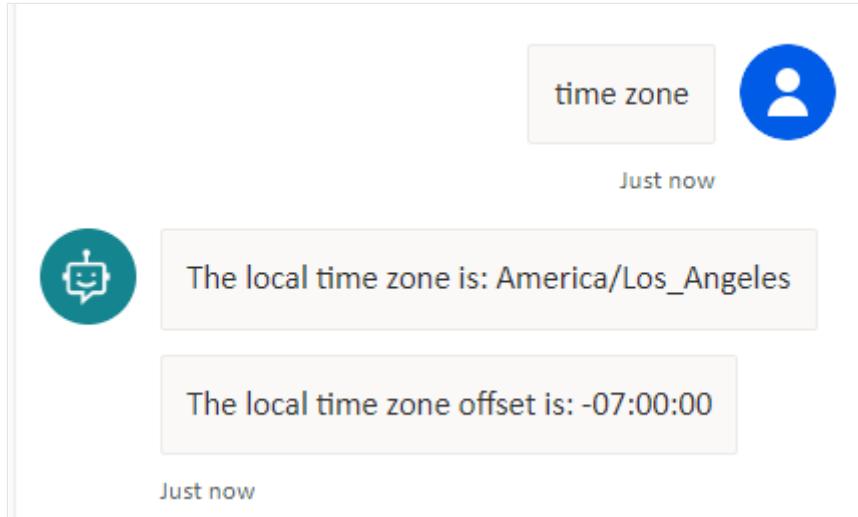
Use these system-level variables to get information about the user's time zone:

- `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.



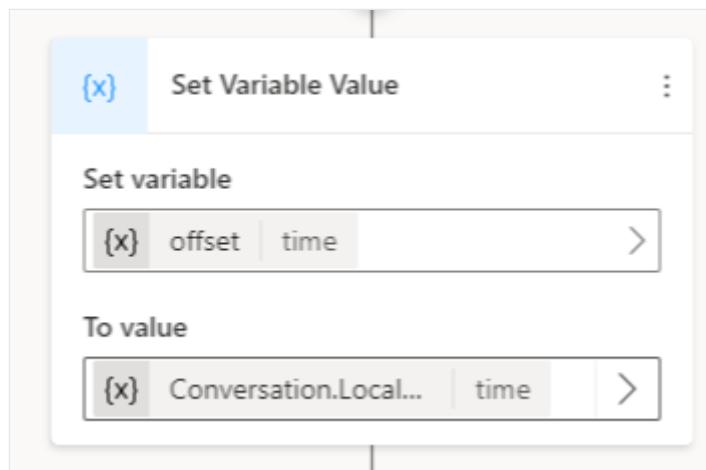
Display the date and time in the local time zone

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.

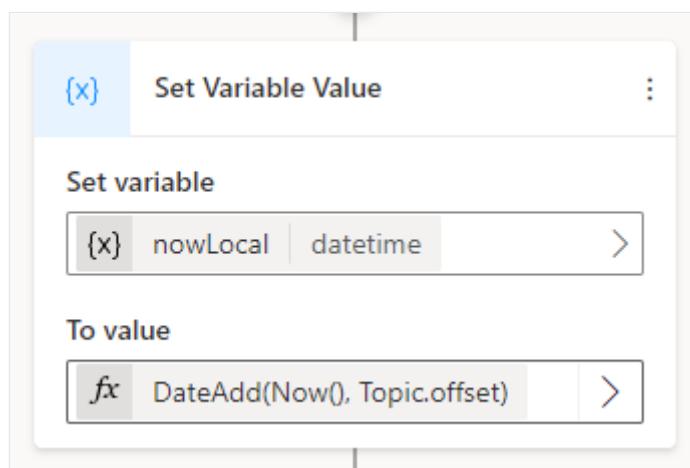
In this example, we get the current date and time using the Power Fx `Now()` function, and then add the time zone offset. It isn't possible to use the

`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`.

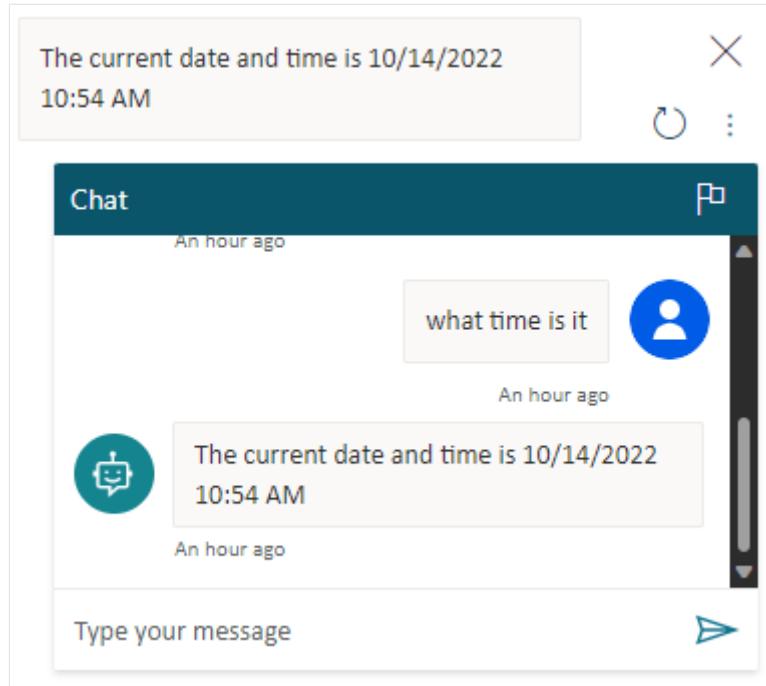
1. Create a topic and add the trigger phrase "what time is it".
2. In the authoring canvas, select **Add node (+)**. Select **Set a variable value**.
 - For **Set variable**, create a variable named `offset`.
 - For **To value**, select the system variable `Conversation.LocalTimeZoneOffset`.



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

Article • 11/19/2024

You can configure your agent to start a conversation 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 might impact 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 default system greeting topic. You can, however, create a new topic to be used as the greeting. You need to divert the default system greeting topic to a new topic.

Important

You may install and use the sample code included in this documentation only for use with the Copilot Studio product. 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

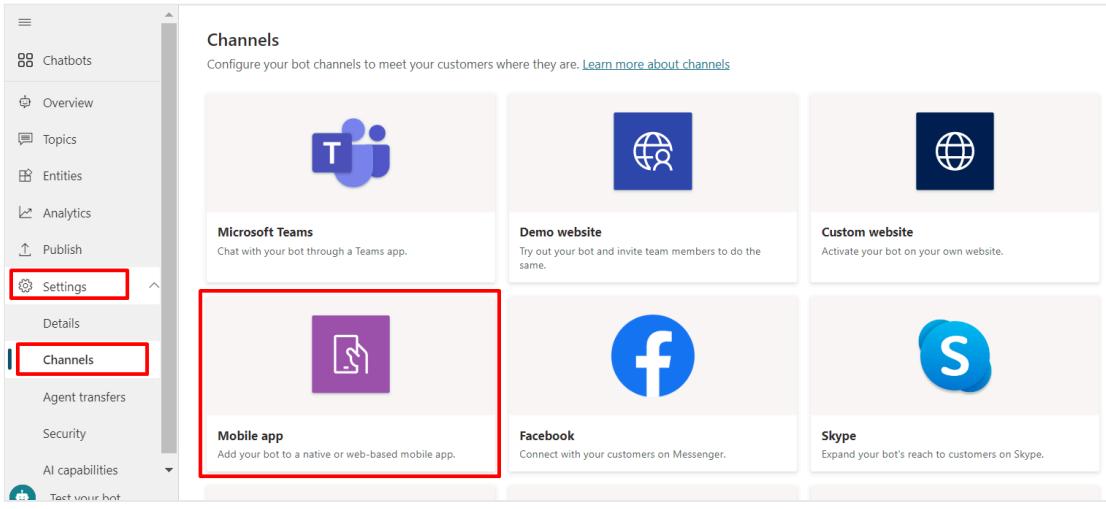
purpose, and non-infringement.

Retrieve token endpoint

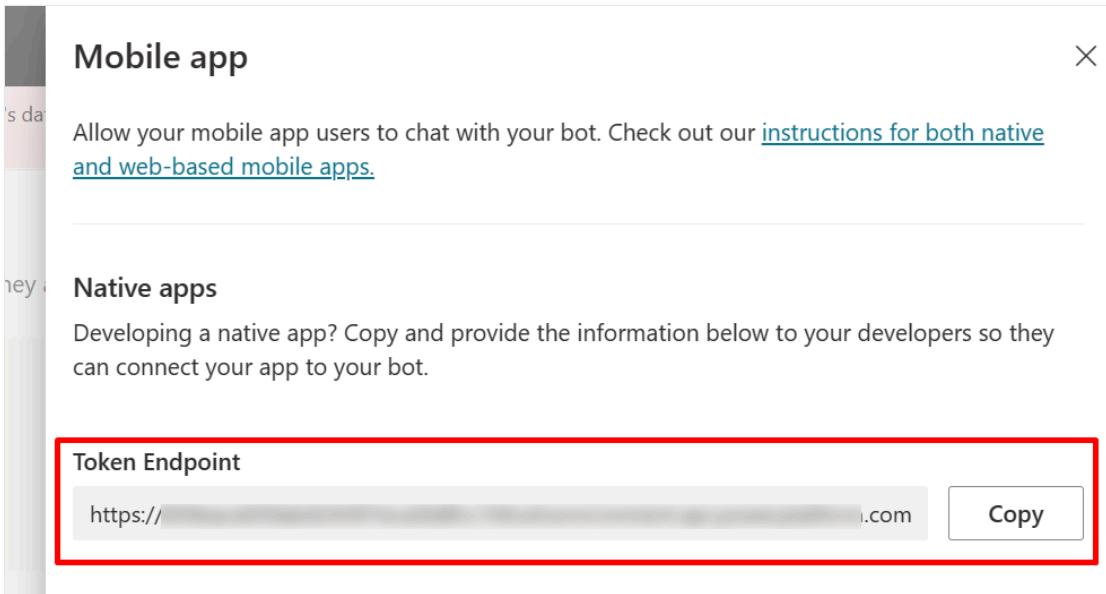
To customize your canvas, whether it's the default canvas or a custom one you connect to, you need to retrieve your agent details.

1. In the navigation menu under **Settings**, select **Channels**.

2. Select **Mobile app**.



3. Next to **Token Endpoint**, select **Copy**.



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 bot canvas.

1. [Create and publish an agent.](#)

2. Copy and paste the HTML code shown here and save it as *index.html*.

You can also copy and paste the code into the [w3schools.com HTML try it editor](#). You still need to add your token endpoint.

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Contoso Sample Web Chat</title>
    <!-- This styling is for the Web Chat demonstration purposes.
    It is recommended that style is moved to a separate file for
    organization in larger projects -->
    <style>
        html, body {
            height: 100%;
        }

        body {
            margin: 0;
        }

        h1 {
            font-size: 16px;
            font-family: Segoe UI;
            line-height: 20px;
            color: whitesmoke;
            display: table-cell;
            padding: 13px 0px 0px 20px;
        }

        #heading {
            background-color: black;
            height: 50px;
        }

        .main {
            margin: 18px;
            border-radius: 4px;
        }

        div[role="form"]{
            background-color: black;
        }

        #webchat {
            position: fixed;
            height: calc(100% - 50px);
            width: 100%;
```

```

        top: 50px;
        overflow: hidden;
    }

</style>

</head>
<body>
    <div>
        <div id="heading">
            <!-- Change the h1 text to change the bot name -->
            <h1>Contoso Bot Name</h1>
        </div>
        <div id="webchat" role="main"></div>
    </div>

    <script src="https://cdn.botframework.com/botframework-
    webchat/latest/webchat.js"></script>

    <script>
        const styleOptions = {

            // Add styleOptions to customize Web Chat canvas
            hideUploadButton: true
        };

        // Add your BOT token endpoint below
        var theURL = "<BOT TOKEN ENDPOINT>";

        var environmentEndPoint =
theURL.slice(0,theURL.indexOf('/powervirtualagents'));
        var apiVersion = theURL.slice(theURL.indexOf('api-
version')).split('=')[1];
        var regionalChannelSettingsURL =
`${environmentEndPoint}/powervirtualagents/regionalchannelsettings?
api-version=${apiVersion}`;

        var directline;
        fetch(regionalChannelSettingsURL)
            .then((response) => {
                return response.json();
            })
            .then((data) => {
                directline = data.channelUrlsById.directline;
            })
            .catch(err => console.error("An error occurred: " +
err));

        fetch(theURL)
            .then(response => response.json())
            .then(conversationInfo => {
                window.WebChat.renderWebChat(
                    {
                        directLine:
                window.WebChat.createDirectLine({

```

```

        domain: `${directline}v3/directline`,
        token: conversationInfo.token,
      }),
      styleOptions
    },
    document.getElementById('webchat')
  );
}
.catch(err => console.error("An error occurred: " +
err));

</script>
</body>
</html>

```

3. In the *index.html* file you created, enter your token endpoint at the line `var theURL = "<YOUR TOKEN ENDPOINT>";`.
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 are receiving responses from your agent and that the agent is working correctly.

If you encounter problems, make sure you've published your agent, and that your token endpoint is inserted in the correct place. The token endpoint should be after the equals sign (=) at the line `var theURL = "<YOUR TOKEN ENDPOINT>"`, and surrounded by double quotation marks (").

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 system greeting topic. You can also create a new topic and divert the default system 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 or create a new greeting topic, you should include some sort of indication that the user is talking to an agent (or "virtual agent"). Such an indication will help the user understand they are not talking to a human.

We recommend you modify the system greeting topic so that you don't have to edit the *index.html* code.

Modify the system greeting topic (recommended)

1. In the navigation menu, select **Topics**, then select the **Greeting** topic row.

The screenshot shows the 'Topics' section of the Microsoft Bot Framework interface. On the left is a navigation sidebar with options like Chatbots, Overview, Topics (which is selected and highlighted with a red box), Entities, Analytics, Publish, and Settings. The main area is titled 'Topics' with a count of 'Existing (12)'. It lists various topics with their names and icons. The 'Greeting' topic is highlighted with a red box. The table structure is as follows:

Type	Name
⚙️	Confirmed Failure
⚙️	Confirmed Success
⚙️	End of Conversation
⚙️	Escalate
⚙️	Goodbye
⚙️	Greeting
💬	Lesson 1 - A simple topic

2. Edit the text inside the **Message** nodes. You can also [add or delete additional 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 user topic

⚠️ Warning

Using a user topic to start a conversation will increase 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 user topic is triggered. For more information, see [View billed sessions \(preview\)](#).

1. In the navigation menu, select **Topics**.
2. Select **New topic**, and give it a name.
3. Add the text inside the **Message** node.
4. Select **Save** when you've finished editing the message.
5. In the navigation menu, select **Topics**, then select the **Greeting** topic row.
6. [Delete the message nodes](#) on the **Greeting** topic.
7. To automatically divert the agent to a user topic, select **Add node (+)** to add a node, and then [Go to another topic](#).
8. In the flyout menu, select the user topic you created above.
9. Select **Save** when you are finished editing the message.
10. [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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

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**.

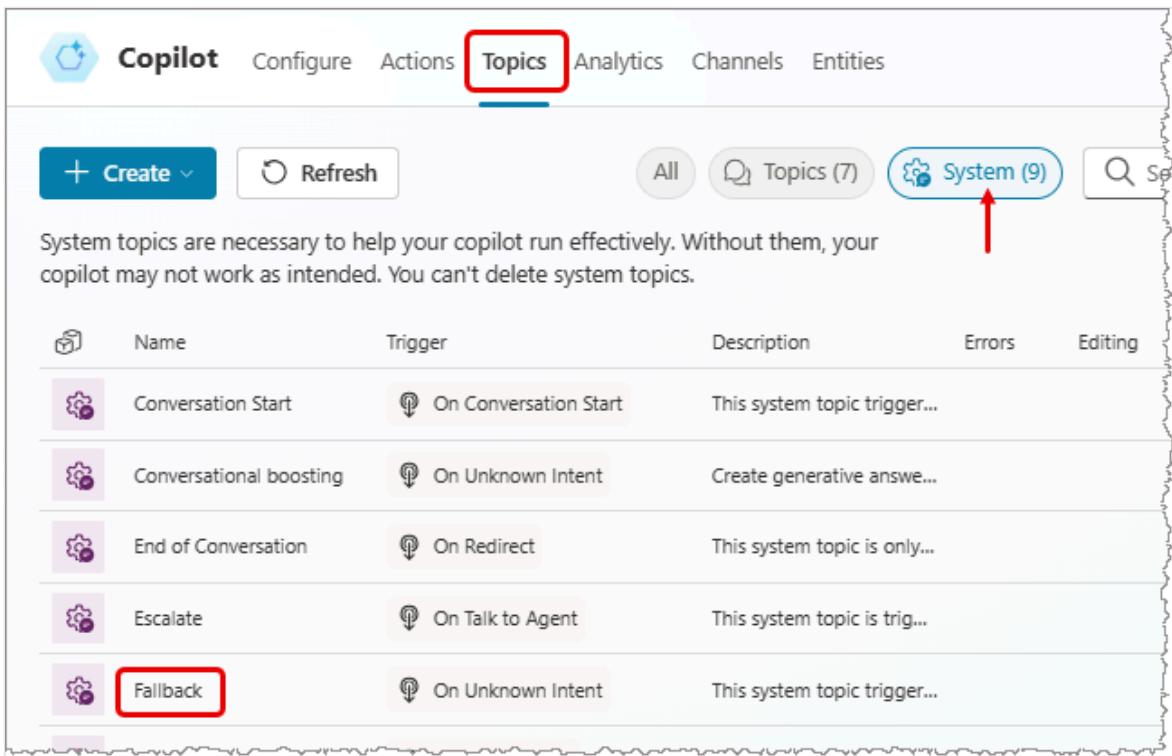
You can customize how the agent behaves when user input doesn't trigger any topic. For example, you can build a catchall topic to capture unrecognized user intent, create a topic to call back-end systems, or route a topic to existing services.

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.

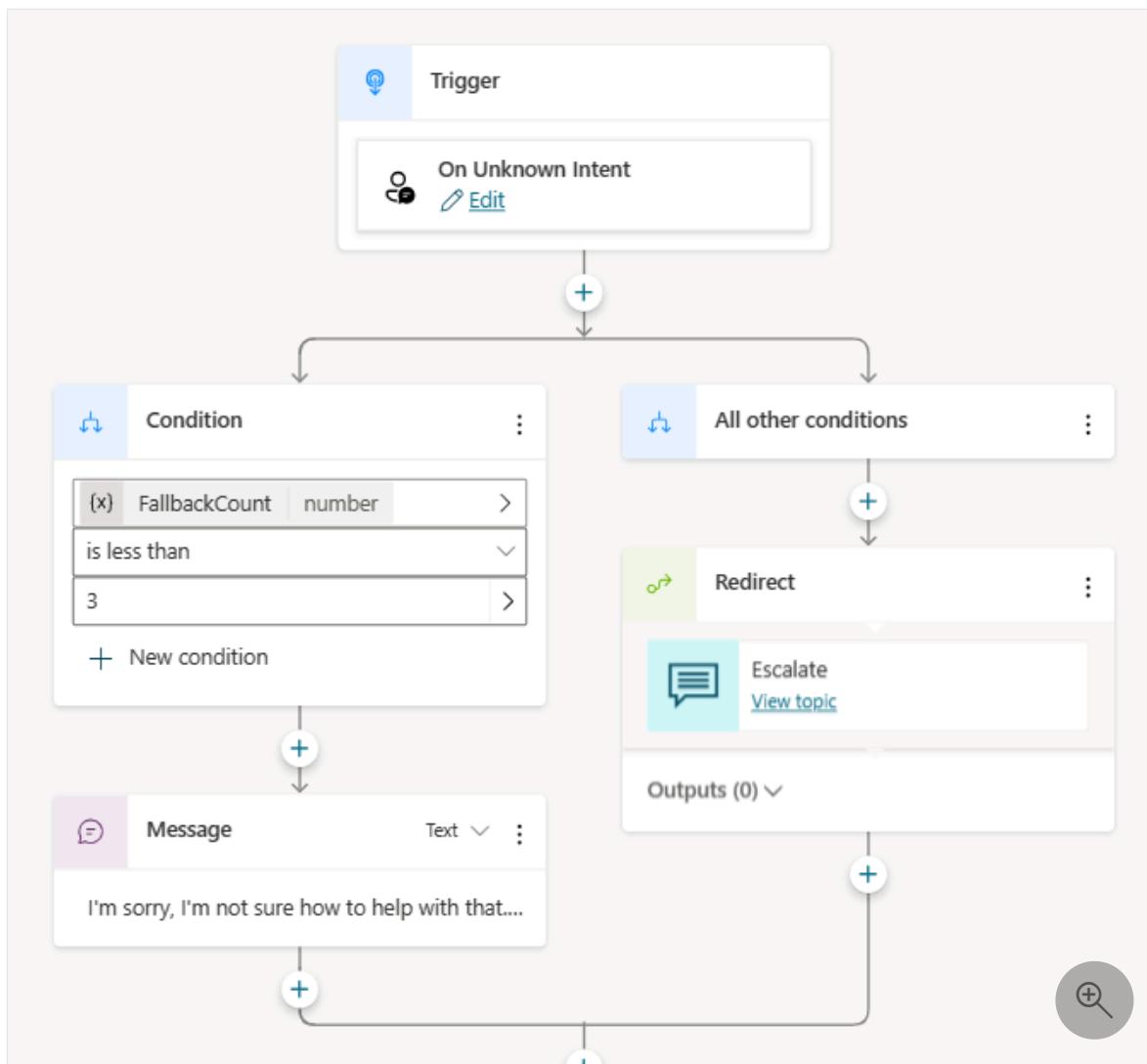


The screenshot shows the Microsoft Copilot Studio interface. At the top, there is a navigation bar with tabs: Copilot, Configure, Actions, **Topics** (which is highlighted with a red box), Analytics, Channels, and Entities. Below the navigation bar, there are several buttons: '+ Create' (blue), 'Refresh' (grey), 'All' (grey), 'Topics (7)' (grey), 'System (9)' (blue with a red arrow pointing to it), and a search bar. A message below the buttons 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 is a table with columns: Name, Trigger, Description, Errors, and Editing. The table lists five system topics: Conversation Start (On Conversation Start), Conversational boosting (On Unknown Intent), End of Conversation (On Redirect), Escalate (On Talk to Agent), and Fallback (On Unknown Intent). The 'Fallback' row has a red box around its name. The 'System (9)' button also has a red arrow pointing to it.

Name	Trigger	Description	Errors	Editing
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...		
Fallback	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](#).

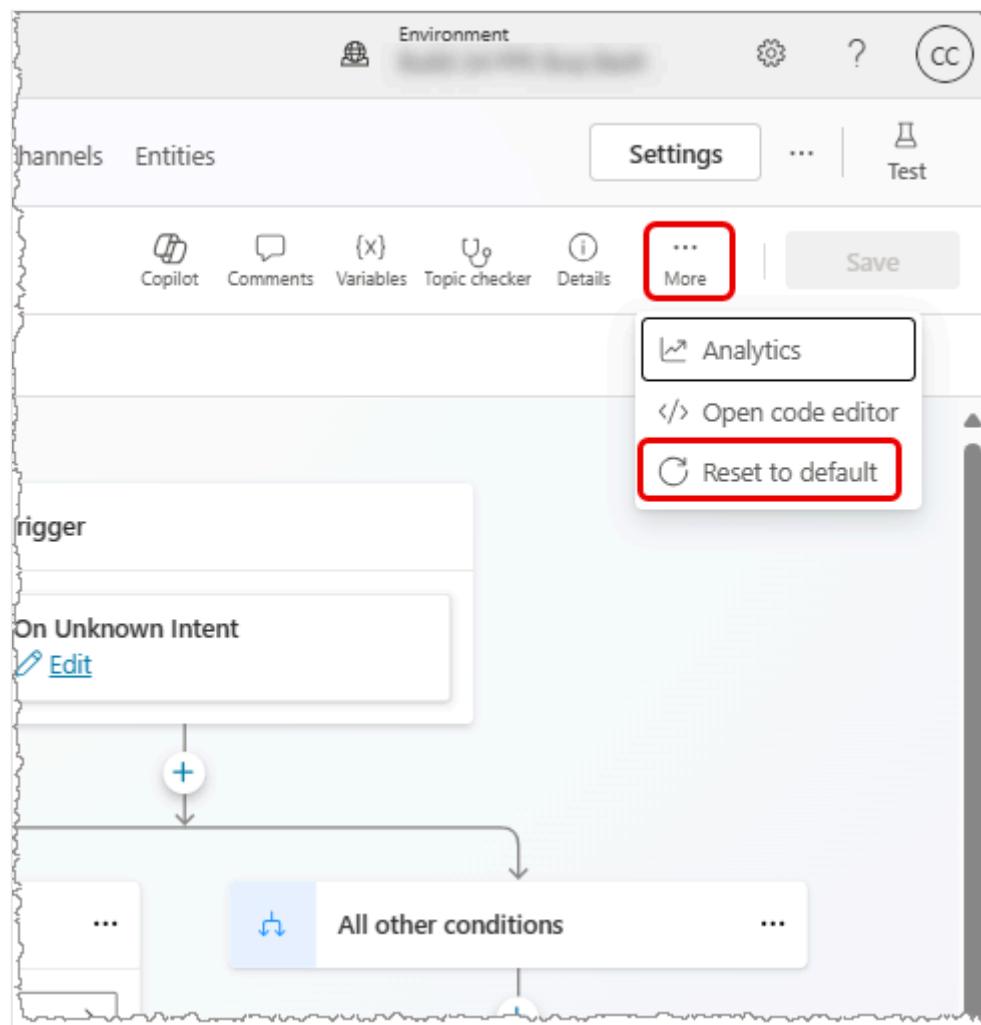
ⓘ Note

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

Article • 11/19/2024

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](#) that's based on the [Bot Framework Web Chat canvas](#).
This approach requires extensive developer knowledge. It's useful for organizations that want a completely custom experience.

ⓘ Important

You may 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.

Web app

After you create and [publish an agent](#), your customers can [use the agent's Web Chat canvas to interact with it](#).

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 (when it's [shared in Microsoft Teams](#)) directly from the portal.

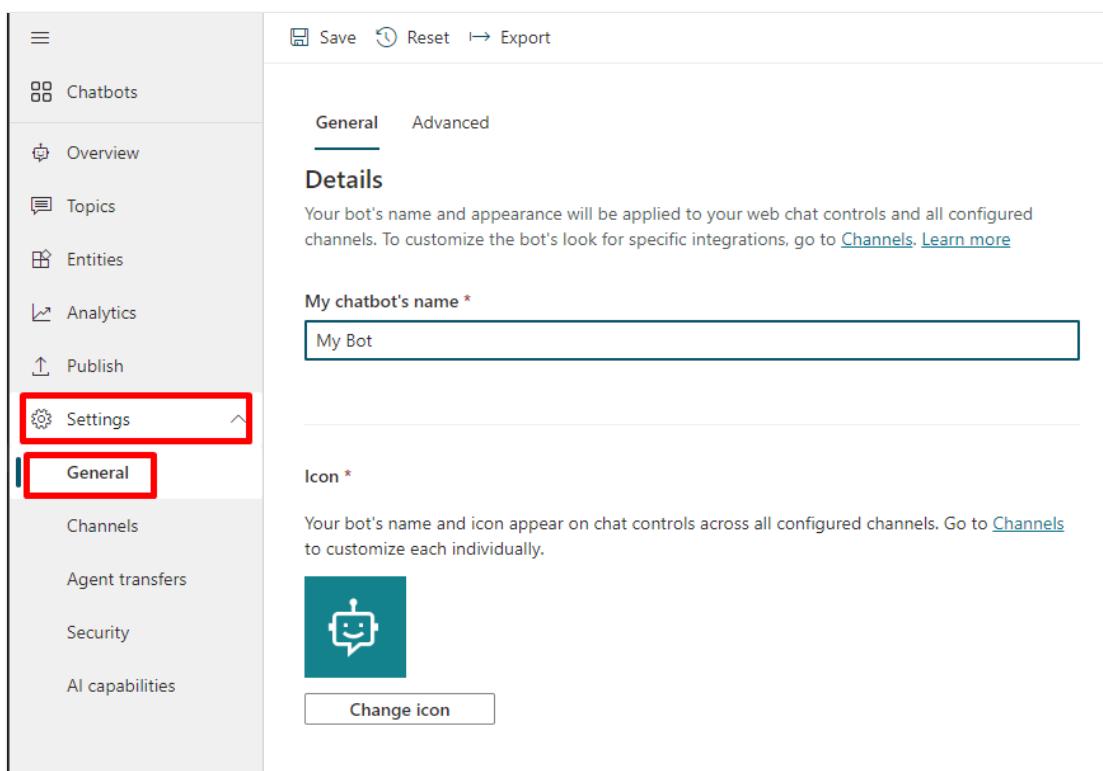
Change the agent name and icon

ⓘ Important

If your agent is connected to [Omnichannel for Customer Service](#), its name is defined by the **Display name** property in the Azure portal registration.

You can change the agent's name and icon. This will affect the icon in all channels where you publish your agent.

1. In the navigation menu, under **Settings**, select **Details**.
2. Change the agent's name and icon. Review the recommendations on [Microsoft Teams icon formats](#).
3. Select **Save** to commit your changes.



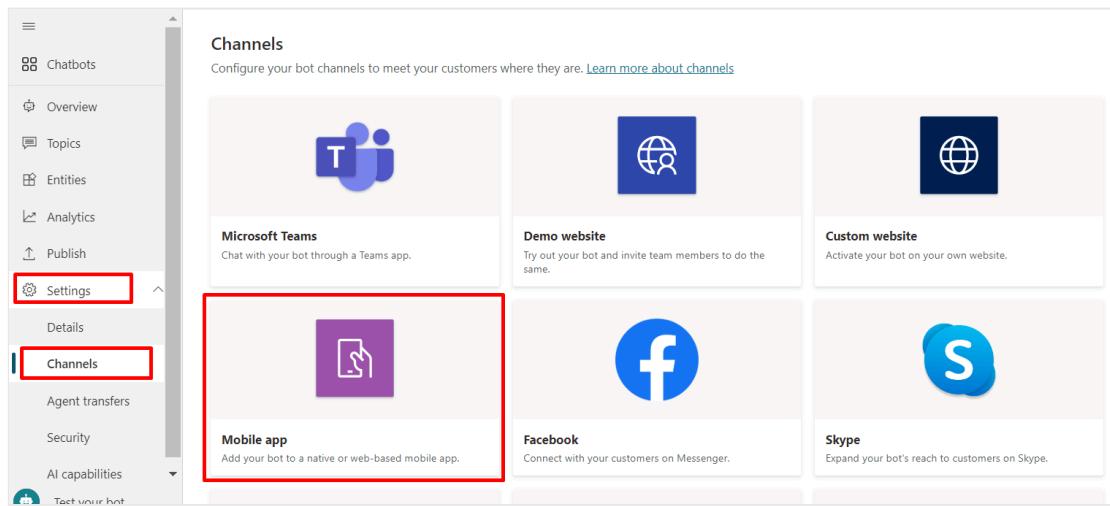
i Important

After updating your agent's icon, it may take up to 24 hours for the new icon to appear everywhere.

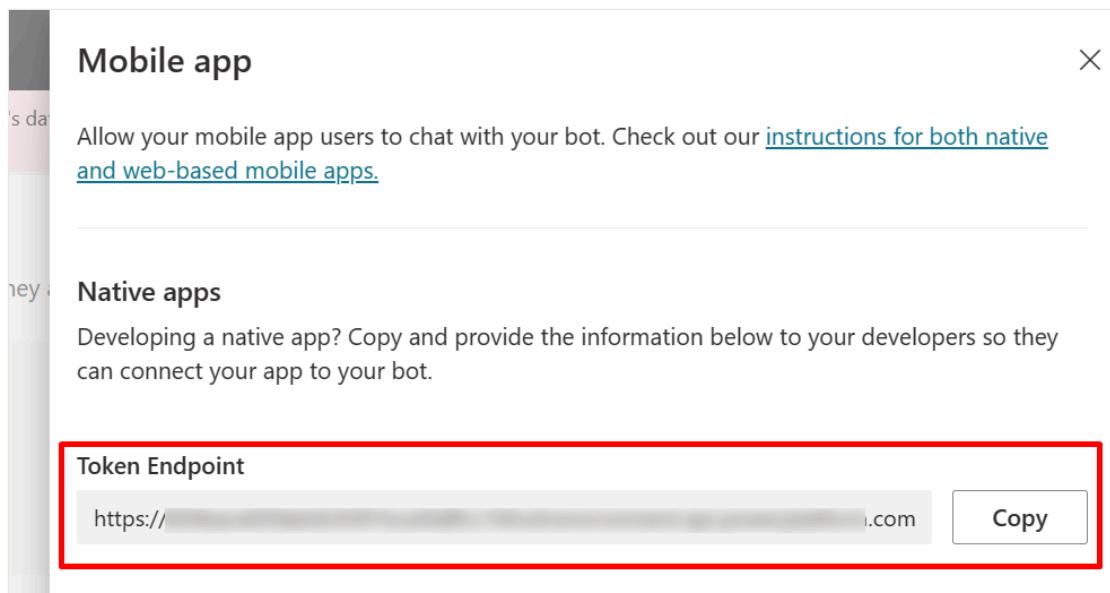
Retrieve token endpoint

To customize your canvas, whether it's the default canvas or a custom one you connect to, you need to retrieve your agent details.

1. In the navigation menu under **Settings**, select **Channels**.
2. Select **Mobile app**.



3. Next to **Token Endpoint**, select **Copy**.



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 and paste the HTML code below and save it as *index.html*.

You can also copy and paste the code below into the [w3schools.com HTML try it editor](#). You'll still need to add your token endpoint.

```
<!doctype html>
<html lang="en">
  <head>
```

A screenshot of a modal dialog titled 'HTML'. It contains a single line of HTML code: `<!doctype html><html lang="en"><head>`. This code is used to define the document type and language for a web page.

```
<title>Contoso Sample Web Chat</title>
<!--
    This styling is for the Web Chat demonstration purposes.
    It is recommended that style is moved to a separate file for
organization in larger projects.

    Please visit https://github.com/microsoft/BotFramework-
WebChat for details about Web Chat.
-->
<style>
    html,
    body {
        height: 100%;
    }

    body {
        margin: 0;
    }

    h1 {
        color: whitesmoke;
        font-family: Segoe UI;
        font-size: 16px;
        line-height: 20px;
        margin: 0;
        padding: 0 20px;
    }

    #banner {
        align-items: center;
        background-color: black;
        display: flex;
        height: 50px;
    }

    #webchat {
        height: calc(100% - 50px);
        overflow: hidden;
        position: fixed;
        top: 50px;
        width: 100%;
    }
</style>
</head>
<body>
    <div>
        <div id="banner">
            <h1>Contoso agent name</h1>
        </div>
        <div id="webchat" role="main"></div>
    </div>

    <!--
        In this sample, the latest version of Web Chat is being
used.
    -->
```

In production environment, the version number should be pinned and version bump should be done frequently.

```
Please visit https://github.com/microsoft/BotFramework-  
WebChat/tree/main/CHANGELOG.md for changelog.  
-->  
<script crossorigin="anonymous"  
src="https://cdn.botframework.com/botframework-  
webchat/latest/webchat.js"></script>  
  
<script>  
  (async function () {  
    // Specifies style options to customize the Web Chat  
    // canvas.  
    // Please visit https://microsoft.github.io/BotFramework-  
    // WebChat for customization samples.  
    const styleOptions = {  
      // Hide upload button.  
      hideUploadButton: true  
    };  
  
    // Specifies the token endpoint URL.  
    // To get this value, visit Copilot Studio > Settings >  
    Channels > Mobile app page.  
    const tokenEndpointURL = new URL('<AGENT TOKEN  
ENDPOINT>');  
  
    // Specifies the language the agent and Web Chat should  
    // display in:  
    // - (Recommended) To match the page language, set it to  
    document.documentElement.lang  
    // - To use current user language, set it to  
    navigator.language with a fallback language  
    // - To use another language, set it to supported Unicode  
    locale  
  
    // Setting page language is highly recommended.  
    // When page language is set, browsers will use native  
    font for the respective language.  
  
    const locale = document.documentElement.lang || 'en'; //  
    // Uses language specified in <html> element and fallback to English  
    // (United States).  
    // const locale = navigator.language || 'ja-JP'; // Uses  
    // user preferred language and fallback to Japanese.  
    // const locale = 'zh-HAnt'; // Always use Chinese  
    // (Traditional).  
  
    const apiVersion = tokenEndpointURL.searchParams.get('api-  
version');  
  
    const [directLineURL, token] = await Promise.all([  
      fetch(new  
        URL(`/powervirtualagents/regionalchannelsettings?api-  
version=${apiVersion}`), tokenEndpointURL)  
    ]);
```

```

        .then(response => {
            if (!response.ok) {
                throw new Error('Failed to retrieve regional
channel settings.');
            }

            return response.json();
        })
        .then(({ channelUrlsById: { directline } }) =>
directline,
            fetch(tokenEndpointURL)
            .then(response => {
                if (!response.ok) {
                    throw new Error('Failed to retrieve Direct Line
token.');
                }

                return response.json();
            })
            .then(({ token }) => token)
        ]);

        // The "token" variable is the credentials for accessing
        the current conversation.
        // To maintain conversation across page navigation, save
        and reuse the token.

        // The token could have access to sensitive information
        about the user.
        // It must be treated like user password.

        const directLine = WebChat.createDirectLine({ domain: new
URL('v3/directline', directLineURL), token });

        // Sends "startConversation" event when the connection is
        established.

        const subscription =
directLine.connectionStatus$.subscribe({
            next(value) {
                if (value === 2) {
                    directLine
                    .postActivity({
                        localTimezone:
                        Intl.DateTimeFormat().resolvedOptions().timeZone,
                        locale,
                        name: 'startConversation',
                        type: 'event'
                    })
                    .subscribe();
                }
            }
        });

        // Only send the event once, unsubscribe after the
        event is sent.
        subscription.unsubscribe();
    }
}

```

```
        }
    });

    WebChat.renderWebChat({ directLine, locale, styleOptions
}, document.getElementById('webchat'));
})();
</script>
</body>
</html>
```

3. In the *index.html* file you created, enter your token endpoint at the line `const tokenEndpointURL = "<YOUR TOKEN ENDPOINT>";`.
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 are receiving responses from it and that it's working correctly.

If you encounter problems, make sure you've published your agent, and that your token endpoint has been inserted in the correct place. It should be after the equals sign (=) at the line `const tokenEndpointURL = "<YOUR TOKEN ENDPOINT>"`, and surrounded by double quotation marks (").

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 a number of 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'
};
```

- 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

- Update the *index.html* file with following sample code:

JavaScript

```
const styleOptions = {
  backgroundColor: 'lightgray'
};
```

- Change `backgroundColor` to any color you wish. You can use standard CSS color names, RGB values, or HEX.

Change the agent name

- Update the `<h1>` text in the *index.html* file with the following:

HTML

```
<body>
  <div id="banner">
    <h1> Contoso agent
    name</h1>
  </div>
```

- Change the text to whatever you want to call the agent. You can also insert an image, although you may 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. Our guidance here is intended for experienced IT professionals, such as IT admins or developers who have a good understanding of developer tools, utilities, and IDEs.

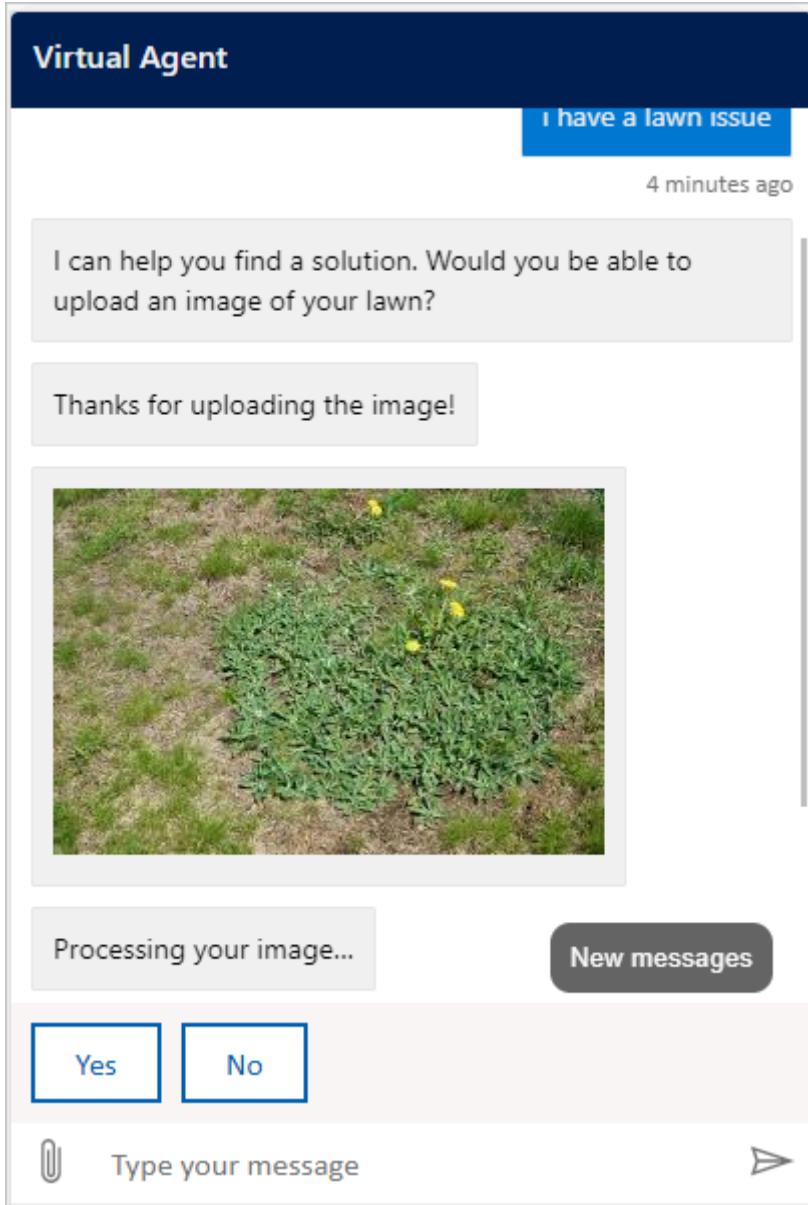
Pick a sample to customize

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:

The screenshot shows a mobile application interface for flight tracking. At the top, there is a header with the text "card flight tracking" and a timestamp "22 minutes ago". Below this, a section titled "Showing flight, tracking" contains a blue airplane icon and the word "Flight Status" followed by the word "DELAYED" in red. A table lists passengers and their seat assignments: Sarah Hum (14A), Jeremy Goldberg (14B), and Evan Litvak (14C). Below this, another table provides flight details: Flight KL0605 departs at 10:10 AM and arrives at 12:00 AM. The departure city is Amsterdam (AMS) and the arrival city is San Francisco (SFO). At the bottom of the screen, there is a text input field with the placeholder "Type your message" and a send button.

- [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

In order to host your custom canvas, deploy all files to a web app.

Related content

- [Bot Framework Web Chat ↗](#)
 - [Generating a Direct Line token](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

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

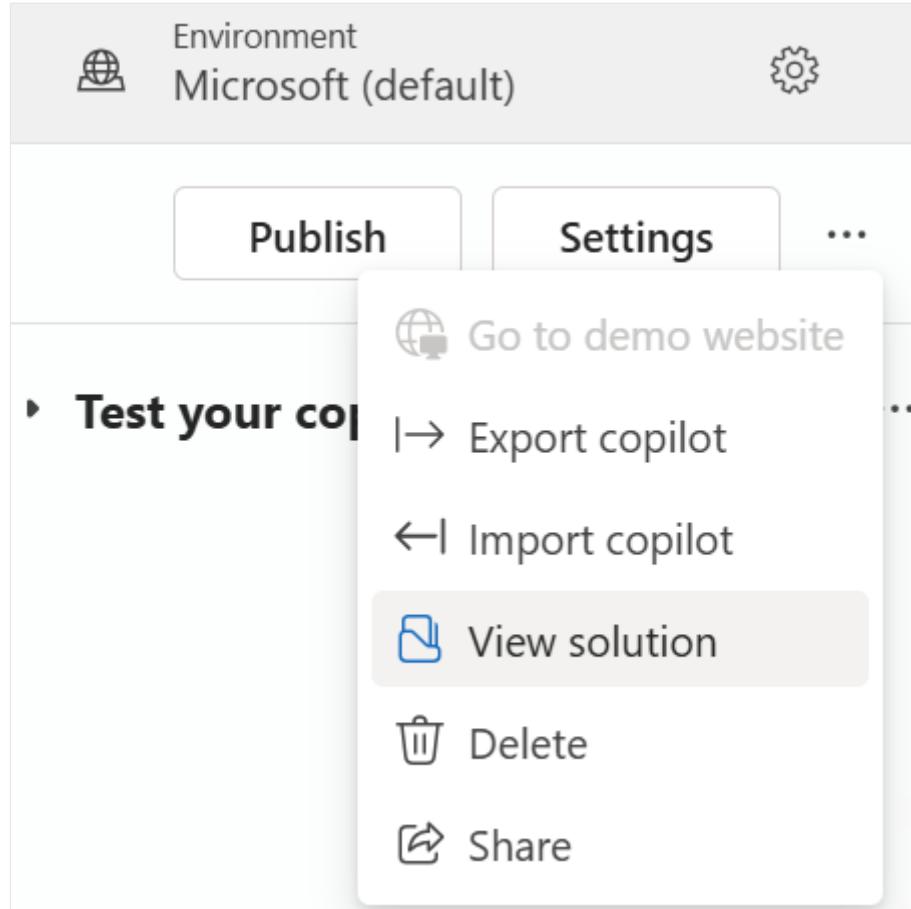
A solution acts as a "carrier" for your agents. When you create an agent, it's automatically added to a default solution, [which you can change](#).

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" item 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 highlighted with a blue border.

To access the solution explorer:

- From the side navigation pane in Copilot Studio:
 1. In Copilot Studio, select the menu icon (...) on the side navigation pane, and then select **Solutions**.

The Power Platform interface is shown, featuring a sidebar on the left with icons for Home, Create, Copilots, Library, and three dots for more options. The main area displays the following components:

- Power Platform**: A section containing four items:
 - Power Apps**: Represented by a purple and pink icon.
 - Power Automate**: Represented by a blue arrow icon.
 - Power Pages**: Represented by a purple and blue icon.
 - Power Platform Admin Center**: Represented by a green and blue icon.
- Learn more about AI integration**: A section containing two items:
 - Azure Cognitive Service**: Represented by a blue and pink icon.
 - Azure Open AI**: Represented by a blue starburst icon.
- Solutions**: A section containing one item:
 - Solutions**: Represented by a blue icon.

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.

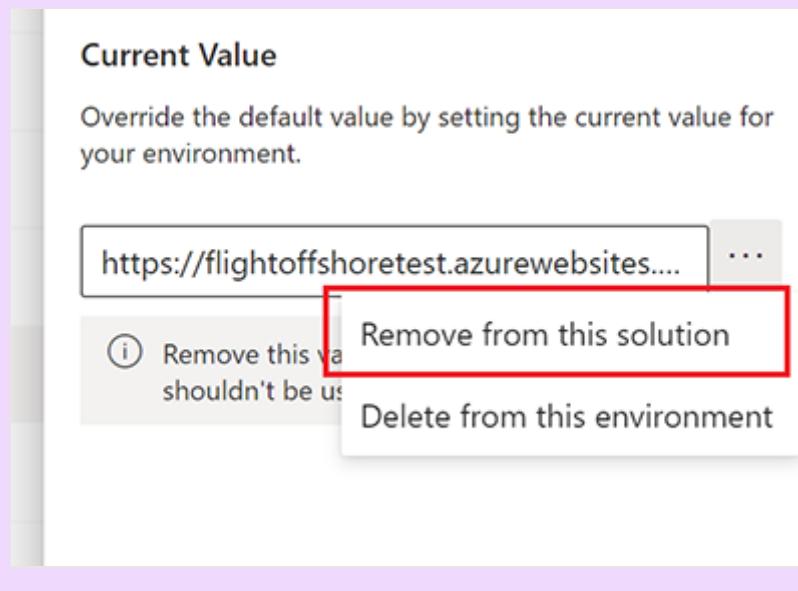
Create a custom solution and add your agent

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)

Article • 11/19/2024

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](#).

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 ellipsis (...) 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 ellipsis (...) 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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Share agents with other users

Article • 11/19/2024

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.

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

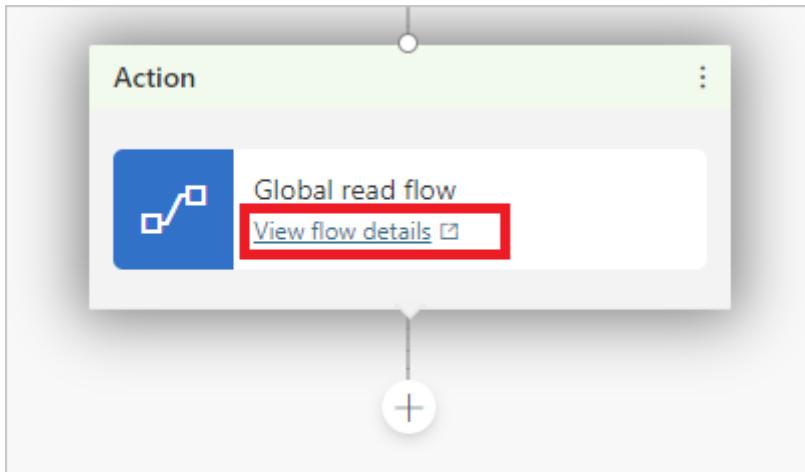
You can [add actions to an agent using flows in Power Automate](#). However, sharing an agent doesn't automatically share the flows in the agent.

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.

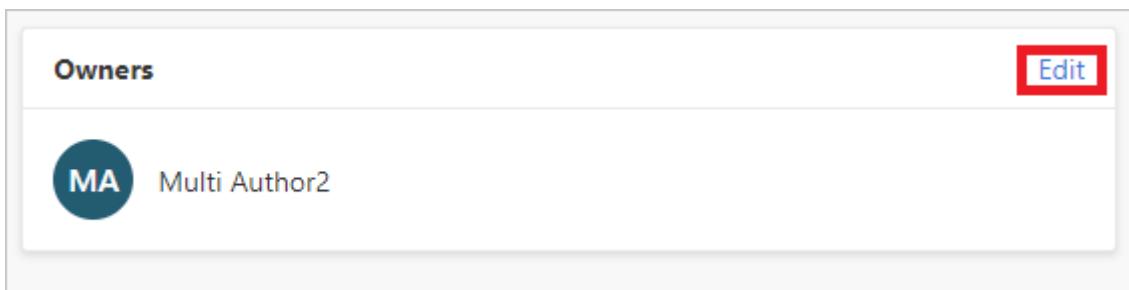
Test your agents to make sure users who chat with them have the required permissions to run the [Power Automate flows](#).

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 ↗](#)

Conversational language understanding integration overview

Article • 11/19/2024

You can integrate your [conversational Language Understanding \(CLU\)](#) model with a Copilot Studio agent. Conversational language understanding is one of the custom features offered by Azure Cognitive Service for Language. It is a cloud-based service that applies machine-learning intelligence to enable you to build natural language understanding component to be used in an end-to-end conversational application. Language service APIs are available (but not required for you to integrate with Copilot Studio) and the [Azure Language Studio](#) provides natural language processing (NLP) features for analyzing conversational text.

For projects created with this capability, agent creators can now access existing CLU model intents and entities directly in Copilot Studio. Copilot Studio users can now map CLU intents to trigger agent dialogs in the same way as native Copilot Studio [utterance triggers](#). Entities imported from a CLU model appear on the Copilot Studio [Entities](#) page and can be used alongside Copilot Studio prebuilt entities. Bot creators can add new [question nodes](#) in a topic and select available entities and intents from the imported CLU model. CLU entities are bound to equivalent Copilot Studio objects. Users specify the name, the data type, and the [JSON structure](#) of a custom data type, if desired for each entity.

Note

For existing agents that you want to take advantage of CLU integration, you'll need to map the CLU model to Copilot Studio, 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 Copilot Studio.

Prerequisites

- A fully [trained CLU model](#). This includes intents for each of [Copilot Studio's system topics](#) and for [any custom topics](#) you may want to author.
- Enabled deployment of the model (including the key and prediction URL)
- CLU project name and deployment name
- [Copilot Studio account](#)
- [Existing Copilot Studio agent](#)

- Ensure that your Copilot Studio language connector is using maker authentication

Key concepts

The concepts discussed in this article help you understand how to integrate CLU models with Copilot Studio projects. For more information, see [Get started with conversational language understanding integration](#).

CLU connectors

A *connector* is a wrapper around an API that allows the Azure Cognitive Service to talk to Copilot Studio. It provides a way for users to connect their accounts and apply a set of prebuilt actions and triggers to build their apps and workflows. For more information, see [Azure Cognitive Service for Language - Connectors](#).

Power Platform connectors allow Microsoft services to talk to the CLU API. For more information, see [Power Platform and Azure Logic Apps connectors documentation - Connectors](#). Although you can import any connector types, Copilot Studio doesn't currently validate connectors.

Connections in Copilot Studio

A [connection](#) is a stored [authentication credential](#) 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 need to set up a connection for it. We offer a standard way of achieving this capability in Power Platform by way of a Connection reference. For more information, see [Connection references in solutions - Power Apps](#).

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 does not 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
- System variable(s) that support the reason for triggering the recognizer

[Expand table](#)

Property name	Type	Description
DisplayName	String	The display name for the recognized intent; localized in the current language and to be shown in the DYM 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 variables, including system variables, see [Work with variables](#).

Mix and match entity component types

CLU entities include relevant information extracted from NLU utterances. An entity can be extracted using different methods. They can be learned through context, mixed and matched from a list, or detected by a prebuilt recognized entity. For a complete list of entity component types, see [Component Types](#).

 **Note**

You can also use Copilot Studio [prebuilt entities](#) along with CLU entities.

Related content

Other agent-building features that CLU integration supports include:

- System topics: [Use system topics](#)
 - Entities and slot filling: [Use entities and slot filling in agents](#)
 - CLU best practices: [Conversational language understanding best practices - Azure Cognitive Services](#)
 - Topic flow interruptions: [Design and control conversation flow - Bot Service](#)
 - Power Fx expressions: [Create expressions using Power Fx](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Get started with conversational language understanding integration

Article • 11/19/2024

This article includes the step-by-step procedures you need to get started with [conversational language understanding \(CLU\) integration](#). Let's get started.

To create an agent using an existing CLU model:

1. [Prepare your environment](#).
2. [Prepare your conversational language understanding project](#).
3. Map CLU intents and entities.
 - [Manually map intents](#).
 - [Manually map entities](#).
 - [Use the wizard to automatically map CLU intents and entities](#).

Prepare your environment

Before you begin, make sure that you have your CLU model project name, deployment name, Cognitive Services account key, and site URL for Azure Cognitive Service for Language.

1. Create a [Copilot Studio agent](#).

(!) Note

Ensure the [environment for your agent](#) has been created in a [region supported by Copilot Studio](#) with an [equivalent CLU region](#).

Not all regions supported by Copilot Studio have an equivalent CLU region.

2. With your agent open, select **Settings** on the top menu bar.
3. Select **Language understanding**.
4. If you see the following, then your environment *isn't* connected to Azure Cognitive Service for Language. Select **Manage connections** in Power Apps, to [create a CLU connection](#).

Natural language understanding



Azure Cognitive Service for Language

Want to reuse an existing Conversational language understanding (CLU) model? You can use CLU in place of the native Microsoft Copilot Studio NLU.

[Learn more](#)

[Manage connections in Power Apps](#)

In the Natural language understanding area of the Language settings page, Microsoft Copilot Studio NLU is selected by default.

5. Select the desired CLU connection from the **NLU Resource** list and select **Save**.

The screenshot shows the 'Language' settings page. At the top, there are 'Save' and 'Refresh' buttons. Below that is a section titled 'Language' with a note about customizing copilot language settings. The main focus is the 'Natural language understanding' section. It states that the copilot has been enabled to use an external NLU system and offers to select an external model or use Microsoft Copilot Studio built-in NLU. A 'Learn more' link is provided. A dropdown menu labeled 'NLU Resource *' is open, showing 'my CLU connection' as the selected option. A link to 'Manage connections in Power Apps' is also visible.

A window appears, prompting you to save a snapshot of your agent.

6. Select **Save snapshot**, and save the resulting ZIP archive, named *botContent.zip* by default, to the desired location. The ZIP archive contains a single YML file named *botContent.yml* with your agent content, including trigger phrases and messages.

7. Select **Yes, delete my trigger phrase** and then select **Continue**.
8. Enter the name and model deployment information for the appropriate Azure Cognitive Service for Language project, and select **Save**.

 **Note**

Changing the NLU configuration for your agent back to 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.

Now that you're done preparing your environment, you're ready to map intents and entities to CLU model data.

Prepare your conversational language understanding project

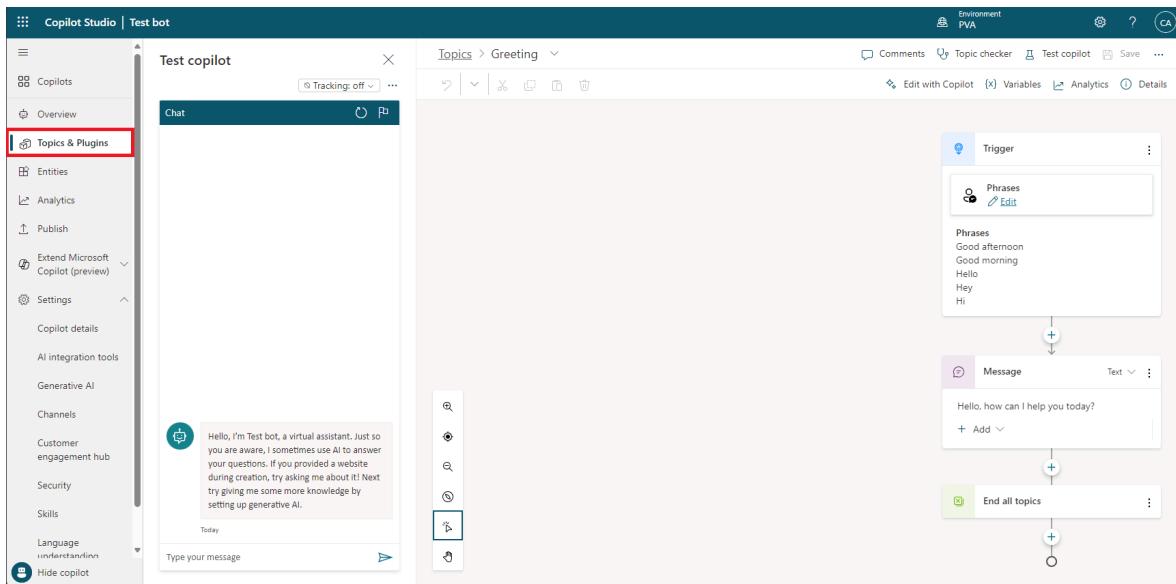
1. If you don't have an Azure subscription, create a [free account](#) before you begin.
2. Create a language resource in [Language Studio](#), with the [conversational language understanding](#) feature enabled.
3. To ensure Copilot Studio functions correctly, you need to create intents for [Copilot Studio system topics](#) and for [any custom topics](#) you want to create.
4. Author [any custom entities](#) you want to use in your Copilot Studio project.
5. Train and deploy the project in the same region as your Copilot Studio resource.

Map CLU intents and entities to Copilot Studio prebuilt data types

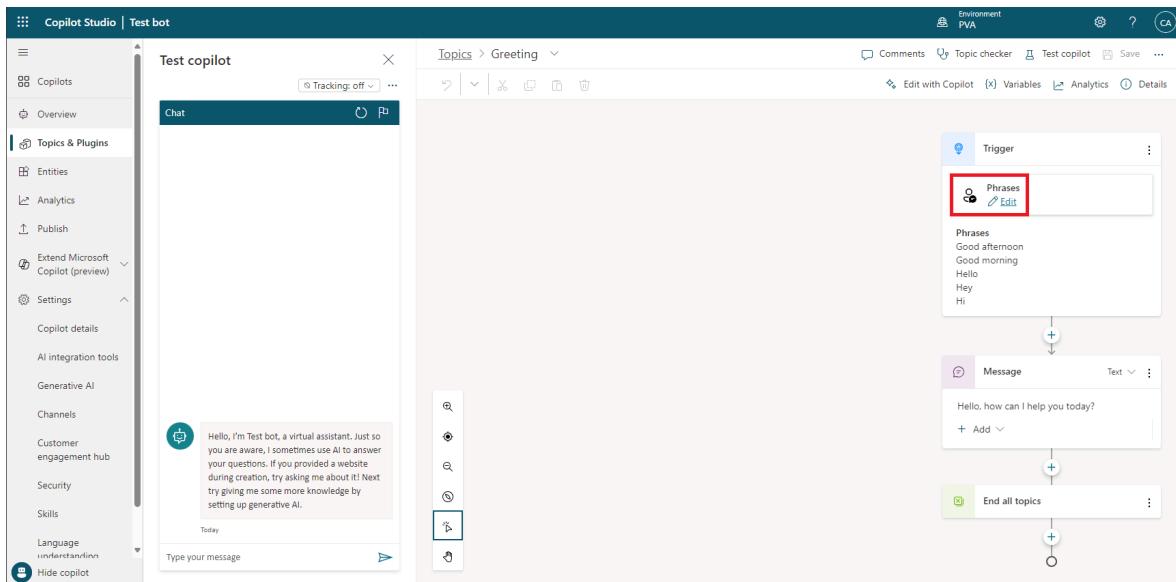
You create a new Copilot Studio agent by mapping existing topics to CLU intents. You can manually map intents and entities, or for bulk mapping, see [Use the Bulk tool wizard](#).

Manually map intents

1. Open your Copilot Studio agent.
2. From the left navigation, select **Topics and plugins** and select an intent.



3. In the *Phrases* section of the **Trigger** node for the intent, select the **Edit** link.



4. In the *Add phrases* box, enter the name of the CLU intent that you want to map to.

[←](#) On Recognized Intent

X

Phrases

To start learning, the bot needs 5-10 short trigger phrases.

[Show writing tips](#)

Add phrases [?](#)

Enter text

+

To add phrases in bulk, paste in line-separated phrases or use Shift+Enter to create line separation

Good afternoon

Good morning

Hello

Hey

Hi

[> Advanced](#)

ⓘ Note

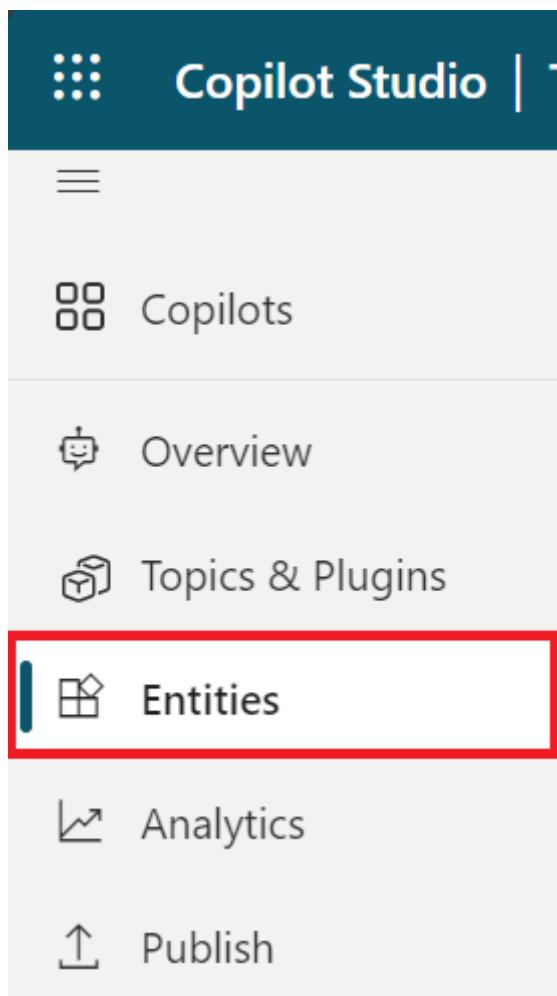
The CLU intent name must be typed exactly as stored in the CLU model, including matching case as uppercase and lowercase characters.

5. To save the intent, click the + (plus) sign to the right of the *Add phrases* box.
6. Repeat for the other intents.

Manually map entities

1. Open your agent.

2. From the left navigation, select **Entities**.



3. Select **New entity** on the **Entities** page.

A screenshot of the "Entities" page. At the top, there's a header with a "New entity" button (highlighted with a red box) and a "Register an external entity" button. Below the header is a table with three rows. The columns are labeled "Name", "Description", and "Method".

Name	Description	Method
MyRegesx	aaa	Regex
billingQuestion	billingQuestion2	External
Temperature	Temperature from CLU	External

4. In the **New entity** pane, Add a *name*, *description* (optional), and *Data Type* for your entity.

Untitled

X

Name *

Entity name (required)

The display name of a closed list type cannot be empty

Description

Description (optional)

Method

External

The bot will use external LU services to extract entities.

[Learn how to use entities](#)

Data Type

String

String

Boolean

Number

Record

Table

From sample data

5. To access CLU prebuilt entities, from the Data Type drop-down menu, select **From sample data**, and select one of the CLU prebuilt JSON code snippets.
6. (Optional) If you want to import JSON schema from a sample file, select **Get schema from sample JSON** to access the external file.

Use the Bulk tool wizard

You create a new agent by bulk mapping existing topics to CLU intents using the **Map intents and entities** wizard. With the wizard, your agent can automatically utilize existing intents from your CLU model rather than trigger phrases to determine which topic to map to.

Begin by uploading your CLU model data file and automatically map CLU intents to your Copilot Studio agent.

1. From the left navigation, select **Language**, then **Add topics and entities from model data**.
 2. In the **File name** option, select **Choose file** to select the data file from which you want to upload your CLU model.
- Your model data file appears in the **Preview** window.
3. Select **Next** to continue.
 4. select **Add intents and entities**.
 5. In the **Map existing topics to CLU intents** screen, select a CLU intent to map to the corresponding topic.

Add intents and entities X

Upload your model data file

Map existing topics to CLU intents

Create new topics for CLU intents

Review

Map existing topics to CLU intents

Your bot will use intents in your CLU model, rather than trigger phrases, to determine which topic to fire. Map a CLU intent to each topic you select in the table below.

For PVA topic	Use CLU intent
Lesson 3 - A topic with a condition, ...	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>
Thank you	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>
Lesson 2 - A simple topic with a con...	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>
Lesson 1 - A simple topic	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>
Goodbye	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>
PurchaseTickets	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>
Greeting	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>
Start Over	<input style="border: 1px solid #ccc; width: 100%; height: 30px;" type="button" value="Select an intent"/>

6. Review your selections, then select **Save**.

Add intents and entities

X

Upload your model data file

Map existing topics to CLU intents

Create new topics for CLU intents

Review

Review

Please review your selections.

[Existing topics](#) [New topics](#)

Topic

Intent

Back

Save

Cancel

7. (Optional) For those intents that aren't mapped to an existing topic, you can create new topics for each intent.

In the wizard, you can see a list of CLU intents that are not yet mapped to a topic. In the field **Create a new topic**, enter a name for each of the topics that you want to create, and then select **Next**.

Add intents and entities

X

Upload your model data file

Map existing topics to CLU intents

Create new topics for CLU intents

Review

Create new topics for CLU intents

Your CLU model contains intents that are not used for any PVA topic. Create a new topic for each intent you select in the table below.

For CLU intent

Create a new topic

None

Timetable

RouteQuestion

PurchaseTickets

Goodbye

Greeting

Agent

Start Over

Thank you

CurrentTemperature

Back

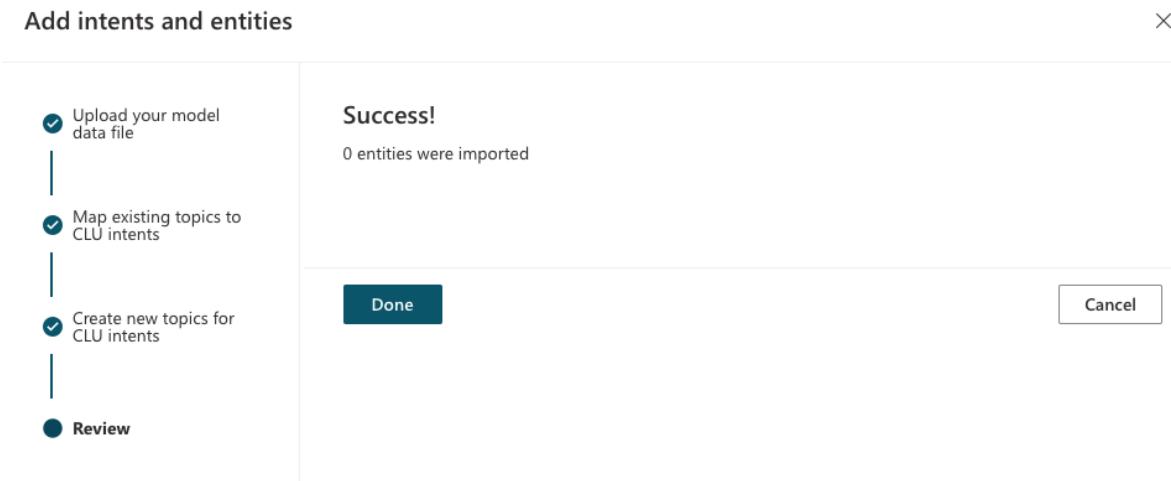
Next

Cancel

ⓘ Note

Any fields that you leave blank are ignored by the wizard.

- Once all CLU intents and entities are mapped to topics, select **Done**.



Once you are done with this procedure, your agent topics now appear in the **Topics** page. For more information, see [Manage topics](#).

Create a CLU connection in Power Apps

- In Power Apps, if the **Connections** page isn't already in focus, select **Connections** in the side navigation pane.
- Select **New connection**.
- 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.
- For the authentication type, select **Api Key**, and enter the required account key.
- Enter the appropriate root site URL, if any.
- Select **Create**. The new connection, with the default name "Azure Cognitive Service for Language," appears in the list of connections for your environment.
- Select the **More commands** icon (...) next to this connection and select **Edit**.
- In the window that appears, replace the default display name with something else, to distinguish this connector from other CLU connectors, and select **Save**.

Related content

- Entity registration for conversational language understanding integration
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Entity registration for conversational language understanding integration

Article • 11/19/2024

This article discusses adding conversational language understanding (CLU) entities to Copilot Studio agents. The entities are composed of the following Boolean, string, and number data types. For more information, see [Data types](#). In most cases, you can use [prebuilt entities](#) for your projects. If you would like to use CLU entity types with custom JSON resolutions, the following schema examples are provided as a reference.

To set up your environment for mapping CLU entities to Copilot Studio agents, see [Get started with conversational language understanding integration](#).

The following data types are available for mapping CLU entities:

- BooleanDatatype: `Choice.Boolean`
- StringDatatype: `Geography.Location`, `Regex`, `List`, `General.Event`,
`General.Organization`, `Phone Number`, `IpAddress`, `Person.Name`, `Phone Number`, `URL`
- NumberDatatype: `Number`

ⓘ Note

Composite entities (entities with multiple components) map to `StringDatatype`.

Schema table

You can use sample JSON code to register [entities](#) for agents you create. Entities resolve to complex data types. You can manually map CLU entities to Copilot Studio data types by copying and pasting the following JSON code blocks for the relevant entity.

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  
}
```

CLU date**Time** entity types

`DateTime` is a special entity type that changes the returned resolution based on the types of user input that are received.

The following examples demonstrate how to configure entities for different types of date and time utterances. You can create your own mappings, based on these examples, depending on the type of result you expect your agent users to provide.

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"  
}
```

Related content

- Get started with conversational language understanding integration

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Copilot connectors and actions overview (preview)

Article • 12/13/2024

[This article is prerelease documentation and is subject to change.]

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 copilot whose functionality they wish to extend:

- Custom agents
- Microsoft 365 Copilot and Copilot agents
- Other copilots that they have maker access to in their organization

Select an agent to extend by selecting **Agents** in the main menu and then choosing from the list of custom agents and Copilot agents.

View existing actions for an agent

After selecting an agent, you can view the actions currently available for the agent within the configuration/authoring panel for the agent.

To view the existing actions for a agent, select the **Actions** tab on the agent page.

The list of available actions appears.

Creating your own new action

You can also create your own new actions.

The steps depend on the type of agent:

- Custom agent: Select + Add an action.

- Copilot agent: Select + Add Action.

You can then select a type of action from a list of available options to start the process of extending the agent with more capabilities.

Copilot Studio guides you through the configuration steps. The details of the steps depend on whether you're extending a custom agent or Copilot agent.

Detailed configuration steps are covered later in the documentation. After completion, the user can publish the action for the agent. This action is now ready for an administrator to review and approve. Once approved, the action is available to all users who have permission to use it, within that specific organization.

Copilot Studio offers multiple ways to make the extension process easy and comprehensive, including:

- Authoring: Author the extension for the agent, which results in an action being created.
- Testing: Validate that the action works with your agent by testing it during or after the publishing process.
- Editing: Save drafts of actions in progress, or edit actions and publish the outcomes for use.
- Export/Import: Take existing completed actions and distribute them to other organizations within your company or to your own customers if you're an ISV. You can also export the package to be submitted for certification with Microsoft to make your action available in the online catalog.
- Feedback: Inform users they're about to make a change to data and ask permission before taking action, provide a card for easy display of results, provide sample questions users might ask the agent.
- Branding: Provide icons and branding information for your action for the purposes of distribution.

 **Note**

Microsoft 365 Copilot can also be extended with [Copilot agents](#).

What happens after you publish your action?

Once an action is published, your agent administrator has to review and enable the action for use. When an administrator enables a agent, they must also determine which groups of users have permission to use it. The location for managing permissions might

differ depending on the agent. More information is available in the administration documentation. Once activated, an action appears in the environment where it was created. The action can also be exported for use in other agents. Exporting creates a package that can be deployed in other environments. Therefore, you don't need to recreate the action for each environment you wish to use it in.

More information on the import and export of actions is available in [Export solutions](#).

Published actions can also be submitted to Microsoft for certification. Certification is done using the Partner Center and involves a review of the action and certification by Microsoft. After that it appears in the action catalog where *all* users of Copilot Studio can use to extend their agents as well.

For more information on certification, see the following documents:

- [Verified publisher certification process](#)
- [Submit your connector for Microsoft certification](#)

Extend a custom agent with an action

To extend a custom agent with an action, see [Use actions with custom agents \(preview\)](#).

Extend a Copilot agent with an action

Copilot Studio provides a quick and easy way to author an action using a step-by-step wizard to guide you through the process.

It can be used to extend Microsoft-branded Copilot agents or custom agents. Once you pick the agent you would like to extend, it walks you through a series of steps to create the action.

The wizard asks you to:

- Choose the type of handler/source you want to use
- Supply authentication information if necessary for type of handler/source you chose
- Select the component actions you want to include in the action
- Define the input and output parameters for each component action
- Provide other enhanced experiences such as:
 - User consent dialogs
 - Adaptive cards to display your results, and conversation starting questions
 - Branding information for certification

After you have completed these steps, you can publish your action. An administrator can then review your action and activate it. Once the action is activated, you can use it within your organization, deploy it to other organizations, or submit it to Microsoft for certification, to make it available to the general public.

Components of an action

Actions created in Copilot Studio can have different components that are brought together to create the action for use in agents.

- Handlers and content sources
- Authentication
- Component actions
- Inputs and outputs
- Enhanced features (user consent, conversation starters, adaptive cards, branding, solution management)

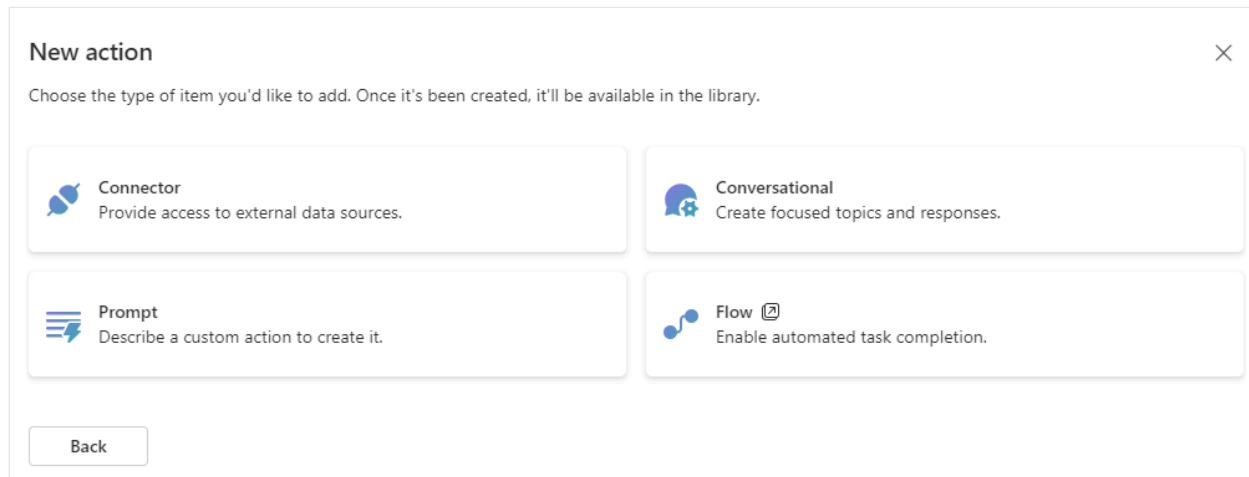
Handlers and content sources - action types

There are multiple input handlers and content sources available to serve as a basis for actions. In Copilot Studio, these sources are called *actions*. What these different types of actions have in common is the basic structure for communicating to the details of the source when submitting queries. The following action types are available:

- **Conversational:** A conversational action built in Microsoft Copilot Studio enables you to extend your agent's behavior. You can create new abilities that aren't natively in Microsoft 365 Copilot, like calling a specific API, applying sophisticated business logic, or accessing data sources that require configuration outside of Microsoft 365 Copilot's access. These abilities are configured similarly to a [standard topic](#) built in Copilot Studio.
- **Connectors:** Connector actions are used within Microsoft Dataverse to connect to other data sources. When created, the connector includes specific component actions that can be used with the connector. It's possible to use Microsoft-certified connectors or custom connectors created within an organization or tenant. Custom connectors let your action retrieve and update data from external sources accessed through APIs. Connectors make it possible to access data from popular enterprise systems such as Salesforce, Zendesk, MailChimp, and GitHub. Connectors are routinely used by makers in their Power Apps and flows.
- **Flows:** Microsoft Power Automate cloud flows can be called from a agent to perform actions, retrieve information, and automate processes across other apps

and data sources using connectors. Flows can be called from within a Microsoft 365 Copilot chat to perform actions or retrieve information across the user's environment.

- **Prompts:** Prompts are AI prompt templates created to customize the generation of text content in relation to user input. These templates can be used for various scenarios like summarization, action items, sentiment or entity extraction, translating text and much more. In addition, you can add your own Dataverse data to augment knowledge sources and get business specific answers. Prompts enable your users to use natural, plain language to get answers and perform actions with Microsoft Copilot. They use natural language understanding (NLU) to understand a user's intent and map it to an associated piece of information, data, or activity.



Not every action type is available for every Microsoft-branded Copilot agent:

[+] Expand table

Feature	Microsoft 365 Copilot	Microsoft 365 Copilot for Sales	Microsoft 365 Copilot for Service
Conversational	Available	Not available	Not available
Connector	Available	Available	Available
Prompt	Available	Not available	Available
Flow	Available	Not available	Not available

You create and configure actions in Copilot Studio. You can also create and edit some types of actions in Power Apps. Learn more in [Create a prompt \(preview\) in Power Automate or Power Apps](#).

Authentication

In order to use the content sources, you have to define authentication for the source. Sources require specific types of authentication. When you select your content, the source provides the needed prompts for authentication.

Here are some examples:

- No authentication: MSN Weather connector. This connector action only retrieves public weather data and has no secure content. As a result, the weather connector is open to be used without providing credentials.
- Flow actions: Cloud flows use the connection and connection settings configured at time of creation.
- Connector actions: If you use a connector to access a data source, it has its authentication method built in to the connector. You don't need to configure it at setup but it's necessary to provide credentials before testing or using the connection.
- Prompt action: Uses your Teams or Power Apps identity. Make sure to share the prompt to the appropriate users or groups.
- Conversational actions: Conversational actions don't need any authentication of their own. However a conversational action can be used to call a connector action, in which case the connector's authentication mechanism is invoked.

Each source has component actions already defined, but when configuring, you can select the component actions you wish to use. You can select any number of component actions to use with the source. For example, with a given content source, you could perform the following actions:

- Retrieve a list of users
- Add a new user
- Update a user's details
- Delete a user

As long as these actions are present in the source, the handler can carry out any of the actions.

Note

Every component action is available to users of the action. You cannot have role based permissions for a specific action within it. For example, if you had an order management extension that allows for retrieving a list of records, updating an order, creating an order, and cancelling an order, then every user with permissions to that extension could take any of those actions. If you need to segment access to

specific actions, you would need to create different extensions with different actions in each with appropriate security roles assigned to each action.

Input and outputs

Inputs and outputs are specific to an action and provide all of the data input options possible for an action and define the results to return. These parameters are provided by the content source (connectors, and so on) and can't be added to or removed. However, descriptions can be updated to provide a better understanding of the inputs and outputs. They're displayed as part of the setup to provide visibility into what needs to be included in a query and what is returned.

Enhanced features

Enhanced features are available when setting up extensions for some agent types.

Enhanced features include:

- Solutions
- User consent
- Conversation starters
- Adaptive Cards
- Testing

Not all enhanced features are available in every agent type.

Solutions

Solutions are essential for application lifecycle management. If saved in a solution an action can be easily moved across environments. By default, the system selects the most suitable solution for you, based on the preferred solution, or a solution where connector components are present. You can also change solution. If you don't specify a solution for your action, the system automatically creates a solution at runtime.

Add a connector action

Build a custom AI action using your connector

Give your action a unique name, description, and solution where it will be saved.

Action name *

Description *

Solution ⓘ
If no solution is selected, one will automatically be created for you.

Select a solution

Back Next Cancel

Learn more about solutions in [Application lifecycle management \(ALM\) basics with Microsoft Power Platform](#).

User consent

Turn on user consent for an action, if you would like the agent to prompt users for confirmation before executing an action. This option helps prevent unintentional actions that could affect data.

Best practices:

- For data retrieve-only actions (get, retrieve, list, and so on), turn off user consent. Such actions only retrieve a set of data and display it. They don't put the data at risk.
- For other actions (create, update, delete, add, remove, and so on), turn on user consent. Such actions can add or change data in the system, so it's a good idea to prompt users to confirm before taking action. You can always change user-consent settings for the action by editing it.

Conversation starters

Conversation starter lets you create commonly used questions that display as clickable buttons. Conversation starters execute the query immediately. They also show types of queries that can be used. These examples can help users come up with more queries on their own using successful natural language framing. Conversation starters are set at the action level rather and are optional. You can edit them at any time.

Conversation starters (optional)

Provide example questions users can use to interact with your action. [Learn more](#)

Sample 1

Status of the order Q3849ZAE2

Sample 2

Get the product warranty details

Sample 3

Order history for the customer

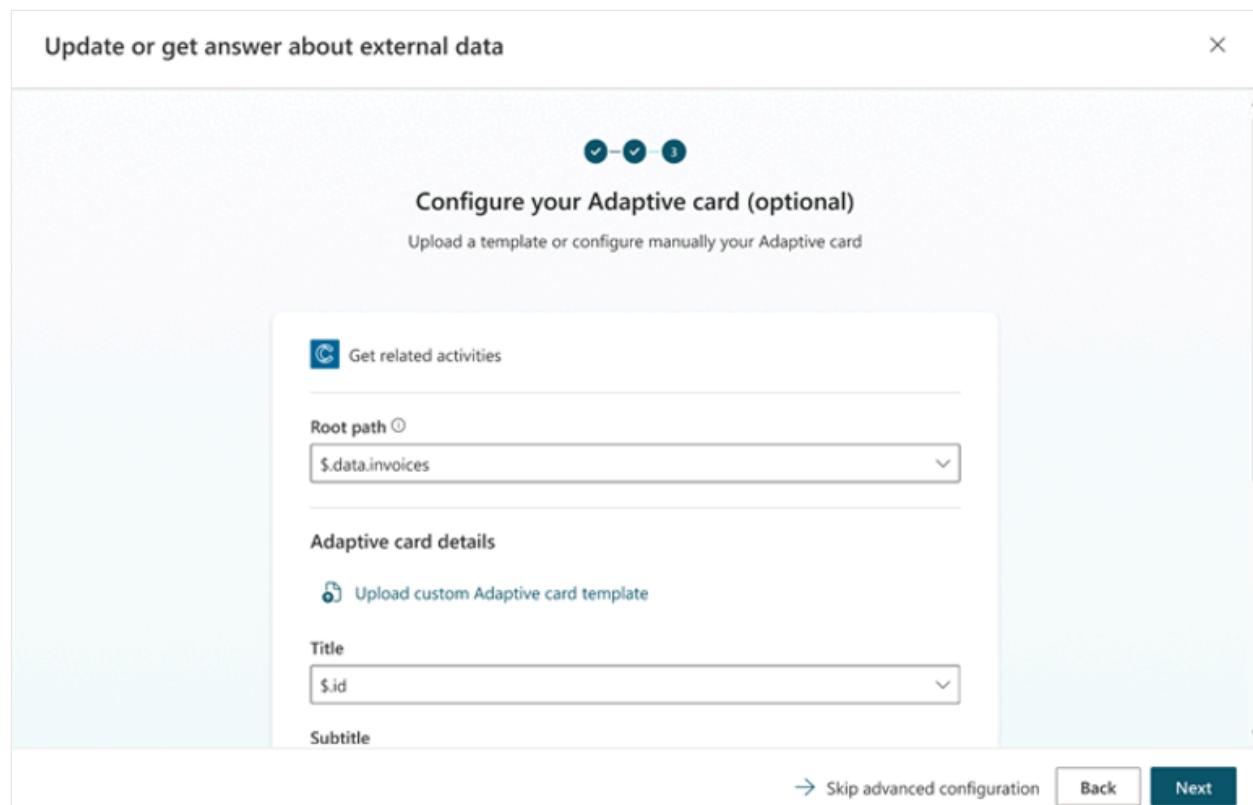
Sample 4

What is the available quantity for the product?

+ Add conversation starter

Adaptive cards

Adaptive cards are an optional component configured at the action level. You can add or modify them at any time by editing the action. Adaptive cards provide an alternative way of displaying results from a agent query.



Adaptive cards come in two types:

- Default: A basic adaptive card that can have a title, a subtitle, a body, and a button that takes the user to a URL location. It provides a more stylized view and a call to

action.

- Custom: Custom adaptive cards are created outside of the wizard and can be uploaded for inclusion in an action. These cards are more customizable and can have multiple fields, multiple buttons, and other options. You can create a custom card in Adaptive Card Designer and export it as a file for upload.

When configuring the default adaptive card, you must select a root path. The root path is a segment of the data source's configuration file and it includes all the fields that can be selected to define the title or body. There can be multiple root paths in a source file, and all available paths are available for selection. After you select a root path, you can choose the desired title and body from lists.

The screenshot shows a user interface for selecting a root path. At the top, there is a label "Root path" followed by a dropdown menu. The dropdown contains the text ".MemberResponseModel" which is underlined, indicating it is the selected option. Below the dropdown is a button labeled "Select root path". A scrollable list box is open, showing a single item: ".MemberResponseModel" preceded by a checkmark and a small downward arrow icon.

Every adaptive card shows the returned values and then lists the references used when compiling the result. Adaptive cards also allow a user to specify the reference view layout. This layout shows the references used in the query for more information. It can show a title, URL, and subtitle. There are also dropdowns that can be populated based on the root path selection.

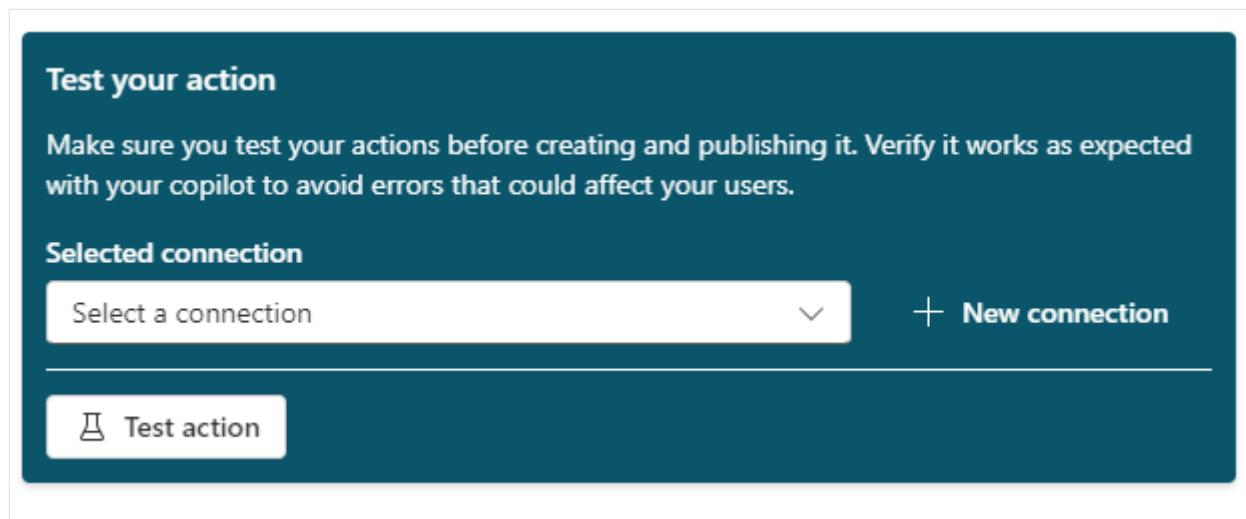
The screenshot shows a user interface for selecting a title. At the top, there is a label "Root path" followed by a dropdown menu containing the text ".DataSetsMetadataV2". Below this is another label "Title" followed by a dropdown menu labeled "Select title". The dropdown menu is open, showing several items: "Select title", ".tabular.source", ".tabular.displayName" (which is highlighted with a gray background), ".tabular.urlEncoding", ".tabular.tableDisplayName", and ".tabular.tablePluralName". A vertical scrollbar is visible on the right side of the dropdown list.

ⓘ Note

When uploading a custom adaptive card template, no edit capabilities are provided in the wizard as the custom card should already include all fields, buttons, and mapped values to be consumed by the agent. Any field left blank in the Default template doesn't appear on the adaptive card within the agent.

Testing

For connector actions, a maker can test the new action inside of Microsoft 365 by sideloading the action. This option appears on the review screen for the action. It prompts the user to select or create a connection, and then creates a sideloaded test version of the action that is available in the Microsoft 365 environment. This test version isn't visible externally and isn't published for admin approval. It's only available for the author to validate that the action works as expected.



Use actions in Microsoft 365 Copilot

Users in your tenant can use conversational and AI actions in their chats with Microsoft 365 Copilot if:

- Your Microsoft 365 tenant admin deploys the Copilot Studio app in the Microsoft 365 admin center.
- Your Power Platform environment admin has enabled the "Microsoft 365 Copilot" setting in Power Platform admin center
- The user enables the connection from within their chat with Microsoft 365 Copilot.

Deploy the Copilot Studio app in Microsoft 365 admin center (admin)

1. Sign in to the Microsoft 365 admin center with your admin account.

2. Expand **Settings** on the side navigation pane and select **Integrated apps**.
3. Go to the **Available apps** tab and select the entry **Microsoft Copilot Studio**. The app's details pane opens.
4. Select **Deploy** to enable the app in chats with Microsoft 365 Copilot.

The screenshot shows the Microsoft 365 admin center interface. On the left, the navigation pane is open with various categories like Billing accounts, Copilot, Support, Settings, and Admin centers. The 'Integrated apps' section is selected. In the main content area, the 'Available apps' tab is selected under 'Integrated apps'. A specific app, 'Copilot Studio', is highlighted. The app's details pane on the right shows the following information:

- Overview**: Shows the app is OK and type is App.
- Basic info**: Shows the app is managed by Microsoft 365 and Teams.
- Description**: Customise Microsoft Copilot for Microsoft 365 or build your own copilot experiences with Copilot Studio.
- Actions**: Includes a 'Remove app' button.
- Assigned users**: Shows the app is assigned to the entire organization.

Tenant admins can enable or disable:

- Showing Conversational AI/topic actions and Prompt actions in the Microsoft 365 Chat app using the "Copilot Studio" app.
- Showing Flow actions in the Microsoft 365 Chat app using the "Power Automate" app.
- Connector actions and API actions in the Microsoft 365 Chat app using the associated apps (the app name will be the connector or action name). Additionally, custom connector actions are available in the Teams store for users to install.

Enable or Disable Microsoft 365 Copilot in Power Platform admin center (admin)

1. Go to the Power Platform admin center using your admin account.
2. Go to **Environments** and select the environment from the list.
3. Select **Settings**.
4. Under **Product**, select **Features**

5. Find the setting for Microsoft 365 Copilot and toggle **on** (or **off**) to allow integration of Copilot agents from this environment.
6. Scroll down and select **Save**.

ⓘ Important

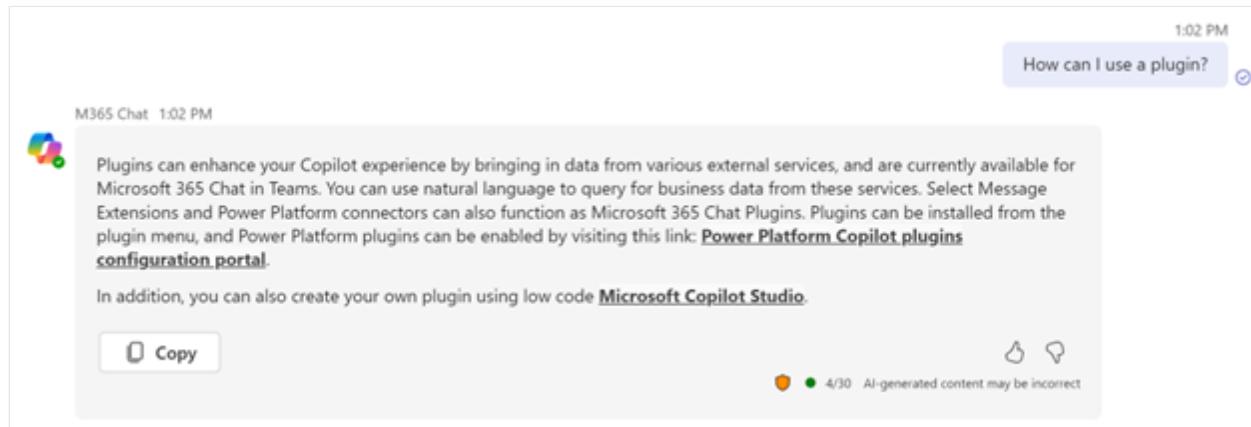
In this release, there are some limitations:

- Integration with Copilot Studio and Power Automate is enabled by default. That means actions created in Copilot Studio will show up for users under "Copilot Studio" or "Power Automate" even if the tenant admins have not explicitly deployed the app for users. This might change in the future.
- Note that, as an admin, you can still block actions from showing up for your users by blocking the app in Microsoft 365 Admin Center.
- In the future, customers can manage the setting for integration with Microsoft 365 Copilot at scale using Environment groups.

Enable the connection in Microsoft Copilot

Microsoft 365 Copilot users need to enable actions before they use them in chat. Available actions can be enabled through the actions menu or through the Power Platform actions configuration portal.

They can get to the portal by asking about actions in their chat with Microsoft 365 Copilot. They receive a short summary about actions and a link to the portal:



Users can also ask directed questions about data connections or how to perform cross-organizational tasks, for example they could ask:

- *How can I use an action?*
- *Tell me about Power Platform Copilot actions*

- How can I get data from an external system?
- How can I get data from Salesforce?

Share actions

By default, actions are only visible and usable in Microsoft 365 Copilot by the person who authored them.

However, the action author can share their actions in the portal where they created them. For example, you can share an AI Builder prompt from the **AI prompts** page by selecting **Share** for the prompt. The same applies for Power Automate flows (from the **Flows** page in Power Automate) or for custom connectors from the **Custom connectors** page.

Related content

[] Expand table

Article	Description
Create conversational actions for Microsoft 365 Copilot	Extend <i>Microsoft 365 Copilot</i> by creating actions that can respond to a user's question with extra data.
Extend a Copilot agent with prompt actions (preview)	Extend a <i>Microsoft-branded Copilot agent</i> with a prompt action.
Extend a Copilot agent with flow actions (preview)	Extend a <i>Microsoft-branded Copilot agent</i> by creating a flow action.
Extend a Copilot agent with connector actions (preview)	Extend a <i>Microsoft-branded Copilot agent</i> by creating a connector action.
Extend Microsoft 365 Copilot with Copilot agents	Extend Microsoft 365 Copilot with a Copilot agent.
Use actions with custom agents (preview)	Extend a <i>custom copilot</i> by creating complex topics for use in copilots you build in Copilot Studio, without doing anything complex.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Extend agent capabilities with Copilot connectors

Article • 11/19/2024

[This article is prerelease documentation and is subject to change.]

Copilot Studio makes it easy to extend the capabilities of agents with actions, Copilot connectors, and sources of data. You can take an existing agent, for example Microsoft 365 Copilot for Sales, which works with your organization's sales data, and extend it using data from an extensive ecosystem of connectors provided by Microsoft.

Alternatively, you can extend a custom agent with a custom connector to use sources such as custom data sources and APIs within your organization. For example, you could:

- Retrieve a list of sales opportunities
- Get information from your order management system about orders associated to the customer
- Check the weather at your customer's current location if you're planning a site visit

Copilot connectors help customers to easily discover and bring multiple enterprise data sources, by leveraging [Microsoft Graph connectors](#) and Power Platform connectors, to expand the capabilities of agents and agent extensions. Using agent connectors, users can add knowledge from their enterprise systems to agents through a no-code experience, and perform actions on these enterprise systems.

Agent connectors include data sources within the productivity cloud in Microsoft 365, business data in Dynamics 365, and analytical data in Microsoft Fabric, as well as non-Microsoft enterprise data sources. Through agent connectors, you can now ground your agent responses with multiple enterprise data sources for more relevant and reliable responses from agents and agent extensions.

- For more information on all the supported knowledge sources using Copilot connectors, see [Knowledge sources overview](#).
- For more information on Copilot connectors that have enterprise data added using Microsoft Graph connectors, see [Add Enterprise data as a knowledge source](#).

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 the systems you're connecting to
- Assistance from the administrator of your agent to enable the extension once it's published

Related content

 Expand table

Article	Description
Create conversational actions for Microsoft Copilot	Extend a <i>Microsoft Copilot</i> by creating actions that can respond to a user's question with extra data.
Extend Microsoft Copilot with connector actions (preview)	Extend a <i>Microsoft Copilot</i> with connector actions.
Use actions with custom agents (preview)	Extend a <i>custom agent</i> by creating complex topics for use in agents you build in Copilot Studio, without doing anything complex.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create conversational actions for Microsoft Copilot

Article • 11/19/2024

ⓘ Note

Actions were previously known as plugins. Some articles and screenshots may refer to actions while we update documentation and training content.

Microsoft 365 Copilot provides functionality to engage with content and resources from across your organization. In some situations, answers and interaction with external systems are required. With Copilot Studio, you can author a conversational topic and publish that as an action. Once your tenant admin approves the action, it can be added to your organization's Microsoft 365 Business Chat experiences.

You can create a conversational action that receives the user's query (stored in the **Activity.Text** variable), processes it by completing one or more operations, and then sends one or more responses back to the user in the chat.

ⓘ Important

You can't use conversational actions created in Copilot Studio as an action with [custom agents](#).

What is a conversational action and how does it work?

The conversational system for Copilot uses orchestration logic and large language models to help the user achieve their goals. This system interprets the user's request, determines the best course of action, and then activates the appropriate system, action, component, or series of components. Finally, the system generates a result and returns it to the user.

A conversational action built in Copilot Studio enables you to extend your app's behavior. You can create new abilities that aren't natively in Copilot, like calling a specific API, applying sophisticated business logic, or accessing data sources.

A traditional component model creates a hard-coded integration point with the copilot. For conversational actions, the maker gives descriptive text to the copilot orchestrator about the action's abilities. The orchestrator can then make its own decision about when the action would be most useful and carry out the logic defined for the action.

Prerequisites

- Customers must have a license to use Microsoft 365 Copilot.
- Your Power Platform tenant must be configured to enable publishing actions, as described in the Microsoft 365 Copilot article [Build plugins using Microsoft business applications](#).
- The Microsoft 365 tenant admin needs to enable the [Microsoft Copilot Studio](#) Integrated Apps category in the Microsoft Admin Center.

ⓘ Note

The tile in the Microsoft Admin Center was previously **Dynamics 365 and Microsoft Copilot Studio**, and was split into two, **Dynamics 365**, and **Microsoft Copilot Studio**. Only the latter is required for this feature to work.

ⓘ Important

In this release, there are some limitations:

- Your topic will not be able to interact (ask questions of) the user, and can only process the incoming query from the user (in the variable **Activity.Text**) and send a response.
- Responses will be limited to text. No media or adaptive cards will be supported.
- In this preview, Copilot Studio can't authenticate the user, so SharePoint sources for generative answers won't be functional.
- This release is limited to English language copilots.
- All users of Microsoft 365 Copilot within your organization will be able to interact with the action, the action will not support authorizing the user itself.
- Copilot Studio integration with Microsoft 365 Copilot is enabled by default. That means that actions created in Copilot Studio will show up for users under "Copilot Studio" or "Power Automate" even if the tenant admins have not explicitly deployed the app for users. This might change in future.

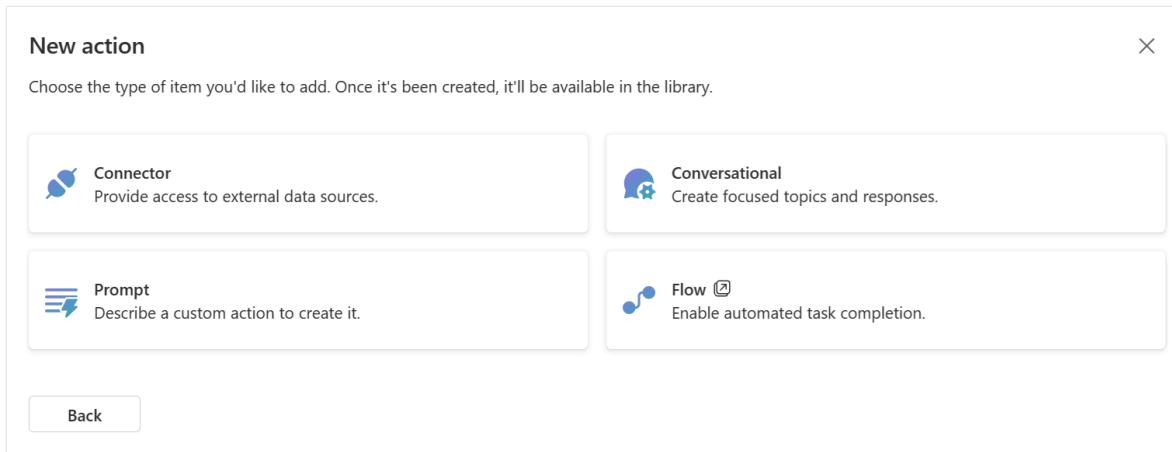
An admin can always block actions from showing up for your users by blocking the app in Microsoft 365 Admin Center.

Create a basic conversational action to publish to Microsoft Copilot

Your topic, when triggered within a chat, uses the system variable **Activity.Text** as an input. **Activity.Text** represents the user's question. The chat processes the question text—for example, passing it to generative answers—and then returns the result as one or more messages. These messages could include the response that comes from generative answers or messages from message nodes. Multiple messages can be included and sent separately to form the output.

The following steps help you create a new conversational action within Copilot Studio. You author basic capabilities and publish them to the Admin Center.

1. On the side navigation pane, select **Agents**.
2. Select Microsoft 365 Copilot. The **Overview** page for Microsoft 365 Copilot appears.
3. Select the **Actions** tab.
4. Select **+ Add action**. A menu of available action types appears.
5. Select **New action**. You are presented with a menu of available action types for the new action.



6. Select **Conversational** to create a conversational action. A configuration window appears letting you set some basic configurations for the conversational action.

7. Choose a name and primary language. Optionally, you can also specify a solution and schema name to use. Otherwise, a default solution and schema name are selected for you.
8. Select **Create** to proceed. Your new conversational action is created. This will take several seconds. When it is done, you are dropped into the conversational authoring canvas.

With your conversational action created, you can next proceed to refine the action using the topics editor. When you are done, you can publish your action.

Refine your conversational action example

Let's look at an example of refining a conversational action.

You can use the topic editor to define the logic for the conversational action. The process is similar to the process for configuring any other topic. This could include calling an action, initiating a generative answers request, or sending a message to the user.

Conversational actions only support logic that doesn't require user interaction.

For this example, let's use a generative answers node to search through web content in a specified web site and provide an answer to the user in the Copilot.

1. Select **Edit the topics** under **Refine** to edit your conversational action.

Conversational action 2

 [Edit details](#)

Next steps: Improve your actions

 [Refine](#)

Your action is configured as a conversational topic.

[Edit the topics](#)

 [Publish](#)

Publish your action to Microsoft Copilot to share with others in your organization.

[Go to publish](#)

Summary

Last modified

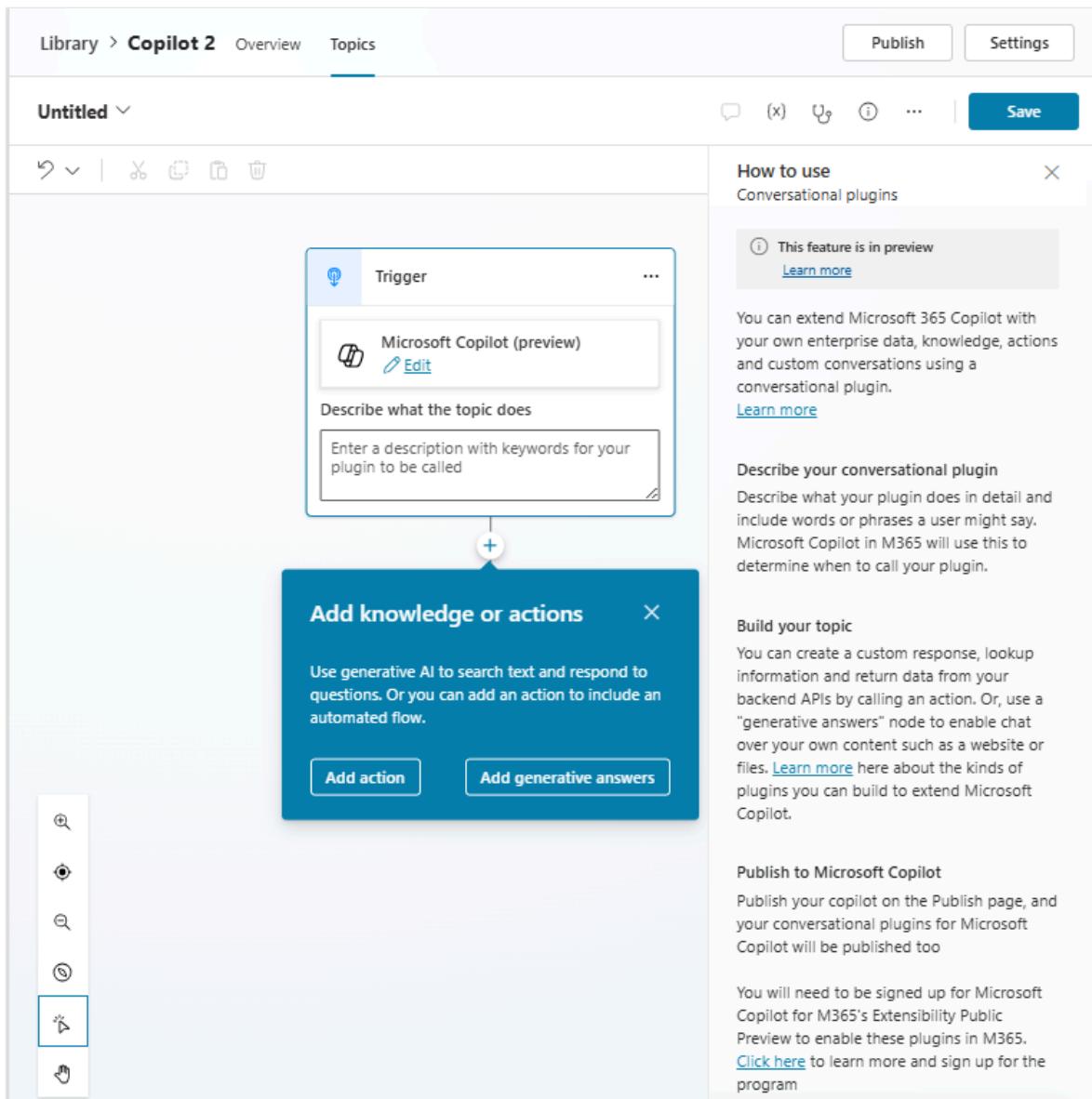
26 minutes ago

Last published

Never

[Go to publish history](#)

You are presented with some guidance on how to configure the conversational action.



2. Before authoring further, set the name of the topic in the top panel above the authoring canvas. This name is visible to the user and admin in the agent and Microsoft 365 admin center respectively.
3. In the **Trigger** node description, provide a clear description of how the conversational action can help the user and what it can do. Unlike other trigger node descriptions, this one has functional purpose and is used by Microsoft 365 Copilot to determine whether to invoke your action or not. Describe what your action does for the user and what it returns.
4. To add the generative answers node, select the plus on the canvas under the **Trigger** node, then select **Generative answers** under the **Advanced** tab of the popup menu.
5. For the **Input** field, invoke the variable picker by selecting the arrow in the **Input** box.

6. Choose the system variable **Activity.Text** which represents the user's last query to Microsoft 365 Copilot. This query was what the user asked in that caused Microsoft 365 Copilot to invoke the action.
7. Select the **Data Sources** editor in the center of the **Create generative Answers** tile. A **Create generative answers properties** configuration window appears.
8. Under **Classic data** and **Data sources** within the **Public websites** field, enter a link to a public website that has content you want for your copilot. This web content gets used to answer user questions. Alternatively, under **Knowledge sources**, select **+ Add knowledge** to open the **Add available knowledge sources** dialog to add a public website there. For more information on adding knowledge, see [Add knowledge to an existing copilot](#).

 **Note**

The content source must be a public website. Make sure Bing can find and index this website.

9. Once the link is added (don't forget to hit "+" after entering the URL), close the properties panel.
10. Optionally, add one or more messages to the user. You can use a message node, or more generative answer nodes, as needed.
11. If your action needs to receive input parameters from a connector or pass output parameters to a connector, you can [use input and output variables](#).
12. Finally, select **Save** to save your conversational action.

Publish your conversational action to Microsoft 365 Copilot

Publishing your conversational action creates a new action in the Dataverse registry for your tenant. Once available there, your tenant admin needs to approve your action to be available to users.

Under the Copilot agent, your conversational action is labeled as *Conversational action name* in the library. With this in mind, make sure to choose your conversational action name for readability.

To publish your conversational action:

1. First, create, edit, and save your conversational action as described in the prior sections.
2. On the action **Overview** page, select **Publish**.
3. Select **Publish**. A message appears prompting you to confirm your intention.
4. Finally select **Publish** again. The publish should complete quickly. The actual availability in the Microsoft 365 admin center can take a few minutes. If an error occurs, try again. If the error continues, file a support ticket in the Power Platform admin center.
5. Your admin can find the **Copilot Studio** integrated app in the [Microsoft 365 admin center](#) under **Settings**, then **Integrated Apps**. Deploy/block is at this category level only.

ⓘ Important

In this release, Copilot Studio integration with Microsoft 365 Copilot is enabled by default. That means, actions created in Copilot Studio will show up for users under "Copilot Studio" or "Power Automate" even if the tenant admins have not explicitly deployed the app for users. This may change in the future.

An admin can always prevent conversational actions from showing up for Microsoft 365 Copilot users by blocking the app in Microsoft 365 admin center.

6. Once your tenant admin approves the integrated app, it should appear in the user's list of actions in their Copilot experience UI.

Once the action is enabled, your user can pose questions in Copilot chat, and see the action invoked and returning the most relevant results.

ⓘ Note

To remove your conversational action from the catalog, in conversational action details, clear the **Status** checkbox, save your conversational action, and then republish, which will remove it from the Dataverse registry and Microsoft 365 Copilot app catalog.

Considerations

Interacting with Microsoft 365 Copilot is likely to be a different experience than you might have had integrating with previous systems.

- Microsoft 365 Copilot will always rewrite your answers with its own voice. It's not possible in this preview to have the content passed through unchanged to the customer.
 - The description of your conversational action is critical to how reliably it will be invoked. The description teaches the orchestrator what your action is good at and what answers it can provide. Be sure to use clear prose when writing the description, and consider experimenting with changes to get the best outcome.
-

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

Extend Microsoft 365 Copilot or Copilot agent with prompt actions (preview)

Article • 11/19/2024

ⓘ Note

Extending with prompt actions is only available for some Microsoft-branded agent experiences.

Open add a prompt action wizard

1. Sign in to [Copilot Studio](#) and select Agents on the side navigation pane.
2. Select the Copilot agent you want to add a prompt action to.

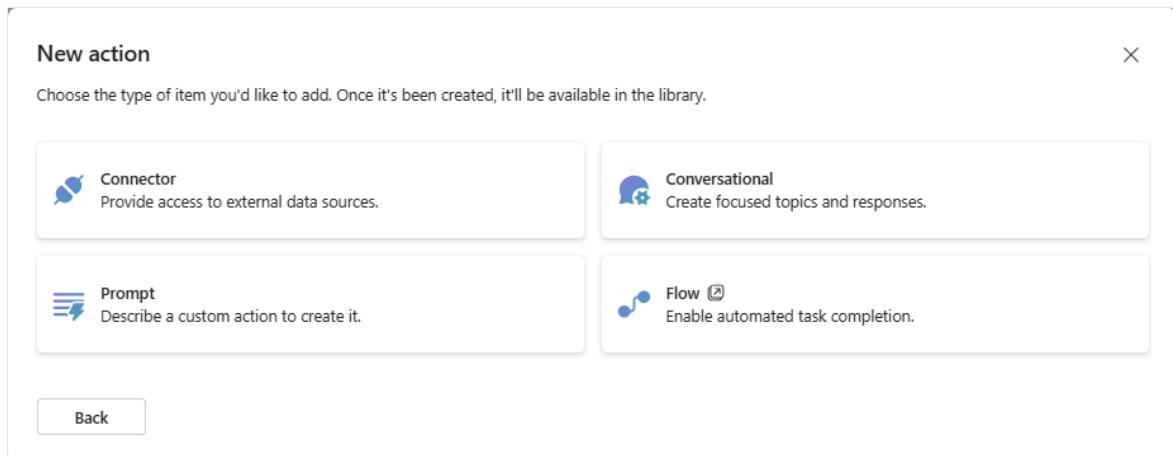
ⓘ Note

For this example, we've chosen **Microsoft 365 Copilot**.

The agent overview page appears.

3. Select the **Actions** tab and then select **+ Add action**.

A New action menu appears.



4. Select **Prompt**.

The **Add a prompt action** wizard opens.

Create a prompt action

In the wizard, follow these steps in sequence.

Action details

1. Enter a name and description for the action.

These details are used by the copilot to perform a semantic matching of the action with the user query.

2. Under **Solution**, a dropdown lists all solutions available within the current environment. If you defined a preferred solution, then it's preselected, otherwise select the solution you want to use or leave it for the wizard to autocreate a new solution for this action.

The screenshot shows the 'Add a prompt action' wizard interface. On the left, a vertical navigation bar lists six steps: 'Action details' (selected), 'Create prompt', 'Select action parameters', 'Review and test', and 'Publish'. The main panel is titled 'Build a custom AI action using a prompt' and contains instructions: 'Provide your actions details and select on which solution you wish to save it.' It has two input fields: 'Action name *' with placeholder 'Enter prompt action name' and 'Description *' with placeholder 'Enter prompt action description'. Below these is a 'Solution' section with a note: 'If no solution is selected, one will automatically be created for you.' A dropdown menu labeled 'Select a solution' is shown. At the bottom right are 'Next' and 'Save and Close' buttons.

3. Select **Next**.

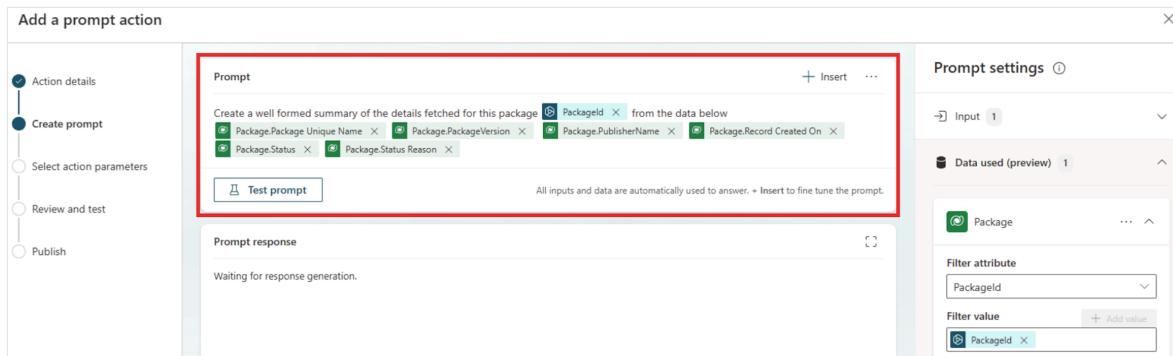
Create a prompt

Describe the task you want your copilot to perform using natural language.

1. In the **Prompt** field, enter instructions and data required to generate response-pertinent details for the use case.

To enable the Copilot agent to pass important pieces of information at runtime to the prompt action, add input variables. Write your own instructions or begin from one of the existing prompt templates.

Examples of prompts are *Summarize text*, *Extract information from text*, and *Classify text*.



2. Try out your prompt by selecting **Test prompt**.

Your prompt is generated and appears in the **Prompt response** field. You might need to revise and iterate on your prompt to meet your needs.

ⓘ Note

Make sure your input variable has test data before you test.

3. When you're satisfied with the prompt response, select **Save custom prompt**.

Select action parameters

This step ensures descriptive input and output names are selected such that Copilot is able to prefill it with information from user utterance and consume the output from the prompt action.

1. In the **Input parameters** section, add the input variables that the prompt action needs to generate the response.

These are the variables that you want to pass to the prompt action at runtime. You can add multiple input variables.

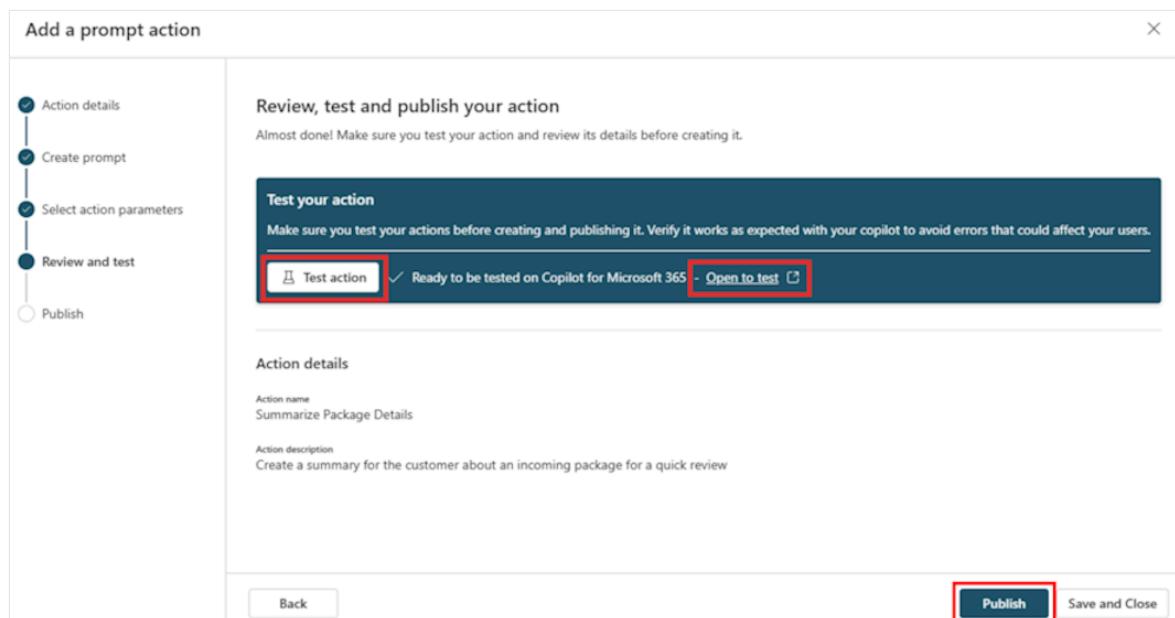
2. In the **Output parameters** section, add the output variables that the prompt action generates.

These are the variables that you want to receive from the prompt action at runtime.

Review and test

This step presents the opportunity for a copilot developer to test the prompt plugin by side-loading it in the Microsoft 365 Copilot.

1. To test the prompt plugin, select **Test action**.
2. Select **Open to test** to open the Copilot chat in a new window. Open the flyout to see the action and toggle it on.
The action appears with a beaker icon under the connector chosen with format *Test-Action name*.
3. You can now test the action by trying the prompts you expect a user to use. You can also use the `-developer on` command to get more information on prompt execution. Once you're satisfied with the results, return to the Copilot Studio window.
4. After you test the prompt and verify that it performs as expected, select **Publish**.



Publish

1. On the **Review and test** step, select **Publish** to see the confirmation message.

Your prompt action is now published to **Microsoft 365 Copilot**. It might take a few minutes for your action to show up in copilot experiences. Meanwhile, you can review information about using and administering actions in [Microsoft 365 Copilot](#).

After you create your action, you need to [enable it for use](#).

For more information about building and using prompt plugins, see [AI Builder prompts](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Extend Microsoft Copilot with flow actions (preview)

Article • 11/19/2024

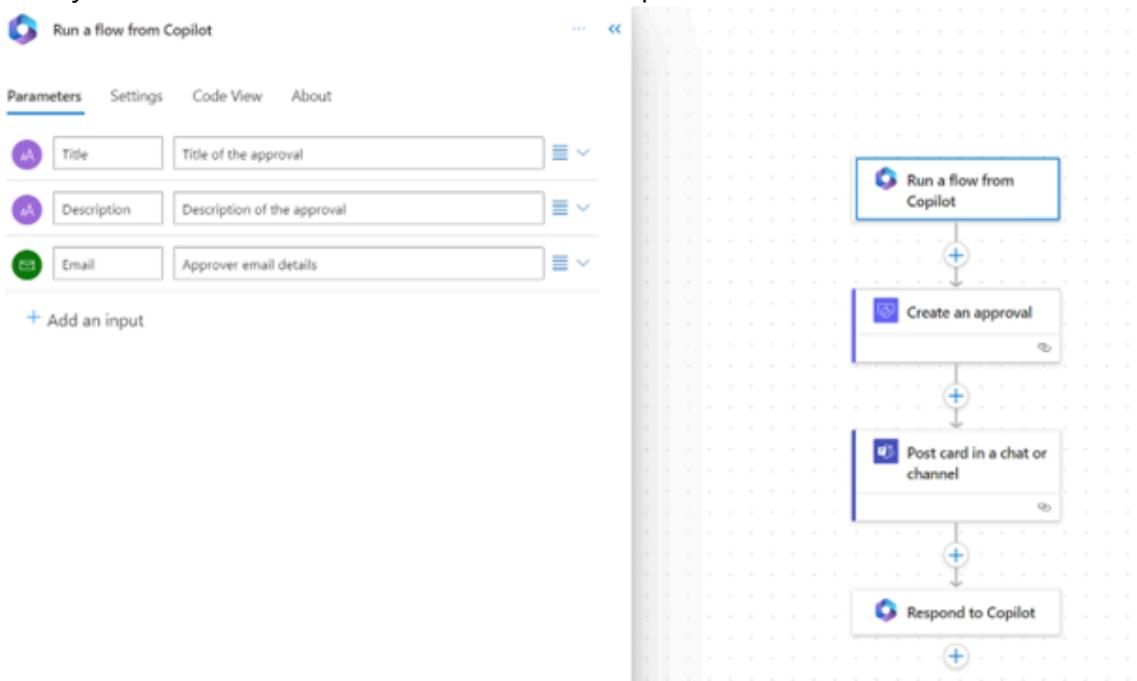
ⓘ Note

Extending with flow actions is not available for all types of Copilots.

Power Automate flow actions let you define flows that can be called from AI surfaces in Power Platform. Flow plugins use the new **Run from Copilot** trigger and **Respond to Copilot** action to define custom processes that can be invoked with natural language.

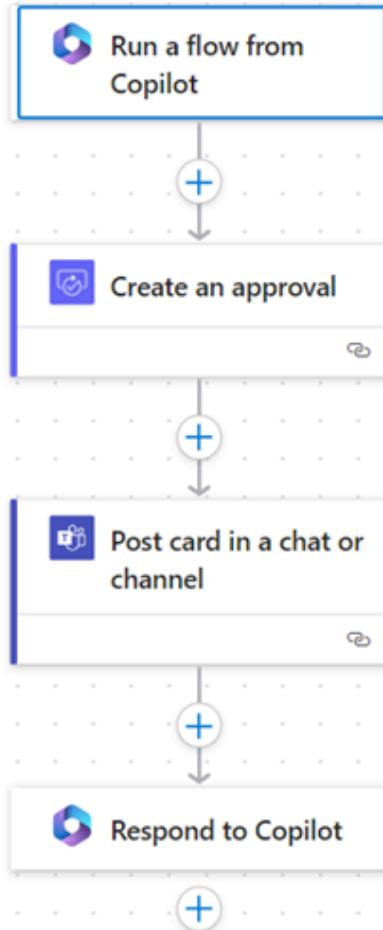
To create a flow action:

1. Sign in to [Copilot Studio](#) and select **Agents** on the side navigation pane.
2. Select one of the agents to extend. The **Overview** page for the agent appears.
3. Select **+ Add action** in the **Overview** page or on the **Actions** page.
4. Select **Flow**. The flow editor automatically opens with the **Run a flow from Copilot** trigger and **Respond to Copilot** action present.
5. You can add inputs to your trigger for information you might want to collect from your users. Provide helpful text descriptions for each input so the AI can correctly identify what to extract from a user's data or inputs.



6. Add actions to your flow such as the **Create an approval** and **Post a message to Teams chat** to create an approval process.

7. Optionally add parameters to the **Respond to copilot** action for any output you want to send back to the copilot.



8. Save your flow with a meaningful flow name and test or run it at least once. The AI uses the title and description of the flow to determine when to invoke the flow. Ensure your flows run correctly, as only tested flows show up as available actions in Microsoft Copilot.

After you create your action, [enable it for use in Microsoft Copilot](#).

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Extend Microsoft 365 Copilot or Copilot agents with connector actions (preview)

Article • 12/16/2024

Connector actions let you define connectors that can be invoked from AI surfaces in Power Platform. For connector actions, you need to identify the actions enabled for use in the Copilot. You can also capture the information that large language models require to effectively identify and utilize the plugin, such as summary and description fields.

Prerequisites

A few prerequisites are needed in order to extend Microsoft-branded copilot experiences with connector actions.

General prerequisites

For all Copilot types, you must have maker-level credentials, a copilot license, and a Microsoft Teams license.

[!NOTE] Actions were previously also called plugins. These actions can be used in agents. Previously created plugins have now evolved to agents. See the [Microsoft 365 Copilot blog](#) for more information.

Microsoft 365 Copilot sales agent and Microsoft 365 Copilot service agent prerequisites

For Microsoft 365 Copilot service agent and Microsoft 365 Copilot sales agent, you must be signed in to an environment that contains the agent that you wish to extend.

Microsoft 365 Copilot sales agent prerequisites

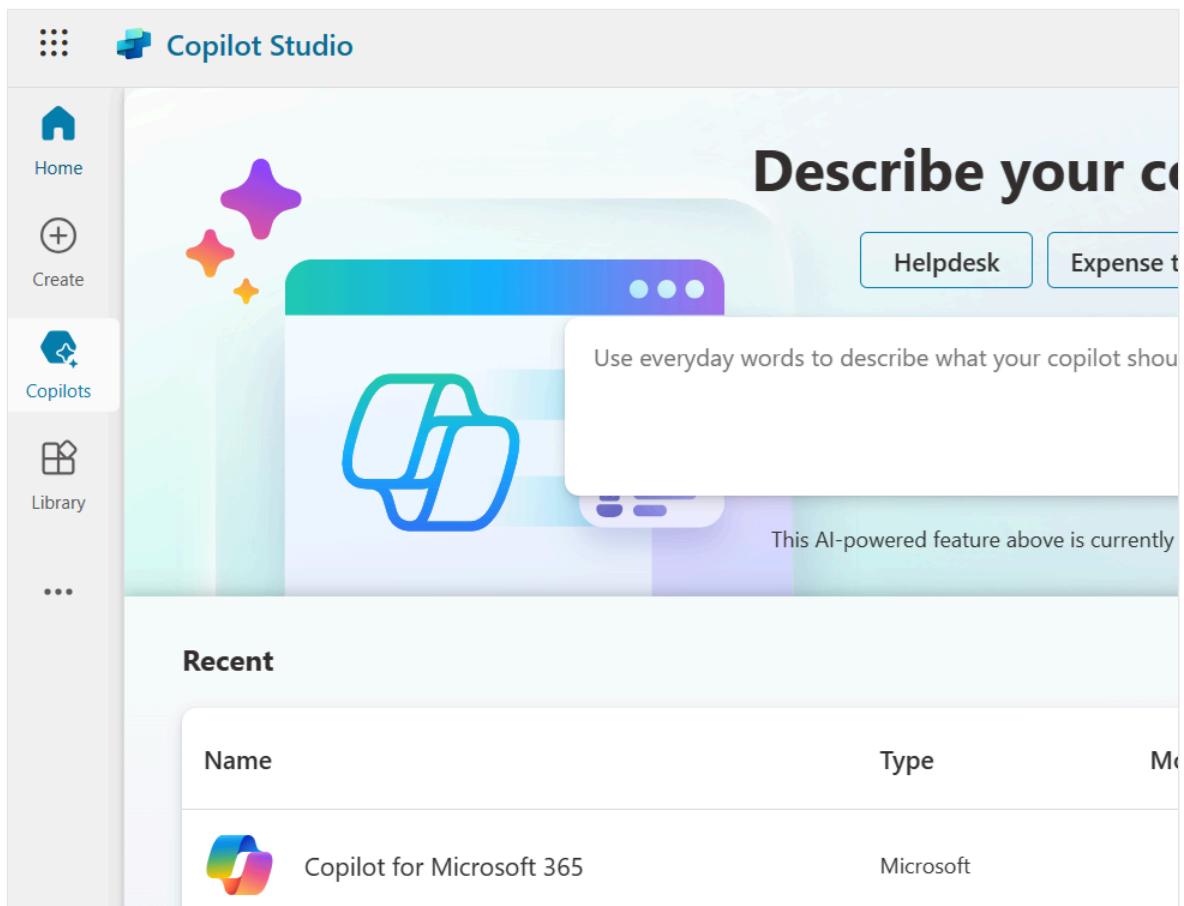
- Salesforce customers must be signed in to their Dataverse environment. This environment is usually named msdyn_viva.
- You must sign in to Microsoft 365 Copilot sales agent through Outlook in the environment you wish to extend at least once before extending.

Add a connector action

Adding a connector action to a Microsoft-branded copilot involves a few steps. The following sections provide guidance on how to add a connector action to Microsoft 365 Copilot, Microsoft 365 Copilot for Sales, and Microsoft 365 Copilot for Service.

Select the agent and choose connector action

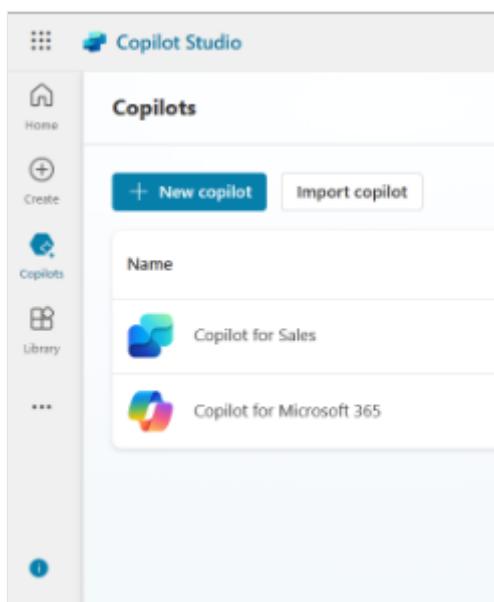
1. On the home page, select **Agents**.



The screenshot shows the Copilot Studio interface. The left sidebar has icons for Home, Create, Copilots, Library, and three dots. The main area has a 'Describe your copilot' input field with 'Helpdesk' and 'Expense' buttons. A message says 'Use everyday words to describe what your copilot should do'. Below it, a note says 'This AI-powered feature above is currently experimental'. The 'Recent' section lists 'Copilot for Microsoft 365' as a Microsoft type agent.

Name	Type	More
Copilot for Microsoft 365	Microsoft	[...]

2. On the agents list in the left hand navigation, select the Copilot agent you wish to extend.



The screenshot shows the 'Copilots' page in Copilot Studio. The left sidebar has icons for Home, Create, Copilots, Library, and three dots. The main area has a 'New copilot' button and an 'Import copilot' button. It lists two agents: 'Copilot for Sales' and 'Copilot for Microsoft 365'.

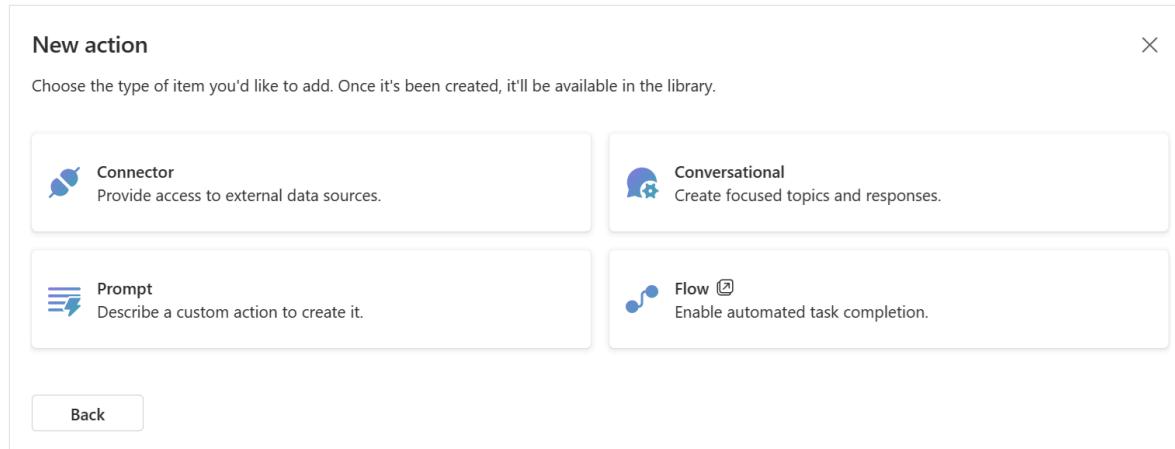
Name
Copilot for Sales
Copilot for Microsoft 365

The overview page for the agent is displayed.

3. Select the **Actions** tab. You can see a list of the actions already present within the Copilot.

4. Next, select **+ Add Action**.

You're provided a list of all supported action types for the Copilot agent.



5. Select **Connector** to extend with a connector action.

The **Select your connector** page is displayed. You're provided a list of available connector types to select from.

Connectors connect copilots to different services. The list includes all certified connectors offered by Microsoft.

Depending on the Copilot, the list might also include custom connectors that are present in the current environment. A search bar is available to help find the connector that you wish to use.

ⓘ Note

At this time, custom connectors cannot be added Microsoft 365 Copilot service agent.

Select connector type and select a solution

1. Find your connector type using the search bar and select it.

Add a connector action

Select your connector

Select the connector you want to use to get information from external sources.

Search

- SharePoint
- OneDrive for Business
- Dynamics 365 (deprecated)

The Build a custom AI action using your connector page is displayed.

2. Write a name and description for your connector.

The connector name must be unique. By default, it takes on the name of the connector type that was chosen. You should add more defining information to ensure it's both a unique name and that it's easy to discern what it does for users who want to enable it later. Consider including in the description the types of actions it allows like "Mail service allows you to retrieve and create messages."

Add a connector action

Build a custom AI action using your connector

Give your action a unique name, description, and solution where it will be saved.

Action name *

Description * ⓘ

SharePoint helps organizations share and collaborate with colleagues, partners, and customers. You can connect to SharePoint Online or to an on-premises SharePoint 2016 or 2019 farm using the On-Premises Data Gateway to manage documents and list items.

Solution ⓘ

If no solution is selected, one will automatically be created for you.

Select a solution

Back

Next

Save and Close

The description might be provided if present in the connector you selected. The description is used by the Copilot agent to identify when the action can be used to get responses back for the user request. Make the description as detailed and

descriptive as you can. Provide specific details using terms you expect users to use when calling on the action. To continue with the mail service example, you might include something like:

"Mail service provides the ability to get lists of existing messages, retrieve folders where messages are stored, list titles of messages and folders, get messages, update message titles and bodies, and set priority for messages."

The description lets the Copilot agent know to use the action without being explicitly instructed, allowing for a more natural language experience.

3. 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, and why they benefit you, see [Solution concepts](#).

Solution ⓘ

If no solution is selected, one will automatically be created for you.

Select a solution

▼

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 action name and default publisher. Storing your action in a solution lets you move it easily across environments.

ⓘ Note

You do not see default solution or 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](#).

4. Choose a solution and select **Next**.

The **Select and configure your plugin action** page is displayed. This page shows all the actions that are available to add for your connector. You can select and configure any actions you want your copilot to support, one at a time. A search bar is included to help you find the action you wish to select. There's no limit to the number of actions you can select for the connector.

Add a connector action

X

- Select connector
- Action details
- Select Actions**
- Configure action
- Select action parameters
- Review and test
- Publish

Select and configure your plugin action

Please select the required action for Copilot for Microsoft 365.

Available actions (124)

Search	
	List permissions for a file or folder
	Remove permission from a file or folder
	Update permission for a file or folder
	Add permission for a file or folder
	Create sharing link for a file or folder
	Render item attachment thumbnail, with key-as-parameter syntax
	Render item thumbnail

Back

Next

Save and Close

Add actions to the connector

Next, you need to select from the available actions to add to the connector.

ⓘ Note

Connectors can only execute the actions you specifically select and configure here. If an action isn't listed, then it's not available. It's not possible to add or remove actions from a connector from within the Copilot Studio actions wizard.

1. Select an action to proceed. The **Configure your plugin action** page is displayed for that chosen action.

Add a connector action

Configure your plugin action

Select the connector action you wish to leverage and enter details to configure your plugin action.

Action name *
Get lists

Action description * ⓘ
Gets SharePoint lists from a site.

Back **Next** **Save and Close**

2. Provide a name and description for your action. The name doesn't have to be unique in this case. These values are most likely going to be populated for you from the connector. However, as with the connector description, you should provide a detailed description so that the agent knows when to invoke the specific action. The description helps allow a more natural language approach to selecting the action to execute. For example, if an action is "List message folders," you might use:

"List message folders allows you to get a list of all message folders available from the mail service. This action retrieves all folders, places them into a list view, and returns them to you."

The action description provides the context clues "list message folders," "get," "all," "retrieve," "list," "mail service," "return," "folders," and "view." Your copilot can use these context clues to identify when the action is the right one for a user request.

3. After you complete the page, select **Next**. The **Review your action's parameters** page appears.
4. Select **Next** to proceed.

This action takes you back to the **Select and configure your plugin action** page. You can see the new action you selected.

Add a connector action

The screenshot shows the 'Select and configure your plugin action' step. On the left, a vertical navigation bar lists steps: 'Select connector' (checkmark), 'Action details' (checkmark), 'Select Actions' (selected), 'Configure action' (radio button), 'Select action parameters' (radio button), 'Review and test' (radio button), and 'Publish' (radio button). The main area is titled 'Select and configure your plugin action' with the sub-instruction 'Please select the required action for Copilot for Microsoft 365.' Below this is a section titled 'Selected actions (1)' containing a single item: 'Get lists'. A large section titled 'Available actions (123)' follows, showing five items: 'List permissions for a file or folder', 'Remove permission from a file or folder', 'Update permission for a file or folder', 'Add permission for a file or folder', and 'Get lists' (which is also listed under 'Selected actions'). At the bottom are 'Back', 'Next', and 'Save and Close' buttons.

5. Select another action and repeat the previous action configuration steps if desired.

Alternatively, select **Next**. The **Review, test, and publish your action** screen is displayed.

Add a connector action

The screenshot shows the 'Review, test and publish your action' step. The left navigation bar shows steps: 'Select connector' (checkmark), 'Action details' (checkmark), 'Select Actions' (selected), 'Configure action' (radio button), 'Select action parameters' (radio button), 'Review and test' (selected), and 'Publish' (radio button). The main area has a title 'Review, test and publish your action' with the sub-instruction 'Almost done! Make sure you test your action and review its details before creating it.' Below this is a dark teal box titled 'Test your action' with the sub-instruction 'Make sure you test your actions before creating and publishing it. Verify it works as expected with your copilot to avoid errors that could affect your users.' It contains a 'Selected connection' dropdown set to 'Select a connection' and a '+ New connection' button, and a 'Test action' button. Below the teal box is an 'Action details' section with 'Action name' set to 'SharePoint-Contoso Marketing' and 'Action description' empty. At the bottom are 'Back', 'Publish', and 'Save and Close' buttons.

The **Review, test and publish your action** page displays the details of all of the configuration steps up to this point. You can review them, and then test your action before publishing.

Test

With your connector action configured, you can now test it.

⚠ Note

At this time Microsoft 365 Copilot service agent does not allow for testing from inside of the connector action creation wizard. When this feature becomes available you will be able to test the plugin from this screen. For now users must publish the plugin and then they can validate it is functional within the Microsoft 365 Copilot service agent directly after it is enabled by an administrator.

1. Select your existing connection from the dropdown or select **+ New connection** if you don't have a connection.

If you select **+ New connection**, a wizard opens where you can add details to build the connection. When you're done, you can select the new connection from the dropdown.

2. When you're done selecting a connection, select **Test action**.
3. When you see a message that the action is ready to test, select **Open to test** to open the copilot chat in a new window. Open the flyout to see the action, and toggle it on.

The action appears with a beaker icon under the connector chosen with format *Test-Action name*.

4. You can now test the plugin by trying the prompts you expect a user to use. You can also use the `-developer on` command to get more information on prompt execution. Once you're satisfied with the results, return to the Copilot Studio window.

Publish

1. Once you're done testing, select **Publish**.

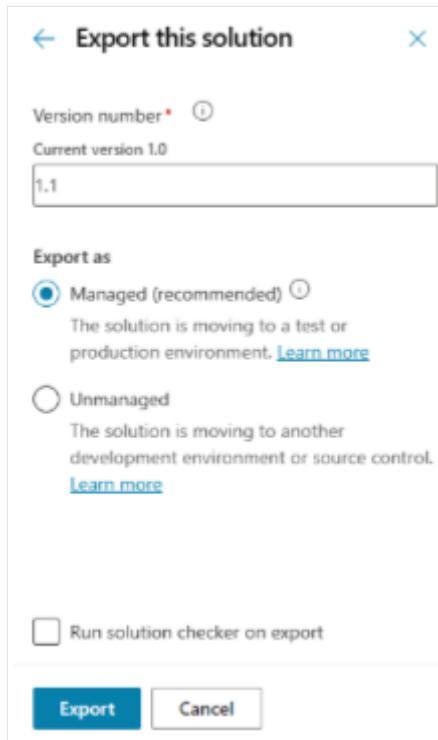
A page is displayed showing your connector in the process of being published. Once completed, you see a page indicating that your connector is published.

2. Select **Go To Details Page** to see the details of the published connector.

⚠ Note

Now that your connector is published an administrator must enable it before it can be used in the copilot. If you don't see the connector in the copilot make sure that the connector is enabled through Microsoft Admin Center or the copilot agents page. Once properly enabled, the connector will show up in the flyout. This step can have a delay of up to 24 hours the first time you do it.

3. You can also easily move a connector from one environment to another using a solution. To do so select **Export solution** on the details page. Exporting the solution lets you export the connector.



4. Change the version number if needed and select if the solution must be *Managed* or *Unmanaged*.

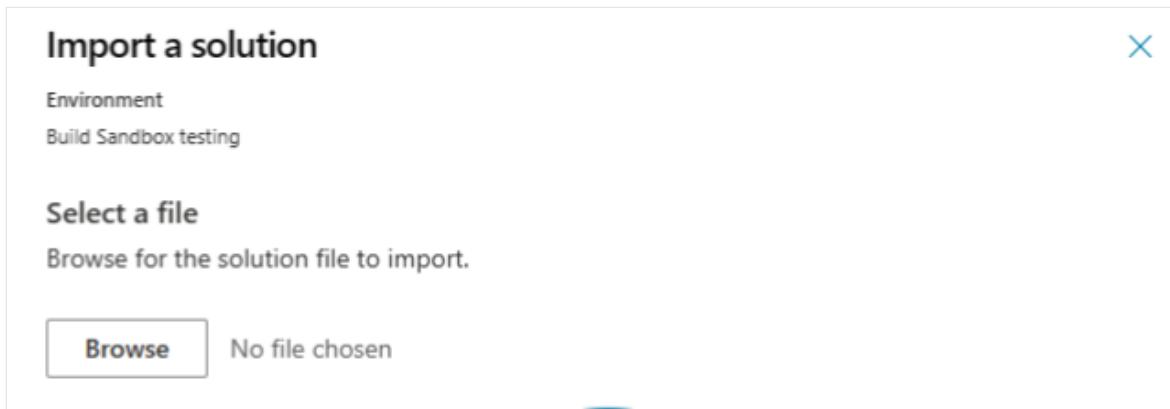
- *Managed* means you don't expect to have to make further changes to the solution.
- *Unmanaged* means you want to make changes to the solution in the target environment.

5. On completion, you see a green message with option to *Download* the solution, which saves a .zip file in your system.

! Note

You aren't required to select **Publish** in the solutions panel.

6. Return to the agent **Actions** page in Copilot Studio and select **Import Solution** in the target environment.



7. Browse to select the .zip file for your exported solution and select **Next** to proceed.
You can see the connector appearing in the **Actions** list.

Delete a connector action

If you need to delete a connector action, perform the following procedure.

Locate solution used by connector action

The procedure to delete a connector action is carried out in the Power Apps Maker portal. However, you need the solution used for the connector action in Copilot Studio.

1. Open Copilot Studio.
2. Select the environment used to create the connector action.
3. Under **Agents**, select the agent where the connector action was created.
4. Select the **Actions** tab to view the list of actions associated with the agent.
5. Select the action you want to delete, and take note of the solution name.

The screenshot shows the 'Copilot for Sales' interface with the 'Actions' tab selected. A message at the top says, '⚠ You have a draft in progress. Select 'Edit draft' to continue from where you left.' Below this, the 'Contoso Hub 6/27' connector is listed with its details: Name (Contoso Hub 6/27), Description (This is a sample connector to extend Copilot for Sales), Type (Custom Connector-Plugin), and Solutions (Contoso Hub 6/27 Solution).

Delete connector in Power Apps Maker portal

1. Go to the Maker portal (<https://make.powerapps.com>) and select the environment you used in Copilot Studio to create actions.
 2. In the navigation panel, select **Solutions** to view the list of solutions in the environment.
 3. Locate the solution you used to create the action and select it.
- A list of objects associated with the solution appears.
4. Find the connector action you want to delete. The action is of type *AIPlugin*.

💡 Tip

If you don't see your action in the solution, move the action from *draft* state to *ready* state by publishing it.

5. Before deleting the action, you need to delete the dependency on the *AICopilot_AIPlugin* table. First, select **Advanced > Show dependencies**.

The screenshot shows the Power Apps Maker portal's 'Objects' screen for the 'Contoso Hub - v2 Solution'. A context menu is open over a row for an 'AIPlugin' action named 'msdynapi_574ff27d-e3c8-480a-8d50-3c33a87...'. The menu path 'Advanced > Show dependencies' is highlighted. Other options in the menu include 'Edit', 'Add required objects', 'Remove', 'See solution layers', and 'Managed properties'.

The dependencies for the connector action are displayed.

6. For the *AICopilot_AIPlugin* entry in the dependency list, select **Delete**.

This screenshot shows the 'Dependencies' section in the Power Apps portal. It lists items that depend on a specific managed solution. The table includes columns for Display name, Name, Parent table, Object type, and Dependency type. The 'AIPlugin' entry is selected, and a context menu is open, showing options like 'Open', 'Delete', 'Remove dependency', and 'Open documentation'. The 'Delete' option is highlighted.

7. Return to the list of objects and delete the action by selecting **Remove > Delete from this environment**.

This screenshot shows the 'Objects' list in the Power Apps portal. The 'All' category is selected, showing various objects including 'Contoso Hub' and 'msdynapi_574ff27d-e3c8-480a-8d60-3c33a87d8b6f'. The latter is selected, and a context menu is open, showing 'Remove' and 'Delete from this environment' options. The 'Delete from this environment' option is highlighted.

8. Delete any operations of type *AIPluginOperation* in the table by selecting **Remove > Delete from this environment**.

This screenshot shows the 'Objects' list again. A message at the top indicates that objects have been removed successfully. The 'All' category is selected, showing 'Contoso Hub' and 'Enrich CRM record details'. The 'Enrich CRM record details' entry is selected, and a context menu is open, showing 'Remove' and 'Delete from this environment' options. The 'Delete from this environment' option is highlighted.

9. Return to Copilot Studio to verify that the action is removed from the **Actions** view for the Copilot.

① Note

This procedure removes the action from the Actions list. However, this will not remove any connection created to the data source.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(330310c16e5f800021391ab8f7c7ee51_img.jpg\)](#)

Extend Microsoft 365 Copilot with Copilot agents

Article • 11/19/2024

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.

In Copilot Studio, these more focused assistants are called *Copilot agents*. These agents are authored by giving natural language instructions that generate prompts for Microsoft 365 Copilot. Copilot agents can include references to enterprise data, actions, and suggested prompts.

ⓘ Note

What are called Copilot agents in Copilot Studio are known as *declarative agents* in Microsoft 365 Copilot. For more information about Microsoft 365 Copilot extensibility, see [Microsoft 365 Copilot extensibility](#).

This article covers:

- Authoring a Copilot agent
- Adding actions and knowledge to the agent
- Publishing the agent to Microsoft 365 Copilot and Microsoft Teams

What are Copilot agents, actions, copilot knowledge, and starter prompts?

The following table gives more detailed information about the distinctions between agents, actions, copilot knowledge, and starter prompts.

[+] [Expand table](#)

Name	Definition
Copilot agent	A Copilot agent is created 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. An agent is most often used in the context of

Name	Definition
	extending Microsoft 365 Copilot. A Copilot agent can also be used directly as an agent in Microsoft Teams.
Action	Formerly known as plugins. The mechanism that lets a copilot call out to other systems and retrieve data or perform other actions. Actions are single-turn interactions. Actions can be authenticated, and can include parameters, cards, and other functionalities. See Use actions with custom agents .
Knowledge	Knowledge refers to information that lives within your Microsoft 365 tenant, including SharePoint, and Graph connectors.
Starter prompt	Starter 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 actions invoked via REST API descriptions configured by the agent. Custom actions 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 an agent

Unlike traditional authoring where a maker creates each topic with natural language triggers for even simple tasks, an agent configured to use generative AI has a baseline set of default behaviors. As a maker, you can add knowledge and actions to make the agent more capable for your specific requirements.

To create a Copilot agent for Microsoft 365 Copilot:

1. Sign-in to a Copilot Studio account at <https://copilotstudio.microsoft.com>
2. Navigate to **Agents** in the left side navigation panel.
3. Select **Microsoft 365 Copilot** 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 agent is now created. You can interact with the Copilot 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.

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 Graph

Name	Definition
	connectors
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 Graph connector in Microsoft 365, you can choose a Graph connector from the configured data sources.
 - Web search can be enabled/disabled from the copilot 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 are not returned, please submit feedback to Microsoft for analysis and product improvement.

Add an action to a Copilot agent

An action is used to enable an agent to retrieve information or to perform a task such as updating a record or completing a transaction. Actions can be created from Copilot

connectors (also known as Power Platform connectors). Action types include AI Builder prompts, connectors, flows, and custom connectors. To learn more about Copilot connectors, see: [Learn content on Copilot connectors](#).

Adding an action requires extra planning on the part of the maker vs adding knowledge. Several considerations come into play:

- Should the action 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 action?
- 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 actions to an agent.

Step 1 of 3: Choose an action
Create an action or browse through our list of actions you want to use to get information from external sources. [Learn more](#)

Discover an action
Search for flows and commonly used connector actions

sap

30 actions found

Connectors Custom Connectors Flows Dataverse

Bring SAP window to the front IA-Connect SAP GUI	Get SAP table properties IA-Connect SAP GUI
Call SAP function (V2) SAP ERP	Get SAP tree nodes IA-Connect SAP GUI
Call SAP function (V3) SAP ERP	Maximise SAP window IA-Connect SAP GUI

Cancel

To add an action:

1. Return to the agent's overview page.
2. From the **Actions** card, choose **Add Action**.
3. Search for the desired copilot connector or choose **Create a new flow** or **Add an API for a custom connector**.
4. Assuming a Copilot connector is chosen, a UX is presented with **Connector** and **Configure** sections.

Connector section

- 1. End User Authentication:** Choose the type of authentication preferred for this action. You have a choice between user authentication and copilot author authentication. Other authentication configuration details are automatically handled by the copilot connector implementation.
- 2. Usage Description:** Instructions that tell the Microsoft 365 Copilot how to use the action in the agent.

Configure section

- 1. Plugin Action 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.
- 2. Display Name:** Friendly name for the action as seen in the Microsoft 365 Copilot user experience.
- 3. Description for the agent to know when to use this action:** A full conversational description of when Copilot should use this action, and the sort of questions that trigger it.

Inputs and outputs

Copilot Studio automatically populates a set of inputs and outputs by inspecting the copilot 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.

Details **Inputs** Outputs

Required inputs (2)

Required inputs need to be filled in for an action to run.

Location

How will the copilot fill this input?

Dynamically fill with best option (default) ▾

Display name	Identify as
Location	User's entire response >

Description

The location search query. Valid inputs are City, Region, State, Country, Landmark, Postal Code, latitude and longitude

Units

How will the copilot fill this input?

Dynamically fill with best option (default) ▾

Display name	Identify as
Units	User's entire response >

Description

The measurement system used for all the measure values in the request and response. Valid options are 'Imperial' and 'Metric'.

Finishing and testing

The last tab allows you to review your choices, make any adjustments and then save your changes.

Once saved, the actions are ready to test 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.

In the chat, the first time the user triggers the action, the agent presents a connection card. This card is used to enable the user to sign in to the action with their own credentials.

Please establish the connection. To proceed, please select 'Connect'. Once completed, please select 'Retry'. [Review all permissions](#)

[Connect](#)

[Retry](#)

Selecting **Connect** takes the user to an experience where they can choose the connection for the action they're using and sign in or otherwise connect to it. Once the connection is established, it's durable for the session. After that, the user can close the browser and retry the action.

ⓘ Important

The full screen browser for connection management is a temporary solution and will be addressed before release.

Starter 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 starter prompts experience. Starter prompts present the user with easy prompts to get started.

The screenshot shows the Microsoft Copilot Studio interface. On the left is a vertical sidebar with icons for Home, Create, My Content, Feed, Apps, Copilot (which is selected), Admin, Outlook, Teams, Word, Excel, and an ellipsis. The main area has a light blue background with a white speech bubble icon containing a gear and the text 'Relecloud Drones'. Below it are six cards 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?

Below these cards is a text input field with placeholder text: "OK, what can I help with? Try one of the examples or enter your own prompt." To the right of the input field are "View prompts" and "Need help?" buttons. At the bottom of the input field is a character counter "0 / 2000".

Copilot Studio automatically creates starter prompts when you use the conversational experience to create your agent. You can also edit or create your own at any time with the **Starter prompts** experience on the **Overview** page.

Understanding starter prompts

One way to use the starter 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 starter 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 starter prompts

1. Navigate back to the agent's **Overview** page.
2. Scroll down to **Starter Prompts** and select **Edit**.
3. Add up to six starter prompts and hit save.

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 copilot, an agent isn't automatically deployed to Microsoft 365 Copilot and Teams when published.

Publish your 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 Azure Entra ID environment, corresponding to your agent. This will allow users to interact with the agent in Microsoft Teams.

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 has disabled side-loading or publishing for the tenant, one or more of these features may be blocked. Work with the tenant admin to determine the appropriate distribution mechanism for the tenant.

 Expand table

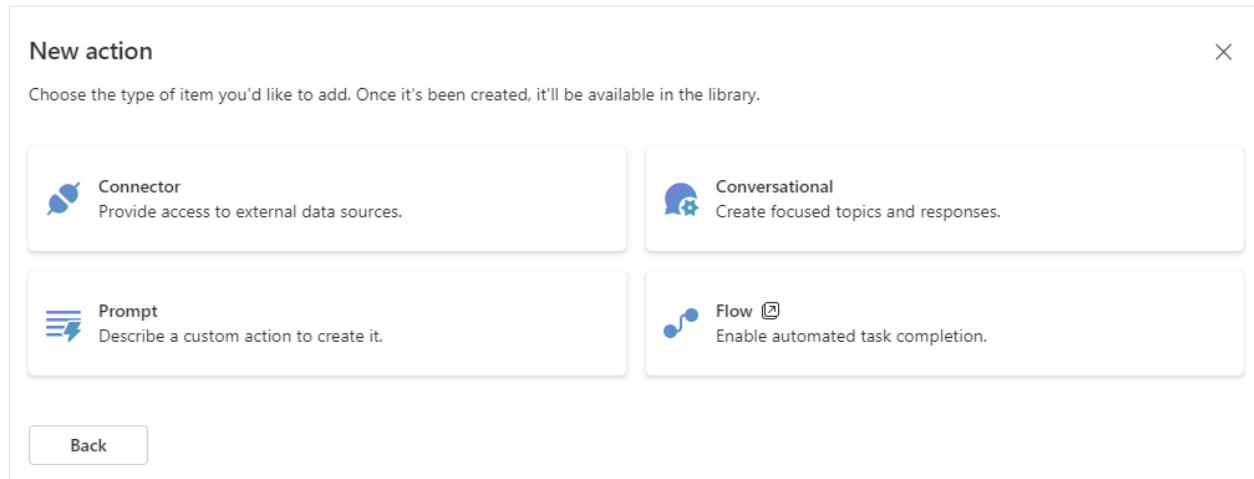
Availability Option	Description
Share Link	You can copy a <i>deep link</i> into Microsoft Teams to invoke the published agent. Opening the link in the tenant opens a chat experience with the agent.
Show to my teammates and shared users	Lets you grant access to others to participate in authoring the agent, 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](#)

Using 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. The user 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. Once selected, the user interacts only with the agent until they end that conversation and return to Microsoft Copilot.

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.



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 a custom 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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Copilot Studio action architecture

Article • 12/13/2024

An **action** is a reusable piece of code that can perform a specific task or provide specific functionality for an agent. For example, an action can help an agent answer a natural language query, execute a workflow, connect to an external system, or provide topic-specific guidance. An **agent** is a conversational or UX-based assistant that helps users accomplish their tasks and goals in a specific domain or application.

An action, in this context, is created in the action authoring experience in the Microsoft Copilot Studio. This feature allows users to create and edit actions using a graphical user interface and publish them to the actions registry.

The actions registry helps you create an action once and use it in multiple agents. The registry provides storage and management for metadata and execution information for actions. Users can apply the power and flexibility of actions to enhance the capabilities of agents without writing code for each agent separately. The various agents interact with the actions registry to discover actions and execution information available for a user. This capability enables AI actions to be created once and reused many times.

An action is born

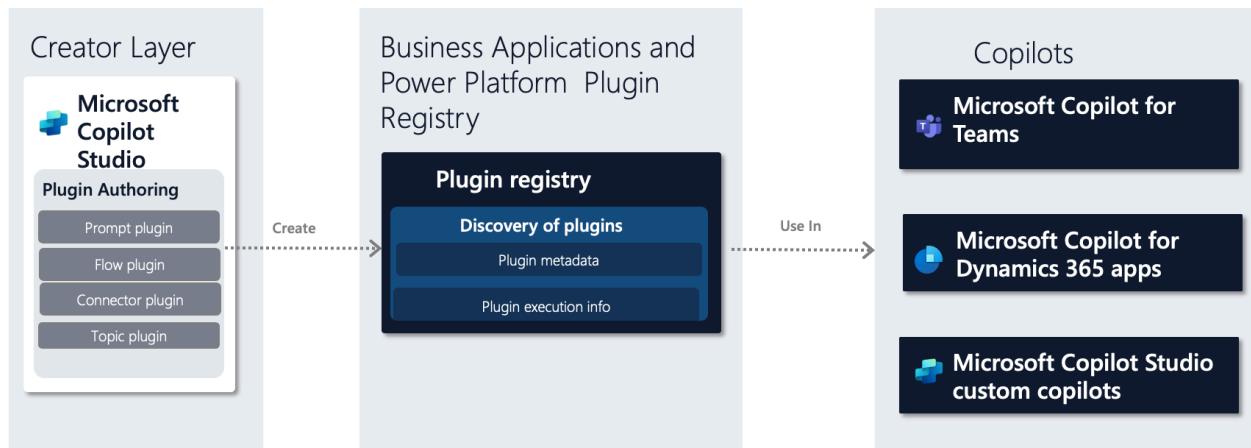


Figure: action Authoring using Microsoft Copilot Studio

Creator Layer

The Creator Layer of Microsoft Copilot Studio is where users can create and edit actions. This layer supports four types of actions: Prompt, Flow, Connector, and Topic.

- **Prompt** actions are actions that can answer natural language queries from users using natural language processing (NLP) and natural language generation (NLG)

techniques. For example, a prompt action can answer a question like "What are the sales opportunities for ACME Corp?" by querying data in Microsoft Dataverse and generating a natural language response that can be customized (such as a bulleted list).

- **Flow** actions are actions that can execute multi-step workflows using Microsoft Power Automate. For example, a flow action can create a new contact in Microsoft Dynamics 365 after manager approval or send an email to a customer with a prescribed template.
- **Connector** actions are actions that can connect to external systems or data sources using Microsoft Power Platform connectors. For example, a connector action can connect to Salesforce or SAP using Power Platform connectors and retrieve or update data. Connector actions can be certified connectors that are shared across tenants or environment-specific custom connectors.
- **Topic** actions are single-turn conversational threads between a user and an agent that can be created to answer a specific user utterance. For example, a topic about store hours with a trigger phrase *check store hours* can return the store hours. These topics can still handle user utterances such as "see store opening hours" thanks to the power of generative AI.

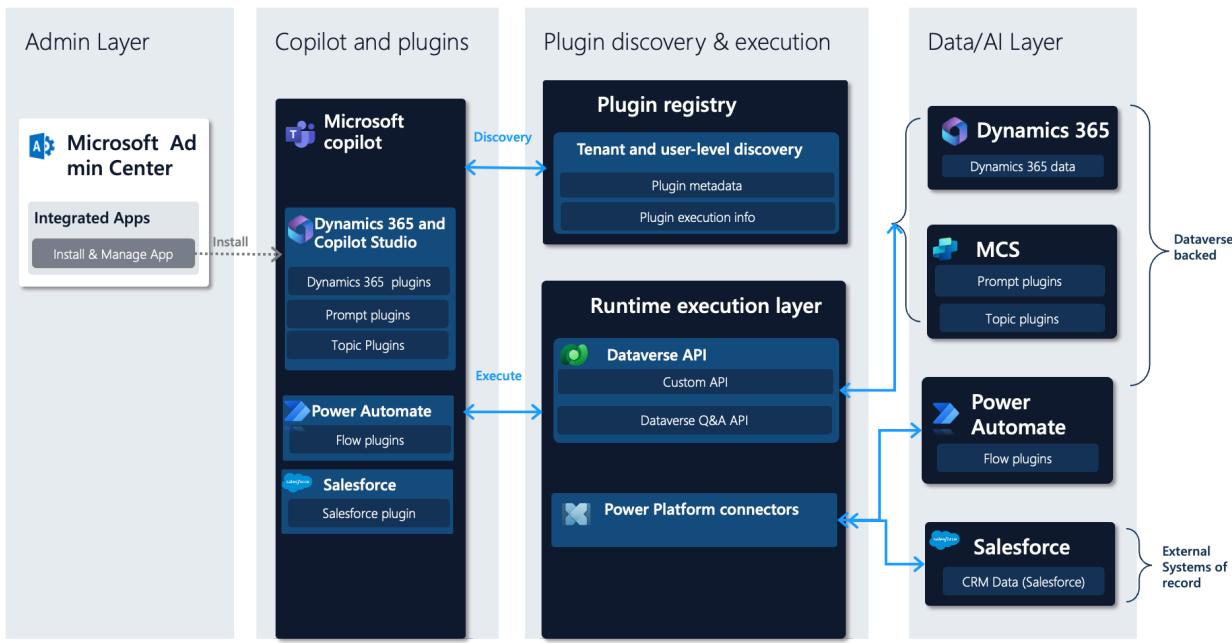
Business applications and Power Platform action registry

The actions registry stores and manages the action metadata and execution information. The registry is a single source for discovery of actions authored from Microsoft Copilot Studio. The registry helps you discover integrated apps available at the tenant level used in the Microsoft Admin center by Microsoft 365 Teams. The registry shows you actions available to a user, or used by agents in Microsoft Copilot Studio. An action has the same security as its underlying artifact, such as a flow and can provide a customized list of actions for a specific user role.

Agents samples

This sample set of the various agents integrates with the actions registry to consume actions. These agents include Microsoft agents for Dynamics 365 apps, Microsoft agents for Teams, and custom agents authored using Microsoft Copilot Studio. The list is expected to grow in the future as more first-party and third-party agents are developed.

An action in Microsoft Copilot



Admin layer

The administration layer is where administrators can use the Microsoft Admin Center to choose the integrated apps available for their tenant. Administrators can allow, deploy, or block apps for specific or all users. An integrated app can have multiple actions. The apps available from Microsoft Copilot Studio-created actions are retrieved from the actions registry and include:

- **Dynamics 365 and Microsoft Copilot Studio:** Includes Dynamics 365 actions from teams such as Microsoft Dynamics 365 Sales and Microsoft Dynamics 365 Field Services. You can create custom prompts and topic actions.
- **Power Automate:** Includes Power Automate flow actions.
- **Connector specific:** A sample of Salesforce and SAP actions are called out here. Every certified connector action created by Microsoft or ISVs shows up as their own integrated app for fine-grained management.

(!) Note

App names are subject to modification.

Action discovery and execution

The actions available to a maker include actions assigned through deployed integrated apps. The actions available to a maker are discovered through the actions registry. Makers are free to further customize the actions based on how they want to use the

action flyout experience. The action execution information gets retrieved from the actions registry and includes the runtime path.

There are two runtime execution paths supported today:

1. **Actions that are Dataverse-based:** Microsoft Dynamics 365 actions that access or modify Dataverse data, Microsoft Copilot Studio topic actions, and prompt actions. This path supports Microsoft Entra ID authentication. The calls are made to the underlying system on behalf of the user.
2. **Actions that use the Power Platform connector ecosystem:** This path helps execute connectors that provide access to internal and external systems of record. It supports explicit configuration of credentials per user such as Salesforce connection credentials that are then maintained per user.

If a maker doesn't find a suitable action for their task, they can navigate to the Microsoft Copilot Studio and create an action that addresses their needs.

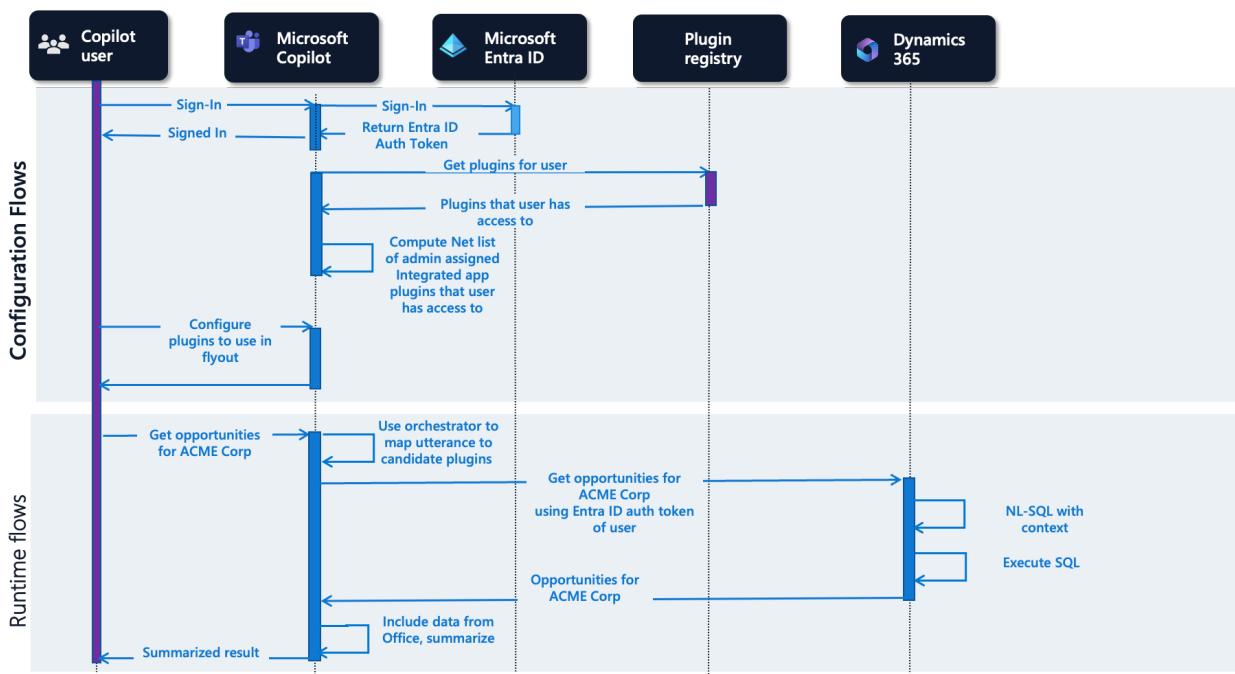
Data/AI Layer

This layer represents the various data stores and systems where the data and business logic reside. These stores and systems include Sales, Field Service and other Dynamics 365 products, Microsoft Copilot Studio, Power Automate, external systems such as Salesforce, SAP, and others.

Data and control flows for some action types

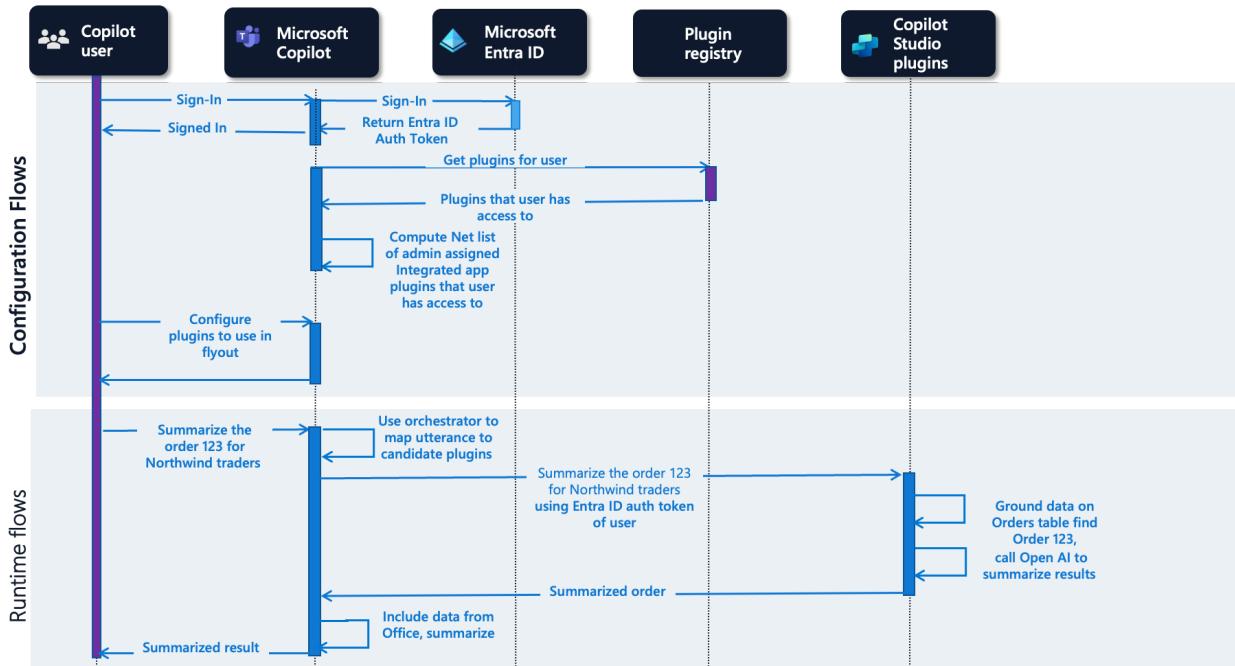
As a prerequisite for all flows, an administrator uses the Microsoft Admin Center to configure the corresponding integrated apps and assigns them to users, including the Copilot user referenced here. In these flows, we first see the **configuration phase** for a user in Copilot for Teams. After the user signs in, the actions applicable for that user are retrieved from the action registry. The actions in a user's flyout belong to the integrated apps their administrator configures for them and the ones they have access to in the action registry. The user can then configure the actions they want to use in the experience in the flyout in their Copilot for Teams experience.

Dynamics 365 actions



In the runtime flow for the Dynamics 365 actions, the orchestrator maps a user's utterance of *Get opportunities for ACME corp* to a set of candidate actions. An action from Dynamics 365 is found to be best candidate and then executed, translating the user's utterance to SQL against the dynamic data the user has access to. The results are then returned to the user. Teams can additionally add data from Microsoft 365 and summarize the results.

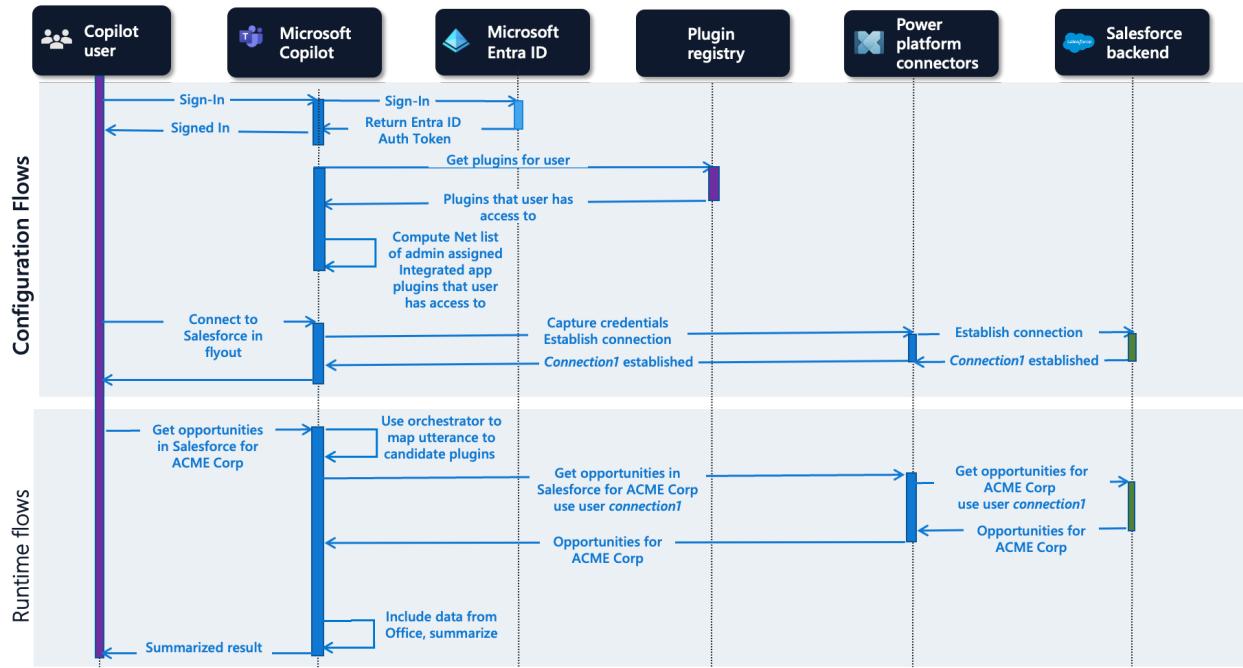
Microsoft Copilot Studio actions



In the runtime flow for Copilot Studio prompt actions, the user's utterance is again mapped to an action. In this case, the matching action is a prompt authored in Microsoft Copilot Studio. The prompt grounds data in Dataverse table *orders* and calls

into OpenAI to summarize the results for order 123 and returns the summary to the user.

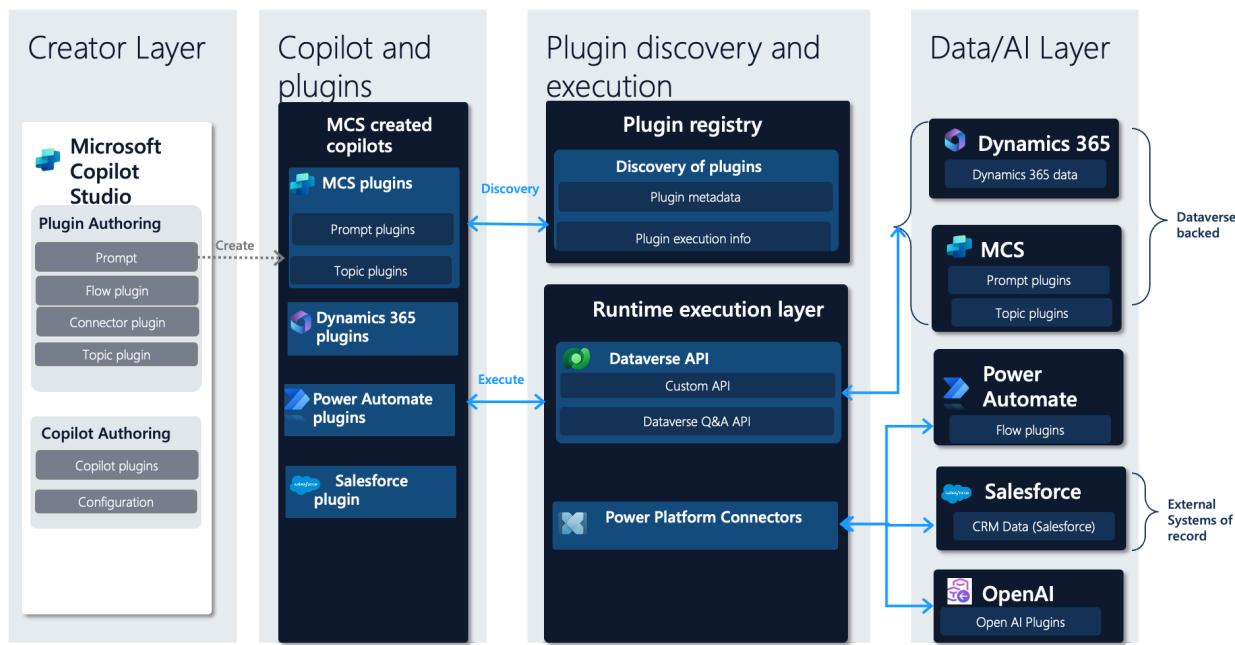
Connector actions



The configuration flow for Salesforce or other connector-based actions starts with a **Connect** experience in the flyout where the user can enter connection information for Salesforce. At runtime, the user utterance is mapped to a candidate action and this action is executed using the Power Platform Connector infrastructure. The results are then returned to the user.

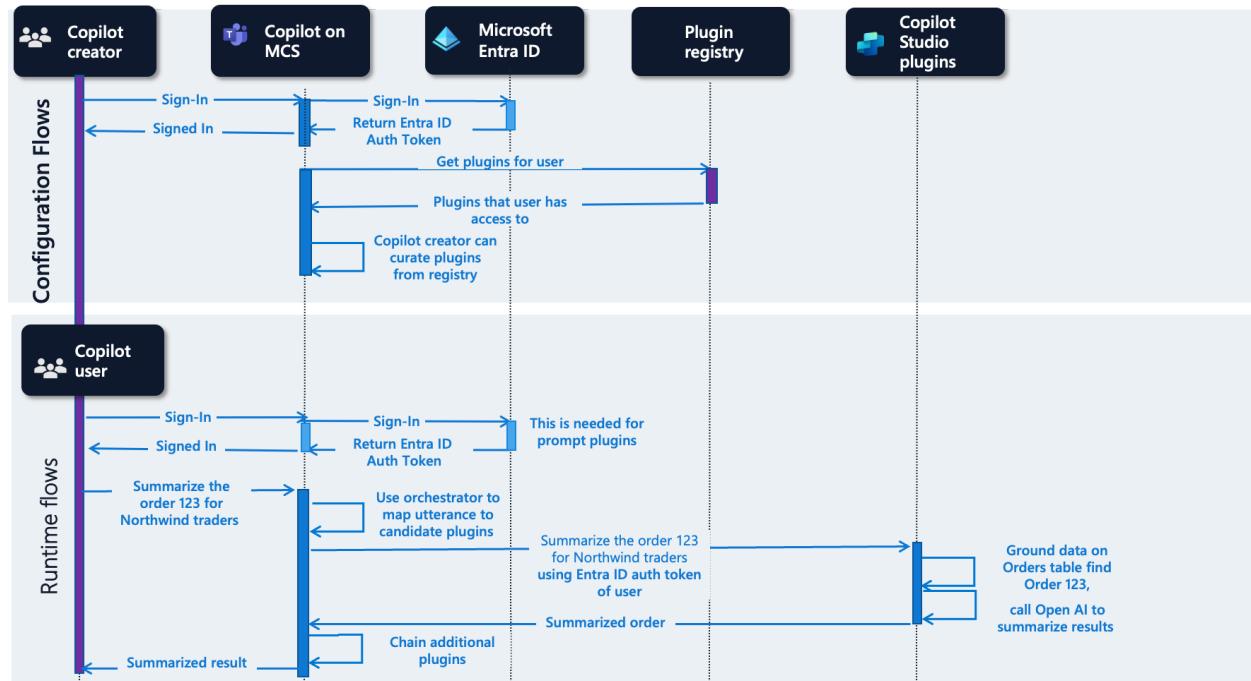
Actions used in Microsoft Copilot Studio agents

Microsoft Copilot Studio can be used to create agents in addition to actions. Dynamics 365 teams or third parties can build these agents.



Similar to the creator layer, the action discovery, execution, and Data/AI layer look like the figure shown. The difference is in how you manage curation of actions for the agent. The agent creator chooses amongst available actions from their Dataverse environment.

Data and Control flows for some action types



Here's an example of a prompt action used from an agent built in Microsoft Copilot Studio. The configuration flow includes an agent creator curating the actions for the agent from the action registry. At runtime, the authenticated agent user summarizes order 123 for Northwind traders. This summary maps the utterance to candidate actions. The matching action is a prompt authored in Copilot Studio. The prompt uses data from the *orders* table in Dataverse. The results are summarized using OpenAI and are returned to the user.

Conclusion

By authoring an action in Microsoft Copilot Studio, you can further the reach of your action across agents in Microsoft Dynamics 365, Microsoft Copilot for Teams, and other third-party agents. Using actions in this way allows the flexibility to create once and deploy across multiple agents.

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Configure a Bot Framework skill for use in Copilot Studio

Article • 04/03/2024

ⓘ Important

Power Virtual Agents capabilities and features are [now part of Microsoft Copilot Studio](#) following significant investments in generative AI and enhanced integrations across Microsoft Copilot.

Some articles and screenshots may refer to Power Virtual Agents while we update documentation and training content.

Copilot Studio lets you extend your copilot by using [Bot Framework skills](#). If you have already built and deployed bots in your organization (using the Bot Framework SDK and pro-code tools) for specific scenarios, you can convert those bots into a skill and register that skill in a Copilot Studio copilot.

ⓘ 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 [Bot Framework skills](#). After a skill has been registered with a Copilot Studio copilot, authors can seamlessly [trigger skill actions in conversation](#).

Prerequisites

- Review restrictions and validation for skills used in Copilot Studio

ⓘ Important

Copilot Studio copilots only support skills built using [Bot Framework SDK version 4.12.0+](#).

Classic chatbots only support skills built using [Bot Framework SDK version 4.7+](#).

If you're using [Bot Framework Composer](#) it's strongly recommended that you use Composer 2.x. If you want to use an existing Composer 1.x copilot, [first migrate it to Composer 2.x](#) before using it as a skill.

To configure a skill, the associated app registration in Microsoft Entra ID for your skill should be set to Multi-tenant configuration.

Flows actions and skills actions

The following table will help determine when to use skills for a conversation.

[] Expand table

	Flow actions	Skill actions
Persona	Copilot authors can build reusable Flows to embed into any copilot conversation	Developers can create, deploy, and host custom skills in their own environment
Conversation	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.
Response	Use Flows to emit a copilot response. For example, show a personalized message or inline images.	Use skills to emit any supported copilot response. For example, show an adaptive card or send random responses.
Actions	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 copilot response.

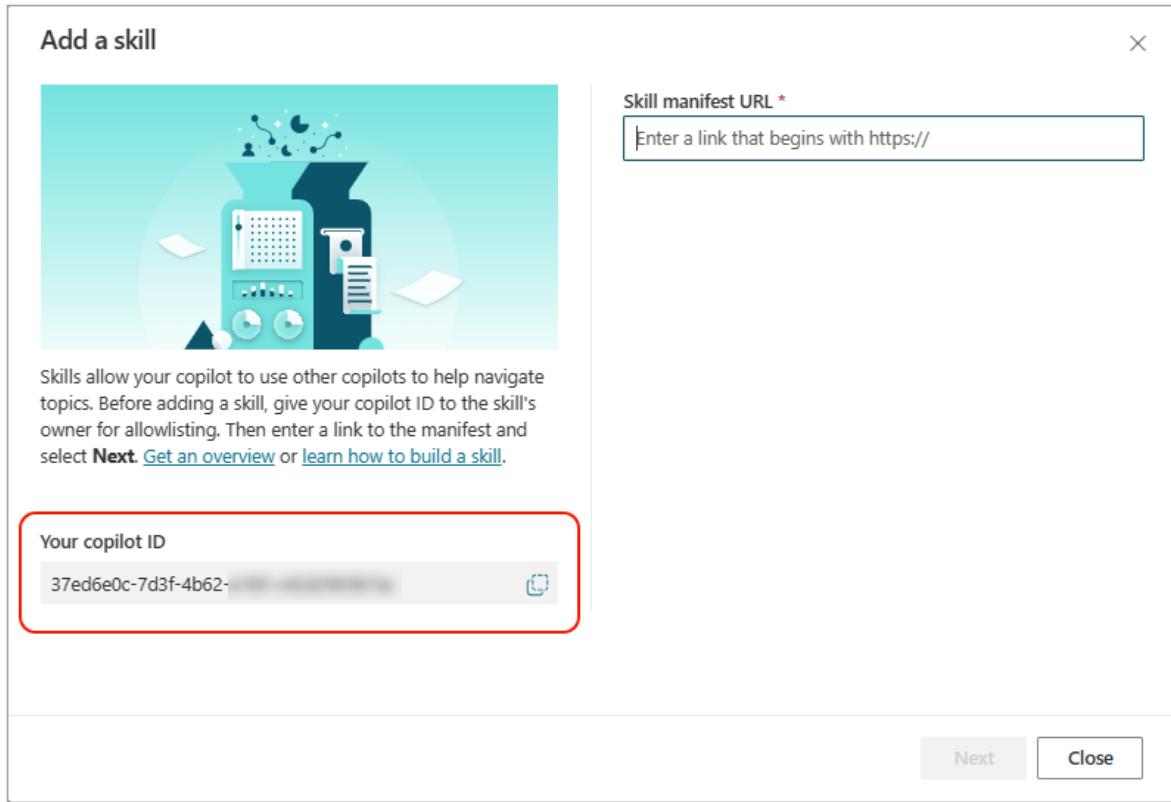
Configure a skill

First, use Copilot Studio to [create a copilot](#) and [create and deploy a skill using pro-code tools](#) into your organization.

Before registering the skill, provide the copilot's ID to your skills developer to authorize the copilot to call actions in the skill. [Learn more about skill allowlist](#). You can get your copilot's ID from the **Add a skill** window.

To add a skill to your copilot:

1. In the navigation pane, expand **Settings** and select **Skills**.
2. At the top of the **Skills** page, select **Add skill**.
3. Copy your copilot 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 copilot will need 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 copilot. 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 copilot. You will need to fix these errors prior to registering a skill.

[] [Expand table](#)

Error message	Troubleshoot / Mitigation
We ran into problems getting the skill manifest. (MANIFEST_FETCH_FAILED)	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. (MANIFEST_MALFORMED)	<p>(a) Check if the manifest is a valid JSON file.</p> <p>(b) Check if the manifest contains required properties For example, (name, msaAppId, single endpoint, activities/id, activities/description, activities/type (only event or message supported)).</p>
There is a mismatch in your endpoints (MANIFEST_ENDPOINT_ORIGIN_MISMATCH)	Check if your skill endpoint matches your Microsoft Entra ID application registration's Publisher domain (preferred) or Home page URL field. Learn more about setting the home page for endpoints .
To add a skill, it must first be registered (APPID_NOT_IN_TENANT)	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:// (URL_MALFORMED, URL_NOT_HTTPS)	Re-enter the link as a secure URL.
The manifest is too large; (MANIFEST_TOO_LARGE)	Check size of the manifest. It must be less than or equal to 500KB.
This skill has already been added to your copilot. (MANIFEST_ALREADY_IMPORTED)	Delete the skill and try registering again.
The skill is limited to 100 actions. (LIMITS_TOO_MANY_ACTIONS)	There are too many skill actions defined in skill manifest. Remove actions and try again.
Actions are limited to 25 inputs. (LIMITS_TOO_MANY_INPUTS)	There are too many skill action input parameters. Remove parameters and try again.
Actions are limited to 25 outputs. (LIMITS_TOO_MANY_OUTPUTS)	There are too many skill action output parameters. Remove parameters and try again.
Your copilot can have a maximum of 100 skills. (LIMITS_TOO_MANY_SKILLS)	There are too many skills added into a copilot. Remove an existing skill and try again.
It looks like something went wrong. (AADERROR_OTHER)	There was a transient error while validating your skill. Retry.
Something went wrong while checking your skill.	Check if your skill endpoint is online and responding to messages.

Error message	Troubleshoot / Mitigation
(ENDPOINT_HEALTHCHECK_FAILED, HEALTH_PING_FAILED)	
This skill has not allow-listed your copilot (ENDPOINT_HEALTHCHECK_UNAUTHORIZED)	Check if your copilot has been added to the skills allowlist.

Use Microsoft Bot Framework skills in Copilot Studio

Article • 07/26/2024

Microsoft Copilot Studio enables you to extend your copilot using Microsoft Bot Framework skills. If you have already built and deployed bots in your organization (using Bot Framework pro-code tools) for specific scenarios, you can convert such a bot to a skill and embed the skill within a Microsoft Copilot Studio copilot.

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 Bot Framework skills.

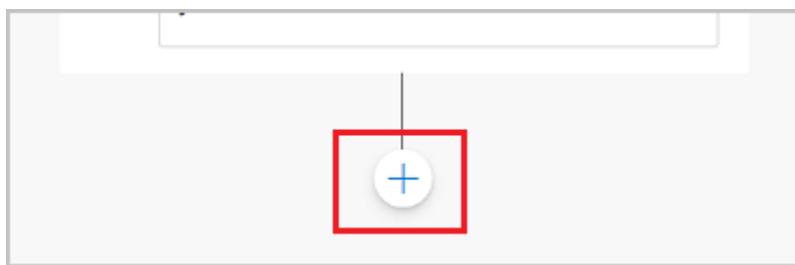
Add a skill to copilot conversations

First, [create a Copilot Studio copilot](#) 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 copilot 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 copilot.

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.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use a classic chatbot as a skill in a Bot Framework bot

Article • 11/19/2024

ⓘ Note

This article includes information applicable to Copilot Studio classic chatbots only. This feature is not available in the new Copilot Studio experience.

When you use a classic chatbot as a [skill](#) with a Bot Framework bot, the Bot Framework bot determines if anything the user says matches the classic chatbot's [trigger phrases](#). If there's a match, the Bot Framework bot [passes the conversation to the classic chatbot](#). The classic chatbot extracts any [entities](#) and triggers the matching topic.

You can also [pass a variable to a Copilot Studio skill topic as an input](#) from Bot Framework Composer and [use the outputs it returns](#).

ⓘ 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 Framework bot to the allowlist for the classic chatbot](#).
2. [Download the skill manifest for the classic chatbot](#).
3. [Use the downloaded skill manifest in Bot Framework Composer to create a connection between the bots](#).
4. [Use the Bot Framework Emulator to confirm the bot works as a skill](#).

Prerequisites

- Understand how Bot Framework skills work.
- Understand how to use Bot Framework Composer.
- Understand how to use skills in Bot Framework Composer.
- Get your Bot Framework bot's app ID.

Add your Bot Framework bot to the allowlist for your classic chatbot

Add the Bot Framework 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. In Bot Framework Composer, open the bot that will use the skill. Select **Configure**, and then select **Development resources**. Copy the ID in **Microsoft App ID**.
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 Framework 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'll get an error. You can only add an app ID for a Bot Framework bot.

Copilot Studio validates the Bot Framework 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've added to the allowlist.
7. Select **Save**.

The Bot Framework bot is shown 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 Framework 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 Framework 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. These are JSON files that include a skill's name, interface, and trigger phrases.

A Bot Framework 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 Framework skill manifest schema](#) and consist of *intents.lu* and *manifest.json* files.

A Bot Framework bot may decide at classic chatbot should handle the user's request, based on the manifest data. It passes to the classic chatbot everything the user said. 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 Framework 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 Framework bot to connect to the published version of your classic chatbot.
 - The published manifest is only available for classic chatbots that have been 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 have never been 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 and Composer dialogs that are added to a classic chatbot. Only topics that were created by copilot authors and Composer intent triggers that authors add to the bot are included.

Connect to a Copilot Studio skill in Composer

Use Copilot Studio skill manifests to create a skill connection in your Bot Framework bot in Composer.

1. In a Bot Framework Composer project, select **Add**, and then select **Connect to a skill**.
2. Browse to and select a Copilot Studio manifest .zip archive, and then select **Next**.
3. Select the topics, or intents, in your classic chatbot that you want to add to your Bot Framework bot, and then select **Next**.
4. Review and edit the trigger phrases for your classic chatbot, and then select **Next**.
5. Make sure **Use Orchestrator for multi-bot projects** is selected, and then select **Continue**.
6. Verify that the Copilot Studio skill is added to your Bot Framework bot correctly, and that a new trigger with your classic chatbot's name appears in the project in Composer.

Locate the topic in the skill manifest

Use an event to trigger a Copilot Studio topic in the skill. To call a specific topic from Composer, refer to the topic by the name of the associated event in the skill manifest.

1. Select the **Connect to a skill** action.

2. Under Skill Dialog Name, select **Show skill manifest**.
3. Find **activities** in the manifest. This property contains nested properties that represent the topics available in your classic chatbot.
4. Find the topic you want to call and look for its **name** property. When the Copilot Studio skill receives an event with this name, it triggers the topic.

In this example, the event activity name is

```
dispatchTo_new_topic_87609dabd86049f7bc6507c6f7263aba_33d.
```

Call a Copilot Studio skill topic

Instead of relying on what a user says to trigger a Copilot Studio skill topic, you can call it directly. You can only directly call Copilot Studio topics that are listed in the skill manifest.

1. In the Composer authoring canvas, select **Add**, then select **Access external resources**, and then select **Connect to a skill**.
2. In the **Skill Dialog Name** list, select your classic chatbot.
3. **Find the event activity's name for the classic chatbot topic** that you want to call.
Copy the value, omitting the quotation marks, and save it to use in the next steps.
4. Select **Close**.
5. In the **Activity** section, select **Show code**.
6. Type or paste the following code. Replace `TOPIC_ACTIVITY_NAME` with your topic's event activity name. Make sure there are no quotation marks in the **name** property.

Bot response

```
[Activity
  type = event
  name = TOPIC_ACTIVITY_NAME
]
```

Pass an input variable to a Copilot Studio skill topic

Bot Framework Composer can pass a variable to a Copilot Studio skill topic as an input. To learn how to create Copilot Studio topics that accept input variables, see [Work with variables](#).

If a Copilot Studio topic that can receive an input variable is listed in your skill manifest, you can pass a Composer variable to it.

1. In the Composer authoring canvas, select **Add**, then select **Access external resources**, and then select **Connect to a skill**.
2. In the **Skill Dialog Name** list, select your classic chatbot.
3. [Find the event activity's name for the classic chatbot topic](#) that you want to call.
Copy the value, omitting the quotation marks, and save it to use in the next steps.
4. Find the topic's **value** property. The **value** property contains a **\$ref** property. Copy that value and save it to use in the next step.

 **Warning**

If a Copilot Studio topic doesn't have inputs, it won't have a **value** property.

5. Find the **definitions** property, then look for a nested property that matches the **\$ref** value you found in the previous step. Note the names and types of the Copilot Studio topic's inputs. You'll use them in the next steps.
6. Select **Close**.
7. Under the **Activity** section, select **Show code**.
8. Type or paste the following code. Replace these values:
 - a. Replace `TOPIC_ACTIVITY_NAME` with your topic's event activity name.
 - b. Replace `PVA_INPUT_VARIABLE` with an input variable from your topic.
 - c. Replace `COMPOSER_INPUT_VARIABLE` with a Composer variable that will provide a value.

Bot response

```
[Activity
  Type = event
  Name = TOPIC_ACTIVITY_NAME
  Value = ${addProperty(json("{}"), 'PVA_INPUT_VARIABLE',
  COMPOSER_INPUT_VARIABLE)}
]
```

In this example, the Composer variable `dialog.storeLocation` provides a value to the input variable `pva_StoreLocation` in the classic chatbot topic `dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f309089_21a34f16`.

Receive an output variable from a Copilot Studio skill topic

It's possible to receive outputs from a Copilot Studio skill topic in Composer. To learn how to create Copilot Studio topics that return outputs, see [Work with variables](#).

1. In the Composer authoring canvas, select **Add**, then select **Access external resources**, and then select **Connect to a skill**.
2. In the **Skill Dialog Name** list, select your classic chatbot.
3. [Find the event activity's name for your classic chatbot topic](#) you want to call. Copy the value, omitting the quotation marks, and save it to use in the next steps.
4. Find the topic's **resultValue** property. The **resultValue** property contains a **\$ref** property. Copy that value and save it to use in the next step.

⚠ Warning

If a Copilot Studio topic doesn't have outputs, it won't have a **resultValue** property.

5. Find the **definitions** property, then look for a nested property that matches the **\$ref** value you found in the previous step. Note the names and types of the Copilot Studio topic's output variables. You'll use them in the next steps.

In this example, the Copilot Studio topic returns two outputs of type String, **pva_State** and **pva_Item**.

6. Select **Close**.
7. In the **Activity** section, select **Show code**.
8. Type or paste the following code. Replace **TOPIC_ACTIVITY_NAME** with your topic's event activity name. Make sure there are no quotation marks in the **name** property.

Bot response

```
[Activity
  type = event
  name = TOPIC_ACTIVITY_NAME
]
```

- Under the code area, set **Property** to a Composer variable to receive and store the output values from the Copilot Studio skill topic.
- In the Composer authoring canvas, select **Add**, then select **Manage properties**, and then select **Set properties**.
- Set **Property** to the Composer value that should store the extracted value. Then set **Value** to the Composer variable you stored the topic's outputs in.

In this example, the output values from `dialog.skillResult.pvaState` and `dialog.skillResult.pvalItem` are extracted into new Composer variables `dialog.State` and `dialog.Item`.

Test your classic chatbot as a skill with a Bot Framework bot

Use the [Bot Framework Emulator](#) to test that your Bot Framework bot is properly calling a classic chatbot as a skill.

Important

Your Bot Framework bot must be added to the classic chatbot's allowlist for the skill connection to work.

Learn More

- [Skills overview](#)
- [Skills manifest](#)
- [Bot Framework Composer documentation](#)
- [Skills in Composer](#)
- [Connect to a remote skill in Composer](#)
- [Test a skill in Emulator](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Overview of Bot Framework Composer

Article • 08/26/2024

ⓘ Note

This article applies to *classic chatbots* only. To extend *copilots* created in Copilot Studio, you don't need to use Bot Framework Composer. All features described in this article are available for copilots, directly in Copilot Studio.

Enhance your copilot by developing custom topics with [Bot Framework Composer](#) and then adding them to your Copilot Studio copilot.

Composer is integrated into Copilot Studio to allow developers and business users to build copilots together. Composer lets you extend your copilot with Bot Framework functionality, including:

- Adaptive dialogs
- Language Generation (LG)
- Regular expressions (RegEx)
- Adaptive Cards

Using Composer with Copilot Studio lets you access conversational memory and context, including the variables defined in Copilot Studio. This memory and context lets you create topics to handle interruptions, cancellations, and context switching. You can also add Adaptive Cards to display interactive elements like images and videos, forms, and more in the copilot.

There are various prebuilt functions and expressions in Composer for you to use, and the ability to define multiple variations of a phrase for the copilot to randomly select at runtime.

Custom topics added with Composer are deployed, hosted, and executed together with the rest of Copilot Studio copilot content, and don't require any more Azure hosting.

ⓘ Important

Composer integration isn't available to users who only have the [Teams Microsoft Copilot Studio license](#). You must have a [trial](#) or full Microsoft Copilot Studio license.

Prerequisites

- [Introduction to Bot Framework Composer](#)

Understand key Composer terms

Composer uses a different set of terms to describe similar concepts used in Copilot Studio. Here are some key terms to understand:

- **dialog:** Dialogs are the basic building block in Composer and serve as containers for triggers. Each dialog represents a portion of the bot's functionality that contains instructions for what the bot does and how it reacts to user input. To learn more about dialogs, see [Dialogs as conversational building blocks in Composer](#).
- **main dialog:** A bot's main dialog is the first dialog that appears in the bot explorer and is created when you import your Copilot Studio copilot into Composer. The main dialog should never be deleted, as it corresponds to your Copilot Studio copilot.
- **trigger:** Triggers are the main component of a dialog and let you catch and respond to events. Each trigger has a condition and a collection of actions to execute when the condition is met. You can think of triggers as the equivalent of *topics* in Copilot Studio. To learn more about triggers, see [Triggers](#).
- **action:** Actions are the main components of a trigger that enable you to send a response, create and assign values to properties, manipulate the conversational flow, manage dialogs, and many other activities. You can think of actions as the equivalent of *nodes* in Copilot Studio.

Composer features not supported with Copilot Studio

The following Composer features aren't supported with Copilot Studio:

[] [Expand table](#)

Unsupported feature	Additional information
LUIS (language understanding) Recognizer	Default Recognizer in Composer is set to Microsoft Copilot Studio Natural Language Understanding (NLU).
Bot Framework Orchestrator	Bot Framework Orchestrator isn't supported in Composer for

Unsupported feature	Additional information
	Copilot Studio.
Custom recognizer	Custom recognizer isn't supported in Composer for Copilot Studio.
Bot skills	Can't invoke Bot skills in Composer content meant for Copilot Studio.
QnA Maker	This feature isn't available in Composer for Copilot Studio.
OAuth Login and User Sign Out	These types of actions aren't supported in Composer for Copilot Studio.
Dialog Inputs and Outputs	Dialog Inputs and Outputs aren't supported in Composer for Copilot Studio.
Entity detection in trigger phrases in Composer	Can't use <code># getUserName my name is {username=username}</code> in Composer.
Testing in Bot Framework Emulator	All Composer content should be published and then tested in Copilot Studio.

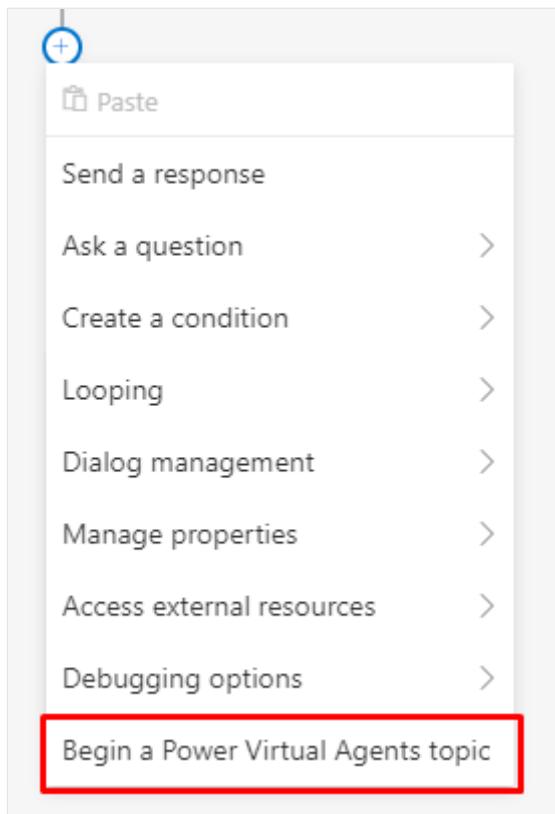
Guidelines for creating content for Copilot Studio in Composer

There are a few key guidelines to follow when using Composer to create content for Copilot Studio:

- The topics from your Copilot Studio copilot are displayed in Composer as dialogs. You can't edit these dialogs in Composer; selecting them opens the relevant topics in Copilot Studio for editing.

The screenshot shows the Microsoft Copilot Studio interface. At the top, there's a blue header bar with the project name "Contoso-Meal-Delivery-Service" and a logo. Below the header is a navigation bar with icons for Home, People, Tools, Search, Cloud, and Help. The main area is a tree view of the project structure. Under the root node "Contoso-Meal-Delivery-Service", there's a folder node "Contoso-Meal-Delivery-Service". Expanding this folder reveals a list of "Power Virtual Agents Topics (13)". This list is highlighted with a red rectangular border. The topics listed are: "Lesson 1 - A simple topic", "Lesson 2 - A simple topic with ...", "Lesson 3 - A topic with a condi...", "Lesson 4 - A topic with a condi...", "Search knowledge article privat...", "Confirmed Failure", and "Confirmed Success".

- Copilot Studio global variables are accessed in Composer by using the `virtualagent` scope. This scope doesn't appear in Composer's property reference menu, but you can access it by entering an expression directly.
- Publishing content in Composer makes the changes available for testing, but doesn't automatically publish your Copilot Studio copilot. To publish your copilot changes to channels, use the [Publish](#) feature in Copilot Studio.
- Use a **Begin a Microsoft Copilot Studio Topic** action to call Copilot Studio topics in Composer.



- You can add new Composer triggers of any kind to your main (also known as "root") dialog.
- You can send messages and ask questions from these Composer triggers.
- You can't use the Regular expression recognizer in the main dialog. Using the Regular expression recognizer in the main dialog prevents you from publishing your Composer content to Copilot Studio. However, you can use the Regular expression recognizer in any other Composer dialog.
- You can add any Composer events and triggers to newly created dialogs.
- On the **Bot Responses** page, the **Common** section should be empty. Using any assets there results in validation errors and prevents you from publishing your Composer content to Copilot Studio.

Name	Responses
(+) New template	

- Each Composer dialog has its own bot response (.lg) and language understanding (.lu) resources. For example, a bot that speaks English (US) and has a dialog named `dialog1` has one `dialog1.en-us.lg` file and one `dialog1.en-us.lu` file.
- The main dialog in Composer should never be deleted, as it corresponds to your Copilot Studio copilot. The main dialog can have zero or more triggers.
- You can capture telemetry in Composer dialogs and triggers and use [Application Insights](#) to analyze it with [Kusto queries for adaptive dialogs](#). For more information, see [Capture telemetry with Application Insights](#).

For a deep dive on Composer dialogs, events, triggers, and how to create advanced bot content with Bot Framework, refer to the following resources:

- [Bot Framework Composer dialogs](#)
- [Bot Framework Composer events and triggers](#)
- [Language generation](#)
- [Conversation flow and memory](#)

References

To learn more about developing in Composer, see these resources:

- [Bot Framework Composer documentation](#)
- [Tour of Composer](#)
- [Tutorial: Create a new bot](#)
- [Tutorial: Add cards and buttons to your bot](#)
- [Tutorial: Add language generation to your bot](#)

- Bot Framework Composer on GitHub ↗

Related content

[Get started with Bot Framework Composer](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Key concepts for Bot Framework Composer users

Article • 07/26/2024

If you're used to designing classic chatbots in Bot Framework Composer, you'll notice some similarities and differences with Copilot Studio. The following table lists some key concepts in Composer and where to find similar concepts in Copilot Studio.

[+] Expand table

Bot Framework Composer concept	Copilot Studio concept	Description
Dialogs and triggers	Topics	Use topics to organize conversation flow or paths. A topic has a set of <i>trigger phrases</i> that indicate when the copilot should start the topic and a set of <i>nodes</i> that describe the topic's conversation path.
Intents	Trigger phrases	Add trigger phrases to a topic for the phrases, keywords, and questions that a customer is likely to type related to a specific issue. Copilot uses natural language understanding to parse what a customer types and find the most appropriate topic.
Actions and prompts	Nodes	Use nodes, such as messages, questions, and conditional branches, on the authoring canvas to create a topic's conversation path.
Bot response variation	Response variations and question variations	Use response and question variations to add variety to your copilot's messages and questions.
Suggested actions	Quick replies	Use quick replies to provide default reply options to the customer.
Entities	Entities	Define and use entities to extract semantic information from what a customer types.
State, storage, and properties	Variables	Use variables to track state.
Formulas and expressions	Power Fx	Use Power Fx to create expressions.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Get started with Bot Framework Composer

Article • 08/26/2024

ⓘ Note

This article applies to *classic chatbots* only. To extend *copilots* created in Copilot Studio, you don't need to use Bot Framework Composer. All features described in this article are available for copilots, directly in Copilot Studio.

To use Bot Framework Composer to extend your Copilot Studio bot, you need to first import your bot into a Composer project.

Prerequisites

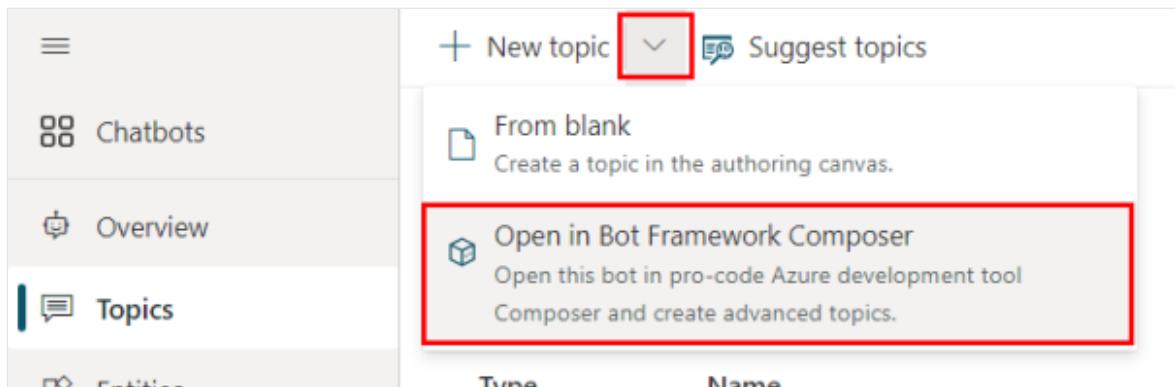
- Extend your bot with Bot Framework Composer

Open your bot in Composer

ⓘ Important

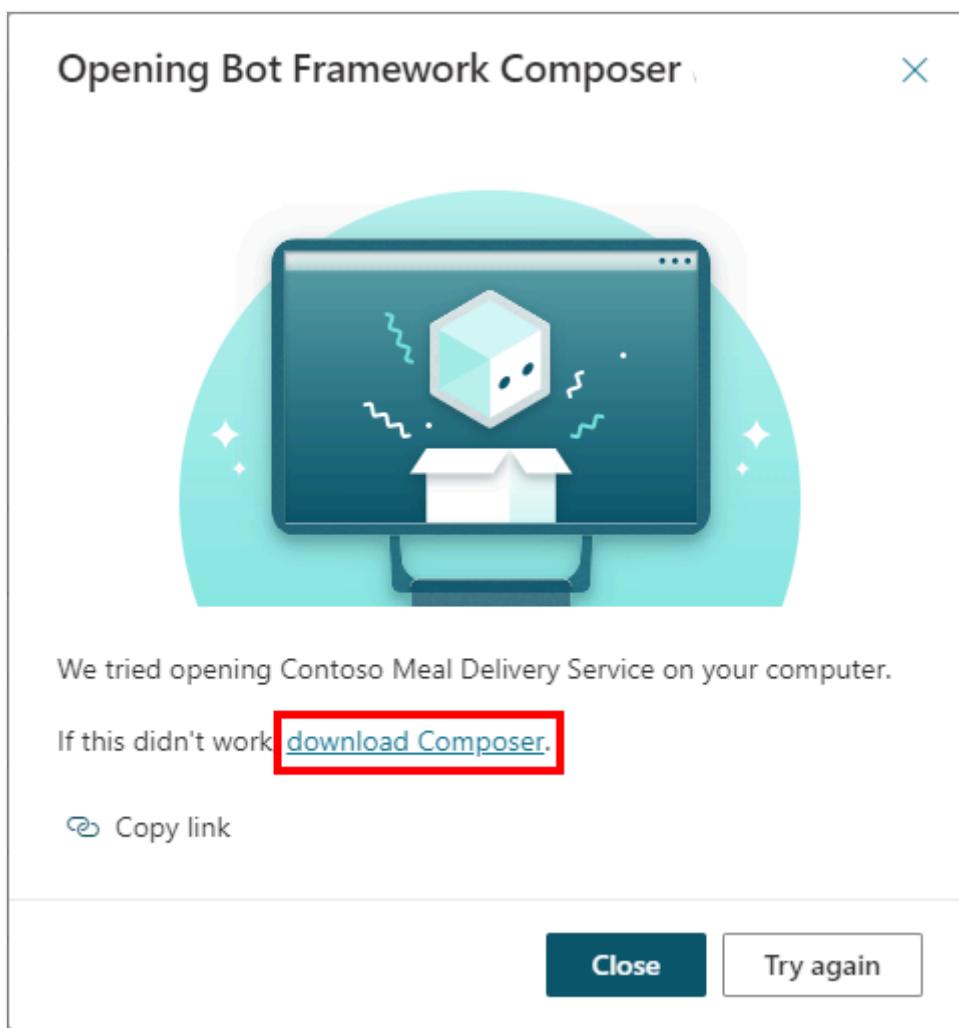
You should always open Composer from within Microsoft Copilot Studio. This will ensure that Composer has all the necessary plug-ins needed for integration with Microsoft Copilot Studio.

1. Go to the **Topics** page for the bot you want to edit and select the down-arrow symbol next to **+ New topic**.
2. Select **Open in Bot Framework Composer** in the dropdown menu to open your bot in Composer.



3. If you don't have Composer installed already:

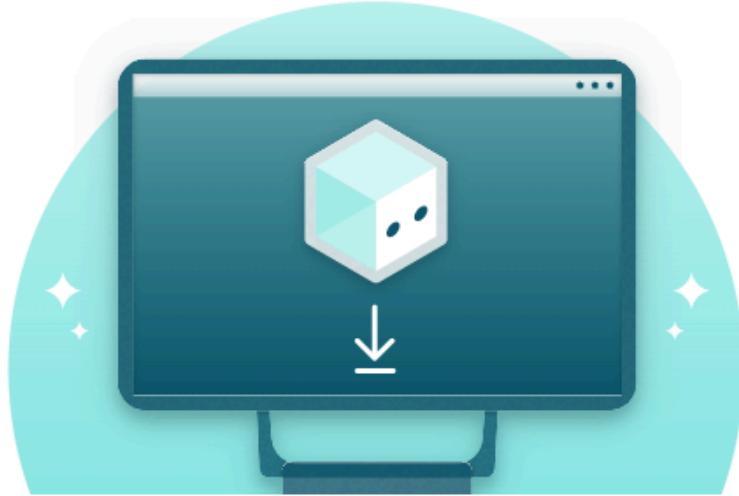
- Select **download Composer**.



- In the **Download Bot Framework Composer** window, select **Download Composer**.

Download Bot Framework Composer

X



Enables developers and multi-disciplinary teams to extend bot capabilities with complex dialogs, adaptive cards, logical expressions, event triggers, interruption handling, and more.

[Learn more](#) about Bot Framework Composer.

Bot Framework Composer is free, and no Azure subscription is needed.

Download for [Windows](#) or [Mac](#).

[Download Composer](#)

Close

- c. When the download completes, run it to install Composer.
- d. Once you've installed Composer, go back to Microsoft Copilot Studio. Go to the **Topics** page and select the down-arrow symbol next to **+ New topic**.

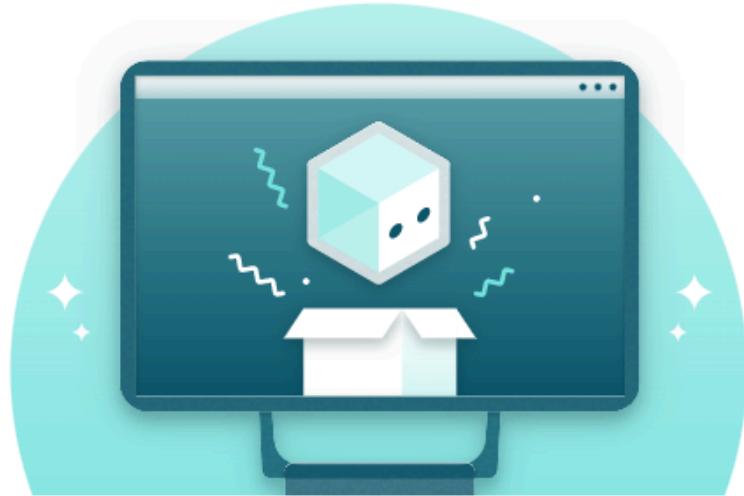
① Note

In some cases, a firewall might be blocking some features in Composer and you might need to explicitly allow access. Check with your admin for details.

4. If Microsoft Copilot Studio is unable to launch Composer automatically, you can launch it manually:
 - a. Select **Copy Link** to copy the link to your clipboard.

Opening Bot Framework Composer

X



We tried opening SSO Bot 08092022 on your computer.

If this didn't work, [download Composer](#).

 Copy link

[Close](#)

[Try again](#)

- b. Open a new tab and paste the link in your web browser's address bar.
5. The first time you open a Microsoft Copilot Studio bot in Composer, it will be imported into a new Composer project. The name you specified in Microsoft Copilot Studio will automatically be used as the name for the project, but you can also enter a custom name.

Import your bot to new project

X

Specify a name, description, and location for your new bot project.

Name *

Contoso-Meal-Delivery-Service

Location

[Create new folder](#)

↓ Name

Date modified

.. a few seconds ago

3D Objects 7 months ago

AppData 7 months ago

[Cancel](#)

[Create](#)

6. Select **Create** to finish importing your bot into Composer.

Import your bot to new project

X

Specify a name, description, and location for your new bot project.

Name *

Contoso-Meal-Delivery-Service

Location

Create new folder

↓	Name	Date modified
	..	a few seconds ago
	3D Objects	7 months ago
	AppData	7 months ago

CancelCreate

Test Composer content within Microsoft Copilot Studio

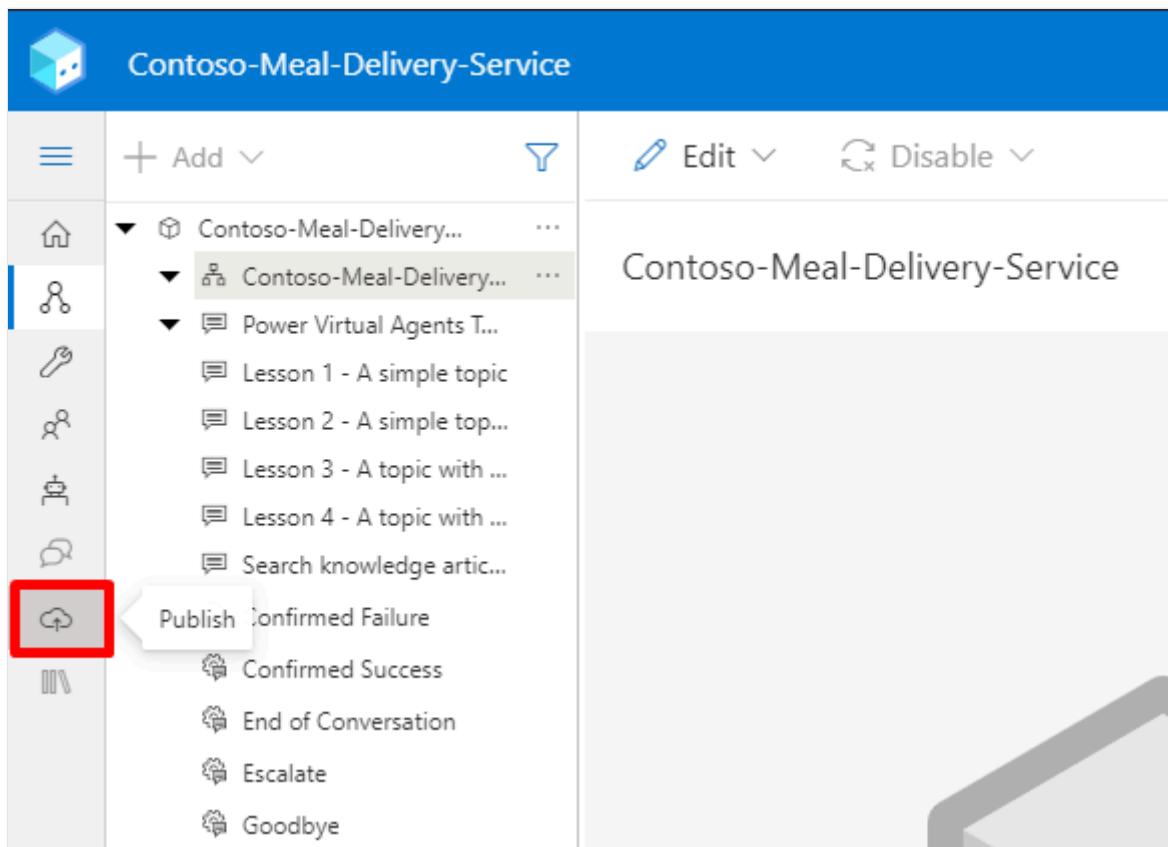
You can't use the Bot Framework Emulator to test Composer content created for Microsoft Copilot Studio bots. Instead you'll need to publish your Composer content to your Microsoft Copilot Studio bot, then use the [Test bot pane](#).

Important

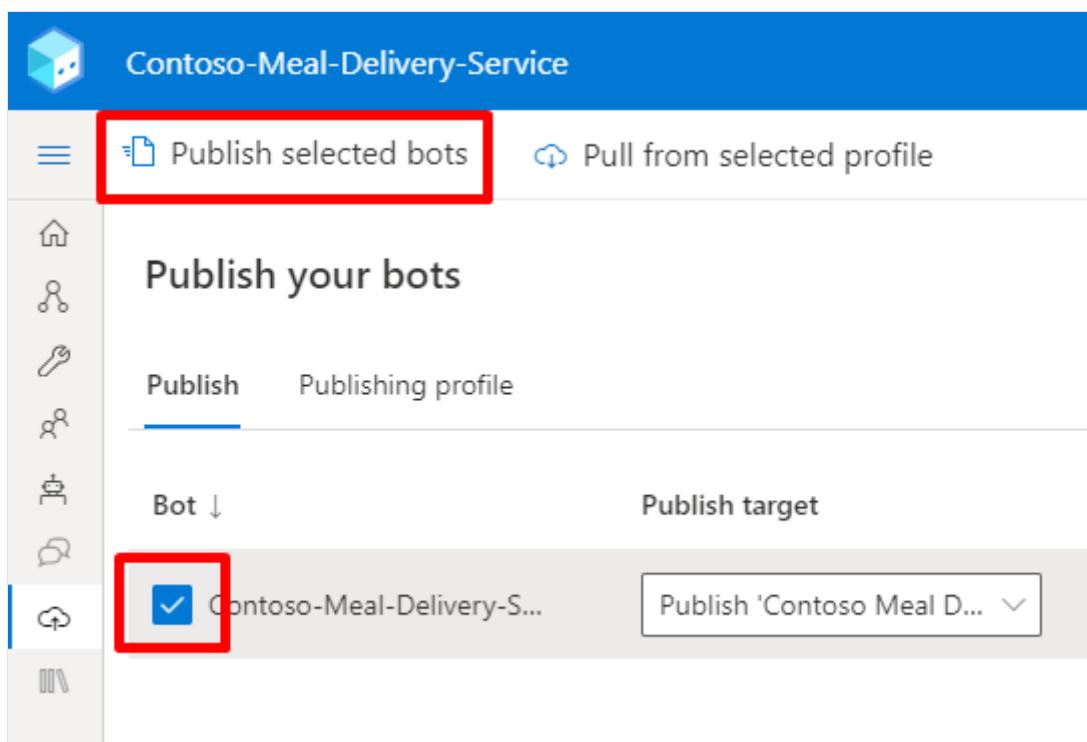
Selecting **Publish** in Composer makes the changes available for testing, but doesn't automatically publish your bot.

[Publish](#) your bot in Microsoft Copilot Studio to update your bot across all the channels it's connected to.

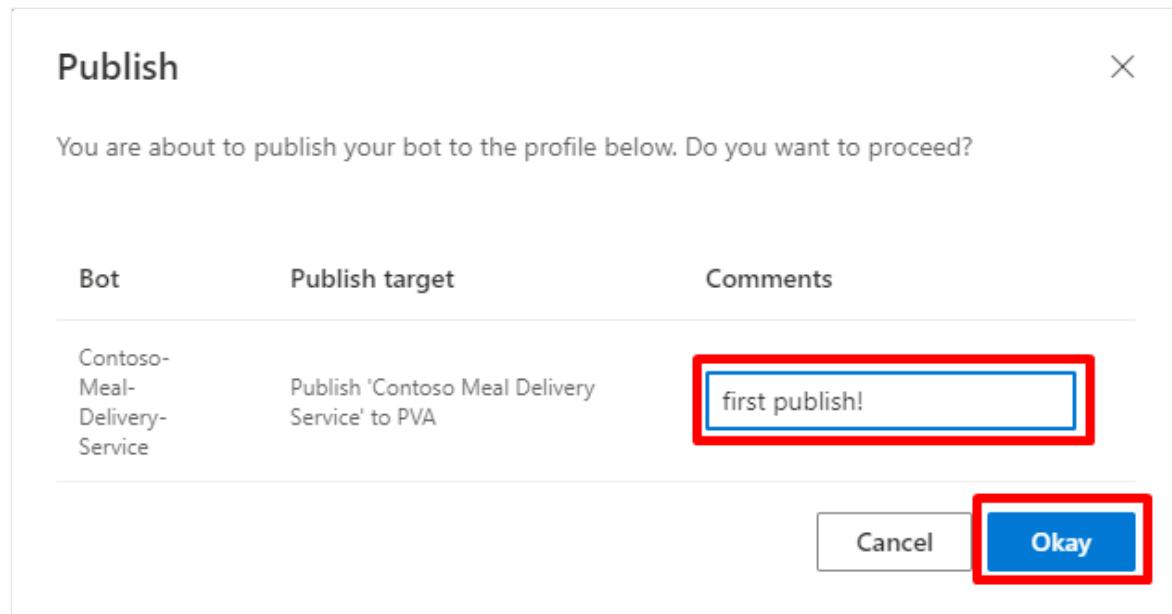
1. Go to the **Publish** page.



2. Select your bot in the **Bot** list, then select **Publish selected bots**.



3. Optionally enter a message for **Comments**, then select **Okay**.



4. Once publishing completes, go to Microsoft Copilot Studio. You may need to refresh the page to see the new topics you've added in Composer.
5. Open the **Test bot** pane and start a conversation by typing in a trigger phrase.

View the publish log

If you're having trouble publishing your Composer content to Microsoft Copilot Studio, you can troubleshoot by reviewing the publish log. The publish log provides a detailed description of any warnings or errors encountered during validation and suggestions on how to fix them.

ⓘ Note

If you're having trouble with your Microsoft Copilot Studio bot, use the [topic checker](#) instead.

1. In Composer, go to the **Publish** page.
2. Select the right arrow (>) next to your bot. You may need to increase the size of the Composer window to make the arrow visible.

The screenshot shows the Microsoft Bot Framework Composer interface. In the top navigation bar, there are icons for Start bot, Publish selected bots, Pull from selected profile, and other settings. Below the navigation, there's a sidebar with icons for Home, Bot, Profile, and others. The main area is titled "Publish your bots" and has tabs for "Publish" and "Publishing profile". Under the "Publish" tab, there's a table showing a single publish record:

Bot	Publish target	Date	Status	Message	Comment
Contoso-Meal-Delivery-S...	Publish 'Contoso Meal D...'	11-03-2022	✓	Publish successful.	first publish!

A red box highlights the "View log" button located at the end of the first row in the "Comment" column.

3. Under Publish history, select View log.

This screenshot is similar to the previous one, but it includes a "Publish history" section below the main table. The "Publish history" section has columns for Time, Date, Status, Message, and Comment. It lists the same publish entry as the main table. A red box highlights the "View log" button located in the "Comment" column of the first row.

4. In the Publish log window, review logs for any errors or warnings.

This screenshot shows the "Publish Log" window. The title bar says "Publish Log" and there's a close button. The main content area displays a JSON-like log entry:

```
  "start": {  
    "line": 40,  
    "character": 1  
  },  
  "end": {  
    "line": 40,  
    "character": 46  
  },  
  "severity": "Error",  
  "source": "BingSearch.en-us.lg",  
  "message": "Error occurred when parsing expression 'AdaptiveCardBingSearch(turn.activity.text)'.  
AdaptiveCardBingSearch does not have an evaluator, it's not a built-in function or a custom  
function."  
}
```

A red box highlights the "View log" button in the top right corner of the log window.

If you encounter issues developing in Composer, you can:

- Go to [Bot Framework Composer issues on GitHub ↗](#).
- See the list of known issues in Composer or [log a new Composer issue ↗](#).

Next steps

To help you get started using Composer, we've created four examples that detail different scenarios:

- Show an Adaptive Card in Copilot Studio
 - Display a multi-select options list in Copilot Studio
 - Use Bing Search as a fallback in Copilot Studio
 - Display a form with a Submit button in Copilot Studio
-

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

Show an Adaptive Card in Copilot Studio

Article • 08/26/2024

ⓘ Note

This article applies to *classic chatbots* only. To extend *copilots* created in Copilot Studio, you don't need to use Bot Framework Composer. All features described in this article are available for copilots, directly in Copilot Studio.

Enhance your bot by developing custom dialogs with [Bot Framework Composer](#) and then adding them to your Microsoft Copilot Studio bot.

In this example, you'll learn how to show an Adaptive Card in Microsoft Copilot Studio by using Composer.

ⓘ Important

Composer integration isn't available to users who only have the [Teams Microsoft Copilot Studio license](#). You must have a [trial](#) or full Microsoft Copilot Studio license.

Prerequisites

- [Introduction to Bot Framework Composer](#).
- [Get started with Bot Framework Composer](#).

Create a new topic

1. In Microsoft Copilot Studio, create a new bot named `Contoso Meal Delivery Service`.
2. Go to the **Topics** page and create a new topic called `Meal delivery options`.
3. Copy and paste the following trigger phrases:

plaintext

what meals can I order
what meal options do you have
what dishes do you deliver
entrees available for delivery

4. On the **Authoring canvas**, select the default **Message** node. Select the three vertical dots, then select **Delete**.
5. Under the **Trigger phrases** node, select **Add node (+)** and then select **Ask a question**. Then in the **Question** node:
 - a. For **Ask a question**, enter `What city are you in?`.
 - b. For **Identify**, select **City**.
6. In **Save response as**, select the automatically created variable. Then in the variable properties pane:
 - a. For **Name**, enter `user_city`.
 - b. For **Usage**, select **Bot (any topic can access)**.

7. Select **Save**.

The screenshot shows the Dialogflow Authoring canvas with a **Question** node selected. The node has the following configuration:

- Ask a question:** What city are you in?
- Identify:** City
- Save response as:** {x} bot.user_city (city)

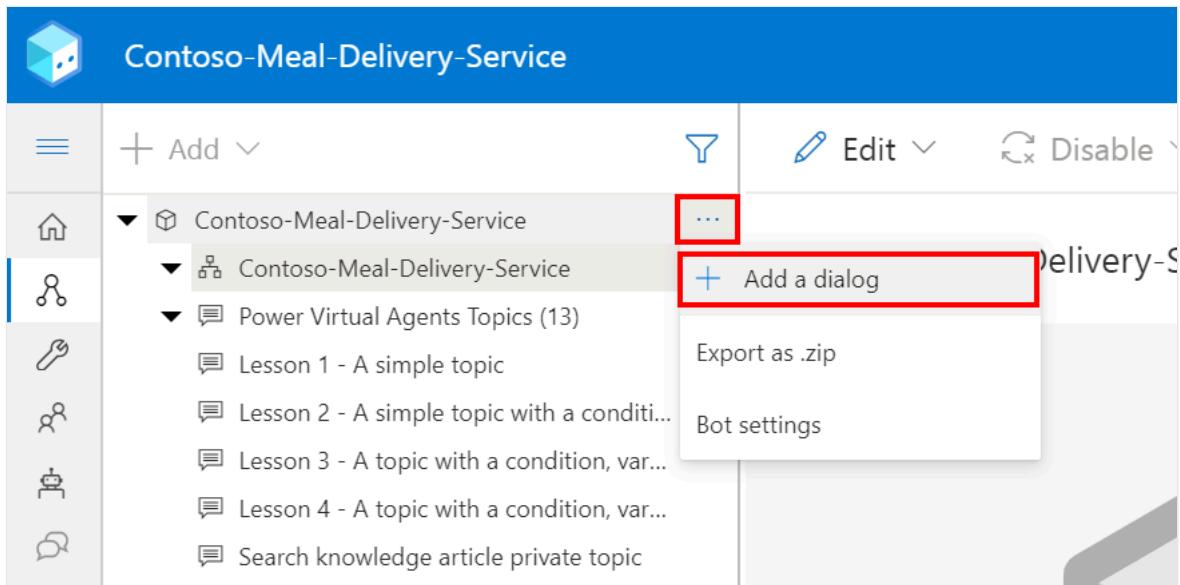
To the right, the **Variable properties** pane is open for the variable `user_city`:

- Name ***: bot. user_city
- Type**: City
- Source**: Go to Source
- Usage**:
 - Topic (limited scope)
 - Receive values from other topics
 - Return values to original topics
 - Bot (any topic can access)
 - External sources can set values

Create an adaptive card

Next, you'll display meal choice images that are available in the user's city using Composer.

1. Open your bot in Composer. For instructions on how to do so, see [Get started with Bot Framework Composer](#).
2. Go to the **Create** page. In your bot, select **More options (...)** then select **+ Add a dialog**.



3. For **Name**, enter `Meals` then select **OK**.

Create a dialog

X

Specify a name and description for your new dialog.

Name *

Meals

Description

Cancel

OK

4. Go to the **Bot Responses** page. In the bot explorer, select **Meals**, then select **Show code**.

The screenshot shows the 'Hello-World' bot's 'Bot Responses' page. On the left, a sidebar menu has 'Bot responses' selected. The main area displays a table of responses:

Name	Responses
New template	
Meals	

A red box highlights the 'Show code' button in the top right corner of the main area. Another red box highlights the 'Meals' row in the table.

5. In the code view, copy and paste the following bot response. Then replace the three example image URLs with your own.

Bot response

```
# Chicken()  
-Chicken
```

```

# Steak()
-Steak

# Tofu()
-Tofu

# SteakImageURL()
-https://www.example.com/steak.jpg

# ChickenImageURL()
-https://www.example.com/chicken.jpg

# TofuImageURL()
-https://www.example.com/tofu.jpg

```

The screenshot shows the Microsoft Bot Framework Emulator interface. The title bar says "Contoso-Meal-Delivery-Service". The main area is titled "Bot Responses" and contains the following code:

```

[import](common.1g)
# Chicken()
-Chicken
# Steak()
-Steak
# Tofu()
-Tofu
# SteakImageURL()
-https://contososcubademo.azurewebsites.net/assets/steak.jpg
# ChickenImageURL()
-https://contososcubademo.azurewebsites.net/assets/chicken.jpg
# TofuImageURL()
-https://contososcubademo.azurewebsites.net/assets/tofu.jpg

```

The code block is highlighted with a red rectangle.

6. Copy and paste the following code into the same code view to add an Adaptive Card that displays three images.

The screenshot shows the Microsoft Bot Framework Emulator interface. The title bar says "Bot response". The code view contains the following JSON code:

```

# adaptivecardjson_meals(location)
```
{
 "type": "AdaptiveCard",
 "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
 "version": "1.1",
 "body": [
 {
 "type": "TextBlock",

```

The code block is highlighted with a red rectangle.

```
 "text": "Meal delivery options for ${location}:",
 "size": "Medium",
 "weight": "Bolder"
 },
 {
 "type": "ColumnSet",
 "columns": [
 {
 "type": "Column",
 "width": "stretch",
 "items": [
 {
 "type": "Image",
 "url": "${SteakImageURL()}",
 "size": "Stretch",
 "spacing": "Medium",
 "horizontalAlignment": "Center"
 }
]
 },
 {
 "type": "Column",
 "width": "stretch",
 "items": [
 {
 "type": "Image",
 "url": "${ChickenImageURL()}",
 "horizontalAlignment": "Center"
 }
]
 },
 {
 "type": "Column",
 "width": "stretch",
 "items": [
 {
 "type": "Image",
 "url": "${TofuImageURL()}",
 "horizontalAlignment": "Center"
 }
]
 }
],
 "type": "ColumnSet",
 "columns": [
 {
 "type": "Column",
 "width": "stretch",
 "items": [
 {
 "type": "TextBlock",
 "text": "${Steak()}",
 "wrap": true,
 }
]
 }
]
 }
]
```

```
 "horizontalAlignment": "Center"
 }
]
},
{
 "type": "Column",
 "width": "stretch",
 "items": [
 {
 "type": "TextBlock",
 "text": "${Chicken()}",
 "wrap": true,
 "horizontalAlignment": "Center"
 }
]
},
{
 "type": "Column",
 "width": "stretch",
 "items": [
 {
 "type": "TextBlock",
 "text": "${Tofu()}",
 "wrap": true,
 "horizontalAlignment": "Center"
 }
]
}
]
}
```
```
```

Contoso-Meal-Delivery-Service

Start bot

Bot Responses

Hide code

Contoso-Meal-Delive

- Common
- Contoso-Meal-Delive
- Meals
- Power Virtual Ag
- Lesson 1 - A simj
- Lesson 2 - A simj
- Lesson 3 - A topi
- Lesson 4 - A topi
- Confirmed Failur
- Confirmed Succes
- End of Conversat
- Escalate
- Goodbye
- Greeting
- Start over
- Thank you

```
12 "SteakImageURL()")
13 -https://contososcubademo.azurewebsites.net/assets/steak.jpg
14 # ChickenImageURL()
15 -https://contososcubademo.azurewebsites.net/assets/chicken.jpg
16 # TofuImageURL()
17 -https://contososcubademo.azurewebsites.net/assets/tofu.jpg
18
19
20 # adaptivecardjson_meals(location)
21 - ``
22 {
23 "type": "AdaptiveCard",
24 "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
25 "version": "1.1",
26 "body": [
27 {
28 "type": "TextBlock",
29 "text": "Meal delivery options for ${location}:",
30 "size": "Medium",
31 "weight": "Bolder"
32 },
33 {
34 "type": "ColumnSet",
35 "columns": [
36 {
37 "type": "Column",
38 "width": "stretch",
39 "items": [
40 {
41 "type": "Image",
42 "url": "${SteakImageURL()}",
43 "size": "Stretch",
44 }
45]
46 }
47]
48 }
49]
50 }
```

7. Copy and paste the following code into same code view to add an activity that will display the Adaptive Card.

Bot response

```
AdaptiveCardMeals(location)
[Activity
 Attachments = ${json(adaptivecardjson_meals(location))}]
]
```

The screenshot shows the Microsoft Bot Framework Composer interface. In the top navigation bar, there is a blue header with the text "Contoso-Meal-Delivery-Service". To the right of the header are several icons: a play button, a gear, a magnifying glass, and a refresh symbol. Below the header, there is a sidebar on the left containing a tree view of bot components. The tree view shows a root node "Contoso-Meal-Delivery-Service" with two children: "Common" and "Contoso-Meal-Delivery-Service". Under "Contoso-Meal-Delivery-Service", there is a "Meals" node, which is expanded to show "Power Virtual Agents 1" and several specific dialog nodes: "Lesson 1 - A simple to", "Lesson 2 - A simple to", "Lesson 3 - A topic with", "Lesson 4 - A topic with", "Confirmed Failure", "Confirmed Success", "End of Conversation", "Escalate", "Goodbye", "Greeting", "Start over", and "Thank you". On the right side of the interface, there is a large code editor window titled "Bot Responses". The code is written in JSON-like syntax, likely for an Adaptive Card. The code includes several numbered lines (93 to 118) and a red box highlighting lines 119 through 122. The highlighted code is as follows:

```
119 # AdaptiveCardMeals(location)
120 [Activity
121 Attachments = ${json(adaptivecardjson_meals(location))}
122]
```

## Display your Adaptive Card

1. Go to the **Create** page. In the bot explorer, select the **Meals** dialog then select the **BeginDialog** trigger.
2. On the authoring canvas, select **Add (+)** then select **Send a response**.
3. Select the new **Send a response** node to open the properties pane. Under **Bot responses**, select **Show code** to switch to the code editor.

The screenshot shows the Microsoft Copilot Studio Composer interface. On the left, a flowchart starts with a 'BeginDialog' event, followed by a 'Send a response' step, and ends with a final event. The 'Send a response' step is highlighted with a blue border. On the right, the 'Send a response' card is displayed with the title 'Send a response', a note 'Send Activity', a 'Learn more' link, and a 'Show code' button which is highlighted with a red box. Below this, there are sections for 'Bot responses' (Text and Responses), and an 'Add alternative' link.

### ⚠️ Warning

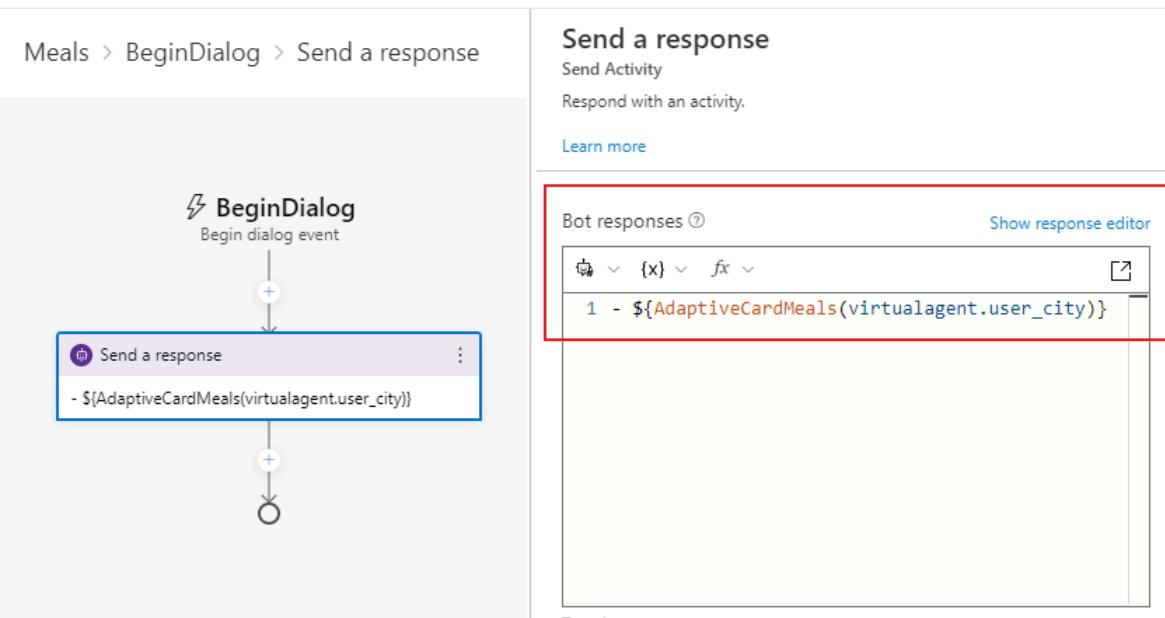
Adding the expression in the next step to the *response* editor instead of the *code* editor will result in the bot responding with raw JSON instead of an Adaptive Card.

4. Copy and paste the following expression into the code editor.

The screenshot shows the 'Bot response' code editor. It contains a single line of code: '- \${AdaptiveCardMeals(virtualagent.user\_city)}'.

Microsoft Copilot Studio global variables are accessed in Composer by using the `virtualagent` scope. This scope won't appear in Composer's property reference menu, but you can access it by entering an expression directly.

In this example,  `${virtualagent.user_city}` refers to the `user_city` global variable that was created in the Microsoft Copilot Studio bot.



## Publish your content

1. [Publish your Composer content](#) to make it available in your Microsoft Copilot Studio bot.

i **Important**

Selecting **Publish** in Composer makes the changes available for testing, but doesn't automatically publish your bot.

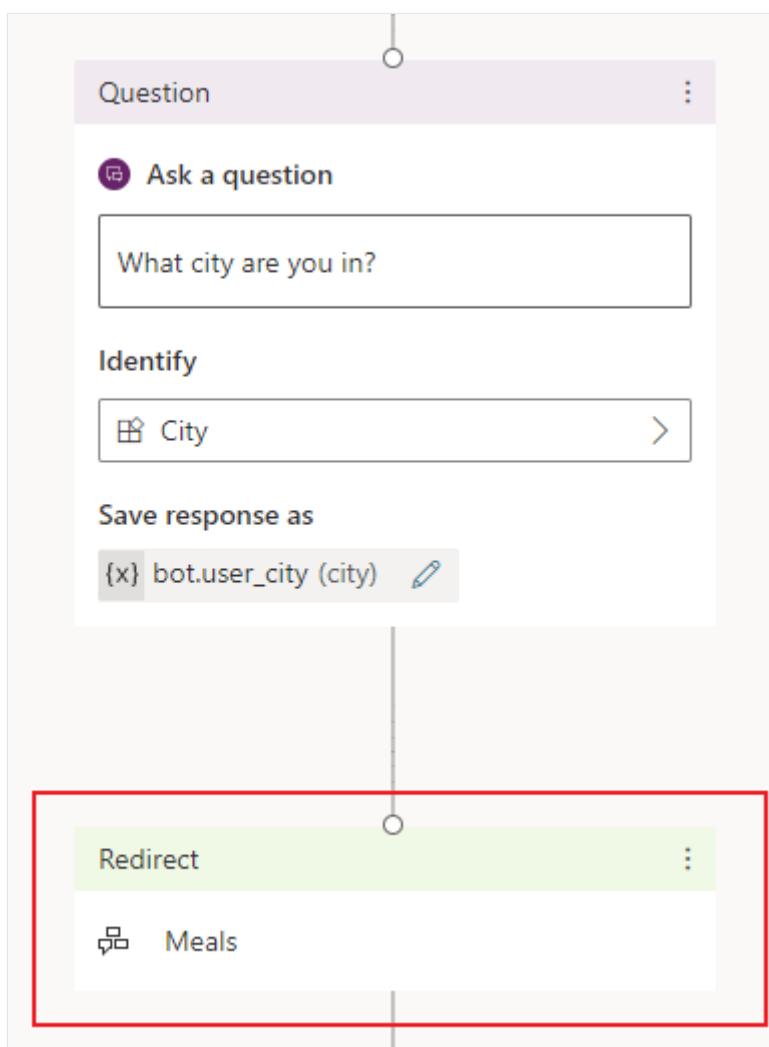
[\*\*Publish\*\*](#) your bot in Microsoft Copilot Studio to update your bot across all the channels it's connected to.

2. Go to the Microsoft Copilot Studio **Topics** page to see your new **Meals** topic.

The screenshot shows the 'Topics' section of the Microsoft Bot Framework interface. On the left, a sidebar includes options like Chatbots, Overview, Topics (selected), Entities, Analytics, Publish, Settings, and Test your bot. The main area displays 'Existing (14)' topics. A table lists topics by Type (e.g., meals), Name, and Trigger phrases. The 'Meals' topic is highlighted with a red border.

| Type                  | Name                  | Trigger phrases                    |
|-----------------------|-----------------------|------------------------------------|
| Meals                 | Meals                 | No trigger phrases                 |
| Meal delivery options | Meal delivery options | (4) entrees available for delivery |

3. Open the **Meal delivery options** topic.
4. Under the **Question** node, select **Add node (+)**. Select **Redirect to another topic** then choose **Meals**.

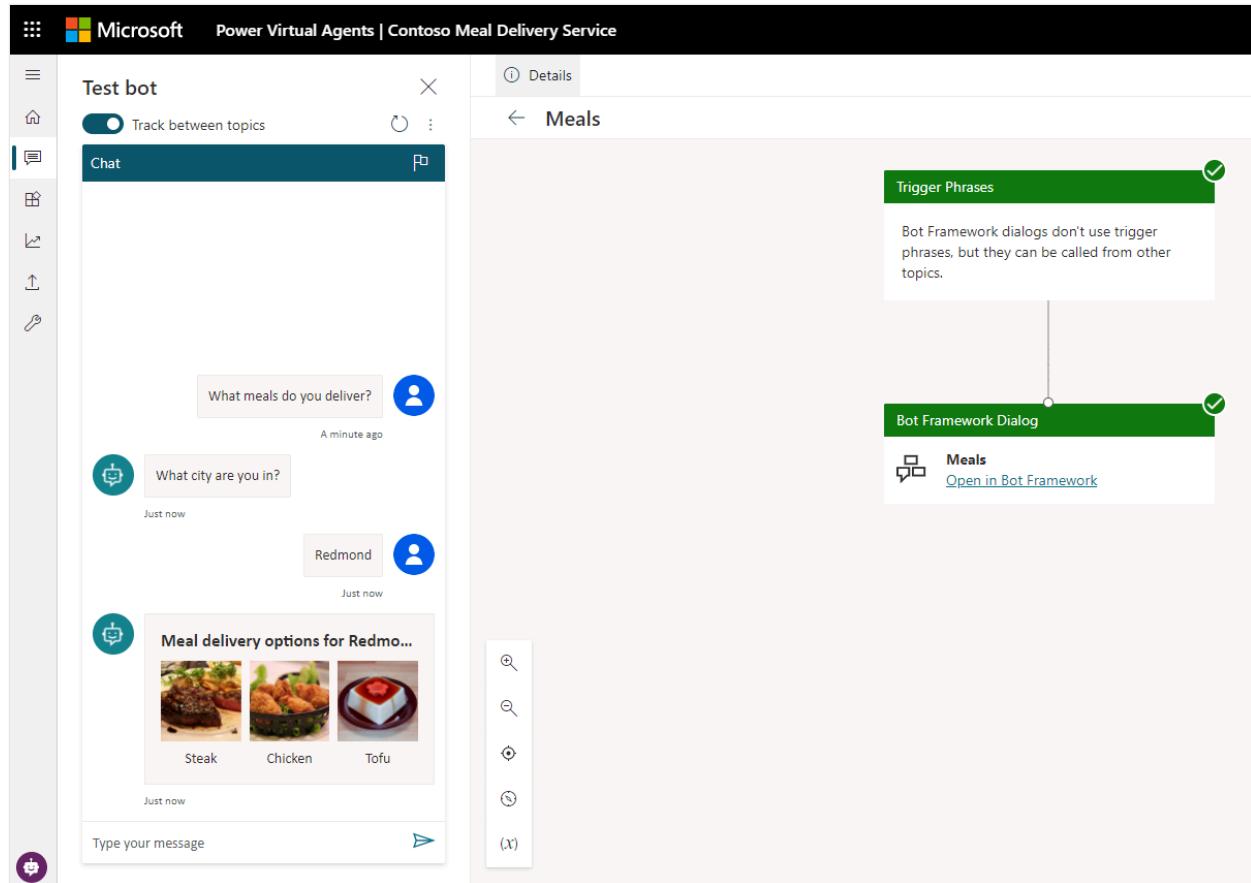


5. Select **Save** to save the changes to your topic.

# Test your bot

To test your changes in Microsoft Copilot Studio, open the **Test bot pane** and make sure **Track between topics** is turned on. Enter the message `What meals do you deliver?` to start your conversation.

Your bot will trigger the **Meal delivery options** topic, which asks the user for their current city. Then the bot will redirect to the Composer dialog **Meals** to display an Adaptive Card.



## Next step

Display a multi-select options list in Copilot Studio.

## Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

# Display a multi-select options list in Copilot Studio

Article • 11/12/2024

## ⓘ Note

This article applies to *classic chatbots* only. To extend *copilots* created in Copilot Studio, you don't need to use Bot Framework Composer. All features described in this article are available for copilots, directly in Copilot Studio.

Enhance your bot by developing custom dialogs with [Bot Framework Composer](#) and then adding them to your Microsoft Copilot Studio bot.

In this example, you'll learn how to display a multi-select list in Microsoft Copilot Studio by using Composer.

## ⓘ Important

Composer integration isn't available to users who only have the [Teams Microsoft Copilot Studio license](#). You must have a [trial](#) or full Microsoft Copilot Studio license.

## Prerequisites

- Show an Adaptive Card in Copilot Studio.

## Create a new dialog

1. In Copilot Studio, open the bot you created in Example 1. If you haven't completed Example 1 yet, see [Show an Adaptive Card in Copilot Studio](#).
2. Open your bot in Composer. For instructions on how to do so, see [Get started with Bot Framework Composer](#).
3. In Composer, go to the **Create** page. In your bot, select **More options (...)** then select **+ Add a dialog**. For **Name** enter `DailySpecials`.

4. In your new **DailySpecials** dialog, select the **BeginDialog** trigger to open the authoring canvas.
5. On the authoring canvas, select **Add (+)**, **Manage properties**, and **Set a property**.
6. In the properties pane:
  - a. For **Property**, enter `conversation.days_array`.
  - b. For **Value**, change the type to **[] array**.
  - c. For **Value**, copy and paste the following array

JSON

```
["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
 "Saturday"]
```

## Configure bot responses

1. Go to the **Bot Responses** page, select **DailySpecials**, then select **Show code**.
2. Copy and paste the following template into the code view to create daily offers for all the days of the week.

Language understanding

```
DailySpecials(day)
- SWITCH: ${day}
- CASE: ${0}
 - All tofu meals are 10% off on Sundays!
 - Every Sunday, all tofu entrees are 10% off.
- CASE: ${1}
 - All steak options are 10% off on Mondays!
 - Enjoy your Monday with a special offer of 10% off on all steak
 dishes!
- CASE: ${2}
 - All the chicken meal options are 10% off on Tuesdays!
 - Tuesday special is 10% off on all the chicken dishes!
- CASE: ${3}
 - All the chicken and tofu meal options are 10% off on Wednesdays!
 - Wednesday special is 10% off on all the chicken and tofu dishes!
- CASE: ${4}
 - On Thursdays, get a free delivery in Seattle, Bellevue, and
 Redmond on all orders over $80!
 - Thursday special is a free delivery on orders over $80 in
 Seattle, Bellevue, and Redmond.
- CASE: ${5}
 - Friday special - get a 10% discount on all dishes and delivery is
```

```
free on all orders over $80!
- Every Friday, we offer 10% off on all meals and a free delivery
on orders over $80!
- CASE: ${6}
- On Saturdays, we have a free delivery on all orders over $50.
- Free delivery on all orders over $50 on Saturdays!
- DEFAULT:
- Holiday special - free delivery anywhere in Seattle, Bellevue and
Redmond on orders over $70 today!
- Holiday Delivery is on us if you are in Seattle, Bellevue and
Redmond and your order is over $70 total!
```

## Prompt for user input

1. Go to the **Create** page. In the bot explorer, under **DailySpecials**, select **BeginDialog**.
2. On the authoring canvas, select **Add (+)**, **Ask a question**, and **Multi-choice**.
3. Select the **User Input (Choice)** action. In the properties pane, under **Responses**, enter `Please select a day:`
4. Select the **User Input** page in the properties pane, then do the following:
  - a. For **Property**, enter `conversation.day_choice`.
  - b. For **Output format**, choose **index** to return the index of the selected option instead of a value.
  - c. For **List style**, choose **heroCard** to display our options list vertically.
  - d. For **Array of choices**, change the type to **Write an expression**.
  - e. For **Array of choices**, enter `=conversation.days_array`.

## Display daily special

1. On the authoring canvas, select **Add (+)** then **Send a response**.
2. In the properties pane, under **Bot responses**, and select **Show code**. Copy and paste the following expression:

Bot response

```
- ${DailySpecials(conversation.day_choice)}
```

3. In the bot explorer, go to the main dialog. Select **More options (...)** then select **Add new trigger**.

4. In the **Create a trigger** window, do the following:

a. For **What is the type of this trigger?**, select **Intent recognized**.

b. For **What is the name of this trigger?**, enter **Specials**.

c. For **Trigger phrases**, copy and paste the following:

Language understanding

-what specials do you have  
-any special deals  
-do you have discounts

d. Select **Submit**.

5. In the bot explorer, select the **Specials** trigger.

6. On the authoring canvas, select **Add (+)**, **Dialog management**, then **Begin a new dialog or topic**.

7. In the properties pane, for **Dialog name**, select **DailySpecials**

## Test your bot

1. [Publish your Composer content](#) to make it available in your Copilot Studio bot.

### i Important

Selecting **Publish** in Composer makes the changes available for testing, but doesn't automatically publish your bot.

[\*\*Publish\*\*](#) your bot in Microsoft Copilot Studio to update your bot across all the channels it's connected to.

2. Go to the Microsoft Copilot Studio **Topics** page to see your new **Specials** and **DailySpecials** topics.

3. Open the **Test bot pane** and make sure **Track between topics** is turned on. Enter the message **Do you have any specials?** to start your conversation.

# Next step

Use Bing Search as a fallback in Copilot Studio.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Use Bing Search as a fallback in Copilot Studio

Article • 09/20/2024

## ⓘ Note

This article applies to *classic chatbots* only. To extend *copilots* created in Copilot Studio, you don't need to use Bot Framework Composer. All features described in this article are available for copilots, directly in Copilot Studio.

Enhance a classic chatbot by developing custom dialogs with [Bot Framework Composer](#), and then adding them to your chatbot in Copilot Studio.

In this example, you learn how to set Bing search as a fallback answer in Copilot Studio by using Composer.

## ⓘ Important

Composer integration isn't available to users who only have the [Teams Microsoft Copilot Studio license](#). You must have a [trial](#) or full Microsoft Copilot Studio license.

## Prerequisites

- Show an Adaptive Card in Copilot Studio
- Display a multi-select options list in Copilot Studio

## Create a fallback trigger

1. In Copilot Studio, open the chatbot from [Example 2](#).
2. Open your chatbot in Composer. For instructions on how to do so, see [Get started with Bot Framework Composer](#).
3. In the bot explorer, go to the main dialog. Select **More options (...)** then select **Add new trigger**.

4. In the **Create a trigger** window, for **What is the type of this trigger?**, select **Unknown intent**. Select **Submit**.
5. Go to the **Bot Responses** page, select **Contoso Meal Delivery Service**, then select **Show code**.
6. Copy and paste the following code into the code view:

```
Bot response

adaptivecardjson_BingSearch(user_utterance)
```
{
    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
    "type": "AdaptiveCard",
    "version": "1.2",
    "fallbackText": "This card requires Media to be viewed. Ask your platform to update to Adaptive Cards v1.1 for this and more!",
    "actions": [
        {
            "type": "Action.OpenUrl",
            "title": "Search with Bing",
            "url": "https://www.bing.com/search?q=${user_utterance}",
            "style": "positive"
        }
    ],
    "body": [
        {
            "type": "Image",
            "url": "https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/am/imageFileData/RWe65Z?ver=2d4e&q=90&m=6&h=201&w=358&b=%23FFFFFF&l=f&o=t&aim=true"
        },
        {
            "type": "TextBlock",
            "text": "${user_utterance}",
            "wrap": true,
            "separator": true,
            "horizontalAlignment": "Center",
            "size": "Medium"
        }
    ]
}
```

```

The screenshot shows the Microsoft Bot Framework Emulator interface. The top navigation bar includes File, Edit, View, Window, Help, and a bot icon labeled "Contoso-Meal-Delivery-Service". On the right side, there are buttons for "Start bot", a list icon, a copy icon, a delete icon, and a refresh icon. The main area is titled "Bot Responses" and contains a code editor. The code is an AdaptiveCard JSON snippet:

```
1 > To learn more about the LG file format, read the documentation at
2 > https://aka.ms/lg-file-format
3
4 # adaptivecardjson_BingSearch(user_utterance)
5 - ``
6 {
7 "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
8 "type": "AdaptiveCard",
9 "version": "1.2",
10 "fallbackText": "This card requires Media to be viewed. Ask your platform to",
11 "actions": [
12 {
13 "type": "Action.OpenUrl",
14 "title": "Search with Bing",
15 "url": "https://www.bing.com/search?q=\${user_utterance}",
16 "style": "positive"
17 }
18],
19 "body": [
20 {
21 "type": "Image",
22 "url": "https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/"
23 },
24 {
25 "type": "TextBlock",
26 "text": "${user_utterance}",
27 "wrap": true,
28 "separator": true,
29 "horizontalAlignment": "Center",
30 "size": "Medium"
31 }
32]
}
```

7. Copy and paste the following code into the same code view:

The screenshot shows the Microsoft Bot Framework Emulator interface with a code editor. The title bar says "Bot response". The code in the editor is:

```
AdaptiveCardBingSearch(user_utterance)
[Activity
 Attachments = ${json(adaptivecardjson_BingSearch(user_utterance))}]
]
```

```

File Edit View Window Help
Contoso-Meal-Delivery-Service
Start bot
Bot Responses
Hide code
Contoso-Meal-Delivery-Service
Common
Contoso Meal Delivery Service
Meals
DailySpecials
Power Virtual Agents Topics (13)
Lesson 1 - A simple topic
Lesson 2 - A simple topic with ...
Lesson 3 - A topic with a conditi...
Lesson 4 - A topic with a conditi...
Meal delivery options
Confirmed Failure
Confirmed Success
End of Conversation
Escalate
Goodbye
Greeting
Start over
Thank you
 "actions": [
 {
 "type": "Action.OpenUrl",
 "title": "Search with Bing",
 "url": "https://www.bing.com/search?q=${user_utterance}",
 "style": "positive"
 }
],
 "body": [
 {
 "type": "Image",
 "url": "https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/a..."
 },
 {
 "type": "TextBlock",
 "text": "${user_utterance}",
 "wrap": true,
 "separator": true,
 "horizontalAlignment": "Center",
 "size": "Medium"
 }
]
}
AdaptiveCardBingSearch(user_utterance)
[Activity
 Attachments = ${json(adaptivecardjson_BingSearch(user_utterance))}]
]

```

8. Go to the **Create** page.

9. In the main dialog, select the **Unknown intent** trigger.

10. On the authoring canvas, select **Add (+)** then **Send a response**.

11. In the properties pane, under **Bot responses**, and select **Show code**.

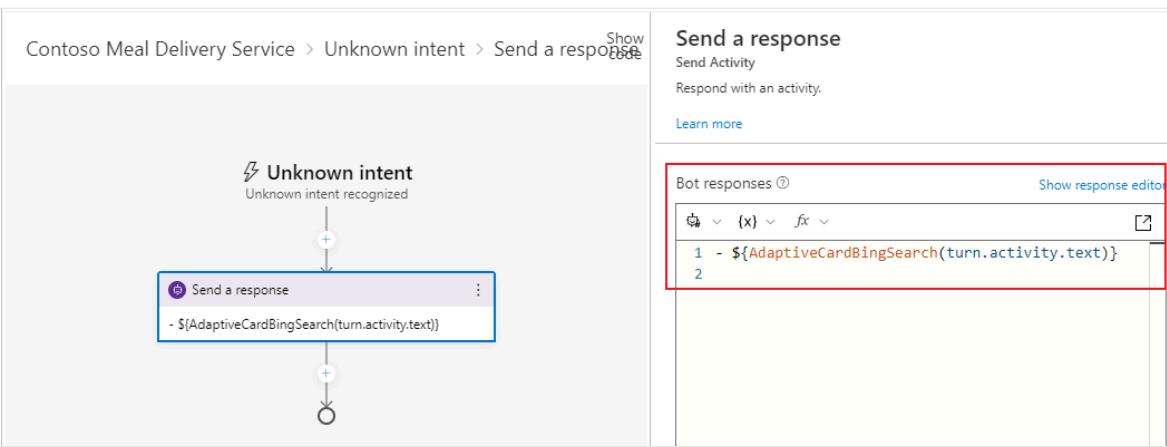
### ⚠ Warning

Adding the expression in the next step to the *response* editor instead of the *code* editor will result in the chatbot responding with raw JSON instead of an Adaptive Card.

12. Copy and paste the following expression:

Bot response

- \${AdaptiveCardBingSearch(turn.activity.text)}



## Test your chatbot

1. [Publish your Composer content](#) to make it available in your chatbot in Copilot Studio.

**ⓘ Important**

Selecting **Publish** in Composer makes the changes available for testing, but doesn't automatically publish your bot.

[\*\*Publish\*\*](#) your bot in Microsoft Copilot Studio to update your bot across all the channels it's connected to.

2. In Copilot Studio, go to the **Topics** page to see your new **OnUnknownIntent** topic.
3. Open the **Test bot pane** and make sure **Track between topics** is turned on. Enter "Is tofu vegan?" to start your conversation.

## Next step

[Display a form with a Submit button in Copilot Studio.](#)

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

# Display a form with a Submit button in Copilot Studio

Article • 11/12/2024

## ⓘ Note

This article applies to *classic chatbots* only. To extend *copilots* created in Copilot Studio, you don't need to use Bot Framework Composer. All features described in this article are available for copilots, directly in Copilot Studio.

Enhance your chatbot by developing custom dialogs with [Bot Framework Composer](#) and then adding them to a copilot created in Copilot Studio.

In this example, you'll learn how to display a form with a **Submit** button in Copilot Studio by using Composer.

## ⓘ Important

Composer integration isn't available to users who only have the [Teams Microsoft Copilot Studio license](#). You must have a [trial](#) or full Microsoft Copilot Studio license.

## Prerequisites

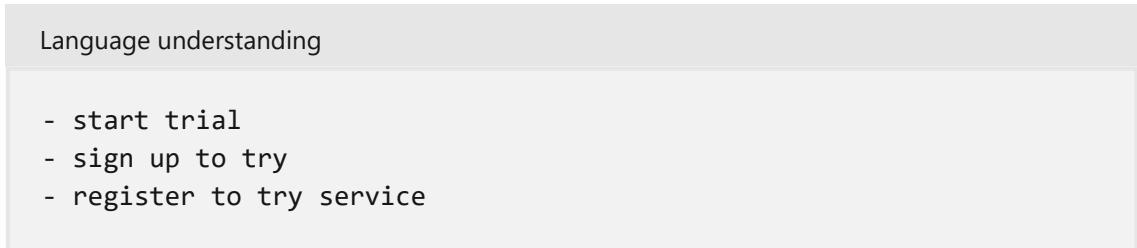
- Show an Adaptive Card in Copilot Studio.
- Display a multi-select options list in Copilot Studio.
- Use Bing Search as a fallback in Copilot Studio.

## Create a new trigger

1. In Copilot Studio, open the copilot from Example 3. If you haven't completed Example 3 yet, see [Use Bing Search as a fallback in Copilot Studio](#).
2. Open your bot in Composer. For instructions on how to do so, see [Get started with Bot Framework Composer](#).
3. In the bot explorer, go to the main dialog. Select **More options (...)** then select **Add new trigger**.

4. In the **Create a trigger** window:

- a. For **What is the type of this trigger?**, select **Intent recognized**.
- b. For **What is the name of this trigger?**, enter **StartTrial**.
- c. For **Trigger phrases**, copy and paste the following:



- d. Select **Submit**.

The 'Create a trigger' dialog box is shown. It contains the following fields:

- What is the type of this trigger?**: Set to **Intent recognized**.
- What is the name of this trigger?**: Set to **StartTrial**.
- Trigger phrases**: A list box containing:
  - + Add entity ▾
  - 1 - start trial
  - 2 - sign up to try
  - 3 - register to try service

At the bottom are **Cancel** and **Submit** buttons.

5. Go to the **Bot Responses** page, select **Contoso Meal Delivery Service**, then select **Show code**.

6. Copy and paste the following into the code view:

Bot response

```
adaptivecardjson_StartTrialForm()
```
{
    "type": "AdaptiveCard",
    "body": [
        {
            "type": "TextBlock",
            "size": "Medium",
            "weight": "Bolder",
            "text": "Register for a meal plan trial:"
        },
        {
            "type": "Input.Text",
            "placeholder": "Please enter your Name",
            "id": "Name"
        },
        {
            "type": "Input.Text",
            "placeholder": "Please enter your Address",
            "id": "Address",
            "isMultiline": true
        },
        {
            "type": "Input.Number",
            "placeholder": "How many weeks",
            "id": "Weeks"
        },
        {
            "type": "ActionSet",
            "actions": [
                {
                    "type": "Action.Submit",
                    "title": "Submit",
                    "style": "positive"
                }
            ]
        }
    ],
    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
    "version": "1.2"
}
```

```

Every input field in this Adaptive card has a unique ID: **Name**, **Address**, and **Weeks**.

7. Copy and paste the following code into the same code view:

Bot response

```
AdaptiveCard_StartTrialForm()
[Activity
 Attachments = ${json(adaptivecardjson_StartTrialForm())}
]
```

```

54
55 "type": "TextBlock",
56 "size": "Medium",
57 "weight": "Bolder",
58 "text": "Register for a meal plan trial."
59 },
60 {
61 "type": "Input.Text",
62 "placeholder": "Please enter your Name",
63 "id": "Name"
64 },
65 {
66 "type": "Input.Text",
67 "placeholder": "Please enter your Address",
68 "id": "Address",
69 "isMultiline": true
70 },
71 {
72 "type": "Input.Number",
73 "placeholder": "How many weeks",
74 "id": "Weeks"
75 },
76 {
77 "type": "ActionSet",
78 "actions": [
79 {
80 "type": "Action.Submit",
81 "title": "Submit",
82 "style": "positive"
83 }
84]
85 },
86],
87 "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
88 "version": "1.2"
89 }
90 ...
91
92 # AdaptiveCard_StartTrialForm()
93 [Activity
94 | Attachments = ${json(adaptivecardjson_StartTrialForm())}
95]

```

## Add conversation logic

1. Go to the **Create** page and select the **StartTrial** trigger.
2. On the authoring canvas, select **Add (+)**, **Ask a question**, then **Text**.
3. In the properties pane, under **Bot responses**, and select **Show code**.

### Warning

Adding the expression in the next step to the *response* editor instead of the *code* editor will result in the bot responding with raw JSON instead of an Adaptive Card.

4. Copy and paste the following expression:

```
Bot response
- ${AdaptiveCard_StartTrialForm()}
```

5. Select the **User Input** tab in the properties pane, then do the following:

- For **Property**, enter `user.name`.
- For **Value**, enter `=turn.activity.value.Name`.

6. On the authoring canvas, select **Add (+)**, **Manage properties**, then **Set properties**.

7. In the properties pane, do the following:

- Under **Assignments**, select **Add new** to add a **Property** box and a **Value** box.
- For **Property**, enter `user.address`
- For **Value**, enter `=turn.activity.value.Address`.
- Select **Add new** a second time to add another **Property** and **Value**.
- For **Property**, enter `user.weeks`.
- For **Value**, enter `=turn.activity.value.Weeks`.

8. On the authoring canvas, select **Add (+)** then **Send a response**.

9. In the response editor, copy and paste the following:

```
Bot response
${user.name} - thanks for starting a ${user.weeks} week trial with us!
Your meals will be delivered to this address: ${user.address}.
```

10. On the authoring canvas, select **Add (+)** then **Begin a Microsoft Copilot Studio Topic**.

11. In the properties pane, for **Dialog name**, select **End of conversation**.

## Test your bot

1. [Publish your Composer content](#) to make it available in your Copilot Studio copilot.

### Important

Selecting **Publish** in Composer makes the changes available for testing, but doesn't automatically publish your bot.

[\*\*Publish\*\*](#) your bot in Microsoft Copilot Studio to update your bot across all the channels it's connected to.

2. Go to the Copilot Studio **Topics** page to see your new **StartTrial** topic.
  3. Open the **Test bot pane** and make sure **Track between topics** is turned on. Enter the message `How do I sign up for a trial?` to start your conversation.
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Key concepts - Publish and deploy your agent

Article • 11/19/2024

With Copilot Studio, you can publish agents to engage with your customers on multiple platforms or channels, such as live websites, mobile apps, or messaging platforms like Microsoft 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.

[https://www.microsoft.com/en-us/videoplayer/embed/RE4mGa9?postJslIMsg=true ↗](https://www.microsoft.com/en-us/videoplayer/embed/RE4mGa9?postJslIMsg=true)

## 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**, it won't be possible for your agent to 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 Microsoft Teams](#) 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:

- [Microsoft Teams](#)
- [Demo Website](#)
- [Custom Website](#)
- [Mobile App](#)
- [Facebook](#)
- [Azure Bot Service channels](#), including:
  - [Skype](#)
  - [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>Microsoft Teams</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> ).<br>Suggested actions are presented as a text-only list; users must retype an option to respond |

## ① Important

Users can't send attachments to Copilot Studio agents. If they try to upload a file (including media, such as images), the agent will say:

*Looks like you tried to send an attachment. Currently, I can only process text. Please try sending your message again without the attachment.*

This applies to all channels, even if the channel or user-facing experience supports attachments (for example, if you're using the Direct Line API or Microsoft Teams).

Attachments can be supported if the message is sent to a skill, where the skill bot supports the processing of attachments. For more information, see [Use Microsoft Bot Framework skills in Copilot Studio](#).

# Next steps

 Expand table

| Article                                                            | Description                                                                         |
|--------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| <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 Microsoft Teams</a> | Use Teams 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).           |

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Publish an agent to a live or demo website

Article • 11/19/2024

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.

[https://www.microsoft.com/en-us/videoplayer/embed/RE4mGa9?postJslIMsg=true ↗](https://www.microsoft.com/en-us/videoplayer/embed/RE4mGa9?postJslIMsg=true)

## 💡 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.

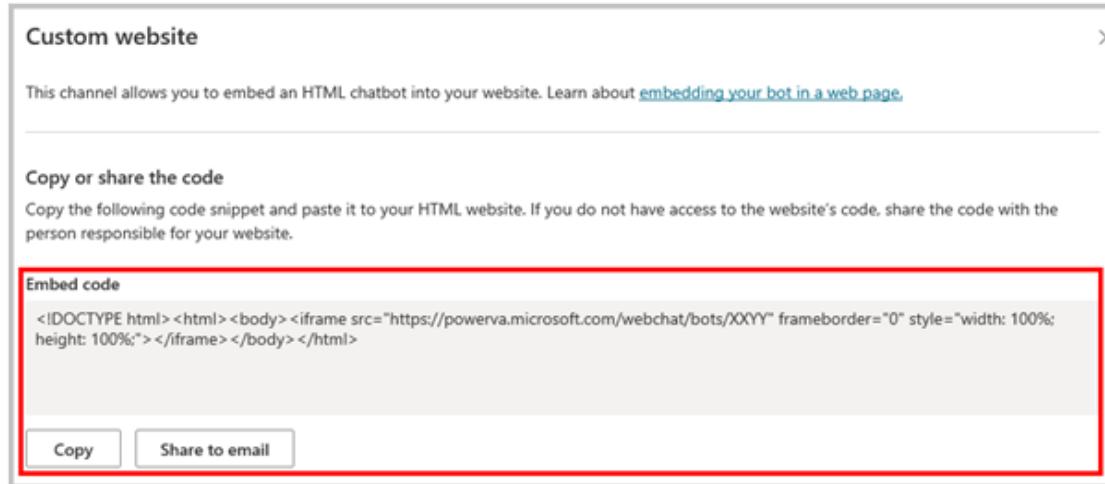
1. Open your agent and on the top menu bar, select **Settings**.
2. On the side navigation pane, select **Security**.
3. Select **Authentication** and then **No authentication**, and then select **Save**.
4. Exit **Settings** and on the top menu bar, select **Channels**.
5. Under **Channels**, select **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. Open your agent and on the top menu bar, select **Channels**.
2. Select **Custom website**, and then either:
  - Select **Copy** to copy the embed code to the clipboard.
  - Select **Share to email** to open a compose window in your default email app, with the code snippet included.



3. Provide the snippet to your web developer to add the agent to your website.

## Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

# 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?



[Provide product feedback ↗](#)

# Connect and configure an agent for Microsoft Teams

Article • 11/19/2024

With your agent published, you can make your agent available to users in Teams. 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.

You can:

- Customize your agent's appearance in Teams
- Install the agent for yourself in Teams
- 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 **Agent extensions** 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

- [Share your agent with other users](#) so they can install and chat with the agent in Teams.
- 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 **Agent extensions** 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 organization** 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.

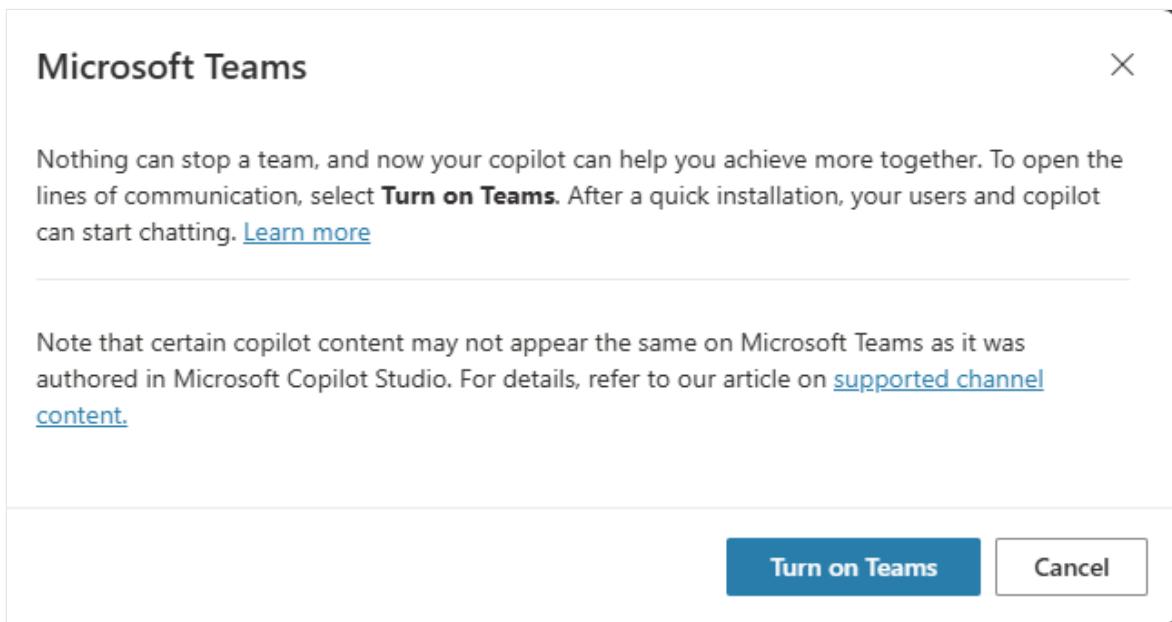
## Connect an agent to the Microsoft Teams channel

After publishing your agent at least once, you can connect it to the Microsoft Teams channel to allow users to chat with it in Teams.

### ⓘ Note

We recommend you [turn on security settings](#) for agents built for Teams or internal employee use, to prevent individuals outside of your organization from using these agents.

1. Open the configuration panel for the Microsoft Teams channel.
2. Select Turn on Teams.



### ⓘ Note

If you do not see this option, the Microsoft Teams channel is already enabled.

## Customize the appearance of an agent for Teams

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 Microsoft Teams channel.
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 have already installed the agent from a shared link or from the [Agent extensions](#) section of the Teams app store. Users must reinstall the agent to obtain the updates.

For agents that have been [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 admin](#) and does not apply for users who install the agent with a direct [installation link](#) or from the Teams app store [Agent extensions](#) section. You can customize the agent avatar at [Customize the look and feel of the agent](#).

## Install an agent as an app in Teams

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 won't appear in conversations that are currently ongoing.

You can type "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 Microsoft Teams channel.](#)
2. Select **Open agent** to have the installation prompt launch in Teams. Select **Add** to add it as an app in Teams. This operation only installs the agent for you.

## Share a link so others can install an agent

1. [Open the configuration panel for the Microsoft Teams channel.](#)
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 installation 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 **Agent extensions** 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 **Agent extensions** section of the Teams app store. The agent does not show for everyone in the organization even if it is 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 **Agent extensions** 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 Microsoft Teams channel.
2. Select **Availability options**.
3. Select **Show to my teammates** and review who has access to the agent.
4. Confirm the **Visible in agent extensions** checkbox is selected and select **Share**.

Your agent becomes available in the **Agent extensions** 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 **Agent extensions** 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 Agent extensions section

You can remove an agent from the **Agent extensions** section. Removing the agent from there doesn't affect users with the agent installed.

1. Open the configuration panel for the Microsoft Teams channel.
2. Select **Availability options**.
3. Select **Show to my teammates** and clear the **Visible in agent extensions** checkbox.
4. Select **Share**. The agent no longer appears in the **Agent extensions** 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 **Agent extensions** 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 Microsoft Teams channel.
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**.

① 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 approved, all new and existing users of the agent in Teams receive the update.

This only applies to the agent details. You can still 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.

① Note

After submitting the agent for admin approval, you should not 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 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 Microsoft Teams channel](#).
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 Microsoft Teams channel](#).
2. Select **Edit details**.
3. Clear the **Allow your users to 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 Microsoft Teams channel](#).
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 Microsoft Teams channel](#).
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.

## 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 **Agent extensions** 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 Microsoft Teams channel.](#)
2. Select **Disconnect from Teams**.
3. Review the information about disconnecting your agent, and select **Disconnect** to continue.

You can always reconnect your agent by selecting **Turn on Teams** in the **Make the agent available to others** window on the Publish page.

## Open the configuration panel for the Microsoft Teams channel

1. Open your agent in Copilot Studio.
2. On the top menu bar, select **Channels**.
3. Select the **Microsoft Teams** tile. The configuration panel appears.

## Known limitations

- [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 Microsoft 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 Microsoft 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.
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# 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

Article • 11/19/2024

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](#)

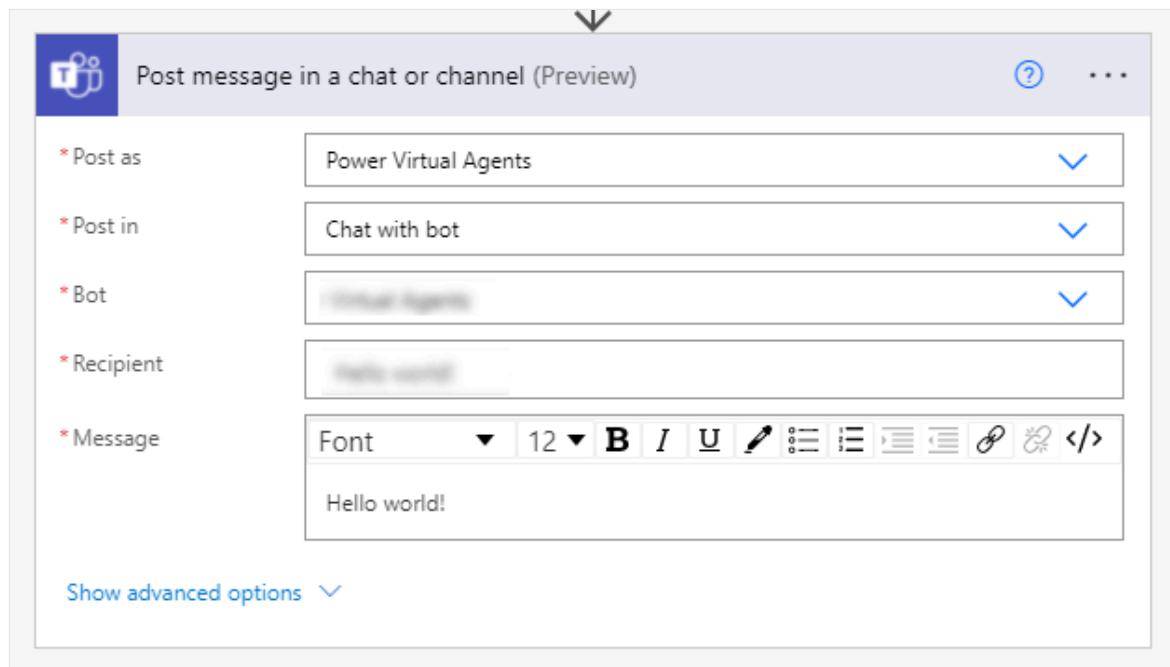
## Known limitations

- 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 **Power Virtual Agents (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.



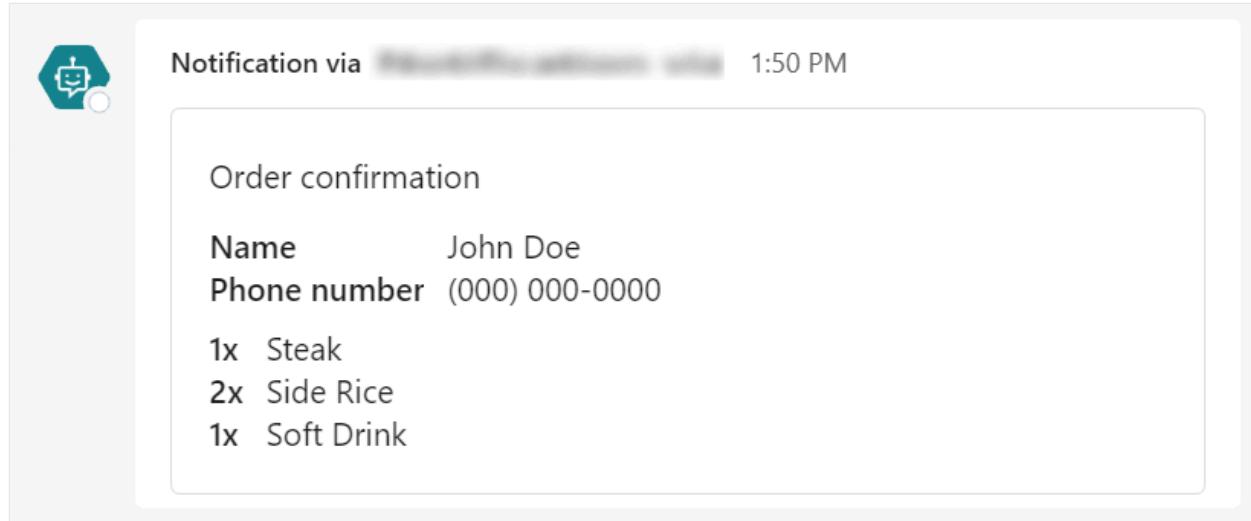
When the flow is run, the recipient will receive the proactive message from the agent in Microsoft Teams.

## 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. To learn how to send Adaptive Cards in an *active* conversation, see [Show an Adaptive Card in Copilot Studio](#).

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.

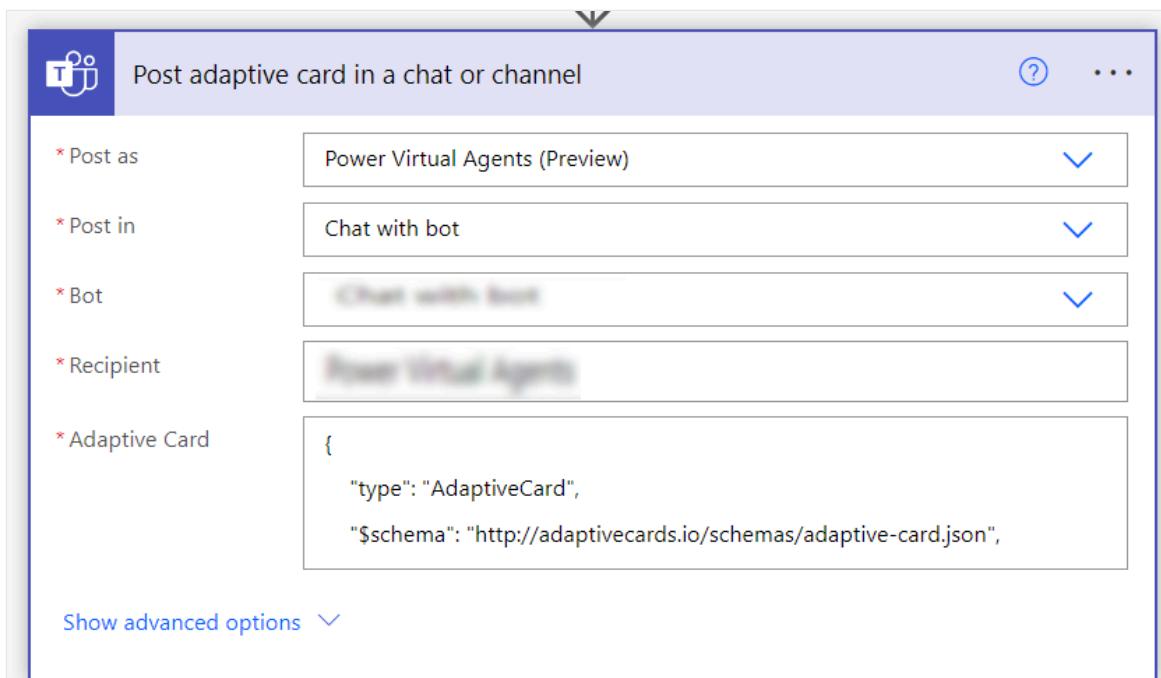
1. In Power Automate, add the Microsoft Teams connector action **Post adaptive card in a chat or channel** at the step where you want to send card in your flow.
2. For **Post as**, choose **Power Virtual Agents (Preview)**.
3. For **Post in**, choose **Chat with bot**.
4. For **Bot**, select the agent 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"
 }
]
}
```

```
]
}
```



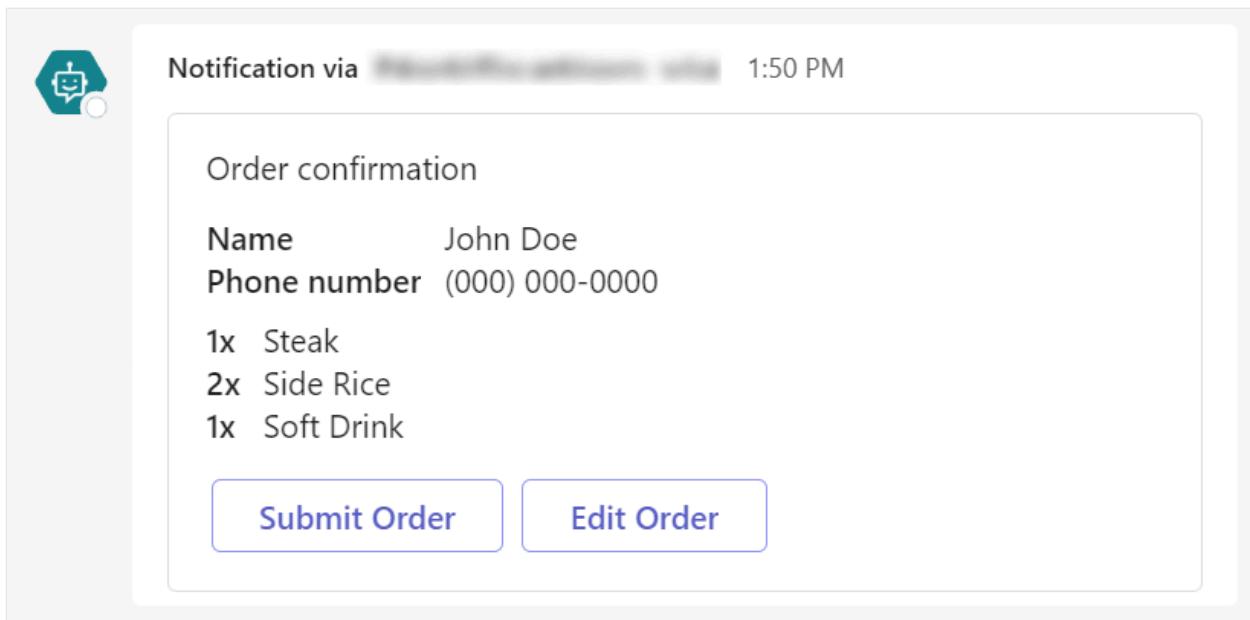
When the flow is run, the recipient will receive the Adaptive Card from the agent in Microsoft Teams.

## Wait for user response

### (i) 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. To learn how to send Adaptive Cards in an *active* conversation, see [Show an Adaptive Card in Copilot Studio](#).

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 **Power Virtual Agents (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"
 },
],
 "actions": [
 {
 "type": "Action.ToggleCard",
 "title": "Edit Order",
 "style": "primary"
 }
]
}
```

```

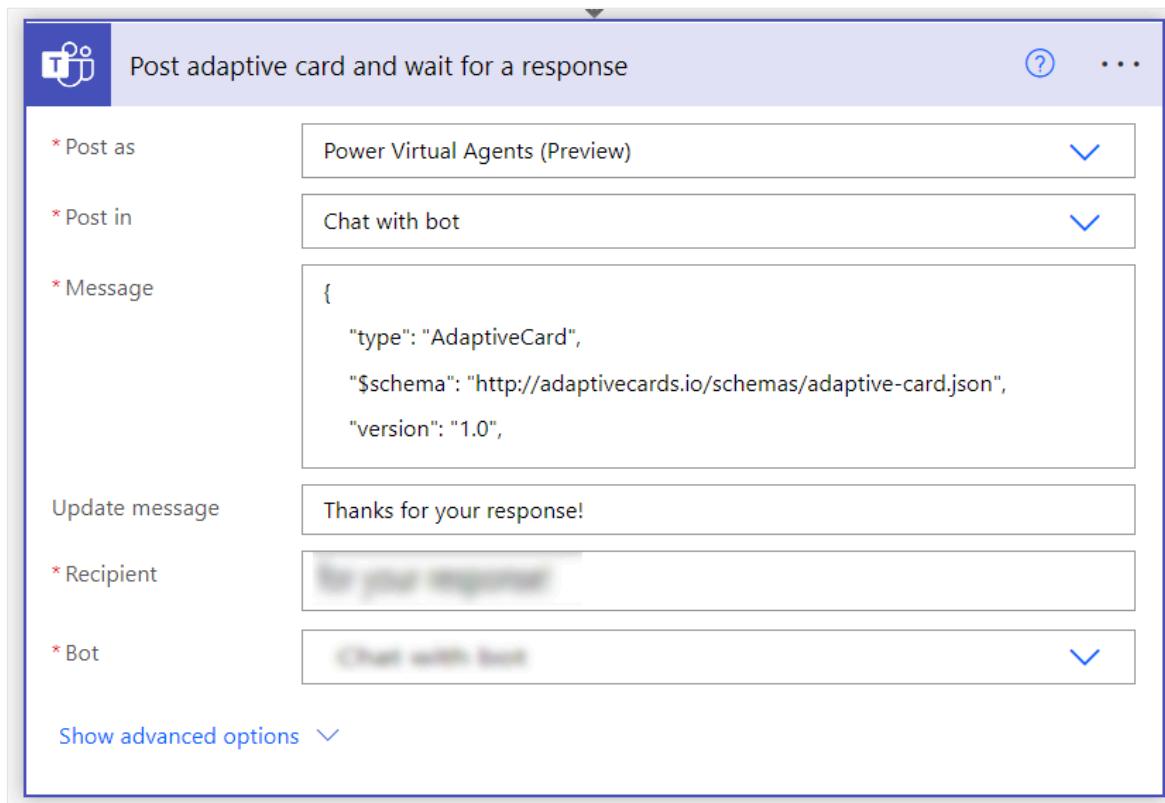
 {
 "type": "FactSet",
 "facts": [
 {
 "title": "Name",
 "value": "John Doe"
 },
 {
 "title": "Phone number",
 "value": "(000) 000-0000"
 }
]
 },
 {
 "type": "Container",
 "items": [
 {
 "type": "FactSet",
 "facts": [
 {
 "title": "1x",
 "value": "Steak"
 },
 {
 "title": "2x",
 "value": "Side Rice"
 },
 {
 "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

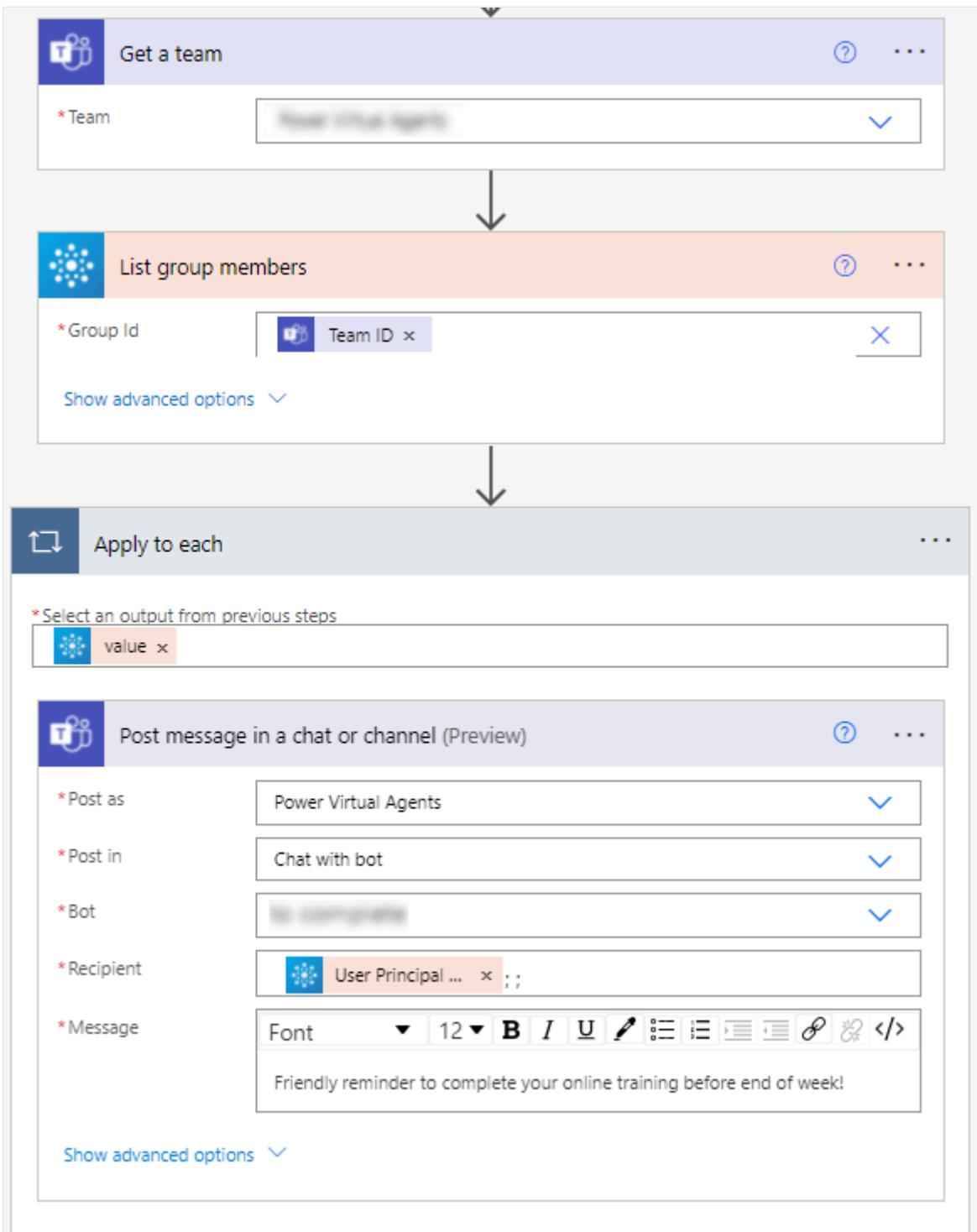
The prerequisites detailed in this topic apply to each recipient.

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 **Power Virtual Agents (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.

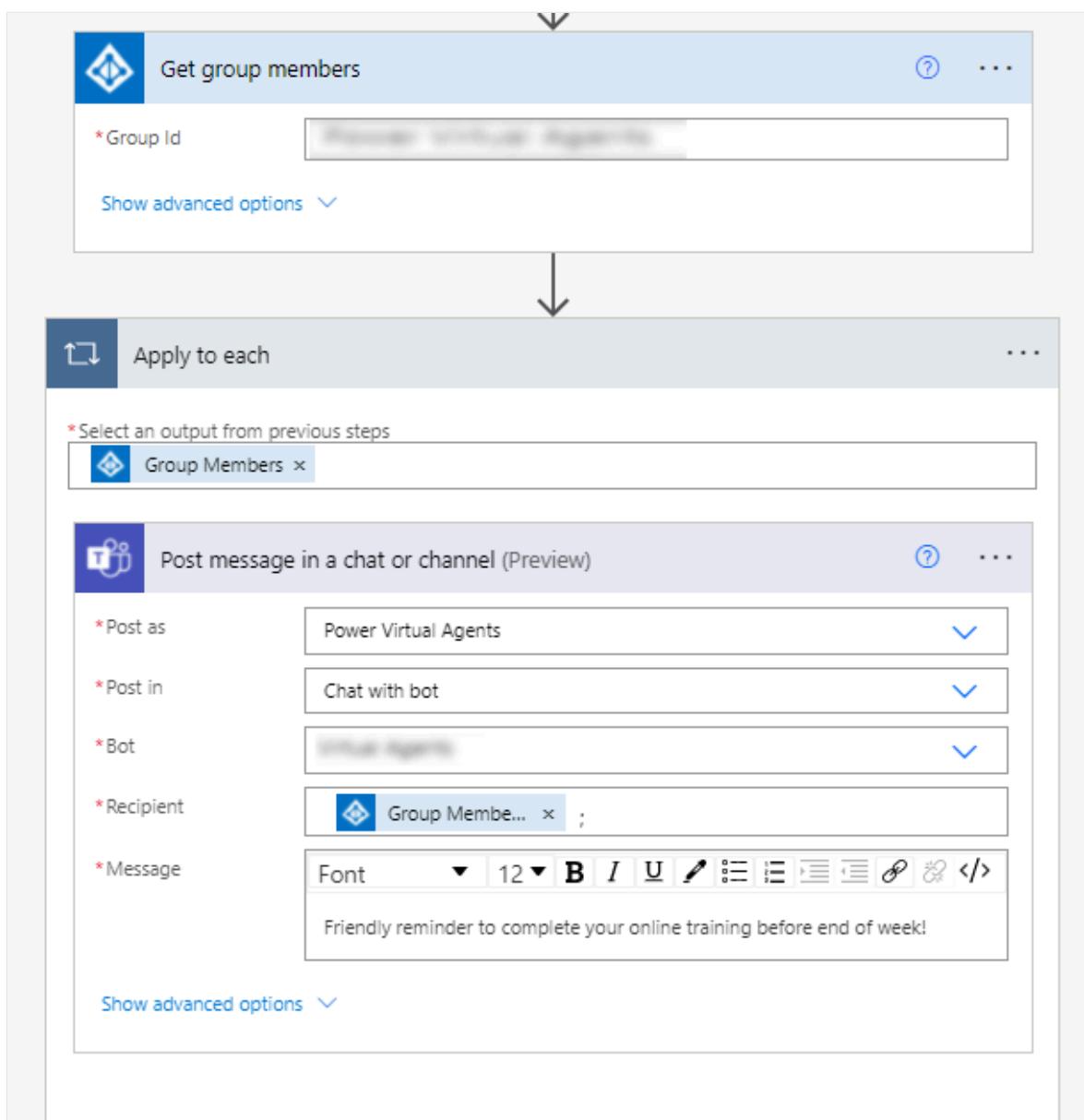
When the flow runs, each user in the team will receive the proactive message in a private chat with the agent.

## 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 **Power Virtual Agents (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

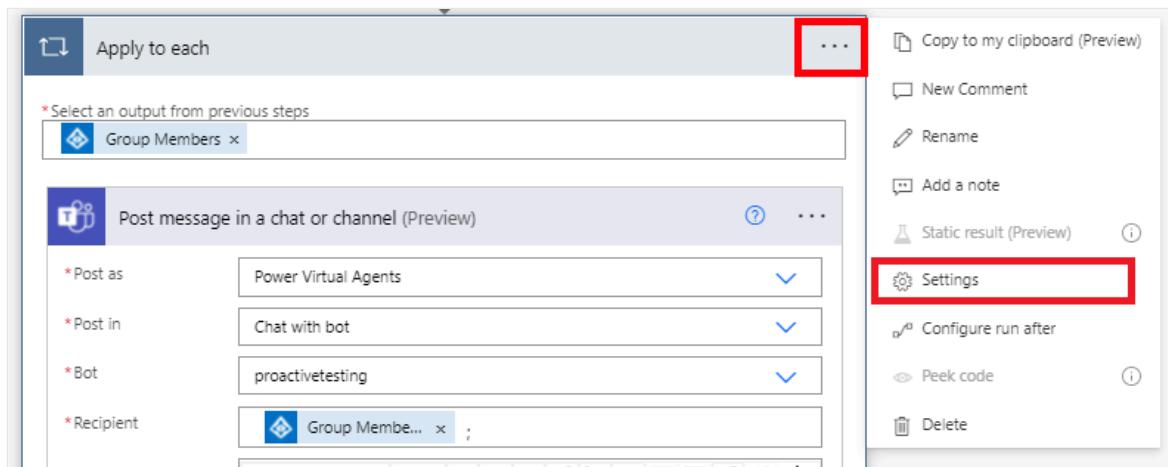
Normally when sending a proactive message to multiple recipients, your agent will send one message after another. However, in some situations, it might be preferable to send the message to multiple recipients at the same time.

## ⚠️ Warning

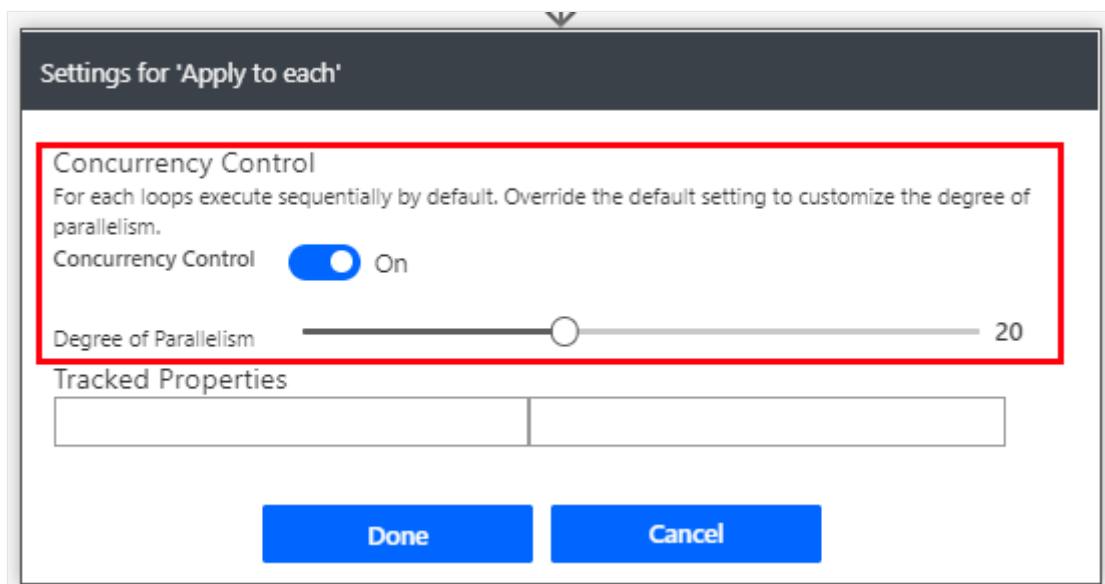
All proactive messages from Copilot Studio are subject to [limits on Power Automate and throttling limits of the Microsoft Teams connector](#).

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**.

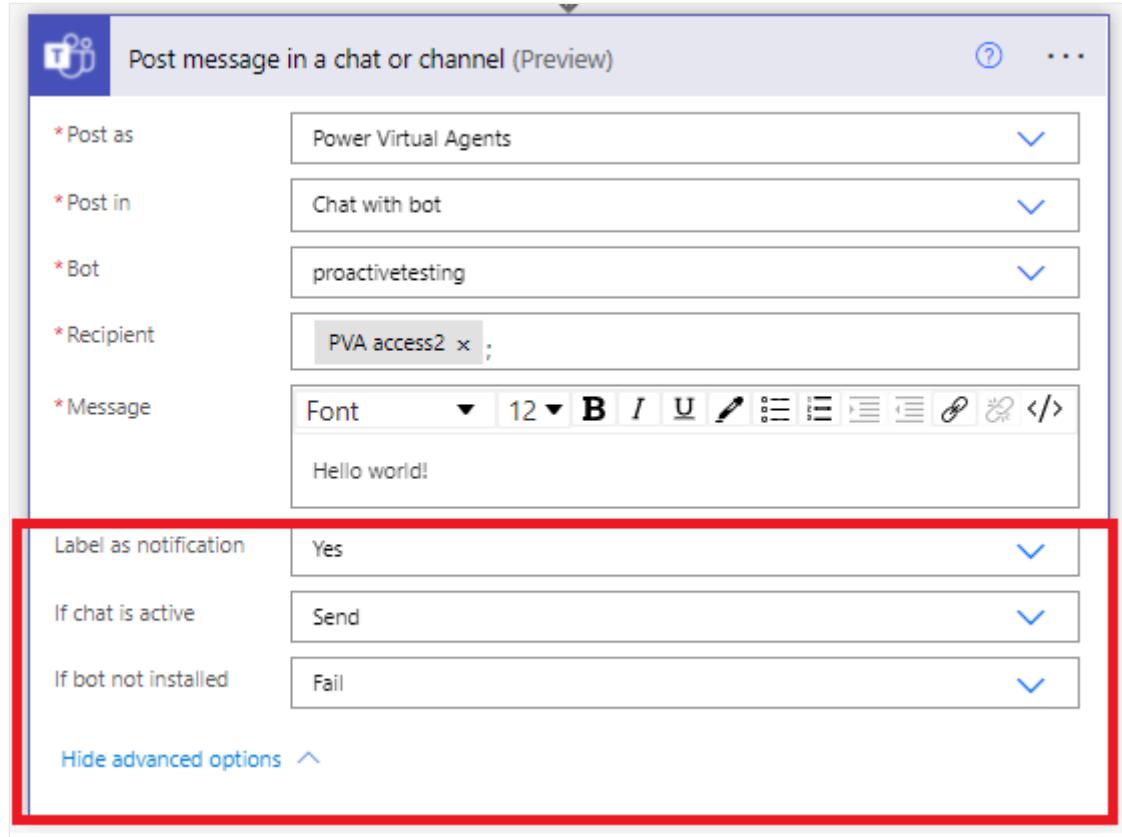


2. Turn on **Concurrency control** and set the degree of parallelism.



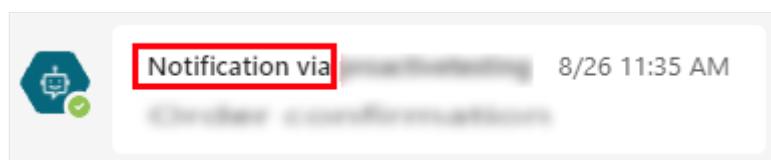
# Configure advanced options for proactive messages

Copilot Studio allows you to control detailed behavior on your agent under **Show advanced options** in the Microsoft Teams connector.



## 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. |

| Status code | Succeeded (Boolean) | Description                                                                                      |
|-------------|---------------------|--------------------------------------------------------------------------------------------------|
| 100         | False               | Message couldn't be delivered because the recipient doesn't have the agent installed.            |
| 300         | False               | Message couldn't be delivered because the recipient is in an active conversation with the agent. |

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Publish an agent to Facebook

Article • 11/19/2024

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:

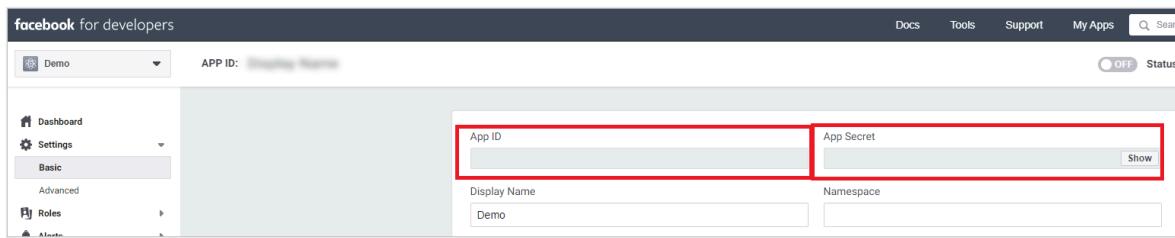
1. Configure Facebook app settings: retrieve your Facebook app information, enable API access, add Facebook Messenger to your app, and configure the Facebook pages your app should appear on.
2. Configure the Facebook publication channel in Copilot Studio.
3. Connect your Facebook app to Copilot Studio using webhooks.

You will then need to submit your app for Facebook review before you can publish your app and make it public.

# Configure Facebook app settings

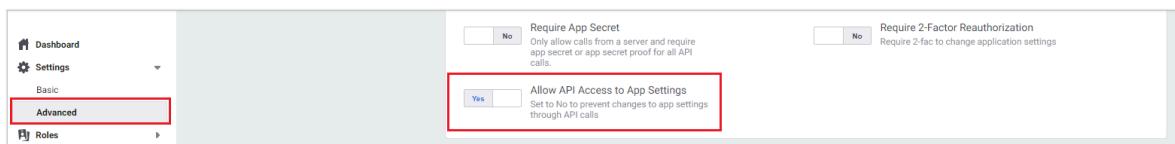
## Retrieve Facebook app information

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](#).



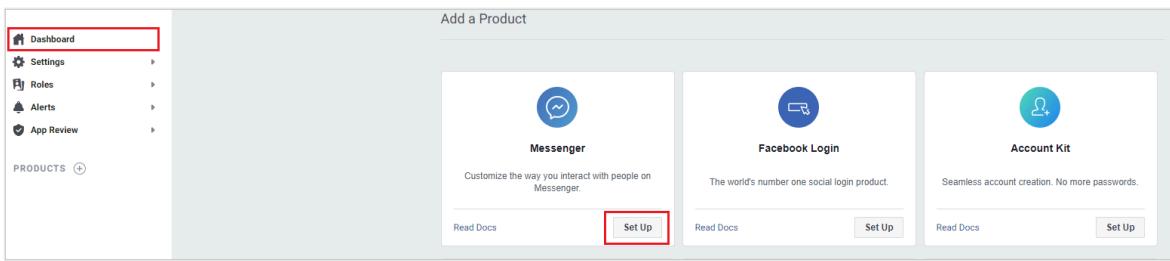
## Enable API access for your Facebook app

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.



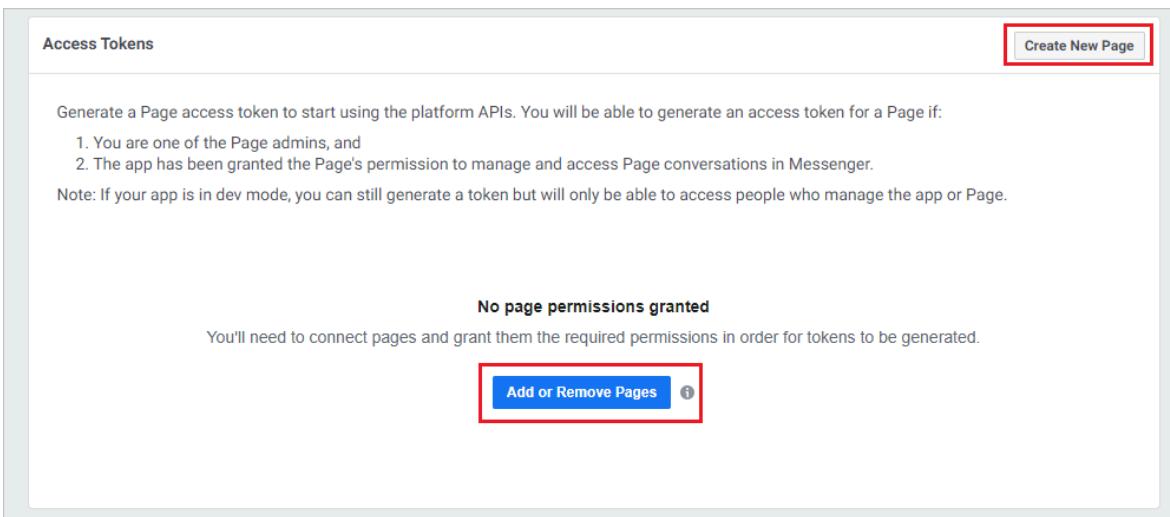
## Add Facebook Messenger to your app

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

YES

[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'll need these 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 *App ID*, *App Secret*, *Page ID*, and *Token* you retrieved earlier into their corresponding fields.

**Facebook**

Set up this channel to let Facebook users chat with your bot. [Learn how](#)

Note that certain bot content may not render the same on Facebook Messenger as it was authored in Virtual Agent. For details, refer to our [supported channel content](#) article.

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 [here](#) and in the [Microsoft Privacy Statement](#).

**Facebook credentials**

Get this information from Facebook and enter it here. Add Facebook page information for each place you want customers to access your bot.

**Facebook app ID \***  
Your Facebook app ID

**Facebook app secret \***  
Your Facebook app secret

**Page ID \***      **Page access token \***

Your Facebook page ID      Your Facebook page access token Delete

+ 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

Add Cancel

5. At least one page is required, but you can select **Add Page** to add additional pages.

## Facebook

X

Set up this channel to let Facebook users chat with your bot. [Learn how](#)

Note that certain bot content may not render the same on Facebook Messenger as it was authored in Virtual Agent. For details, refer to our [supported channel content](#) article.

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 [here](#) and in the [Microsoft Privacy Statement](#).

### Facebook credentials

Get this information from Facebook and enter it here. Add Facebook page information for each place you want customers to access your bot.

**Facebook app ID \***

**Facebook app secret \***

**Page ID \***

**Page access token \***



[+ Add page](#)

### Connection info for Facebook

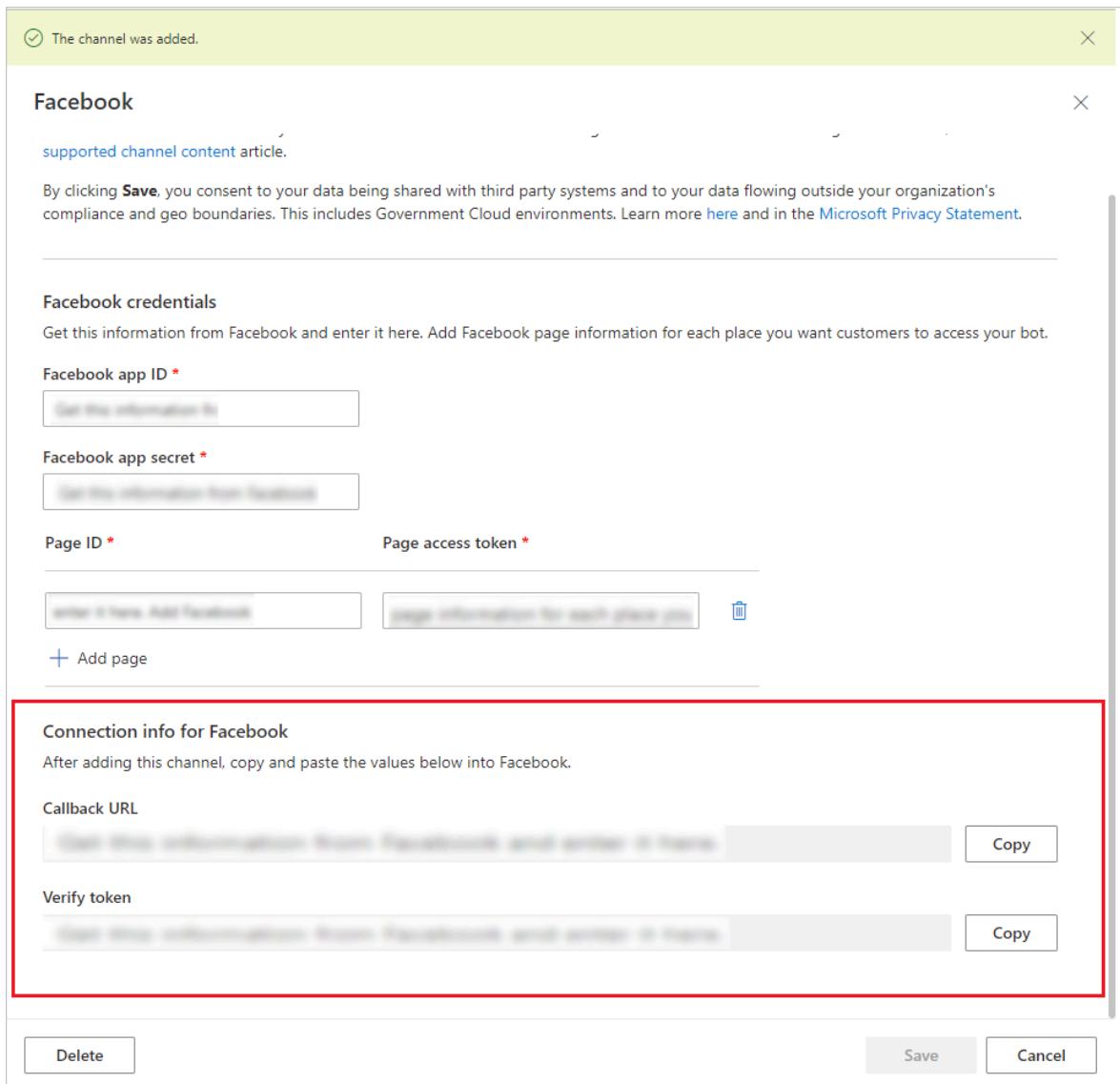
After adding this channel, copy and paste the values below into Facebook.

Callback URL

**Add**

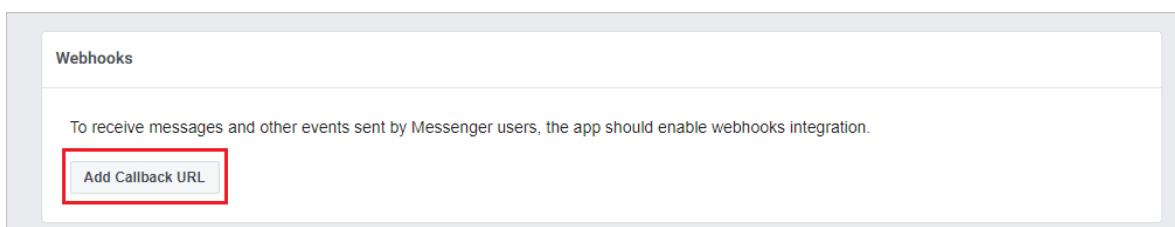
**Cancel**

6. Select **Add** and wait for the success confirmation message. Once successful, copy the *Callback URL* and *Verify token*. You need these when you [connect your Facebook app to Copilot Studio](#).

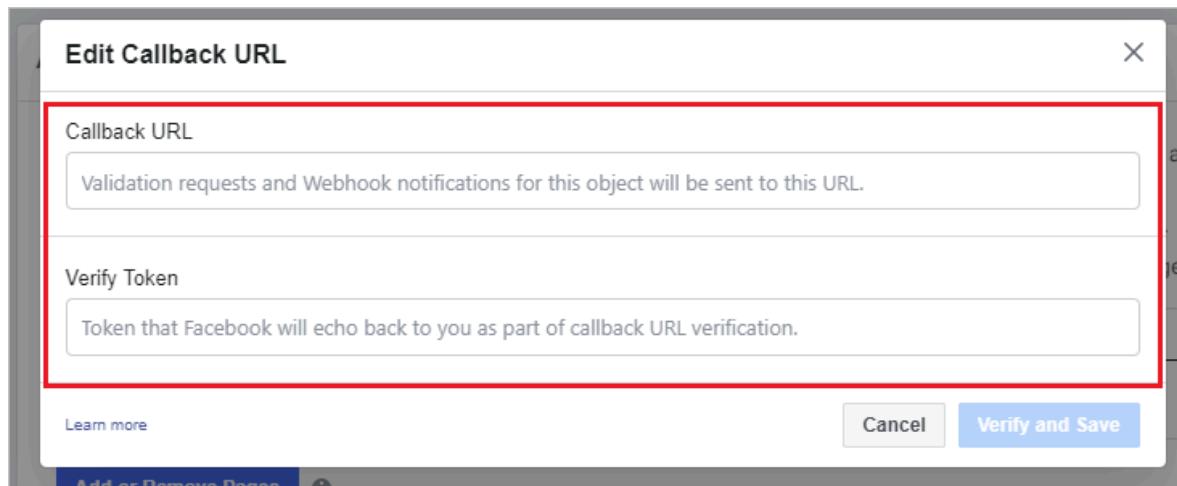


## Connect your Facebook app to Copilot Studio

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.

| Pages ↑        | Webhooks      |                          |
|----------------|---------------|--------------------------|
| (D) [REDACTED] | 0 Fields<br>— | <b>Add Subscriptions</b> |
| (N) [REDACTED] | 0 Fields<br>— | <b>Add Subscriptions</b> |

**Add or Remove Pages** ⓘ

6. Select the following fields:

- **messages**
- **messaging\_postbacks**
- **messaging\_optins**
- **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 need to 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 need to provide those 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 third-party 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 [Sample submissions](#) and [Common rejection reasons](#). Your agent will be tested to ensure it is compliant with [Facebook's Platform Policies](#) before approved by Facebook to become public.

## Make the app public and publish the page

Until the app is published, it is in [Development Mode](#). The agent will not be public and it will work only 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 do not 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 are not 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 may take a few minutes before the agent becomes reachable by users on Facebook pages after the Facebook channel is added.
- After removing the Facebook channel, it may take a few minutes before the agent is removed fully and becomes unreachable on Facebook Messenger.
- After removing a Facebook page, it may take a few minutes before the agent becomes unreachable by visitors to the removed page through Facebook Messenger.

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

# Publish an agent to mobile or custom apps

Article • 11/19/2024

You can connect your agent to a custom app so that the app's users can interact with the agent directly from within your app.

In most cases, your custom app is a mobile-device app that is either a web-based app or a native app or adapter to other services that your business requires.

There are different procedures for connecting to your mobile app, depending on whether your app is a web-based app or a native app.

Connecting your agent to a web-based app is relatively straightforward as it involves copying a code snippet into your app. However, both web-based apps and native or custom apps still require considerable developer expertise to fully integrate the agent into your app. Both procedures are described in this article.

## Prerequisites

- [.NET Core SDK](#) version 2.1.
- Nuget package [Microsoft.Bot.Connector.DirectLine](#).
- An agent created in Copilot Studio that you want to connect to your app.

## Connect your agent to a web-based app

1. In Copilot Studio, in the navigation menu, select **Channels**.
2. Select the **Mobile app** tile to open the configuration window.
3. Copy the code under the **Web-based apps** section and provide it to your app developers to add to your web-based app.

## Mobile app

X

Allow your mobile app users to chat with your copilot. 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 copilot.

#### Token Endpoint

[REDACTED]

 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.

#### Embed code

[REDACTED]

 Copy

## Connect your agent to a native or custom app

### Tip

While this section describes how to connect to a mobile app, the same process could be applied for custom or native apps, such as IoT (Internet of things) apps.

If your goal is to connect to Azure Bot Service channels, besides following the instructions here, your developers can learn more at [Publish an agent to Azure Bot Service channels](#).

### 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.

## Code samples

Code snippets used in this document are from:

- [Connector sample code ↗](#)

## References

The instructions in this document reference the following source material:

- [Bot Framework Direct Line API](#)
- [Direct Line Authentication](#)
- [Context variables available upon handoff](#)
- [Microsoft Bot Framework Activity ↗](#)

## Retrieve your Copilot Studio agent parameters

To connect to the agent you built, you need to retrieve your agent's name and token endpoint to identify it.

1. In Copilot Studio, navigate to the **Overview** page of your agent, and copy your agent's name.
2. Select **Channels > Mobile app**.
3. On the **Mobile app** page, next to **Token Endpoint**, select **Copy**. You need this endpoint for the [Get Direct Line token](#) step.

## Mobile app

X

Allow your mobile app users to chat with your copilot. 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 copilot.

#### 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.

#### Embed code

 Copy

## Get Direct Line token

To start a conversation with your agent, you need a Direct Line token. This token can be obtained by making a GET request to the endpoint indicated within the Copilot Studio screen. This token must then be used as the header for subsequent calls to the directline API.

Example:

rest-api

GET <BOT TOKEN ENDPOINT>

If the request is successful, will be returned a Direct Line token, expiration time and a conversationId for the requested agent. Example:

Json

```
{
 "token":
 "RCurR_XV9ZA.cwA.BKA.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfi
 Cpg4Fv0y8qqb0F5x8qb0F5xn",
 "expires_in": 3600,
 "conversationId": "abc123"
}
```

## Sample code example

The following example uses samples from the [Connector sample code](#) to get a Direct Line token for a Copilot Studio agent.

C#

```
/// <summary>
/// Get directline token for connecting bot
/// </summary>
/// <returns>directline token as string</returns>
public async Task<DirectLineToken> GetTokenAsync(string url)
{
 try
 {
 return await _httpClient.GetFromJsonAsync<DirectLineToken>(url);
 }
 catch (HttpRequestException ex)
 {
 throw ex;
 }
}
```

C#

```
/// <summary>
/// class for serialization/deserialization DirectLineToken
/// </summary>
public class DirectLineToken
{
 public string Token { get; set; }
 public int Expires_in { get; set; }
 public string ConversationId { get; set; }
}
```

The response object is the same as the `GET` request we saw earlier.

JSON

```
{
 "token":
 "RCurR_XV9ZA.cwA.BKA.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfi
 Cpg4Fv0y8qqb0F5x8qb0F5xn",
 "expires_in": 3600,
 "conversationId": "abc123"
}
```

## Use Direct Line to communicate with the agent

After retrieving the Direct Line token, you're ready to have a conversation with your Copilot Studio agent with Direct Line. To start a conversation and send and receive messages, follow the instructions at [Bot Framework Direct Line API](#).

The following example uses samples from the [Connector sample code](#) to start a conversation and send and receive messages from a Copilot Studio agent.

1. Initialize a `DirectLineClient` instance with the Direct Line token and start a conversation:

```
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. Once started, each conversation can be identified and connected using the combination of `token` and `conversationId`. Send a user message to an existing conversation:

```
C#

// Use the retrieved token to create a DirectLineClient instance
// Use the conversationId from above step
// endConversationMessage is your predefined message indicating that
user wants to quit the chat
while (!string.Equals(inputMessage = /*Get_User_Input()*/,
endConversationMessage, StringComparison.OrdinalIgnoreCase))
{
 using (var directLineClient = new DirectLineClient(token))
 {
 // Send user message using directlineClient
 // Payload is a Microsoft.Bot.Connector.DirectLine.Activity
```

```

 await
directLineClient.Conversations.PostActivityAsync(conversationId, new
Activity()
{
 Type = ActivityTypes.Message,
 From = new ChannelAccount { Id = "userId", Name =
"userName" },
 Text = inputMessage,
 TextFormat = "plain",
 Locale = "en-US",
});
}
}

```

3. Retrieve the agent's response using the same `token` and `conversationId`. The retrieved Direct Line response activities contain both the user's and agent's messages. You can filter response activities by your agent's name to get only the agent's response message.

C#

```

// Use the same token to create a directLineClient
using (var directLineClient = new DirectLineClient(token))
{
 // To get the first response set string watermark = null
 // More information about watermark is available at
 // https://learn.microsoft.com/azure/bot-service/rest-api/bot-
framework-rest-direct-line-1-1-receive-messages?view=azure-bot-service-
4.0

 // response from bot is of type
 Microsoft.Bot.Connector.DirectLine.ActivitySet
 ActivitySet response = await
directLineClient.Conversations.GetActivitiesAsync(conversationId,
watermark);

 // update watermark from response
 watermark = response?.Watermark;

 // response contains set of Activity from both user and bot
 // To display bot response only, filter Activity.From.Name
 // equals to your bot name
 List<Activity> botResponses = response?.Activities?.Where(x =>
 x.Type == ActivityTypes.Message &&
 string.Equals(x.From.Name, /*Bot_Name*/,
StringComparison.OrdinalIgnoreCase)).ToList();

 // Display botResponses
}

```

## Refresh Direct Line token

You might need to add code to refresh the Direct Line token if your application has a lengthy conversation with the agent. The token expires but can be refreshed before it expires; learn more at [Direct Line Authentication](#).

The following example uses samples from the [Connector sample code](#) to refresh the token for an existing Copilot Studio conversation:

C#

```
// DirectLine provides a token refresh method
// Requires the currentToken valid when refreshing
string refreshToken = new
DirectLineClient(currentToken).Tokens.RefreshToken().Token;
// create a new directline client with refreshToken
directLineClient = new DirectLineClient(refreshToken);
// use new directLineClient to communicate to your bot
```

## Parse conversation payload from the agent

After you start a conversation with the agent, the conversation JSON payload uses the standard Microsoft Bot Framework Direct Line activity. You can learn more at [Bot Framework Direct Line API](#).

## Handle handoff activity

If your application needs to hand off to a live agent provider, you need to handle the handoff activity. Handoff activity is sent when the "Transfer to agent" node is hit. You can learn more on the [payload of the handoff activity](#).

## Trigger a welcome message

If you want your agent to send the Greeting system topic automatically when a user starts a conversation, you can send an activity with `Type=event` and `Name=startConversation`.

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Publish an agent to Azure Bot Service channels

Article • 11/19/2024

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](#).

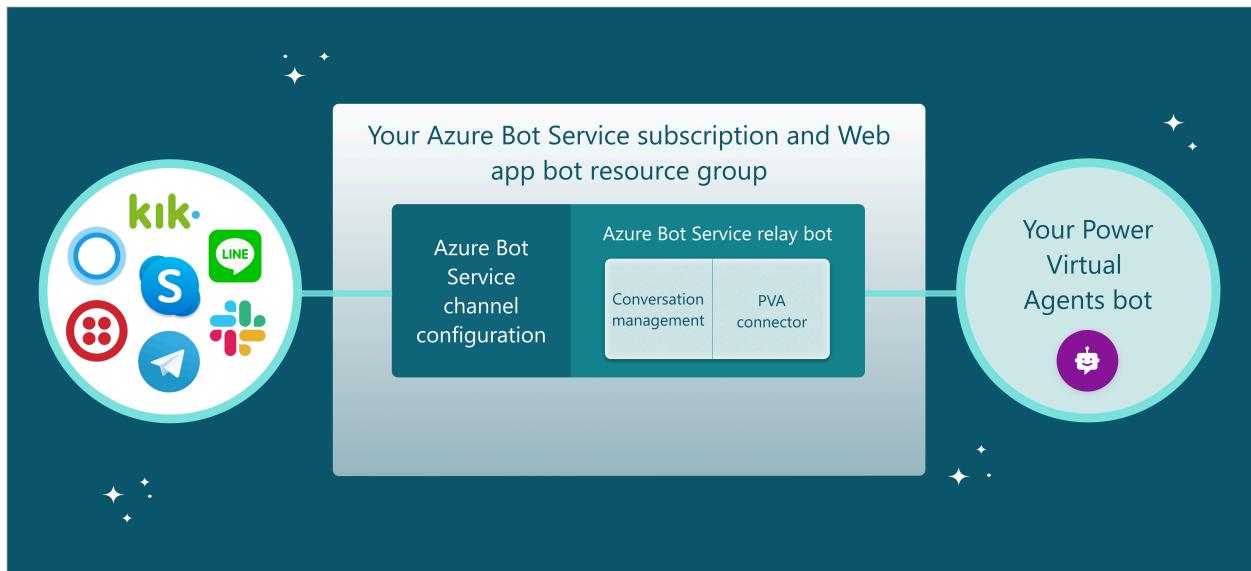
## References

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' section is expanded, showing 'Channels'. The main area displays the 'Contoso Bot' solution, which is highlighted with a red box. Below the title, there is a button labeled 'View Solution (Default Solution)' with an info icon. A large graphic shows a stack of documents with a checkmark, accompanied by the text 'Quickstart with topic suggestions'. Below this, another text block says '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 left sidebar highlights 'Settings' and 'Channels'. The main area lists several channels: 'Mobile app', 'Facebook', 'Skype', 'Cortana', 'Slack', 'Telegram', 'Twilio', 'Line', and 'Kik'. The 'Slack' option is highlighted with a red box. Each channel has a small icon and a brief description below it.

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).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.Conversation
tId, 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](#).

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Agents for customer engagement and handoff

Article • 11/19/2024

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:<br>Unlimited<br>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:<br>Unlimited<br>Classic mode: Limited by the Dataverse file storage allocation                           | None                                           |
| SharePoint                               | Internal | Connects to a SharePoint URL, uses GraphSearch to return results                                                                    | Generative mode:<br>Unlimited<br>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:<br>Unlimited<br>Classic mode: Two Dataverse knowledge sources (and up to 15 tables per knowledge source) | Agent user's Microsoft Entra ID authentication |
| Enterprise data using Copilot connectors | Internal | Connects to Copilot connectors where your organization data is indexed by Microsoft Search                                          | Generative mode:<br>Unlimited<br>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 Open AI, or Custom Data. Instead, from the generative

answers node properties, use the **Classic data** option for [Bing Custom Search](#), [Azure Open AI](#), or [Custom Data](#) sources.

- For websites, you need to confirm which website(s) your organization owns that Bing will search through Copilot Studio.

## 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.              |

# Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Configure handoff to any generic engagement hub

Article • 12/03/2024

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.

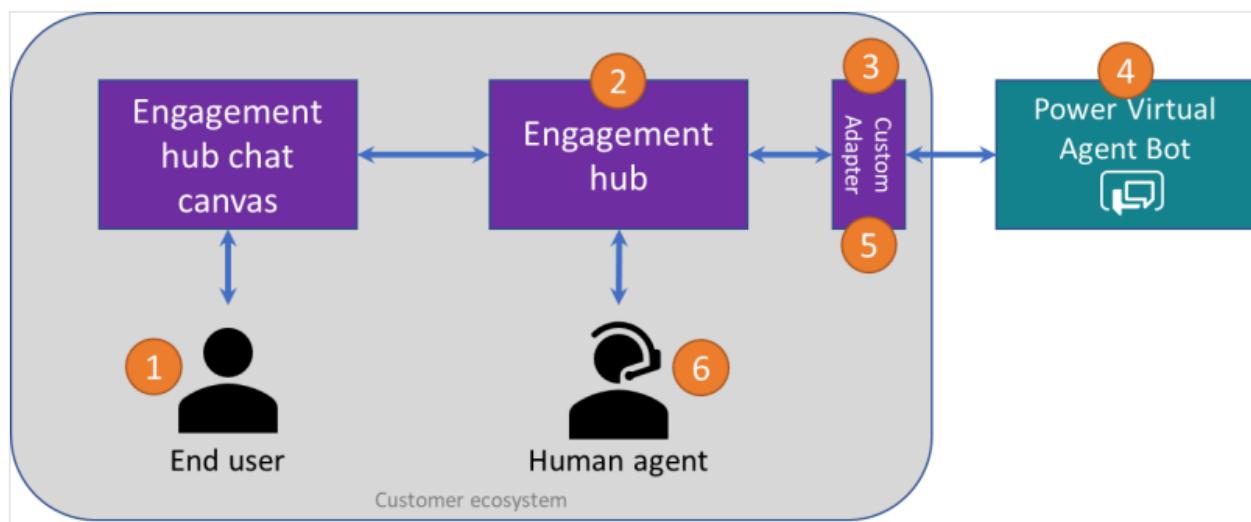
## Prerequisites

- You need to have [built an agent using Copilot Studio](#).
- You need an engagement hub that can interact programmatically using APIs or SDK.

### ⓘ 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>
 }
}
```

---

# Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Configure handoff to Dynamics 365 Customer Service

Article • 11/19/2024

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 voice (in Classic chatbots) and text (messaging) conversations.

## ⓘ 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

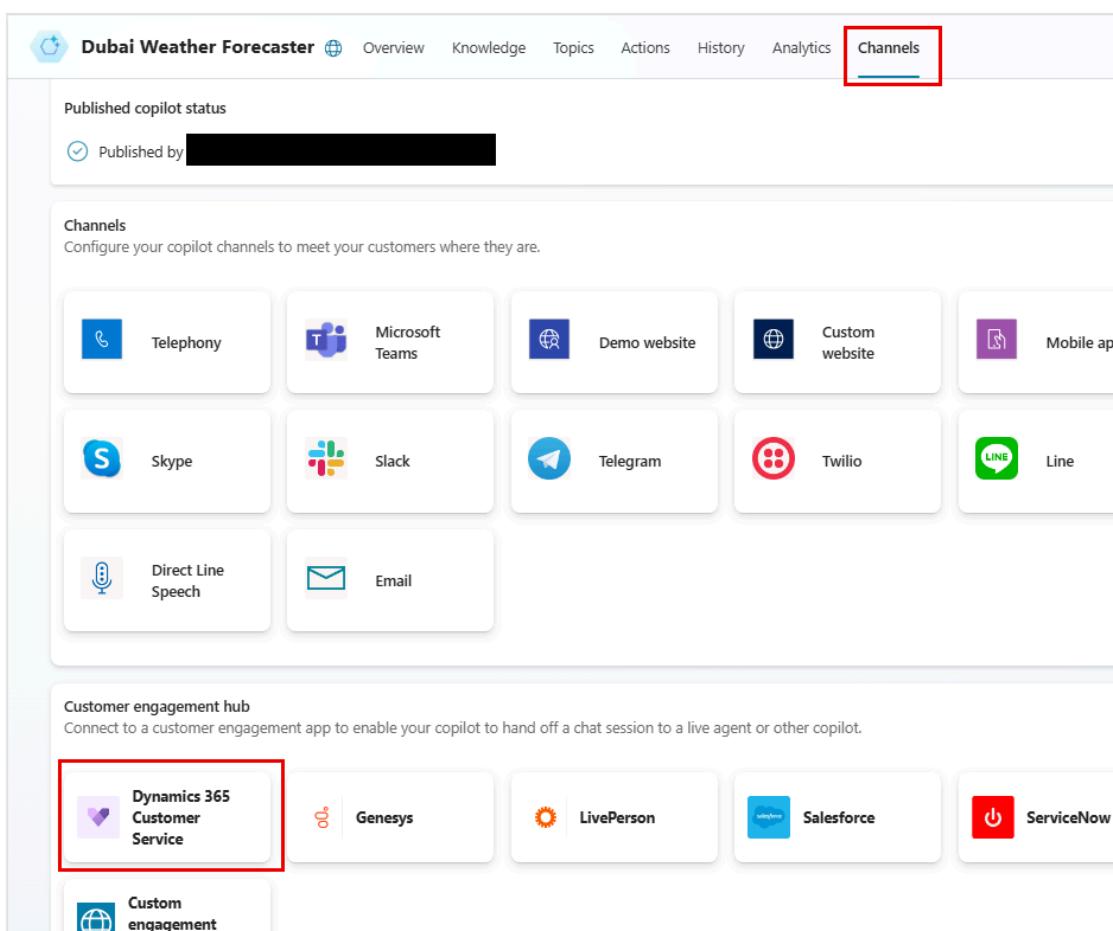
- Sign in with an account that has at least **OC\_Admin** and **Agent Author** roles.
- Have a [product license for Copilot Studio](#) and a [product license for the Chat Add-in for Dynamics 365 Customer service](#).
- Your agent and Dynamics 365 Customer Service must be in the same environment.
- For 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 will fail to connect when you follow the instructions in this article. Ensure your agent's name contains less than 30 characters before you proceed.

# Connect your agent to Dynamics 365 Customer Service

1. In Copilot Studio, open your agent.
2. In the navigation menu, select **Channels**, and then under **Customer engagement hub**, select the **Dynamics 365 Customer Service** tile.



3. Select **Connect**.

- 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 ALM](#).

4. Select **View in Dynamics 365** 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 that's 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

Suppose you've [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 will 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**.

The screenshot shows a list of skills in the Copilot Studio interface. A context menu is open for the skill 'Di ALMBot1'. The menu items are: 'Add required components', 'Managed properties', 'Show dependencies', 'See solution layers' (which is highlighted with a red box), and 'Remove'. Other visible skills include 'Confirmed Success', 'CSAT Rating', 'Email weather condition', 'End of Conversation', 'Escalate', 'Flight Skill Migrated Bot', and 'GetWeather'.

4. Select the unmanaged layer, and then select Remove unmanaged layer.

The screenshot shows the 'Solution Layers' section. 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 navigation path is 'All > Di ALMBot1 > Solution Layers'. The main table lists two solution layers:

| Order | Solution        | Publisher                 |
|-------|-----------------|---------------------------|
| 2     | Unmanaged layer | Default Publisher for org |
| 1     | DiALMSolution   | Di publisher              |

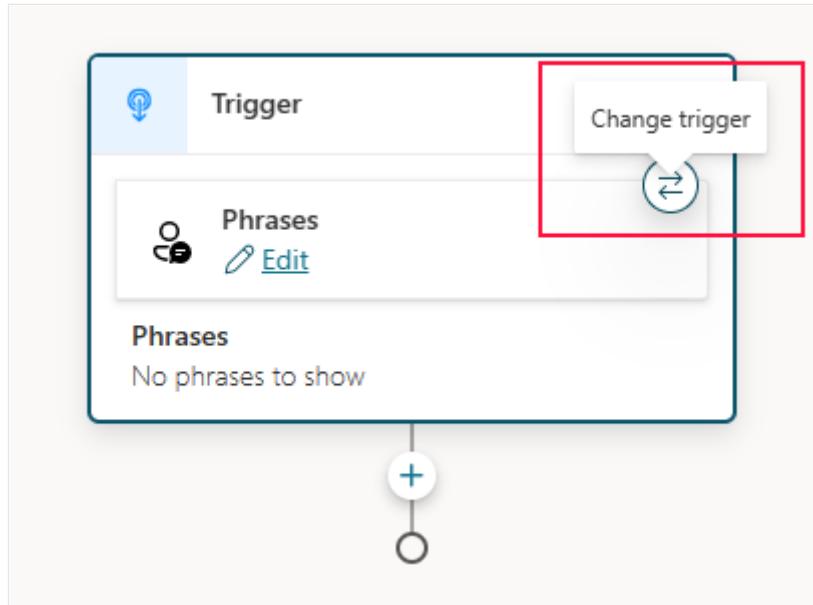
If your agent doesn't have Dynamics 365 capabilities enabled, the message is always displayed.

## Configure inactivity timeout to close conversation

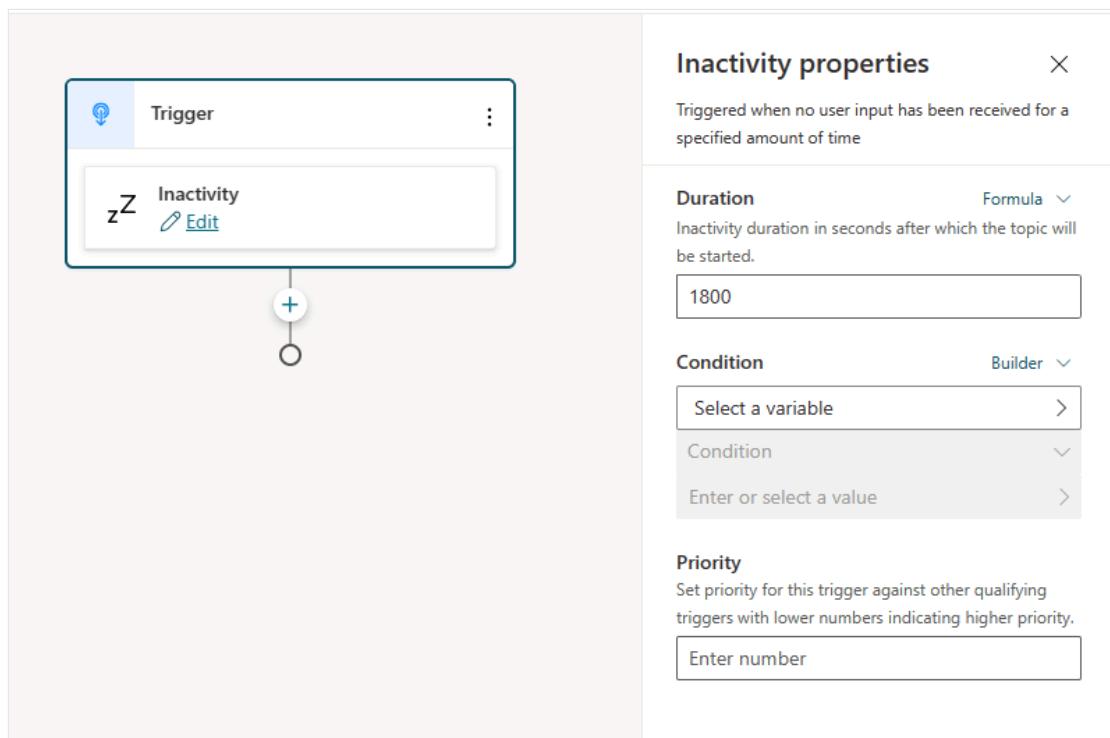
Dynamics 365 Customer Service expects a conversation to end after a period of inactivity. This feature is important to ensure the supervisor view displays the correct ongoing conversations and manages agent loads and metrics correctly. When you create an agent from Dynamics 365 Customer Service, the default agent content includes two topics: **Session Timeout Reminder** and **Session Timeout**. These topics use the **Inactivity trigger** to first remind the user, and then close the conversation after a set inactivity timeout value, which can be configured in the topic. But if you create the agent from the Copilot Studio portal, those topics aren't included in the agent. To create them, follow the steps below:

### Create a Session Timeout topic

1. From the left navigation, select **Topics**.
2. From the top bar, select **New topic** and **From blank**.
3. Hover the trigger **Phrases** and select the **Change trigger** icon.

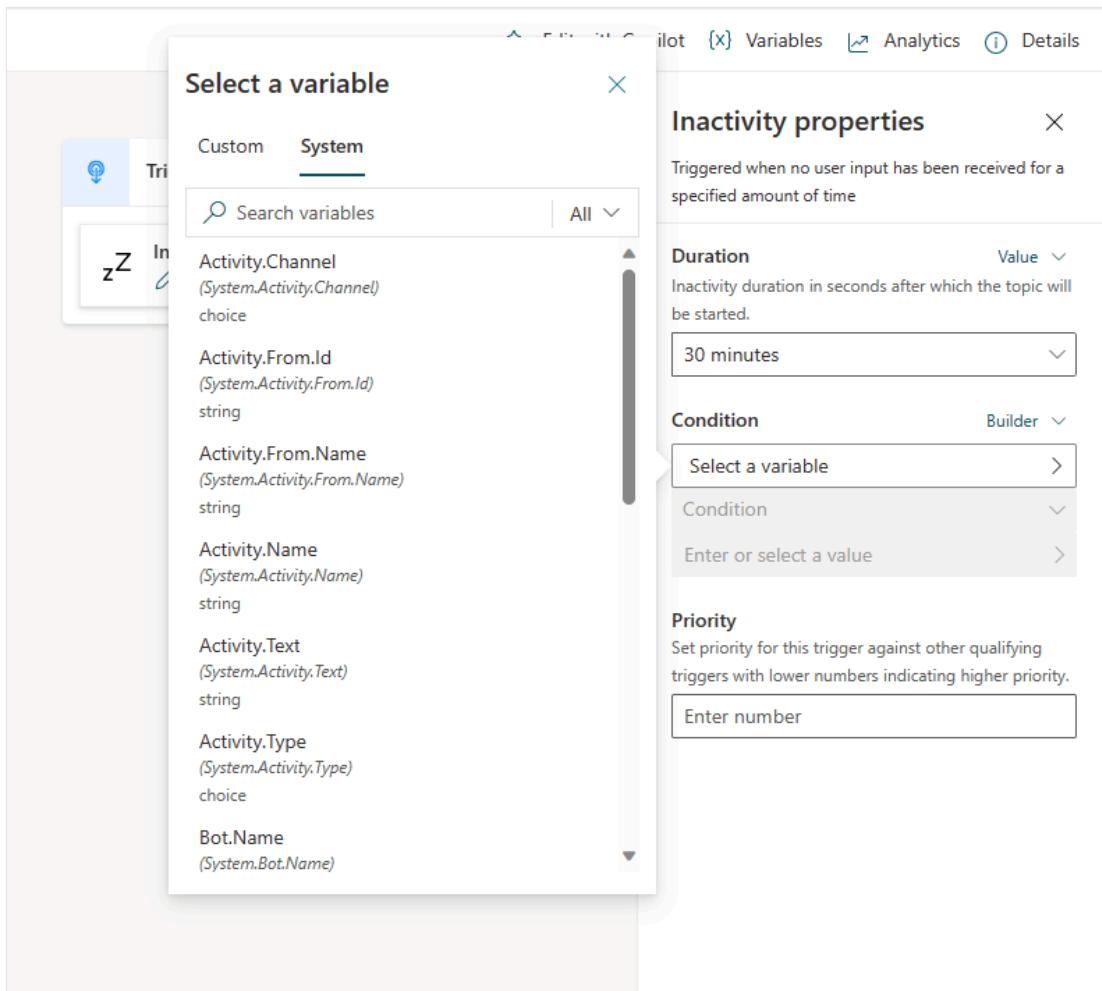


4. Select **Inactivity** from the list.
5. Select **Edit** in the **Trigger** node to configure the inactivity duration.



6. Select a **Value** from the dropdown, or enter an arbitrary value (in seconds) using a **Formula** entry. 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 **Dynamics 365** from the dropdown.

9. Finally, add a message, and at the end, add an End Conversation node so the conversation ends. Click on the (+) sign, select **Topic management**, and then select **End conversation**.

Finally, save and publish your agent.

## Manage your agent's Dynamics 365 capabilities

Select **Settings**, **Agent transfers**, and then select the **Dynamics 365** 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 need to [connect it again](#), starting at Step 2.

## Known limitations

See [limitations when using Copilot Studio with the Chat Add-in for Dynamics 365 Customer Service](#).

## Recommended extensions

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](#).

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

# Hand off to Genesys

Article • 11/19/2024

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.

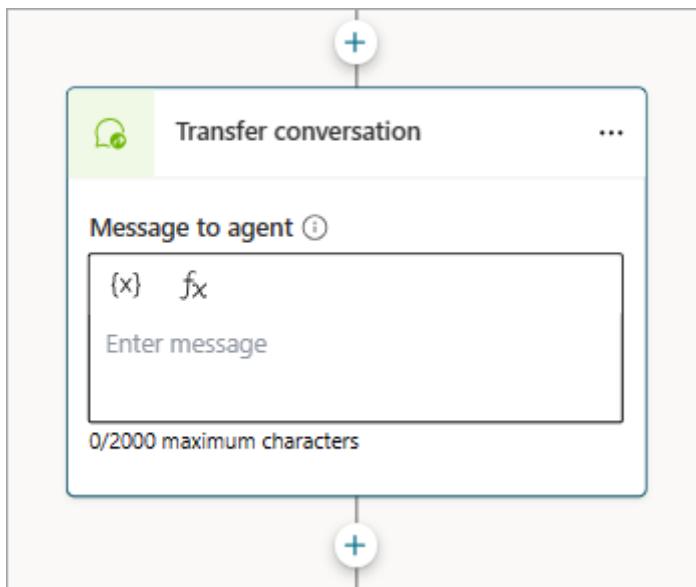
Ensure that the DynamoDB table name created matches the one in the `bot_sessions.py` file.

## 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**: A button at the top left.
- Intents**: A section header. Below it, there is a single intent named **Escalate**.
- Integration**: A section header under Intents. It states: "There are no slots defined for this intent."
- Escalate**: The details for the Escalate intent. It has a slot named **va\_AgentMessage**, which is currently set to **A State.sAgentMessage**.
- Failure Outputs**: A section header. It contains two entries:
  - errorType**: A dropdown menu showing "No output value specified".
  - errorMessage**: A dropdown menu showing "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**.

---

## Feedback

Was this page helpful?



[Provide product feedback](#)

# Hand off to LivePerson

Article • 11/19/2024

Integrating LivePerson with Copilot Studio presents a compelling opportunity for businesses seeking to enhance their customer support and engagement capabilities.

## Prerequisites

Before you begin, ensure that you have:

1. An active LivePerson account.
2. A Copilot Studio agent.
3. 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 at the [LivePerson documentation site](#).
3. To set up handoff to LivePerson, see the [LivePerson documentation - Transfer to agent](#).

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

# Hand off to Salesforce Einstein bot

Article • 11/19/2024

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 PVA 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](#).

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

# Hand off to ServiceNow

Article • 11/19/2024

This document provides step-by-step instructions on how to create and connect an ServiceNow Virtual Assistant to a Copilot Studio agent and escalate to ServiceNow Live Agent. This integration allows you to leverage the capabilities of both platforms and generative answers to enhance your agent experience.

## 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 Vancouver or later required)
- A Copilot Studio agent.
- Admin or developer access to both ServiceNow and Power Platform.

## Steps

Follow these steps to create and connect a ServiceNow Bot to a Copilot Studio agent:

1. [Create a JavaScript function in Microsoft Azure using Visual Studio Code](#).
  - a. Sign in to Microsoft Azure.
  - b. Create a JavaScript function in Microsoft Azure.
  - c. Create an Azure Function for Copilot Studio as Secondary bot with Servicenow Bot Interconnect setup ([KB1112368](#)) KV Function Version3.
  - d. Install the request-promise libraries.
  - e. [Deploy the project to Azure](#).
  - f. From the Azure portal, navigate to **All Resources**.
  - g. Select the name of the function app that you created.
  - h. On the **Overview** page for the function, copy the function URL and keep it for use in a later step. The Azure function URL will be used as the secondary bot endpoint in your ServiceNow instance.
2. [Retrieve Direct Line secret for your bot](#).

3. Add the Direct Line secret key to your Virtual Agent Bot Interconnect instance [.](#)
4. Create a Virtual Agent Bot Interconnect shell topic to call Microsoft Copilot Studio topics [.](#)
5. Create a Virtual Agent conversational custom chat integration configuration [.  
a. Create a new channel or update an existing channel for the integration \[.\]\(#\)  
b. Configure a new provider for the integration \[.\]\(#\)  
c. Set up message authentication \[.\]\(#\)  
d. Create a channel identifier \[.\]\(#\)  
e. Select rich controls for inbound \(user input\) and outbound \(bot response\) transformation \[.  
f. Create and configure a scripted REST API for your custom chat integration \\[.  
g. Create the action scripts \\\[.\\\]\\\(#\\\)\\]\\(#\\)\]\(#\)](#)
6. Transferring Virtual Agent conversations to a live agent

See [Transferring Virtual Agent conversations to a live agent](#) and [Configure handoff to any generic engagement hub](#)

## Step 7: Closing Virtual Agent and Agent Chat conversations [.](#)

---

### Feedback

Was this page helpful?



[Provide product feedback](#)

# Assign licenses and manage access to Copilot Studio

Article • 11/19/2024

This article is written for tenant administrators. If you want to sign up for Copilot Studio as an individual, see the [Sign up for a Copilot Studio trial](#) topic.

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 cannot 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.

More information: [Licensing guide](#)

## ⓘ 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'll need to 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

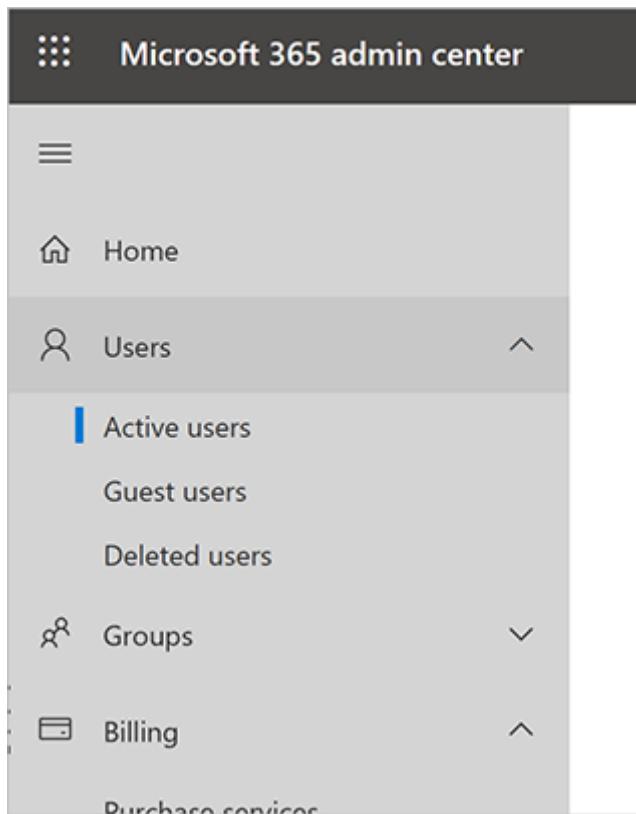
1. After you've purchased a Copilot Studio license from the Microsoft 365 admin center, you need to purchase user licenses to give users access to the product. In the Microsoft 365 admin center, go to **Billing**, and then select **Purchase services**.
2. Scroll down to the **Add-ons** section.
3. Look 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**.

This screenshot shows the "Active users" page in the Microsoft 365 admin center. The title "Active users" is at the top. Below it is a toolbar with icons for "Add a user", "Refresh", "Reset password", "Assign to group", "Manage product licenses" (which is highlighted in blue), and "Manage roles". A table follows, with columns for "Display name ↑", "Username", and "Licenses". It lists two users: "gds" (Unlicensed) and "Alex Wu" (Unlicensed). A checkmark is next to "Alex Wu".

| Display name ↑                              | Username | Licenses   |
|---------------------------------------------|----------|------------|
| gds                                         |          | Unlicensed |
| <input checked="" type="checkbox"/> Alex Wu |          | Unlicensed |

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.

#### ① Note

To simplify user license management, you can assign licenses to an 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 the [Copilot Studio portal](#).

## How to differentiate between tenant and user licenses

If you are 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](#).

See [Quotas, limits, app registration, certificates, and configuration values](#) for more information.

## Using Power Automate with a Copilot Studio license

- [Quotas, limits, app registration, certificates, and configuration values](#)
- [Use Power Automate flows](#)

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Manage Copilot Studio messages and capacity

Article • 11/19/2024

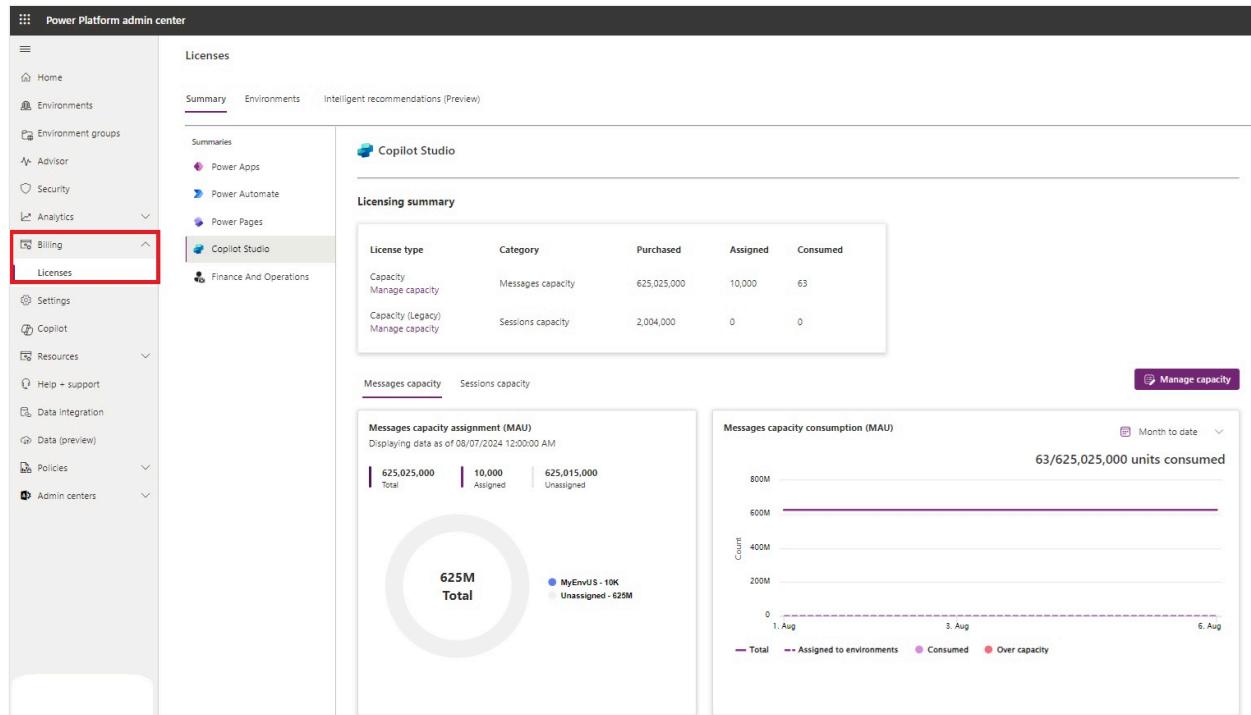
## ⓘ Note

This article is for billed messages in the Copilot Studio license, available for purchase starting December 1, 2023.

A *message* is an interaction between a customer and an agent, representing one unit of consumption. A message is defined as any request or message sent to the agent that triggers an action or a response from the agent.

The purchase of a Copilot Studio license includes a specific number of billed messages capacity. This capacity is pooled across the entire tenant but it must be assigned to an environment in to enable Copilot Studio features for agents in the environment.

Allocation happens in the [Power Platform admin center](#) when you select **Billing > Licenses > Copilot Studio**, and then select **Manage Capacity** in the upper-left menu. Additional information can be found in [Power Platform capacity add-ons](#).



## Message consumption common scenarios

The following table helps illustrate when messages are billed, free, or included in select Microsoft 365 subscriptions. These scenarios are examples only, and not all scenarios are covered.

[\[+\] Expand table](#)

| Scenario                                                                                                        | Is session billed?                               | Topic type     | Channel     | Power Automate flow            | Bot Framework extensibility | Agent stored in      |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------|----------------|-------------|--------------------------------|-----------------------------|----------------------|
| User logs vacation days in a conversation with the agent in Microsoft Teams.                                    | Included with select Microsoft 365 subscriptions | Any topic type | Teams       | Yes (only standard connectors) | Not used                    | Dataaverse for Teams |
| Agent greets user on your website. User escalates to a human agent directly or doesn't interact with the agent. | Yes                                              | System         | Any channel | Not used                       | Not used                    | Any environment type |
| Greeting topic redirects to Bot Framework Composer or a skill topic for advanced operations.                    | Yes                                              | Any topic type | Any channel | Either used or not used        | Yes                         | Any environment type |
| Greeting topic gets information about the user using a Power Automate flow to present a customized experience.  | Yes                                              | Any topic type | Any channel | Yes (any connector)            | Either used or not used     | Any environment type |

| Scenario                                                                                               | Is session billed? | Topic type | Channel     | Power Automate flow     | Bot Framework extensibility | Agent stored in |
|--------------------------------------------------------------------------------------------------------|--------------------|------------|-------------|-------------------------|-----------------------------|-----------------|
| User interacts with the agent on your website to ask about store hours, check order status, and so on. | Yes                | Non-system | Any channel | Either used or not used | Either used or not used     | Dataverse       |

Messages that use the embedded test chat don't count toward billed sessions.

## Overage

In an environment, when consumption exceeds available capacity, the environment is in overage and some features are progressively blocked. We allow some level of overage consumption, similar to a grace period, to avoid blocking business processes.

If your environment has no more capacity, you need to provide capacity to your environment: reallocate existing capacity from the organization (tenant) or environment level. Alternatively, you can purchase more capacity and make it available to your environment.

## Frequently asked questions

### Do Preview features count against usage capacity?

Copilot Studio features that are in preview release status are free to use and don't count against your purchased capacity.

### What happens to the remaining capacity at the end of the month? Can this capacity be used 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 message consumption count reset? Is resetting based on the license acquisition date?**

Credit usage is counted on a monthly basis, starting the first day of each month. Credit usage isn't based on the license acquisition date. Message usage of an environment shouldn't exceed the message allocation.

## **Where can I see message usage for this month?**

Overall organization (tenant) level credit consumption is available in the Power Platform admin center in the **Resources > Capacity > Summary** tab.

The data is aggregated daily. Some data might be delayed; if so, the consumption bar shows all known consumption at the **Last updated** date in the tooltip box.

## **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 messages aren't affected.

---

## **Feedback**

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Block unauthorized self-service agent sign-ups

Article • 10/16/2024

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 Azure PowerShell, in the Set-MsolCompanySettings.md file, set the `-AllowAdHocSubscriptions` parameter to False.

PowerShell

```
Set-MsolCompanySettings -AllowAdHocSubscriptions $False
```

For more information, see [Azure PowerShell](#).

- b. Additionally, if the customer wants to block other trial sign-ups, like Power Apps and Power Automate, run these commands next.

PowerShell

```
Remove-AllowedConsentPlans -Types "Internal"
Remove-AllowedConsentPlans -Types "Viral"
```

For more information, see [Azure 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**.
  - **PowerShell:** For more information, see [Remove Microsoft 365 licenses from user accounts with PowerShell](#).
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Work with Power Platform environments in Copilot Studio

Article • 11/21/2024

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:

- [Regional settings and data locations for organizations](#)
- [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.

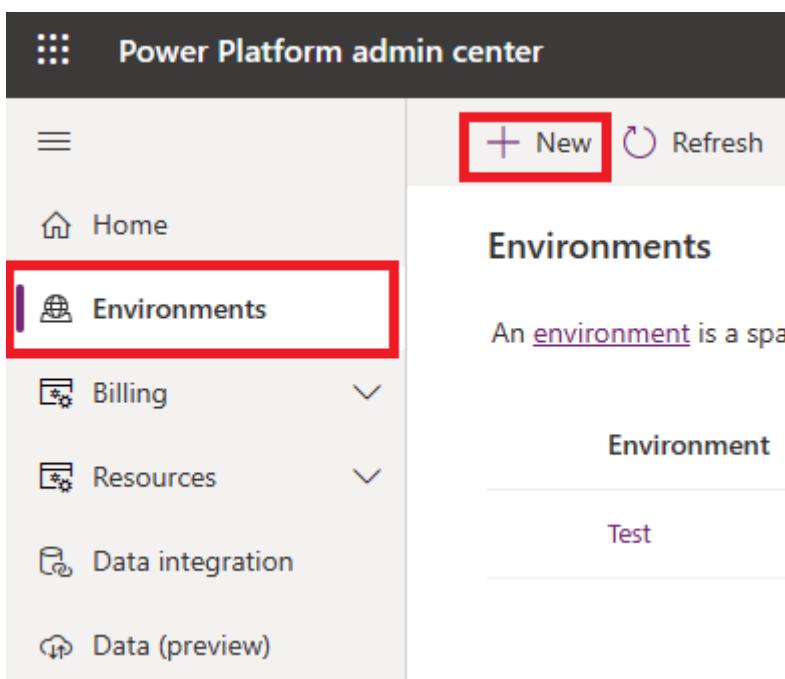
Learn more about the [types of environments](#).

# 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 <https://admin.powerplatform.com> and sign in using your work account.  
Use the same tenant that you use for Copilot Studio.
2. Go to the **Environments** tab and select **+ New** to open the **New environment** pane.



- For **Name**, specify a unique name for the environment.
- For **Type** select the environment type.

## ⚠ Note

A production environment is intended for production scenarios and won't be subject to the same restrictions as a [trial environment](#). If you are trying out Copilot Studio for free, you'll need to make sure you set the environment to **Trial**. The standard limitations for [trial environments](#) apply.

- For **Region** select the [support data region](#) where you want the environment to be created.

- Set Add a Dataverse data store? to Yes.
- Select Next.

New environment X

ⓘ This operation is subject to [capacity constraints](#)

Name \*

Region \*  
 ▼  
A local region can provide quicker data access

Type  ⓘ \*  
 ▼

Purpose

Add a Dataverse data store?  ⓘ  
 Yes

Pay-as-you-go with Azure?  ⓘ  
 No

Next Cancel

### Note

The environment needs to be created in a supported region, otherwise you won't be able to use it when creating an agent. For more information, see [Regional settings and data locations for organizations](#).

3. 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.
4. 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 the [Copilot Studio portal](#) 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 the [Copilot Studio portal](#) 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 the Copilot Studio portal, 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. For more information, see [Regional settings and data locations for organizations](#).

## Related content

- [Power Platform - Environments overview](#)
- [Power Platform - Configure environment security](#)

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

# Disable or limit sharing of agents

Article • 12/17/2024

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 doesn't impact any existing users who already have access to the app, flow, or agent prior to the application of the sharing rules. However, if an app, flow, or agent is out of compliance after rules are set, only un-sharing is allowed until the app, flow, or agent is compliant with the new rules.

After sharing rules are set in the Power Platform admin center, it may take up to an hour for them to begin enforcement.

Sharing rules in Dataverse for Teams environments do not impact sharing to a Team when you select Publish to Teams. However, when a user attempts to share with individuals or groups in a Team other than the one bound to the environment, the sharing limits are enforced.

See more about setting sharing limits for canvas apps, flows, and agents in the Power Platform administration documentation article, [Limit sharing](#).

## Use the Editor and Viewer roles to control sharing

You control sharing with individuals by setting sharing controls in the Power Platform Admin Center. This allows administrators to enforce how makers can share **Editor** or **Viewer** assignments with other individuals for their agents in Copilot Studio. See the Power Platform doc [Limit sharing](#) for instructions on setting assignments.

# 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 ^  
▼

## 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



The remainder of this article describes how the assignments impact 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                                                                                                                                                                                                               |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Let people grant Editor permissions when agents are shared</b> | 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 won't affect the ability to share with viewers by owners or editors.                                                                                 |
| <b>Let people grant Viewer permissions when agents are shared</b> | 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 won't prevent owners/editors from sharing their agents with individuals as editors. |
| <b>Only share with individuals (no security groups)</b>           | 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.                                                                                                                               |
| <b>Limit number of viewers who can access each agent</b>          | 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)

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Deploy and enable actions in Copilot Studio (preview)

Article • 11/19/2024

[This article is prerelease documentation and is subject to change.]

Users in your tenant can use conversational and AI actions in their chats with a Copilot agent if:

- Your Microsoft 365 tenant admin deploys the Copilot Studio app in the Microsoft 365 admin center.
- The user enables the connection from within their chat with a Copilot agent.

## Deploy the Copilot Studio app (admin)

1. Sign in to the Microsoft 365 admin center with your admin account.
2. Expand **Settings** on the side navigation pane and select **Integrated apps**.
3. Go to the **Available apps** tab and select the entry **Microsoft Copilot Studio**. The app's details pane opens.
4. Select **Deploy** to enable the app in chats with a Copilot agent.

The screenshot shows the Microsoft 365 Admin Center interface. On the left, the navigation pane is open with sections like Billing accounts, Copilot, Support, and Settings. Under Settings, the Integrated apps section is selected. The main content area is titled "Integrated apps" and shows the "Available apps" tab. A card for "Copilot Studio" is displayed, stating "Plugin for Microsoft Copilot is enabled". The "Basic info" section shows it is an App type, OK status, and associated with Microsoft 365 and Teams. The "Actions" section includes "Remove app" and "Edit users". At the bottom, there is a table with one row for "Copilot Echo Service".

Tenant admins can enable or disable:

- Conversational AI/topic actions and Prompt actions in the Microsoft 365 Chat app using the "Copilot Studio" app.
- Flow actions in the Microsoft 365 Chat app using the "Power Automate" app.
- Connector actions and API actions in the Microsoft 365 Chat app using the associated apps (the app name will be the connector or action name). Additionally, custom connector actions are available in the Teams store for users to install.

 **Important**

In this release, there are some limitations:

- Integration with Copilot Studio and Power Automate is enabled by default. This means that plugins created in Copilot Studio will show up for users under "Copilot Studio" or "Power Automate" even if the tenant admins have not explicitly deployed the app for users. This might change in the future.
- Note that, as an admin, you can still block plugins from showing up for your users by blocking the app in Microsoft 365 Admin Center.
- The number of Power Platform environments enabled for integration is currently limited. Reach out to Microsoft support if you have a large number of Power Platform environment (>100) and want specific Power Platform environment to be enabled.

In the future, customers can use the Power Platform admin center to enable or disable plugins in an environment for a Copilot agent.

## Enable the connection in an agent

Agent users need to enable plugins before they use them in chat. Available plugins can be enabled through the plugins menu or through the Power Platform plugins configuration portal.

Users can get to the portal by asking about plugins in their chat with a Copilot agent. They receive a short summary about plugins and a link to the portal:

M365 Chat 1:02 PM

How can I use a plugin? 

Plugins can enhance your Copilot experience by bringing in data from various external services, and are currently available for Microsoft 365 Chat in Teams. You can use natural language to query for business data from these services. Select Message Extensions and Power Platform connectors can also function as Microsoft 365 Chat Plugins. Plugins can be installed from the plugin menu, and Power Platform plugins can be enabled by visiting this link: [Power Platform Copilot plugins configuration portal](#).

In addition, you can also create your own plugin using low code [Microsoft Copilot Studio](#).

 Copy    
 4/30 AI-generated content may be incorrect

Users can also ask directed questions about data connections or on how to perform cross-organizational tasks. For example:

- *How can I use a plugin?*
- *Tell me about Power Platform agent plugins*
- *How can I get data from an external system?*
- *How can I get data from Salesforce?*

## Related content

- [Connectors, actions, and extensions overview \(preview\)](#)

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(93639a87d44eb16a4fb60db38d585a0a\_img.jpg\)](#)

# Configure data loss prevention policies for agents

Article • 11/19/2024

Organizational data is the most important asset administrators are responsible for safeguarding. The ability to build automation to use that data is a large part of their company's success.

You can rapidly build and roll out your high-value agents for your users. You can connect your agents with many data sources and services. Some of these sources and services might be external, non-Microsoft services, and might even include social networks.

It's easy to overlook the potential for exposure. This sort of exposure can result from data leakage or connections to services and audiences that shouldn't have access to the data.

Administrators can govern agents in your organization using data loss prevention (DLP) policies with existing and Copilot Studio connectors. DLP policies are created in the [Power Platform admin center](#). To create a DLP policy, you need to be a [tenant admin](#) or have the [Environment Admin](#) role.

## Prerequisites

- Review concepts about [DLP policies](#)

## Copilot Studio connectors

Copilot Studio connectors can be classified within a DLP policy under the following data groups, which are presented in the Power Platform admin center when reviewing DLP policies:

- Business
- Non-business
- Blocked

You can use the connectors in DLP policies to protect your organization's data from any malicious or unintentional data exfiltration by your agent makers.

 **Important**

By default, DLP enforcement for agents is disabled in all tenants. Learn about [enabling enforcement](#).

Once enabled, Copilot Studio supports DLP enforcement at real time. For example, both makers and users will see DLP enforcement errors when a policy is applied.

The connectors need to be in a single data group as data can't be shared among connectors that are in different groups.

Several Copilot Studio connectors are available in the Power Platform admin center. These connectors can be configured for DLP as follows:

[ ] [Expand table](#)

| Connector name                                                   | Description                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Application Insights in Copilot Studio                           | Block agent makers from <a href="#">connecting agent 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.<br><a href="#">agent users must authenticate themselves</a> to chat with the agent.<br>For more information, see <a href="#">Data loss prevention 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="#">DLP connector endpoint filtering</a> to allow or deny endpoints.                                                                                                            |
| 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="#">DLP connector endpoint filtering</a> to allow or deny endpoints.                                                                                                       |
| Knowledge source with documents in Copilot Studio                | Block agent makers from publishing agents configured with documents as a knowledge source.                                                                                                                                                                                                   |
| 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.                                                                                                                                                                                                                           |

| Connector name                     | Description                                                                                                                                                                                                                                          |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Skills with Copilot Studio         | Block agent makers from using skills in Copilot Studio agents.<br>For more information, see <a href="#">Data loss prevention example - Block skills in agents</a> and <a href="#">Data loss prevention example - Block HTTP requests in agents</a> . |
| Event triggers with Copilot Studio | Block agent makers from using event triggers in Copilot Studio agents.<br>For more information, see <a href="#">Data loss prevention example - Block event triggers in agents</a> .                                                                  |

## Example DLP policy configurations

To help you get started with Copilot Studio agent governance, we created the following examples that detail different scenarios:

- [Data loss prevention example - Require user authentication in agents](#)
- [Data loss prevention example - Block SharePoint knowledge source in agents](#)
- [Data loss prevention example - Block Power Platform Connectors in agents](#)
- [Data loss prevention example - Block HTTP requests in agents](#)
- [Data loss prevention example - Block skills in agents](#)
- [Data loss prevention example - Block event triggers in agents](#)
- [Data loss prevention example - Block channels to disable agent publish](#)

## Use PowerShell to enable and administer DLP enforcement for agents in your organization

You can configure whether DLP policies should be applied to your agents with the `PowerAppDlpErrorSettings` and `PowerVirtualAgentsDlpEnforcement` PowerShell cmdlets.

You can:

- Confirm if DLP is enabled for agents in your tenant.
- Enable or disable DLP in an auditing mode (`-Mode SoftEnabled`) so agent makers can see errors, but aren't prevented from performing actions that would be blocked if DLP enforcement was fully enabled.
- Enable or disable DLP enforcement, to show DLP enforcement errors, and prevent agent makers from publishing DLP-affected bots or configuring DLP-related settings.
- Exempt specific agents from DLP enforcement.

- Add and update the learn-more and contact email links that are shown to agent makers when they encounter DLP in the Copilot Studio web and Teams apps.

### **ⓘ Important**

Before using the PowerShell cmdlets, or the example scripts shown here, ensure you [install the following modules using PowerShell](#).

- Microsoft.PowerApps.Administration.PowerShell
- Microsoft.PowerApps.PowerShell -AllowClobber

You need to be a [tenant admin](#) to use the cmdlets.

Typically, you would use these cmdlets in accordance with a DLP rollout process, which might consist of the following steps, in order:

1. Add or update the learn-more and admin contact email links that are shown in DLP errors for agent makers.
2. Determine which (if any) agents currently have DLP policy enforcement enabled.
3. Use auditing or "soft" mode so makers can see DLP errors in the Copilot Studio web and Teams apps.
4. Mitigate risk by contacting makers and informing them about the best course of action for their app or flow.
5. Enable DLP enforcement for agents to prevent DLP-affected tasks and features.

You might also decide to exempt one or more agents from DLP policy enforcement, depending on the agent's use case and requirements.

## **Add and update the learn-more and admin contact email links**

You can configure an email and learn-more link using the `Set-PowerAppDlpErrorSettings` PowerShell cmdlet. Your agent makers will see this information when they experience DLP errors.

**Publish status**

Verify or modify the availability of your copilot

**1 error** ^

ⓘ You have errors in your draft that prevent publishing.

⊗ Due to a recent data loss prevention policy change, some issues are preventing your copilot from working correctly. Download the file to review the error details and contact your admin(admin@contoso.com). [Learn more](#)

Download

To add the email 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`.

#### ⊗ Caution

These settings apply to all Power Platform apps within the specified tenant.

## Enable and configure DLP enforcement for agents

You can enable, disable, configure, and audit DLP enforcement within Copilot Studio with the `PowerVirtualAgentsDlpEnforcement` cmdlet.

In any of the following examples, replace (or declare) <tenant ID> with your tenant's ID.

You can scope to agents created after a certain date by replacing <date> with a date in the format MM-DD-YYYY. To remove the scope, delete the `-onlyForBotsCreatedAfter` parameter and its value.

## Confirm DLP enforcement for agents

By default, DLP enforcement for agents is disabled in all tenants.

You can run the following PowerShell cmdlet to check if DLP for Copilot Studio is enabled for a tenant.

PowerShell

```
Get-PowerVirtualAgentsDlpEnforcement -TenantId <tenant ID>
```

(!) Note

If you haven't configured Copilot Studio DLP, the results from the cmdlet will be empty.

## Use auditing or "soft" mode to see DLP errors in the Copilot Studio web or Teams apps

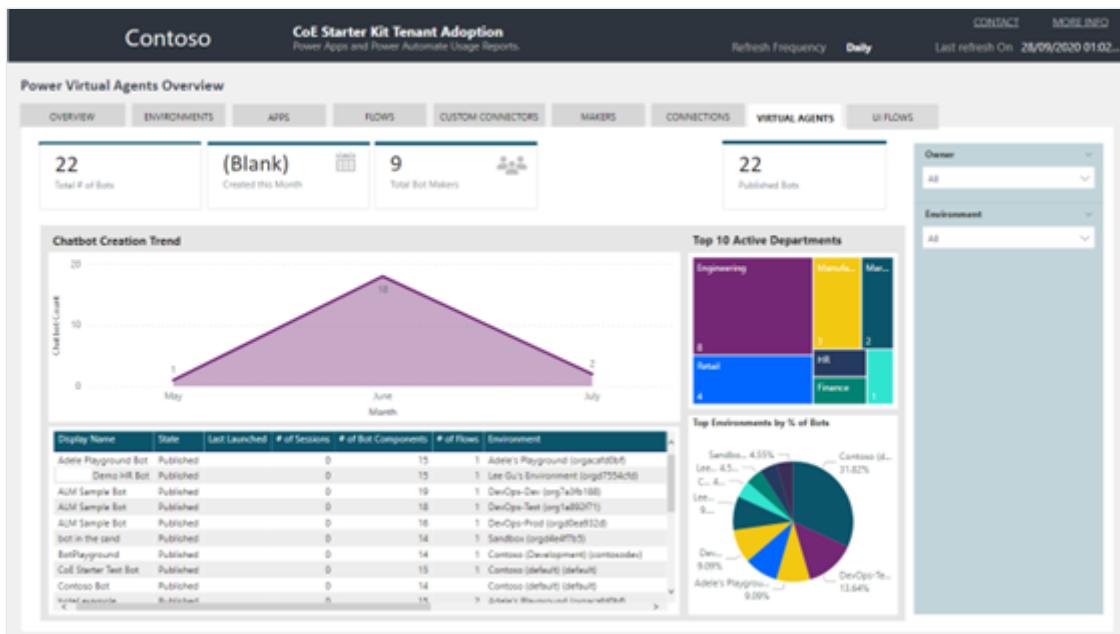
Run the following PowerShell script to enable DLP policies in auditing mode. Agent makers will see DLP-related errors when configuring agents in the Copilot Studio web and Teams apps, but they won't be blocked from performing DLP-related actions. Additionally, makers are unable to publish agents while the *soft* mode is enabled.

PowerShell

```
Set-PowerVirtualAgentsDlpEnforcement -TenantId <tenant ID> -Mode SoftEnabled
```

To find agents that could be impacted by your organization's existing DLP policies, you can:

1. Use the [Center of Excellence \(CoE\) Starter Kit](#) to get a list of agents in your organization. Go to the Copilot Studio overview page on the CoE Dashboard to see the agents and environment names in your organization.



2. Run a campaign with the agent makers in your organization to address DLP errors or updated DLP policies. You can download all agent DLP errors by selecting **Details** in the error notification banner and selecting **Download** from the error message details.

## Enable DLP enforcement for agents

### *ⓘ* Important

Before enabling DLP enforcement, ensure you know which agents will show errors to your agent users due to DLP policy violations.

If you run into issues, you can exempt an agent from DLP policies or disable DLP enforcement while your makers fix the agent to comply with DLP policies.

You can run the following PowerShell command to enforce DLP policies in Copilot Studio. Agent makers will be prevented from performing DLP-impacted actions, and users will see errors if they trigger.

PowerShell

```
Set-PowerVirtualAgentsDlpEnforcement -TenantId <tenant ID> -Mode Enabled -OnlyForBotsCreatedAfter <date>
```

## Exempt a bot from DLP policies

If you've enabled DLP enforcement for your tenant but you need to exempt an agent from showing DLP errors to makers and users, you can run the following PowerShell script.

Make sure to replace <environment ID>, <bot ID>, <tenant ID>, and <policy ID> with the appropriate IDs for the agent you want to exempt.

### Tip

You can find the <environment ID> and <bot ID> from the agent's URL.

The <policy ID> is listed alongside the error details in the **Download details** file.

You can download that file by selecting **Download details** on the error notification banner in Copilot Studio.

PowerShell

```
$environmentId = "<environment ID>"
$botId = "<bot ID>";
$tenantId = "<tenant ID>"
$policyName = "<policy ID>"

Ensure the DLP commands are installed
if (-not (Get-Command "Get-PowerAppDlpPolicyExemptResources" -ErrorAction
SilentlyContinue))
{
 Write-Host "Please ensure the Power Apps DLP commands are available:
https://docs.microsoft.com/power-platform/admin/powerapps-
powershell#environments-commands" -ForegroundColor Red
 return;
}
Set up the PVA resource information
$pvaResourceId = "$environmentId+$botId"
$pvaResourceType = "Bot"
$exemptBot = [pscustomobject]@{
 id = $pvaResourceId
 type = $pvaResourceType
}
Write-Host "Getting exempt resources"
$resources = Get-PowerAppDlpPolicyExemptResources -TenantId $tenantId -
PolicyName $policyName
if (-not $resources)
{
 $resources = [pscustomobject]@{ exemptResources = @($exemptBot) }
 Write-Host "No exempt resources configured yet"
}
$resources = New-PowerAppDlpPolicyExemptResources -TenantId $tenantId -
```

```
PolicyName $policyName -NewDlpPolicyExemptResources $resources
Write-Host "Added bot to exempt resources"
```

## Disable DLP enforcement for agents

The following command will disable DLP enforcement in agents.

PowerShell

```
Set-PowerVirtualAgentsDlpEnforcement -TenantId <tenant ID> -Mode Disabled
```

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Data loss prevention example - Require user authentication in agents

Article • 11/19/2024

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 for Authenticate with Microsoft 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.

## 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  
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 loss prevention (DLP) 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 DLP 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 DLP-related connectors, see [Configure data loss prevention policies for agents](#).

# Configure DLP to require authentication in the Power Platform admin center

## Select or create a policy

1. In the [Power Platform admin center](#), under Policies, select Data policies.
2. Create a new policy, or choose an existing policy to edit:
  - If you want to create a new policy, select **New policy**.
  - If you want to choose an existing policy to edit, select the policy and select **Edit policy**.
3. Enter a name for the policy then select **Next**. You can change the name later.

## Choose an environment

1. Choose one or more environments to add to your policy.
2. Select **+ Add to policy**.
3. Select **Next**.

## Add the connector

1. Use the search box to find the **Chat without Microsoft Entra ID authentication in Copilot Studio** connector.
2. Select the connector's **More actions** icon (:), and then select **Block**.
3. Select **Next**.
4. If you're a tenant admin, or an environment admin for multiple environments, you see the **Scope** step. Choose one or more environments that your DLP policy applies to.

### (!) Note

If the policy has a tenant scope, DLP policy will apply to all agents.

5. Review your policy, then select **Update policy** to apply the DLP changes.

## Confirm policy enforcement in Copilot Studio

You can confirm that this connector is being used in the DLP policy from the Copilot Studio web app.

First, open your agent from the environment where the DLP 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 DLP download spreadsheet details to make appropriate updates to the DLP 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

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Data loss prevention example - Block SharePoint knowledge sources

Article • 11/19/2024

You can use data loss prevention (DLP) policies to prevent agent authors for connecting to data. Doing so can help prevent data exfiltration.

For more information about other DLP policy configurations, see [Configure data loss prevention policies for agents](#).

## Configure DLP to block or configure endpoints using SharePoint in Power Platform admin center

### Select or create a policy

1. In the [Power Platform admin center](#), under **Policies**, select **Data policies**.
2. Create a new policy, or choose an existing policy to edit:
  - If you want to create a new policy, select **New policy**.
  - If you want to choose an existing policy to edit, select the policy and select **Edit policy**.
3. Enter a name for the policy then select **Next**. You can change the name later.

### Choose an environment

1. Choose one or more environments to add to your policy.
2. Select **+ Add to policy**.
3. Select **Next**.

### Add the connector

1. Use the search box to find the connector you want to block. You can see connectors that are already blocked on the **Blocked** tab.

2. Select the connector's **More actions** icon (:), and then select **Block**.

3. Select **Next**.

4. Review your policy, then select **Update policy** to apply the DLP changes.

If admins wants to allow or deny SharePoint endpoints their makers can use as knowledge sources in Copilot Studio, they can use [DLP connector endpoint filtering](#) instead of blocking it.

## Confirm policy enforcement

You can confirm that this connector is being used in the DLP policy from Copilot Studio.

1. Open your agent from the environment where the DLP 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 DLP violation.

In the details file, a row appears for each violation. If a knowledge source has a DLP 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.

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Data loss prevention example - Block Power Platform connectors in agents

Article • 11/19/2024

You can use data loss prevention (DLP) policies to prevent agent authors from configuring connectors. Doing so can help prevent data exfiltration.

For more information about other DLP policy configurations, see [Configure data loss prevention policies for agents](#).

## Configure DLP to block Power Platform connectors in the Power Platform admin center

### Select or create a policy

1. In the [Power Platform admin center](#), under Policies, select Data policies.
2. Create a new policy, or choose an existing policy to edit:
  - If you want to create a new policy, select **New policy**.
  - If you want to choose an existing policy to edit, select the policy and select **Edit policy**.
3. Enter a name for the policy then select **Next**. You can change the name later.

### Choose an environment

1. Choose one or more environments to add to your policy.
2. Select **+ Add to policy**.
3. Select **Next**.

### Add the connector

1. Use the search box to find the connector you want to block. You can see connectors that are already blocked on the **Blocked** tab.

DLP Policies > Edit Policy

Policy name: Di

Prebuilt connectors

Custom connectors

Scope

Environments

Review

Assign connectors ⓘ

Business (0) Non-business (1145) | Default Blocked (4)

Blocked connectors can't be used where this policy is applied.

|             | Name        | Blockable |
|-------------|-------------|-----------|
| Salesforce  | Salesforce  | Yes       |
| MSN Weather | MSN Weather | Yes       |
| ServiceNow  | ServiceNow  | Yes       |
| Zendesk     | Zendesk     | Yes       |

2. Select the connector's **More actions** icon (:), and then select **Block**.
3. Select **Next**.
4. Review your policy, then select **Update policy** to apply the DLP changes.

## Confirm policy enforcement

You can confirm that this connector is being used in the DLP policy from Copilot Studio:

1. Open your agent from the environment where the DLP policy is applied and go to the authoring canvas.
2. Create a new topic and add a **Call 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 DLP violation.

In the details file, a row appears for each violation. If a connector has a DLP violation, a row appears for each connector.

### (!) Note

Classic chatbots don't support Power Platform connectors.

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Data loss prevention example - Block HTTP requests in agents

Article • 11/19/2024

Agent makers in your organization can make HTTP requests with the [Send HTTP request node](#) or by [extending classic chatbots with Bot Framework Composer](#).

You can use data loss prevention policies to prevent agent makers from configuring HTTP requests to help prevent data exfiltration.

For more information about other DLP-related connectors, see [Configure data loss prevention policies for agents](#).

## Configure DLP to block HTTP requests in the Power Platform admin center

### Select or create a policy

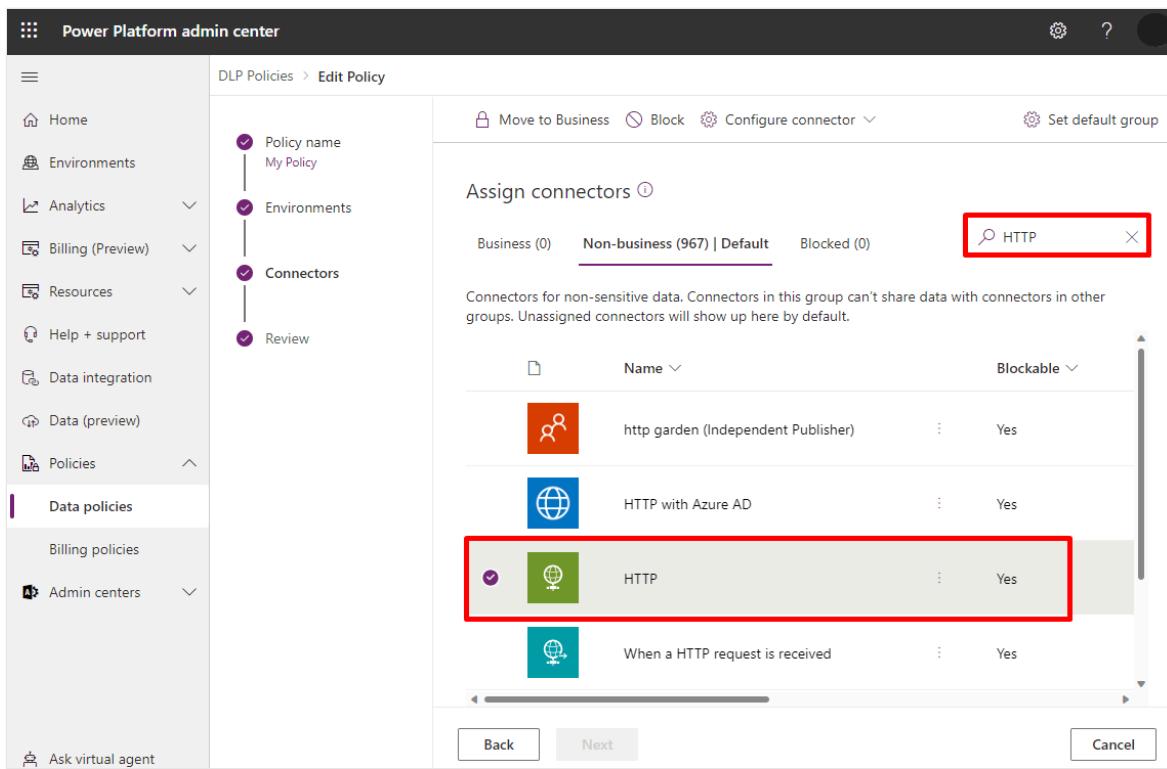
1. In the [Power Platform admin center](#), under Policies, select Data policies.
2. Create a new policy, or choose an existing policy to edit:
  - If you want to create a new policy, select New policy.
  - If you want to choose an existing policy to edit, select the policy and select Edit policy.
3. Enter a name for the policy then select Next. You can change the name later.

### Choose an environment

1. Choose one or more environments to add to your policy.
2. Select + Add to policy.
3. Select Next.

### Add the connector

1. Use the search box to find the HTTP connector.



2. Select the connector's **More actions** icon (:), and then select **Block**.
3. If admins want to allow or deny specific HTTP endpoints, they can use [DLP connector endpoint filtering](#) instead of blocking all HTTP calls.
4. Select **Next**.
5. Review your policy, then select **Update policy** to apply the DLP changes.

## Confirm policy enforcement

Web App

You can confirm that this connector is being used in the DLP policy from the Copilot Studio.

First, open your agent from the environment where the DLP 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 DLP policy, or if the HTTP policy isn't blocked but an endpoint is denied.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(d863efe41a848c99510344ae622091cd\_img.jpg\)](#)

# Data loss prevention example - Block skills in agents

Article • 11/19/2024

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 restrict their use to help prevent data exfiltration.

You can use the Skills with Copilot Studio connector in Power Platform data loss prevention (DLP) policies to stop agent makers from adding skills to their agent.

See the [Configure data loss prevention policies for agents](#) topic for information about other DLP-related connectors.

## Configure DLP to block skills in the Power Platform admin center

### Select or create a policy

1. In the [Power Platform admin center](#), under **Policies**, select **Data policies**.
2. Create a new policy, or choose an existing policy to edit:
  - If you want to create a new policy, select **New policy**.
  - If you want to choose an existing policy to edit, select the policy and select **Edit policy**.
3. Enter a name for the policy then select **Next**. You can change the name later.

### Choose an environment

1. Choose one or more environments to add to your policy.
2. Select **+ Add to policy**.
3. Select **Next**.

### Add the connector

1. Use the search box to find the **Skills with Copilot Studio** connector.
2. Select the connector's **More actions** icon (:), and then select **Block**.
3. Select **Next**.
4. Review your policy, then select **Update policy** to apply the DLP changes.

## Confirm policy enforcement in Copilot Studio

You can confirm that this connector is being used in the DLP policy from Copilot Studio.

Open your agent from the environment where the DLP 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.

---

## Feedback

Was this page helpful?



[Provide product feedback ↗](#)

# Data loss prevention example - Block channels to disable agent publish

Article • 11/19/2024

You can use data loss prevention (DLP) policies to block an agent from publishing to the channels. Copilot Studio supports four channels - Direct Line, Team, Omnichannel, and Facebook.

## Block all channels to disable agent publish

Admin can block the four channels in the Power Platform admin center using a DLP policy.

The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with various options like Home, Environments, Advisor, Security, Analytics, Billing, Settings, Copilot (preview), Resources, Help + support, Data integration, Data (preview), and Policies. The Policies option is currently selected. The main area is titled "DLP Policies > Edit Policy". A policy named "Security Build Demo (do not delete)" is being edited. Under "Prebuilt connectors", "Custom connectors", and "Scope", checkboxes are checked. In the "Assign connectors" section, the "Blocked (4)" tab is selected. A note says "Blocked connectors can't be used where this policy is applied." Below is a table listing four connectors, each with a red border around it:

| Name                                      | Blockable |
|-------------------------------------------|-----------|
| Microsoft Teams channel in Copilot Studio | Yes       |
| Direct Line channels in Copilot Studio    | Yes       |
| Facebook channel in Copilot Studio        | Yes       |
| Omnichannel in Copilot Studio             | Yes       |

Makers of agents see a disabled **Publish** button and **Channels** tiles.

 Security Copilot //Build Overview Knowledge Topics Actions Analytics Channels

1 error is preventing your copilot from being published. 1 error may be preventing your copilot from working as intended. [Details](#)

**Publish status**  
Verify or modify the availability of your copilot [Publish](#)

**1 error** [^](#)

**Due to a recent data loss prevention policy change, some issues are preventing your copilot from working correctly. Download the file to review the error details and contact your admin(admin@contoso.com). [Learn more](#)**

[Download](#)

**Published by Diganta Kumar 5/16/2024, 2:57 PM** [1 error](#) [▼](#)

**Channels**  
Configure your copilot channels to meet your customers where they are.

**Some channel aren't available due to your organization's data loss prevention policy changes. Contact your admin with questions. [Learn more](#)**

|                                                                                                   |                                                                                                      |                                                                                                  |                                                                                              |                                                                                              |
|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  Microsoft Teams |  Demo website       |  Custom website |  Mobile app |  Facebook |
|  Skype           |  Slack              |  Telegram       |  Twilio     |  Line     |
|  GroupMe         |  Direct Line Speech |  Email          |                                                                                              |                                                                                              |

## Only one channel is allowed and configured

The agent can be published by makers only if they configure the channel that admins have allowed. If they don't configure the allowed channel except Direct line (on by default), or if admins don't allow any channel, the agent cannot be published.

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Data loss prevention example - Block event triggers in agents

Article • 11/19/2024

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 may 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 loss prevention (DLP) policies to stop agent makers from adding event triggers to their agent.

For more information about DLP-related connectors, see [Configure data loss prevention policies for agents](#).

## Configure DLP to block event triggers in the Power Platform admin center

### Select or create a policy

1. In the [Power Platform admin center](#), under **Policies**, select **Data policies**.
2. Create a new policy, or choose an existing policy to edit:
  - If you want to create a new policy, select **New policy**.
  - If you want to choose an existing policy to edit, select the policy and select **Edit policy**.
3. Enter a name for the policy then select **Next**. You can change the name later.

### Choose an environment

1. Choose one or more environments to add to your policy.
2. Select **+ Add to policy**.
3. Select **Next**.

## Add the connector

1. Use the search box to find the Microsoft Copilot Studio connector.
2. Select the connector's **More actions** icon (:), and then select **Block**.
3. Select **Next**.
4. Review your policy, then select **Update policy** to apply the DLP changes.

DLP Policies > View Policy

Policy name: Block1pConnector

Prebuilt connectors

Custom connectors

Scope

Environments

Review

Set default group

Assign connectors

Business (0)   Non-business (2232) | Default   **Blocked (1)**   Exempt (39)   Search connectors

Blocked connectors can't be used where this policy is applied.

| Name                               | Blockable | Endpoint |
|------------------------------------|-----------|----------|
| Microsoft Copilot Studio (Preview) | Yes       | No       |

## Confirm policy enforcement in Copilot Studio

You can confirm that this connector is being used in the DLP policy from Copilot Studio.

Open your agent from the environment where the DLP 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

+ Add trigger

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.

When a file is modified → Trigger your agent with certain message upon event: When a file is modified.   ...

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

# View Copilot Studio audit logs

Article • 11/19/2024

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.

This article lists and describes the Copilot Studio activities that are logged and available using the Microsoft Purview compliance portal.

## Important

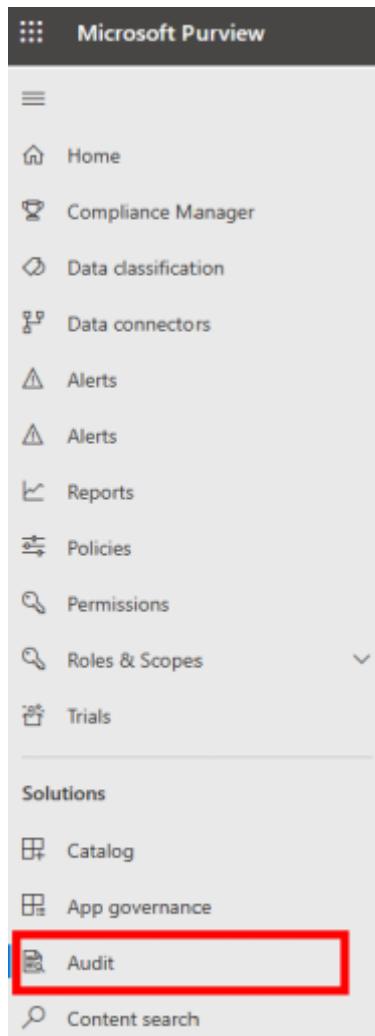
- Administrative activities for Copilot Studio are enabled by default on all tenants. You can't disable activity collection.
- At least one user with an assigned Microsoft 365 E5 or greater license, as required by Microsoft Purview. More information: [Auditing solutions in Microsoft Purview](#)

## View Copilot Studio audit logs in Microsoft Purview compliance portal

You can find logs of Copilot Studio activities in the [Microsoft Purview compliance portal](#). These logs are also accessible to developers via the [Office 365 Management API](#).

## 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. Select **Audit** from the **Solutions** category.



Admins can filter for specific activities in the **Activities** list. Copilot Studio activities are listed here.

## See audited events

All logging is done at the SDK layer, so a single action can trigger multiple logged events. Here's a list of events that you can audit.

[Expand table](#)

| Category | Event         | Description                                                        |
|----------|---------------|--------------------------------------------------------------------|
| 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                           |

| Category             | Event                     | Description                                                                           |
|----------------------|---------------------------|---------------------------------------------------------------------------------------|
| 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 (e.g. topic, skill, etc.) for an agent in Copilot Studio    |
| Agent Component      | BotComponentDelete        | The deletion of a component (e.g. topic, skill, etc.) 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                                  |
| 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                       |

## Get details on the base schema for logging

Schemas define the Power Automate fields that are sent to the Microsoft Purview compliance portal. Some fields are common to all applications that send audit data to Microsoft 365, while others are specific to Power Automate. The base schema contains these common fields.

## Common 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 operation.                                                      |
| 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 the 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 |

| Name                                    | Type                            | Mandatory | Description                                                                           |
|-----------------------------------------|---------------------------------|-----------|---------------------------------------------------------------------------------------|
| BotComponentId                          | Edm.String                      | No        | A unique identifier of the agent components, such as topics, entities                 |
| BotComponentSchemaName                  | Edm.String                      | No        | A unique string identifying the components of agent, such as topics, entities         |
| BotComponentType                        | Edm.String                      | No        | Type of the agent component, such as topics, entities                                 |
| BotComponentUpdateDetails               | Edm.ComplexType<br>(Collection) | No        | Details of the properties updated of the agent component                              |
| AIPluginOperationId                     | Edm.String                      | No        | A unique identifier for the operation with the AI plugin                              |
| AIPluginOperationName                   | Edm.String                      | No        | Name of the operation with the AI plugin                                              |
| 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 |

## 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 for Sharepoint data sources (preview)

Article • 11/19/2024

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, Personal, Public, General, Confidential, and Highly Confidential. For more information, see [Learn about sensitivity labels](#).

Microsoft Copilot Studio supports the sensitivity labels for SharePoint data sources for knowledge or generative answers classic. This capability from Microsoft Purview strengthens your data security and compliance for Copilot Studio.

## 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 SharePoint knowledge source, the copilot users see a shield icon with the highest sensitivity label from the files (such as Word, Excel, PowerPoint) used to ground the responses. The users also see a sensitivity label for each file in the response references.

Here are examples of how agent users might see a sensitivity labels in Copilot Studio web and Teams chats.

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].

The screenshot shows the Copilot Studio interface with a sidebar containing '2 references' and a 'Copilot Studio.docx' file. The sidebar also lists 'General' and 'Secure Generative Answers Share...' files, both of which have 'Confidential\Any User' sensitivity labels applied. A red box highlights the 'Confidential\Any User' label for the 'Copilot Studio.docx' file.

The screenshot shows a Microsoft Teams conversation. A message from 'ahkamal test bot' says 'Hello, now can I help you today?'. Below it, a message from 'Highly Confidential\Any User' says 'Label for highly confidential data that is accessible by any user in the tenant'. The timestamp is 'Yesterday 9:36 PM'. On the right, a user asks 'Where is Cairo?'. Below the messages, a card for 'Cairo' provides information about it being the capital of Egypt and its location. A red box highlights the 'Highly Confidential\Any User' label in the message from the bot.

## ⓘ Important

Microsoft Purview sensitivity labels in Copilot Studio are enabled by default when a maker configures an agent with a SharePoint knowledge source.

## 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](#)

# Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# 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 ↗

# Copilot Studio for US Government customers

Article • 12/13/2024

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       |
| 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 |

| <b>Microsoft Personnel</b>                                    | <b>Description</b>                                                                                                                                                                                            |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Screening and Background Checks</b> <sup>1</sup>           |                                                                                                                                                                                                               |
| 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 |
| admin.powerplatform.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.

| Feature or capability                         | Available in GCC | Available in GCC High |
|-----------------------------------------------|------------------|-----------------------|
| Copilot Studio Analytics <sup>1</sup>         | ✗                | ✗                     |
|                                               | No               | No                    |
| Copilot Studio Microsoft Teams app experience | ✗                | ✗                     |
|                                               | No               | No                    |
| Teams channel in the Copilot Studio web app   | ✓                | ✗                     |
|                                               | Yes              | No                    |
| Transfer to agents                            | ✓                | ✗                     |
|                                               | Yes              | No                    |

1. Alternatively, you can create [Custom Analytics using a Power BI dashboard \(blog\)](#).

## Requesting support

Having a problem with your service? You can create a support request to get the issue resolved.

More information: [Contact Technical Support](#)

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

# Manage extensions for Microsoft 365 Copilot for Sales (preview)

Article • 11/19/2024

[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.

## 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 any of the following roles in Dataverse:

- Microsoft Copilot administrator
- 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](#)

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Configure user authentication in Copilot Studio

Article • 11/19/2024

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 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:

- Azure Active Directory
- Azure Active Directory v2
- Azure Active Directory v2 with Certificates
- 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, <code>https://login.microsoftonline.com/common/oauth2/v2.0/authorize</code>                                                                                                                                          |
| 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 ( <a href="#">?</a> <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> ).                                                                                                      |

| Field name                            | Description                                                                                                                                                                                                                                                               |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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, <code>https://login.microsoftonline.com/common/oauth2/v2.0/token</code> .                                                                                                                                                      |
| 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've signed 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 privilege access control principle</a> . |
| 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 *actions* configured to [require user credentials](#), don't turn off authentication at the agent level, since this would prevent these actions from working.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Configure user authentication with Microsoft Entra ID

Article • 12/04/2024

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 any organizational tenant (Any Microsoft Entra ID directory - Multitenant)** and **personal Microsoft accounts (e.g. Skype, Xbox)**.
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`, and select **Configure**.

This action takes you back to the **Platform configurations** page.
4. Under **Redirect URIs** for the **Web** platform, select **Add URI**.
5. Enter `https://europe.token.botframework.com/.auth/web/redirect`, and select **Save**.

### Note

The authentication configuration pane in Copilot Studio might show the following redirect URL:

`https://unitedstates.token.botframework.com/.auth/web/redirect`. Using that URL makes the authentication fail; use the URI instead.

6. In the **Implicit grant and hybrid flows** section, select both **Access tokens (used for implicit flows)** and **ID tokens (used for implicit and hybrid flows)**.
7. Select **Save**.

## Generate a client secret

1. Under **Manage**, select **Certificates & secrets**.
2. In the **Client secrets** section, select **New client secret**.
3. (Optional) Enter a description. One is provided if left blank.
4. Select the expiry period. Select the shortest period that's relevant for the life of your agent.

5. Select **Add** to create the secret.
6. Store the secret's **Value** in a secure temporary file. You need it when you configure your agent's authentication later on.

### 💡 Tip

Don't leave the page before you copy the value of the client secret. If you do, the value is obfuscated and you must generate a new client secret.

## Configure manual authentication

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 **Azure Active Directory v2**.
  - **Client ID:** Enter the application (client) ID that you copied earlier from the Azure portal.
  - **Client secret:** Enter the client secret you generated earlier from the Azure portal.
  - **Scopes:** Enter `profile openid`.
6. Select **Save** 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.

**API permissions**

« Refresh

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of 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                   |
|------------------------|-----------|-------------------------------|
| 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**.

Request API permissions

Select an API

Microsoft APIs    APIs my organization uses    My APIs

Commonly used Microsoft APIs

**Microsoft Graph**

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.

**Azure Communication Services**

Rich communication experiences with the same secure CPaaS platform used by Microsoft Teams

**Azure DevOps**

Integrate with Azure DevOps and Azure DevOps server

**Azure Service Management**

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.

## Application

Your application needs to access the API on behalf of 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 ⓘ<br>View users' email address                                    | No                     |
| <input type="checkbox"/> offline_access ⓘ<br>Maintain access to data you have given it access to | No                     |
| <input checked="" type="checkbox"/> openid ⓘ<br>Sign users in                                    | No                     |
| <input checked="" type="checkbox"/> profile ⓘ<br>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 'Expose an API' blade in the Azure portal. On the left, there's a navigation menu with items like Overview, Quickstart, Integration assistant (preview), Manage (Branding, Authentication, Certificates & secrets, Token configuration, API permissions), and Owners. The 'Expose an API' item is highlighted with a red box. The main area has sections for Application ID URI (Set), Scopes defined by this API (with a note about defining custom scopes), and Authorized client applications (with a note about authorizing client applications). A prominent red box highlights the '+ Add a scope' button under 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<br><code>Test.Read</code> |
| Who can consent?           | Select <b>Admins and users</b>                                                       |
| Admin consent display name | Enter a name that makes sense in your environment, such as<br><code>Test.Read</code> |
| Admin consent description  | Enter <code>Allows the app to sign the user in.</code>                               |
| 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

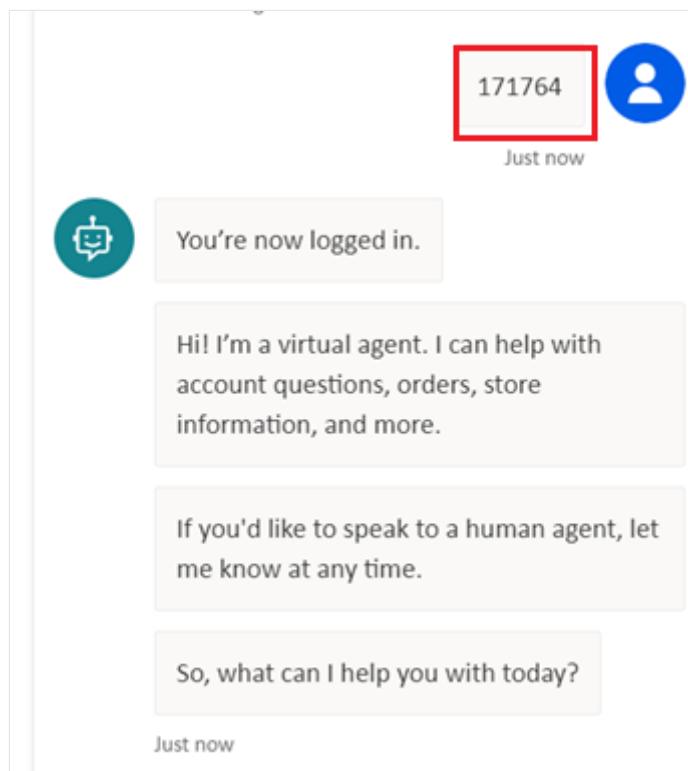
- `https://[OrgURL]/user_impersonation` for Prompt Nodes and Dataverse structured data
- For example, Dataverse Structure Data or Prompt Node should have the following scopes: `profile openid Sites.Read.All Files.Read.All`  
`https://myorg123.com/user_impersonation`

## Test your agent

1. Publish your agent.
2. In the **Test agent** pane, 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.



## Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

# Configure web and Direct Line channel security

Article • 11/19/2024

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.

Any access to the agent that doesn't provide this security measure won't work.

1. Under your agent, under **Settings**, select **Security**. Then select the **Web channel security** tile.

The screenshot shows the Microsoft Copilot Settings page. On the left, there is a sidebar with the following navigation items:

- Copilots
- Overview
- Topics & Plugins
- Entities
- Analytics
- Publish
- Extend Microsoft Copilot (preview)
- Settings** (highlighted with a red box)
- Copilot details
- AI integration tools
- Generative AI
- Channels
- Customer engagement hub
- Security** (highlighted with a red box)
- Skills

The main content area is titled "Security" and contains the following tiles:

- Sharing**: Invite people to collaborate on your copilot.
- Authentication**: Verify a user's identity during a chat.
- Web channel security** (highlighted with a red box): Review other enhanced security options.
- Allowlist**: Let other copilots call your copilot as a skill.

2. Turn the **Require secured access** switch to **Enabled**.

## Web channel security

X

Microsoft Copilot Studio provides several channels by default, some of which use Direct Line to facilitate communication between the copilot and your client application. The Direct Line channels are accessible by default without the need to configure a secret. [Learn more about Direct Line](#)

### Secrets and tokens

A Direct Line secret is a master key that can be used to initiate conversations with the associated copilot. A secret can also be used to obtain a token, which is the preferred and secure method if you're writing an application where the client runs in a web browser or mobile app. [Learn more about Direct Line secrets and tokens](#)

Secret 1

.....



**Regenerate**

**Copy**

Secret 2

.....



**Regenerate**

**Copy**

### Secured access



Require secured access has been enabled.

X

Enabling this renders the Demo website unavailable as well as any Direct Line channel not using a secret or token.

#### Require secured access

 Enabled

### ⚠ 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 will be in effect. You do not 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 switching **Require secured access** to **Disabled**. Disabling secured access can take up to two hours to propagate.

## Disable require secured access

This action renders the Demo website available, as well as any Direct Line channel not using a secret or token.

This action can take up to two hours to take effect.

**Disable**

**Cancel**

## 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 that's connected using the original secret or a token obtained from that secret will be 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.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfi
Cpg4Fv0y8qqb0F5x8qb0F5xn",
 "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 hasn'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.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfiCp
g4Fv0y8qqb0F5x8qb0F5xn
```

## 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.y8qb0F5xPGfiCpg4Fv0y8qqb0F5x8qb0F5xniaJrC8xpy8qb0F5xnR2
 vtCX7CZj0LdjAPGfiCpg4Fv0",
 "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
- 

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

# Configure single sign-on with Microsoft Entra ID

Article • 11/19/2024

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 four main steps to configuring SSO for Copilot Studio:

1. Create an app registration in Microsoft Entra ID for your custom canvas.
2. Define a custom scope for your agent.
3. Configure authentication in Copilot Studio to enable SSO.
4. Configure your custom canvas HTML 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     |

<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 has been either:

- Published to a [Power Apps portal](#).
- Published to a [SharePoint website](#) as an iframe.

However, SSO is supported for an agent that has been published to a SharePoint website as an [SPFx component](#) ↗.

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. Follow the instructions to create an authentication app registration again, to create a second app registration, which serves as your canvas app registration.
3. Add the canvas app registration ID to the authentication 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 registration blade, 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 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 **Web**.
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

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

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.

### ⓘ Note

The code in the GitHub repo requires that the user select a login button or login from a different site. To enable automatic login, add the following code to the beginning of `async function main()`:

#### HTML

```
(async function main() {
 if (clientApplication.getAccount() == null) {
 await
 clientApplication.loginPopup(requestObj).then(onSignin).catch(function (error) {console.log(error) });
 }
 // Add your BOT ID below
 var theURL =
```

1. Go to the [Overview](#) page in Azure portal and copy the **Application (client) ID** and **Directory (tenant) ID** from your canvas app registration.

Microsoft Azure

Home > App registrations > ContosoSalesCanvas

### ContosoSalesCanvas

Search (Ctrl+ /) Overview Quickstart Integration assistant (preview) Manage Branding

Delete Endpoints

Display name : ContosoSalesCanvas  
Application (client) ID :  
Directory (tenant) ID :  
Object ID :

The screenshot shows the Microsoft Azure portal's 'App registrations' section. A specific application named 'ContosoSalesCanvas' is selected. The 'Overview' tab is highlighted with a red box. To the right, application details are listed: Display name (ContosoSalesCanvas), Application (client) ID, Directory (tenant) ID, and Object ID. The 'Endpoints' button is also visible.

## 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:

```
C#

var clientApplication;
(function (){
 var msalConfig = {
 auth: {
 clientId: '00001111-aaaa-2222-bbbb-3333cccc4444',
 authority: 'https://login.microsoftonline.com/7ef988bf-
xxxx-51af-01ab-2d7fd011db47'
 },
 };
});
```

## 3. Set the `theURL` variable to the token endpoint URL you copied earlier. For example:

```
C#

(async function main() {

 var theURL = "https://<token endpoint URL>"
});
```

## 4. Edit the value of `userId` to include a custom prefix. For example:

```
C#
```

```
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.

## Test your agent using your web page

1. Open your web page in your browser.
2. Select **Login**.



### ⓘ Note

If your browser blocks popups or you are using an incognito or private browsing window, you're prompted to log in. Otherwise, the log in completes using a validation code.

A new browser tab opens.

3. Switch to the new tab and copy the validation code.
4. Switch back to the tab with your agent, and paste the validation code into the agent conversation.

## Related content

- [Azure App Registration](#)

# Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Configure single sign-on with Microsoft Entra ID for agents in Microsoft Teams

Article • 11/19/2024

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 Microsoft Teams channel app ID

1. In Copilot Studio, open the agent for which you want to configure SSO.
2. Under the settings for the agent, select **Channels**. Select the **Microsoft Teams** tile.
3. If the Microsoft Teams channel isn't connected to your agent yet, select **Turn on Teams**. For more information, see [Connect an agent to the Microsoft Teams channel](#).
4. Select **Edit details**, expand **More**, and then select **Copy** next to the **App ID** field.

## Add your Microsoft 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**.

The screenshot shows the 'Expose an API' blade in the Azure portal. On the left, there's a sidebar with icons for Overview, Quickstart, and Integration assistant. The main area has a search bar, a 'Got feedback?' link, and sections for 'Application ID URI' and 'Scopes defined by this API'. A red box highlights the 'Set' button next to the 'Application ID URI' input field.

3. Enter `api://botid-{teamsbotid}` and replace `{teamsbotid}` with your Teams channel app ID that you found earlier.

The screenshot shows a 'Set the App ID URI' dialog box. It has a title 'Set the App ID URI', an input field labeled 'Application ID URI' containing 'api://botid-', and two buttons at the bottom: 'Save' and 'Discard'. A red box highlights the 'Application ID URI' input field.

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 'Overview', 'Quickstart', 'Integration assistant', 'Branding & properties', 'Authentication', 'Certificates & secrets', 'Token configuration', 'API permissions' (which is selected and highlighted), and 'Expose an API'. The main area is titled 'Request API permissions' and shows the 'OpenId permissions (2)' section. Under this section, four permissions are listed: 'email', 'offline\_access', 'openid', and 'profile'. The 'openid' and 'profile' checkboxes are checked and highlighted with red boxes. At the bottom right of the dialog, there are 'Add permissions' and 'Discard' buttons, with the 'Add permissions' button 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**.

The screenshot shows the Azure portal's 'Expose an API' configuration page. On the left, a sidebar lists several options: Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API (which is selected and highlighted in grey), App roles, Owners, and Roles and administrators. To the right of the sidebar, there is a search bar labeled 'Search (Ctrl+ /)' and a 'Got feedback?' link. Below the sidebar, the 'Application ID URI' field is empty. A section titled 'Scopes defined by this API' contains the instruction 'Define custom scopes to restrict access to c' and 'API can request that a user or admin conser'. It also includes the note 'Adding a scope here creates only delegated application type. [Go to App roles](#)'. A red box highlights the 'Add a scope' button, which has a plus sign icon and the text '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 'Expose an API' blade in the Azure portal. On the left, there's a sidebar with links like 'Overview', 'Quickstart', and 'Integration assistant'. Under 'Manage', there are links for 'Branding & properties', 'Authentication', 'Certificates & secrets', and 'Token configuration'. The main area has a search bar and a 'Scopes' section containing 'api://'. Below that is a list of 'Authorized client applications' with a note: 'Authorizing a client application indicates this API.' At the bottom right of this section is a button labeled '+ Add a client application', which is highlighted with a red box.

2. In the **Client ID** field, enter the client ID for Microsoft Teams mobile/desktop, which is `1fec8e78-bce4-4aaf-ab1b-5451cc387264`. Select the checkbox for **the scope that you created earlier**.

The screenshot shows the 'Add a client application' dialog. It has fields for 'Client ID' (containing '1fec8e78-bce4-4aaf-ab1b-5451cc387264') and 'Authorized scopes' (containing 'api://botid-1fec8e78-bce4-4aaf-ab1b-5451cc387264 /Test.Read'). A checkbox next to the scope is checked. At the bottom are 'Add application' and 'Cancel' buttons.

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 \*

(Input field)

Client secret \*

.....

Token exchange URL (required for SSO) [Learn more about SSO](#)

f22611d/Test.Read

Tenant ID

(Input field)

Scopes ⓘ

profile openid

Save

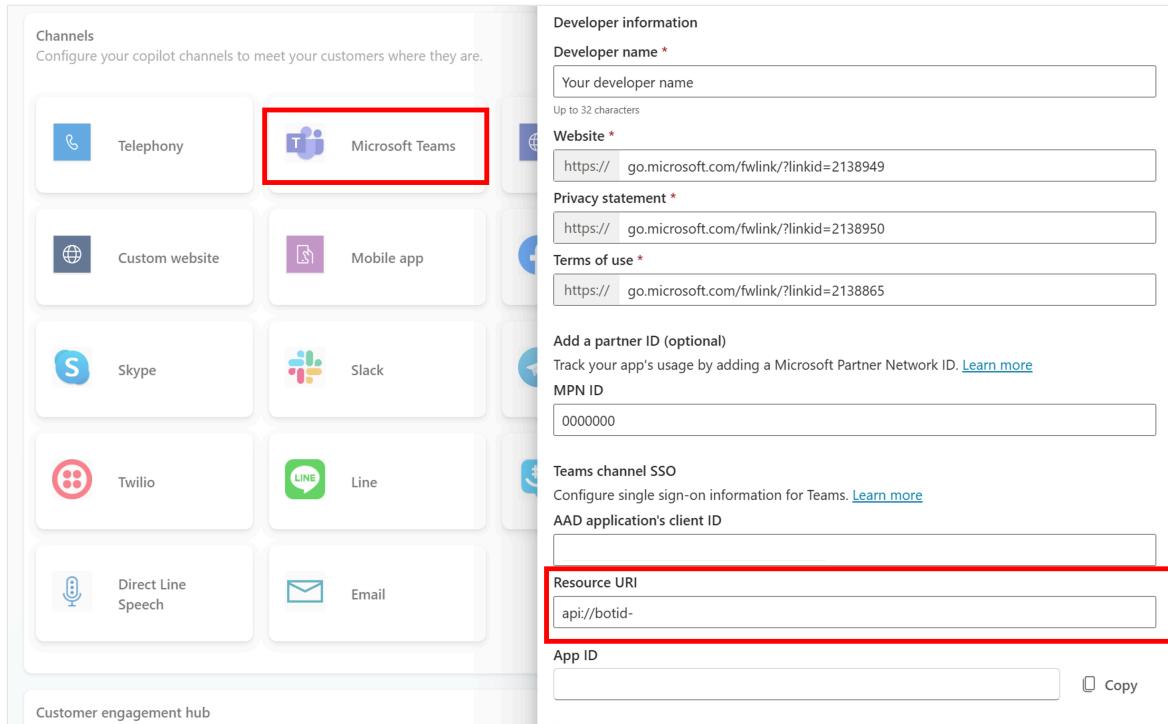
# Add SSO to your agent's Microsoft Teams channel

1. In Copilot Studio, under the settings for the agent, select **Channels**.
2. Select the **Microsoft Teams** 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 **Open the agent in Teams**, to start a new conversation with your agent in Microsoft 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.

## Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

# Configure single sign-on with a generic OAuth provider

Article • 11/19/2024

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 a configuration interface for an HTTP request. On the left, there is a large blue-bordered box containing the configuration details. On the right, a detailed view of the "HTTP Request properties" is shown.

**HTTP Request Properties (Right Panel):**

- Headers:** A section titled "Give additional information with values." It contains a key "Authorization" and a value expression "fx "Bearer " & System.User.Ac...". Below the value is a collapsed section with the expanded value "Bearer " & System.User.AccessToken".
- Body:** A section titled "Decide what to send or send nothing." It contains a dropdown menu set to "No content".
- Request timeout (milliseconds):** A dropdown menu set to "30000".
- Response headers:** A section with a dropdown menu set to "Select a variable".

**HTTP Request Configuration (Left Panel):**

- URL:** https://my.api
- Method:** Get
- Headers and body:** An "Edit" button.
- Response data type:** Record (selected from a dropdown).
- Edit schema:** A link to edit the schema.
- Save response as:** A dropdown menu currently set to "apiResponse record".

## 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

Article • 11/19/2024

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

- [Work with variables](#)
- [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**

It's recommended 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. 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 may 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 [Work with variables](#).

 [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](#).

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Configure user authentication for actions

Article • 12/03/2024

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 in a given zip code. OR when 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 log in to the experience. They are 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 may 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.

## About data access and permission management

Copilot Studio does not store any credentials and reprompts users for access if the token for access expires or is revoked on the service side. Additionally, they can manually access this connection page and refresh or revoke permissions once they are done talking to your agent.

# Supported channels

The following table details the [channels](#) that currently support user authentication for actions.

[\[+\] 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     |

<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 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](#).

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Key concepts – Analytics in Copilot Studio

Article • 11/19/2024

Use analytics to understand how well your agent is performing and to identify areas for improvement.

The Analytics tab in Copilot Studio shows you a clean overview of the key metrics for your agent, and you can drill down into each metric to get more details.

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.

## Tip

By default, the Analytics tab opens to the **Summary** page in the traditional analytics experience.

Toggle the upgraded experience on and off with the **New Analytics** switch at the top of the page.

## Sessions and conversations

Analytics for agents in Copilot Studio are based on 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.
- *Analytics sessions* track user engagement for your agent.  
Analytics sessions try to capture how well your agent handles user tasks.
- *Billed sessions* are used to measure resource consumption.  
Billed sessions are independent of analytics sessions.

Analytics sessions track user engagement with your agent and try to capture how well your agent handles user tasks.

A single conversation can contain one or more analytics sessions, and each session is associated with a specific topic. The last non-system topic triggered—or the first system topic triggered, if no non-system topics were triggered—is associated with the analytic session.

A session will time out after 30 minutes of inactivity.

## Related content

[+] [Expand table](#)

| Article                                                                                     | Description                                                                                                                                                              |
|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Review and improve agent effectiveness (new analytics)</a>                      | Use aggregated metrics to identify how users are engaging with the agent, how well the knowledge sources are being utilized, and overall user satisfaction and feedback. |
| <a href="#">Analyze agent performance and usage (traditional analytics)</a>                 | Review overall analytics for your agents.                                                                                                                                |
| <a href="#">Analyze your agent's customer satisfaction metrics (traditional analytics)</a>  | Identify overall trends in customer satisfaction, and where satisfaction is falling behind.                                                                              |
| <a href="#">Analyze topic usage (traditional analytics)</a>                                 | See how individual topics are doing and how well they're working at resolving issues.                                                                                    |
| <a href="#">Analyze session information (traditional analytics)</a>                         | Review and monitor how sessions are doing, including analytics from the topics triggered during a session.                                                               |
| <a href="#">View billed sessions (traditional analytics)</a>                                | Understand how sessions are being billed, based on their usage.                                                                                                          |
| <a href="#">Understand conversation transcripts from Power Apps (traditional analytics)</a> | Export conversation transcripts from Dataverse and create custom reports.                                                                                                |

### ① Note

Conversation transcripts in Dataverse are unavailable for download on the Copilot Studio app in Teams. To review and export transcripts in Dataverse, you'll 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](#).

Transcripts are available for download about 20 minutes after an analytic session ends, as conversation transcripts 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.

In the classic version of Copilot Studio, conversation transcripts in Dataverse are available approximately six hours after the analytic session starts.

 Note

Agent analytics features aren't yet available for Copilot Studio US Government Cloud plans. In the meantime, you can use a [custom analytics solution to create a Power BI dashboard](#)  to get analytics in these plans.

---

## Feedback

Was this page helpful?

 Yes

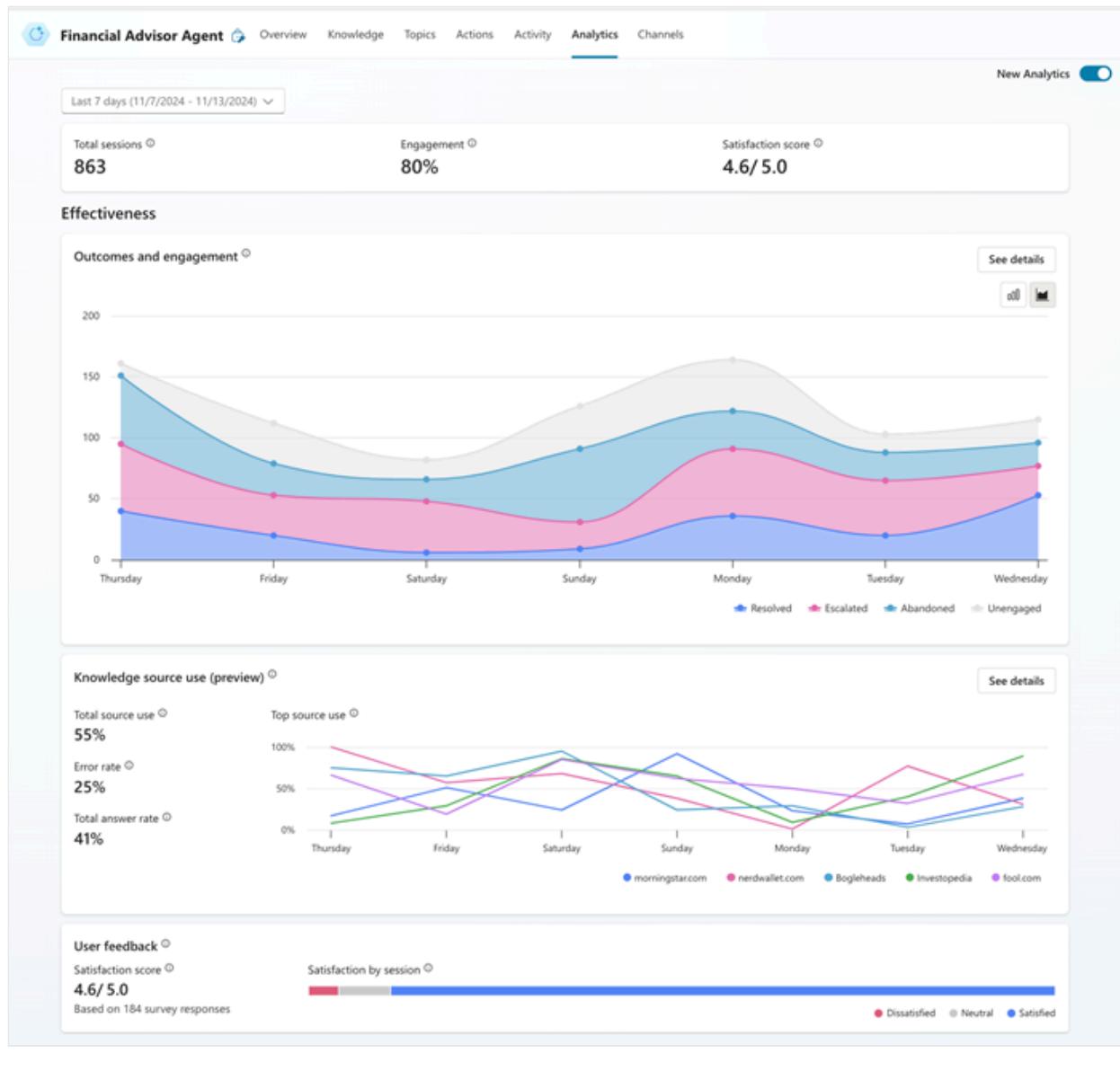
 No

[Provide product feedback](#) 

# Review and improve agent effectiveness in Copilot Studio

Article • 11/19/2024

The upgraded analytics page in Copilot Studio provides an aggregated insight into the overall effectiveness of your agent.



...

## 💡 Tip

By default, the Analytics tab opens to the **Summary** page in the traditional analytics experience.

Toggle the upgraded experience on and off with the **New Analytics** switch at the top of the page.

There are three core areas to focus on when reviewing and improving agent effectiveness:

- **Outcomes and engagement:** Knowing the end result of a conversation helps you begin to identify where your agent is succeeding and where it needs improvement.
- **Knowledge source use:** Seeing how often your knowledge sources are used helps you understand how well your agent is able to provide answers to user questions.
- **User feedback:** Reviewing user feedback helps you identify new user scenarios and issues and make improvements based directly on what your users are asking for.

### 💡 Tip

By default, the page shows you key performance indicators for the last seven days. To change the time period, use the date controls at the top of the page.

## Outcomes and engagement

The **Outcomes and engagement** 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 piechart breakdown of sessions outcomes, along with the [custom 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 non-system 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 can have 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:** A session ends and is considered *resolved* when either:
  - The **Confirmed Success** topic is triggered.

- The **End of Conversation** topic is triggered and the user confirms that the interaction was a success or lets the session time out.
- **Abandoned:** A session ends and is considered *abandoned* when an engaged session times out and it didn't reach a resolved or escalated state.

See the guidance documentation [Measuring copilot engagement](#) for suggestions and best practices on how to measure and improve engagement.

## Knowledge source use

The **Knowledge source use** section shows a chart and metrics that track how often your knowledge sources are used in conversations between your agent and users.

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 to provide answers to user questions.

The chart displays the top 5 knowledge sources used over the date period defined at the top of the **Analytics** page.

To open a side panel with a list of all knowledge sources used in the specified time period, along with trend indicators, select **See details** on the chart.

The **User feedback** 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 the survey.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Analyze agent performance and usage in Copilot Studio

Article • 11/19/2024

The **Summary** tab of the **Analytics** page gives you a broad overview of your agent's performance. It uses artificial intelligence (AI) technology to show you which topics are having the greatest effect on escalation rate, abandon rate, and resolution rate.

## 💡 Tip

Not seeing the **Summary** tab? The new upgraded analytics experience replaces the existing set of tabs with a more actionable view of your agent's performance, user feedback, and knowledge source use.

To switch between the traditional and upgraded analytics experiences, use the **New Analytics** switch at the top of the page.

Analytics sessions track user engagement with your agent and try to capture how well your agent handles user tasks. A single conversation can contain one or more analytics sessions. Each session is associated with a specific topic. The last non-system topic triggered—or the first system topic triggered, if no non-system topics were triggered—is associated with the analytic session.

The summary charts measure how well your agent is doing and help you find the topics that affect your agent's performance the most. A session will time out after 30 minutes of inactivity.

- 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.
- A session becomes *engaged* when one of the following occurs:
  - a non-system 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 can have one of the following outcomes: *resolved*, *escalated*, or *abandoned*.

- 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.)

- A session ends and is considered *resolved* when either:
  - The **Confirmed Success** topic is triggered.
  - The **End of Conversation** topic is triggered and the user confirms that the interaction was a success or lets the session time out.
- A session ends and is considered *abandoned* when an engaged session times out and it didn't reach a resolved or escalated state.

The summary tab includes various charts with graphical views of your agent's key performance indicators. The remainder of this article describes each chart in more detail.

### 💡 Tip

By default, the page shows you key performance indicators for the last seven days. To change the time period, use the date controls at the top of the page.

## Summary charts



The summary charts summarize the key performance indicators for the specified time period and the percent change over the period.

[+] Expand table

| Description     | Details                                                        |
|-----------------|----------------------------------------------------------------|
| Total sessions  | The total number of sessions within the specified time period. |
| Engagement rate | The percentage of total sessions that became engaged.          |
| Resolution rate | The percentage of engaged sessions that were resolved.         |
| Escalation rate | The percentage of engaged sessions that were escalated.        |
| Abandon rate    | The percentage of engaged sessions that were abandoned.        |

| Description | Details                                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CSAT        | The graphical view of the average of customer satisfaction (CSAT) scores for sessions in which customers respond to an end-of-session request to take the survey. |

## Engagement over time chart

The **Engagement over time** chart provides a graphical view of the number of engaged and unengaged sessions over time.

[\[+\] Expand table](#)

| Description | Details                                 |
|-------------|-----------------------------------------|
| Engaged     | The daily number of engaged sessions.   |
| Unengaged   | The daily number of unengaged sessions. |

## Session outcomes over time chart

The **Session outcomes over time** chart provides a graphical view of the daily resolution rate, escalation rate, and abandon rate over the specified time period.

[\[+\] Expand table](#)

| Description | Details                               |
|-------------|---------------------------------------|
| Resolved    | The daily rate of resolved sessions.  |
| Escalated   | The daily rate of escalated sessions. |
| Abandoned   | The daily rate of abandoned sessions. |

## Resolution rate drivers chart

The **Resolution rate drivers** chart displays topics in order of their impact on the resolution rate over the specified time period.

[\[+\] Expand table](#)

| Description | Details                                                                                                                                                                              |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Topic       | A Copilot Studio topic.                                                                                                                                                              |
| Rate        | The percentage of engaged sessions for the topic that were resolved.                                                                                                                 |
| Impact      | The topic's resolution rate impact score. The resolution rate impact score is the overall resolution rate including the topic minus the overall resolution rate excluding the topic. |

The chart displays the impact as a percentage. Topics with a positive impact percentage are topics with a higher than average resolution rate—they increased the overall resolution rate of your agent.

Increase the resolution rate of topics with high negative impact percentages to more quickly improve the overall resolution rate for your agent.

To see additional information about each topic, select the **Detail** link to display the topic details page. For more information, see [Analyze topic usage](#).

## Escalation rate drivers chart

The **Escalation rate drivers** chart displays topics in order of their impact on the escalation rate over the specified time period.

[ ] [Expand table](#)

| Description | Details                                                                                                                                                                              |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Topic       | A Copilot Studio topic.                                                                                                                                                              |
| Rate        | The percentage of engaged sessions for the topic that were escalated.                                                                                                                |
| Impact      | The topic's escalation rate impact score. The escalation rate impact score is the overall escalation rate including the topic minus the overall escalation rate excluding the topic. |

The chart displays the impact as a percentage. Topics with a positive impact percentage are topics with a greater than average escalation rate—they increased the overall escalation rate of your agent.

Lower the escalation rate of topics with high impact percentages to more quickly decrease the escalation rate for your agent.

To see additional information about each topic, select the **Detail** link to display the topic details page. For more information, see [Analyze topic usage](#).

# Abandon rate drivers chart

The **Abandon rate drivers** chart displays topics in order of their impact on the abandon rate over the specified time period.

 Expand table

| Description | Details                                                                                                                                                                  |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Topic       | A Copilot Studio topic.                                                                                                                                                  |
| Rate        | The percentage of engaged sessions for the topic that were abandoned.                                                                                                    |
| Impact      | The topic's abandon rate impact score. The abandon rate impact score is the overall abandon rate including the topic minus the overall abandon rate excluding the topic. |

The chart displays the impact as a percentage. Topics with a positive impact percentage are topics with a greater than average abandon rate—they increased the overall abandon rate of your agent.

Lower the abandon rate of topics with high impact percentages to more quickly decrease the abandon rate for your agent.

To see additional information about each topic, select the **Detail** link to display the topic details page. For more information, see [Analyze topic usage](#).

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Analyze your agent's customer satisfaction metrics (preview)

Article • 11/19/2024

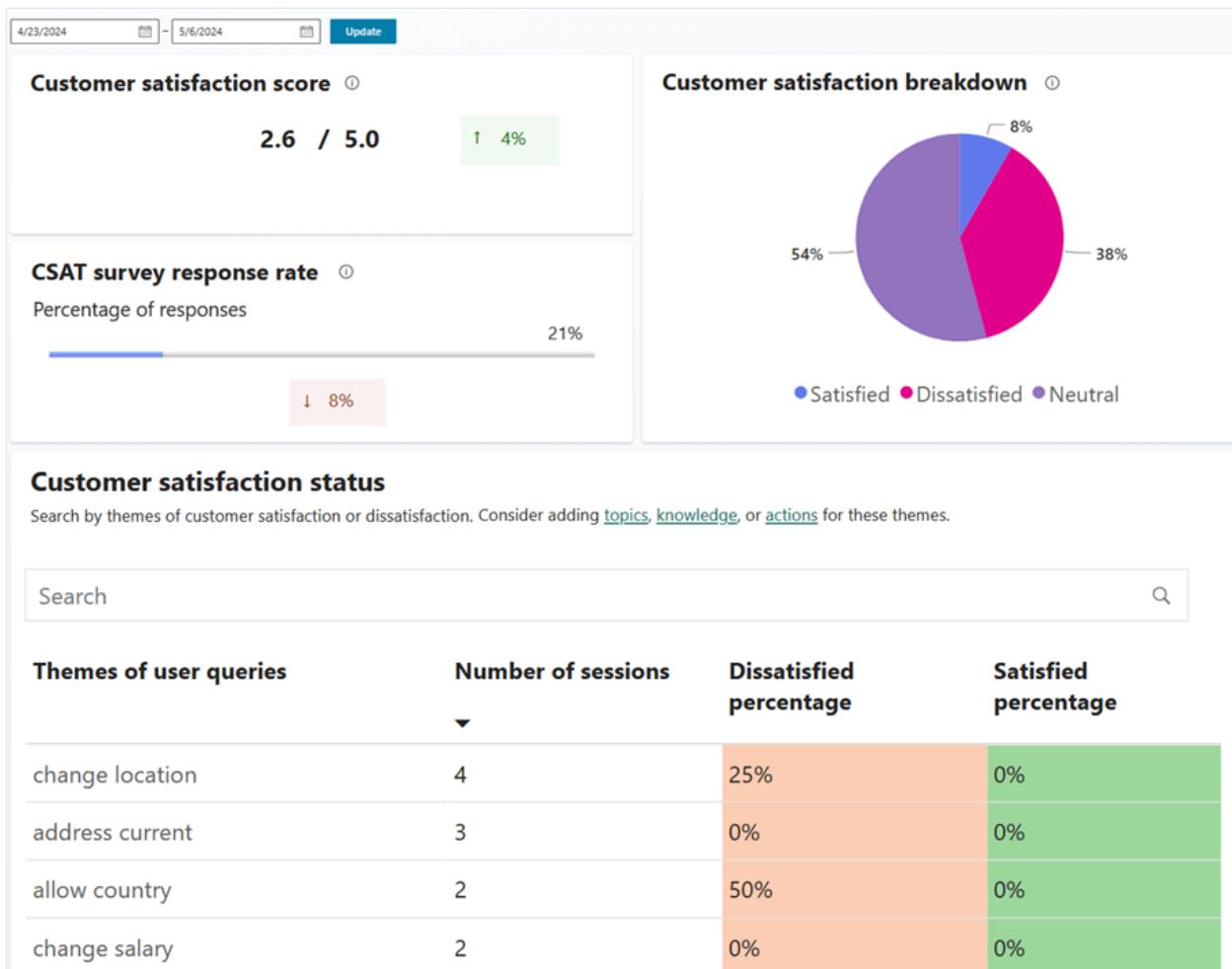
The **Customer Satisfaction** tab of the **Analytics** page provides a detailed view of customer satisfaction (CSAT) survey data, including the average CSAT score, primary user query themes, and actionable insights on drivers of satisfaction or dissatisfaction with your agent's responses.

## Tip

Not seeing the **Customer Satisfaction** tab? The new upgraded analytics experience replaces the existing set of tabs with a more actionable view of your agent's performance, user feedback, and knowledge source use.

To switch between the traditional and upgraded analytics experiences, use the **New Analytics** switch at the top of the page.

By default, the page shows key performance indicators for the last seven days. To change the time period, use the date pickers at the top of the page. You can retrieve data for any period within the last 45 days.



## Customer satisfaction score

The **Customer satisfaction score** chart provides a graphical view of the average of CSAT scores for sessions in which customers respond to an [end of conversation](#) request to take the survey. The CSAT survey asks customers to rate their experience on a scale of 1 to 5. If a customer responds to more than one survey in the same session, only the most recent is used.

This chart also provides a period-over-period change indicator. For example, when you select a three-day period, the indicator shows the percentage change relative to the three days before the selected period. The period-over-period indicator only appears if your agent has CSAT survey data available for the prior period. If there are no CSAT survey data available for the same trailing period of time relative to what is selected in the filter, then the period-over-period indicator doesn't appear.

## CSAT survey response rate

The **CSAT survey response rate** chart shows the number of [end of conversation](#) CSAT surveys that were presented and the percentage of surveys that were completed.

# Customer satisfaction breakdown

The Customer satisfaction breakdown chart shows that percentage of sessions that were satisfied, dissatisfied, or neutral in the selected time period. The [Customer satisfaction status pane](#) provides more detail on the various signals used to determine session satisfaction status.

## Customer satisfaction status

The Customer satisfaction status chart provides key insights on the themes users search for, and the satisfaction level of the users about the agent's responses. Sessions with similar themes are grouped together. The chart shows the number of sessions for each theme during the selected period, and the percentage of these sessions that were satisfied or dissatisfied. Sessions that weren't satisfied and not dissatisfied are considered neutral sessions and don't appear on this chart.

The theme of a given session is derived using ML models. Before being sent to the analytics dashboard, themes are processed to remove any personal data or sensitive information, such as phone numbers. Additionally, if themes contain profane or harmful language, these themes are masked.

To see the specific satisfaction or dissatisfaction drivers, hover over each segment of the chart. If any of the criteria are true for a given session, then the session is classified as satisfied or dissatisfied:

- A session is considered **dissatisfied** if:
  - The user gave two stars or less at the [end of conversation](#) survey.
  - The user was asked to rephrase their query twice or more than twice in the ([system fallback topic](#)).
  - The user abandoned the session.
  - The user escalated the session to a live agent.
  - The overall sentiment of the user about their conversation with the agent is classified as negative. Sentiment is determined using a publicly available ML model fine-tuned for sentiment analysis.
- A session is considered **satisfied** if:
  - The user gave four stars or more at the [end of conversation](#) survey.
  - The user wasn't asked to rephrase their query more than once in the ([system fallback topic](#)).
  - The session was resolved.
  - The overall sentiment of the user about their conversation with the agent is classified as positive.

Sessions that don't meet any of the above criteria are considered neutral sessions and don't appear on this chart.

On the [Summary tab](#) of the [Analytics](#) page, you can use the Information icons to learn more about engagement, escalation, abandon, and resolution rates.

## Theme and session sentiment extraction

Copilot Studio uses natural language processing (NLP) techniques to extract themes and assign sentiment to a given agent session.

For every session, Copilot Studio extracts themes from the first user utterance. Individual sessions with similar themes are aggregated, and appear as a single item on the [Customer satisfaction status](#) chart.

To evaluate session sentiment, the underlying NLP model is trained on public English language datasets. This process analyzes the text of the session to determine if the overall sentiment is positive, negative, or neutral. The process also preprocesses user queries to remove false positives. For example, this preprocessing ensures that a query such as "what is the best option?" isn't classified as positive solely because the word "best" appears in the query.

---

## Feedback

Was this page helpful?



[Provide product feedback ↗](#)

# Analyze topic usage in Copilot Studio

Article • 11/19/2024

The topic analytics pane provides a view into the performance of an individual topic and how it can be improved.

To display the topic analytics pane, select the **Detail** link for the topic in one of the following charts on the [Summary](#) or [Customer Satisfaction](#) tabs:

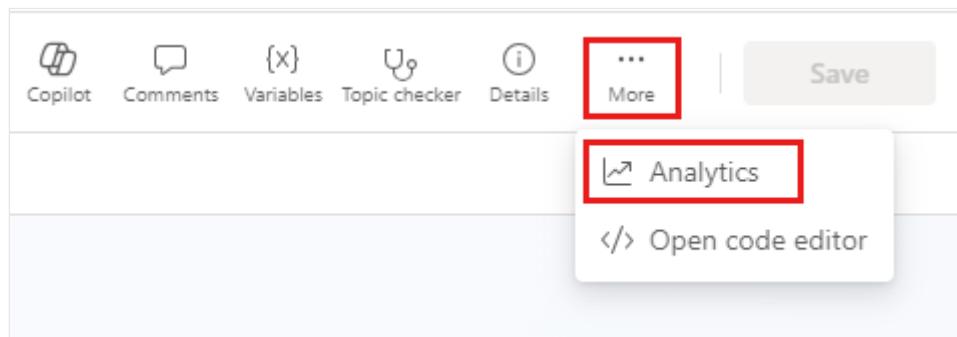
- Escalation rate drivers (Summary tab)
- Abandon rate drivers (Summary tab)
- Resolution rate drivers (Summary tab)

## Tip

Not seeing the **Summary** and **Customer Satisfaction** tabs? The new upgraded analytics experience replaces the existing set of tabs with a more actionable view of your agent's performance, user feedback, and knowledge source use.

To switch between the traditional and upgraded analytics experiences, use the **New Analytics** switch at the top of the page.

Alternatively, open the desired topic from the [Topics page](#), and select **More > Analytics**.



The topic details page includes various charts with graphical views of a topic's key performance indicators. For information about each chart, see:

- Analyze topic usage in Copilot Studio
  - [Topic summary charts](#)
  - [Impact summary charts](#)
  - [Topic Volume by Day chart](#)

## Topic summary charts

The Topic summary charts summarize the topic's performance indicators for the specified time period and the percent change over the period.

[Expand table](#)

| Description     | Details                                                                              |
|-----------------|--------------------------------------------------------------------------------------|
| Total sessions  | The total number of sessions within the specified time period.                       |
| Average CSAT    | The average customer satisfaction (CSAT) survey score for the specified time period. |
| Resolution rate | The percentage of engaged sessions that are resolved.                                |
| Escalation rate | The percentage of engaged sessions that are escalated.                               |
| Abandon rate    | The percentage of engaged sessions that are abandoned.                               |

## Impact summary charts

The Impact summary charts summarize the effects of the topic on key performance indicators for the specified time period.

[Expand table](#)

| Description            | Details                                                                                                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CSAT impact            | The topic's customer satisfaction impact score. The customer satisfaction impact score is the overall average CSAT survey score including the topic minus the overall average CSAT survey score excluding the topic. |
| Resolution rate impact | The topic's resolution-rate impact score. The resolution-rate impact score is the overall resolution rate including the topic minus the resolution rate excluding the topic.                                         |
| Escalation rate impact | The topic's escalation-rate impact score. The escalation-rate impact score is the overall escalation rate including the topic minus the escalation rate excluding the topic.                                         |
| Abandon rate impact    | The topic's abandon-rate impact score. The abandon-rate impact score is the overall abandon rate including the topic minus the abandon rate excluding the topic.                                                     |

## Topic Volume by Day chart

The Topic volume by day chart provides a graphical view of the number of sessions for the topic over the specified time period.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Capture telemetry with Application Insights

Article • 11/12/2024

This article discusses how you can capture telemetry data from your Microsoft Copilot Studio copilot for use in [Azure Application Insights](#).

In addition to the native analytics features within Microsoft Copilot Studio, you can send telemetry data to Application Insights. Telemetry offers insights into your copilot by tracking:

- Logged messages and events sent to and from your copilot
- 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](#) ↗.

Web App

## Connect your Copilot Studio copilot to Application Insights

To connect your copilot to Application Insights, you first need to add your instrumentation key to the project.

1. Navigate to the **Copilot details** page under **Settings**.
2. Switch to the **Advanced** tab.
3. Within the **Application Insights** section, populate the **Connection string** setting. See the [Azure Monitor documentation](#) ↗ to find out how to locate your connection string.
4. 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 Microsoft 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 Microsoft 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 the Microsoft Copilot Studio test canvas. 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 Microsoft 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                               |
| designMode    | Conversation happened within the test canvas | True / False                                     |

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

# Download conversation transcripts in Copilot Studio

Article • 11/19/2024

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.

This article covers downloading conversation transcripts in Copilot Studio. To download transcripts in Power Apps, see [Download conversation transcripts in Power Apps](#).

By default, you can download up to seven days of agent conversation transcripts from the past 30 days directly from the Copilot Studio portal.

## Prerequisites

- To view transcripts, 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 **Sessions** tab.

### 💡 Tip

Not seeing the **Sessions** tab? The new upgraded analytics experience replaces the existing set of tabs with a more actionable view of your agent's performance, user feedback, and knowledge source use.

To switch between the traditional and upgraded analytics experiences, use the **New Analytics** switch at the top of the page.

4. Select a row to download the session transcripts for the specified time frame.

Summary Customer satisfaction Sessions Billing +1 more 7/12/2024 - 7/18/2024 Update

Download sessions from the past 28 days, up to 7 days at a time (starting at midnight UTC).

Sessions for download

[7/12, 12:00 AM EDT - 7/12, 7:59 PM EDT](#)

[7/16, 8:00 PM EDT - 7/17, 7:59 PM EDT](#)

[7/17, 8:00 PM EDT - 7/18, 4:54 PM EDT](#)

### !Note

Sessions are split into rows representing 1-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, the sessions are represented by a single row with up to 2,500 sessions. Otherwise, sessions are split into rows that represent 1-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:

text

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.
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Download conversation transcripts in Power Apps

Article • 11/19/2024

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.

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 [Analyze session information in Copilot Studio](#).

By default, Power Apps downloads conversation transcripts from the last 30 days. [You can change the retention period](#).

## ⓘ Note

If you no longer have access to your environment, you'll need to [create a support request](#) in the Power Platform admin center.

## Prerequisite

- To view conversation transcripts, you need the transcript viewer security role. Only admins can grant this role by [assigning the Transcript viewer security role during agent sharing](#).

## View and export agent conversation transcripts from the Power Apps portal

You can view and export agent conversation transcripts from the Power Apps portal.

First, sign in to <https://www.powerapps.com> with your credentials.

### View conversation transcripts

1. In the side pane, select the **Tables** node, and then select **All**.

The screenshot shows the 'Tables' section of the Power Apps portal. On the left, there's a sidebar with 'Tables' highlighted. The main area displays a table of existing tables, with the 'All' filter selected. The columns include Name, Type, Managed, Customizable, and Tags. Some rows are partially visible.

| Name                     | Type     | Managed | Customizable | Tags     |
|--------------------------|----------|---------|--------------|----------|
| AAD User                 | Virtual  | Yes     | Yes          | Standard |
| Account                  | Standard | Yes     | Yes          | Core     |
| Action Card              | Standard | Yes     | Yes          | Standard |
| Activity                 | Standard | Yes     | Yes          | Standard |
| Activity File Attachment | Standard | Yes     | Yes          | Standard |
| Address                  | Standard | Yes     | Yes          | Standard |
| AI Builder Dataset       | Standard | Yes     | No           | Managed  |
| AI Builder Dataset File  | Standard | Yes     | Yes          | Standard |

2. Enter *conversation* in the Search box.

3. Select ConversationTranscript under Tables.

A page for the ConversationTranscript table opens.

## Export conversation transcripts

1. In the side pane, select the Tables node, and then select All.

This screenshot is identical to the one above, showing the 'Tables' section of the Power Apps portal with the 'All' filter selected and the 'Tables' node highlighted in the sidebar.

| Name                     | Type     | Managed | Customizable | Tags     |
|--------------------------|----------|---------|--------------|----------|
| AAD User                 | Virtual  | Yes     | Yes          | Standard |
| Account                  | Standard | Yes     | Yes          | Core     |
| Action Card              | Standard | Yes     | Yes          | Standard |
| Activity                 | Standard | Yes     | Yes          | Standard |
| Activity File Attachment | Standard | Yes     | Yes          | Standard |
| Address                  | Standard | Yes     | Yes          | Standard |
| AI Builder Dataset       | Standard | Yes     | No           | Managed  |
| AI Builder Dataset File  | Standard | Yes     | Yes          | Standard |

2. Enter *conversation* in the Search box.

3. Select ConversationTranscript under Tables.

4. Select Export > Export data.

It takes a few minutes for the data to be compiled for export.

The screenshot shows the Power Apps interface with the navigation bar at the top. In the center, there's a table named 'ConversationTranscript'. The 'Table properties' section shows 'Name' and 'Primary column' both set to 'ConversationTranscript'. On the right, there are tabs for 'Schema', 'Data experiences', and 'Forms'. At the top right, there's a 'Export' dropdown menu with a red box around it, and a red arrow points to the 'Export data' option under it.

5. To download the content, select **Download exported data**.

A screenshot of a confirmation message. It says: 'ConversationTranscript was exported successfully. Click the link below to download the exported data.' Below this, there is a button labeled 'Download exported data' which is highlighted with a red box.

## Download agent conversation transcripts

1. In the Power Apps navigation menu, select **Dataverse > Tables**, and then select **All**.
2. Enter **conversation** in the **Search** box.
3. Select the **ConversationTranscript** table.
4. Select **Export > Export data**.

The screenshot shows the Power Apps interface with the navigation bar at the top. In the center, there's a table named 'ConversationTranscript'. The 'Table properties' section shows 'Name' and 'Primary column' both set to 'ConversationTranscript'. On the right, there are tabs for 'Properties' and 'Tools'. At the top right, there's a 'Export' dropdown menu with a red box around it, and a red arrow points to the 'Export data' option under it. Other options like 'Link to Azure Synapse' are also visible.

5. 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 zipped 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.

[\[+\] Expand table](#)

| Field                      | Description                                                                                                                                                                                                                                                                                    | Example                                                                                                                         |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Content                    | The entire transcript in JSON format                                                                                                                                                                                                                                                           | Refer to the <a href="#">Content field</a> section.                                                                             |
| 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-eeee3333ffff", "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:

[+] Expand table

| 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>                                                                                                                                                                                                                                                                                                                         | Contains fields <code>id</code> and <code>role</code> : <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> |
| Notes:<br>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.<br>For security and privacy, the ID is hashed before it's written to the transcript. |                                                                                                                                                                                                                                                                                                                                                                       |
| <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 data</code>                                                                                                                                                                                                                      |

| Key         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| text        | The text for <code>message</code> activities                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| value       | Fields specific to the activity based on the value type; this field is where most of the useful information exists                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| channeldata | <ul style="list-style-type: none"> <li>Contains channel data: <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> <li><code>enableDiagnostics</code></li> <li><code>clientTimestamp</code></li> <li><code>clientActivityId</code></li> </ul> </li> </ul> </li> </ul> |
| name        | 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 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                                                                               |

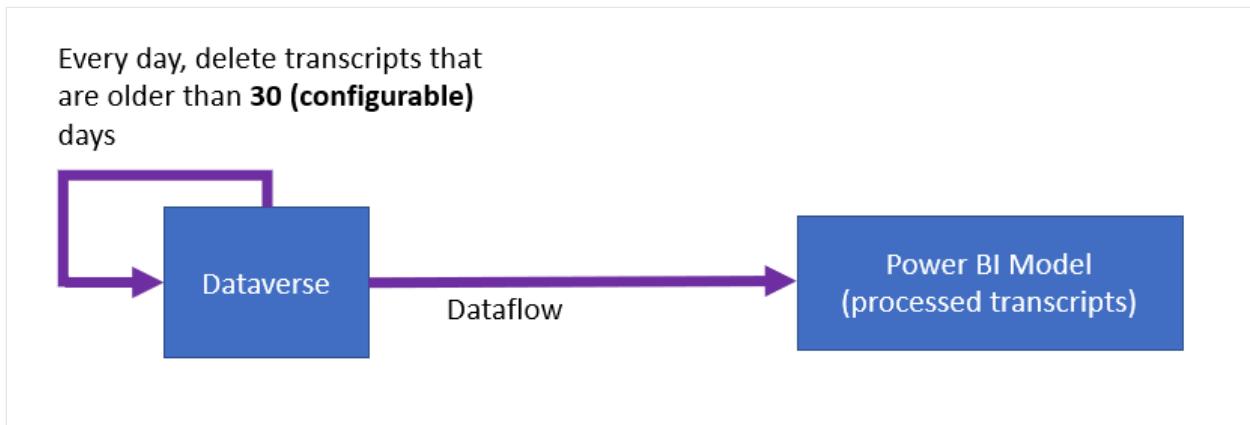
| Activity value type | Description                                                                                                                                                                       |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SessionInfo         | The type ( <code>unengaged</code> or <code>engaged</code> ), outcome ( <code>Escalated</code> , <code>Resolved</code> , <code>Abandon</code> ), and the turn count of the session |
| VariableAssignment  | A value is assigned to a variable                                                                                                                                                 |

## Build custom Power BI reports from conversation transcripts

A useful way to approach building custom reports from conversation transcripts is to look at the number of messages that need to be processed.

### Lightweight approach

Connect Power BI to Dataverse with a Power BI dataflow [↗](#). Other infrastructure isn't required and all records are retrieved from Dataverse with every refresh.



[\[\] Expand table](#)

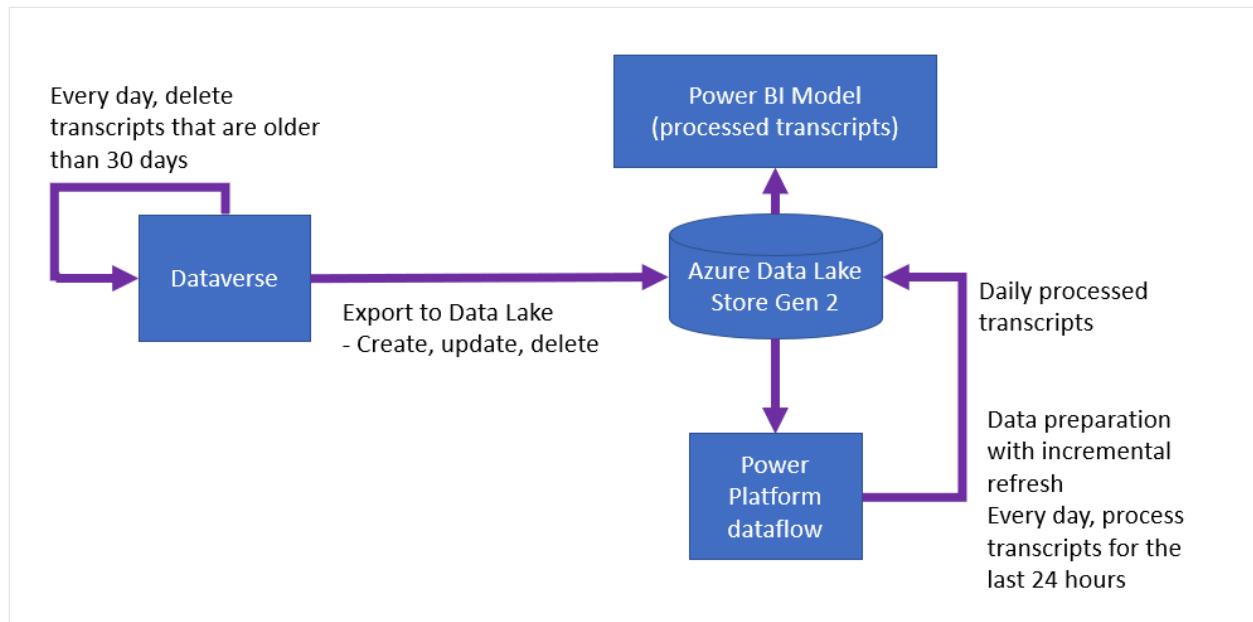
| Pros                    | Cons                                                                                        |
|-------------------------|---------------------------------------------------------------------------------------------|
| Easiest implementation. | Refresh time might increase as transcripts increase.<br>High cost for storage in Dataverse. |

### Standard Azure Data Lake Storage approach

Export Dataverse data to Azure Data Lake Storage. This method requires an Azure Data Lake Storage account, but no other Azure infrastructure.

Use a Power Platform dataflow on the transcripts to prepare the data, and extract the data into a structured form.

Power BI refreshes data from the dataflow. Power Platform and Power BI manage all compute infrastructure. Records for a certain day are retrieved from Dataverse with every refresh.

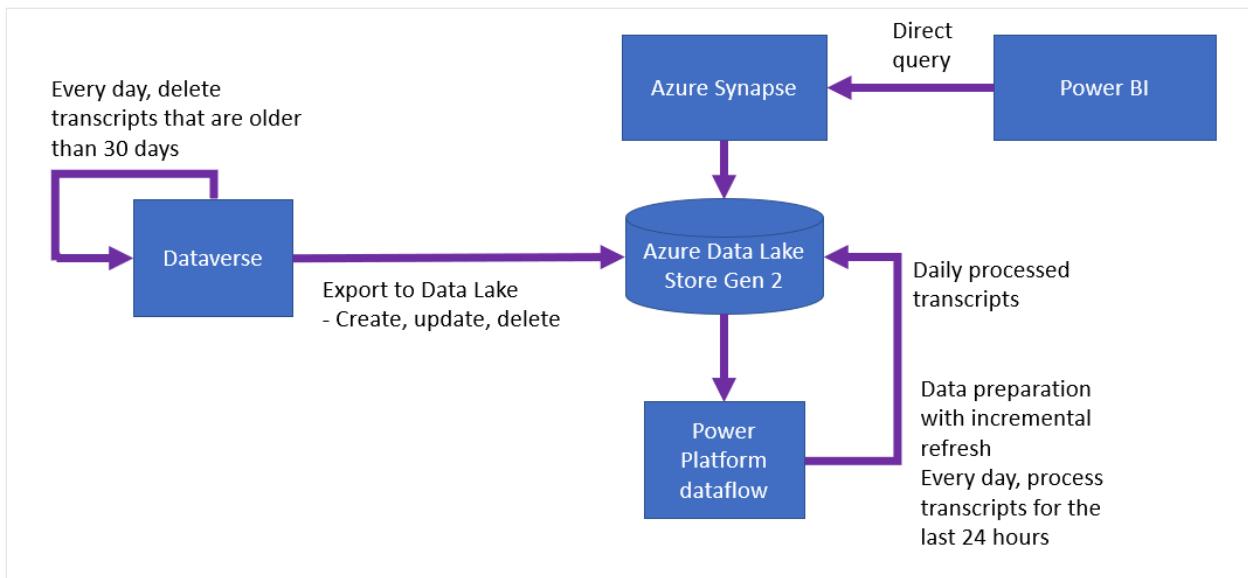


[+] [Expand table](#)

| Pros                                                                                                                                                                      | Cons                                                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>Low storage cost in Azure Data Lake Storage.</li><li>Moderately simple implementation.</li><li>No Azure Compute required.</li></ul> | <ul style="list-style-type: none"><li>Refresh time might increase as transcripts increase each day.</li></ul> |

## Azure Data Lake Storage + Synapse approach

The data size in the Standard Azure Data Lake Storage approach can cause slow refreshes or other operational problems. In that case, you can use [Azure Synapse Analytics](#) and integrate it into Power BI.



[ ] [Expand table](#)

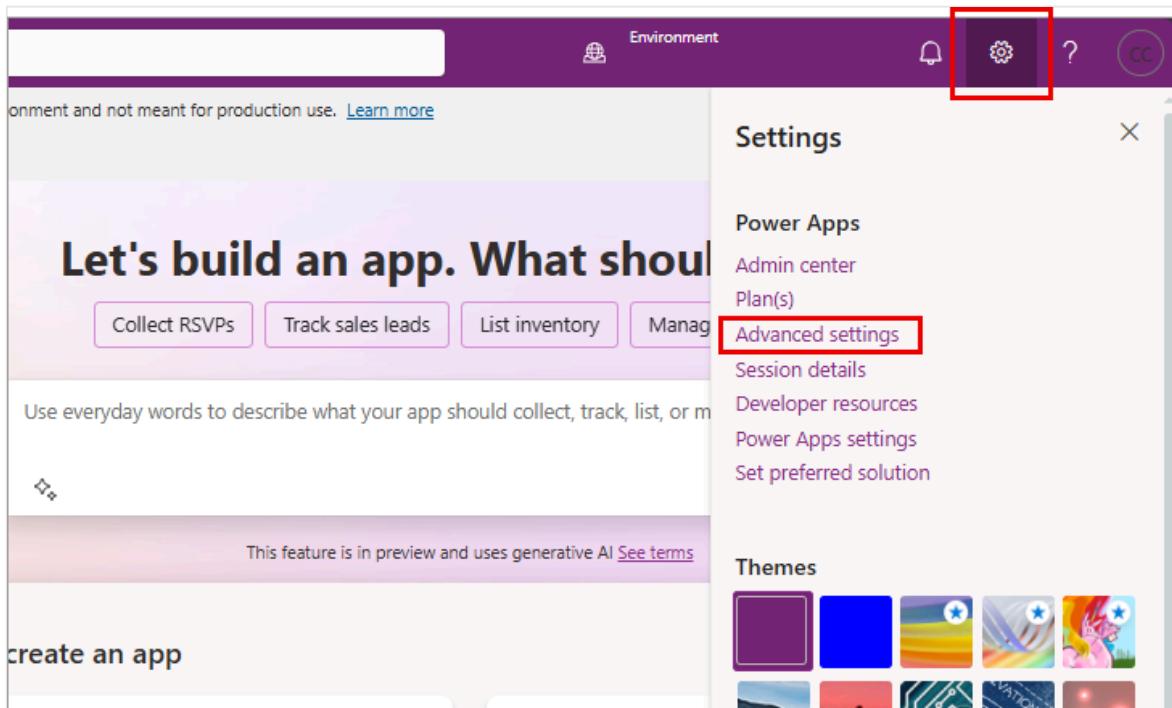
| Pros                                                                                                           | Cons                            |
|----------------------------------------------------------------------------------------------------------------|---------------------------------|
| Refresh time is constant.<br>Moderately simple implementation.<br>Low storage cost in Azure Data Lake Storage. | Compute cost for Azure Synapse. |

## 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.

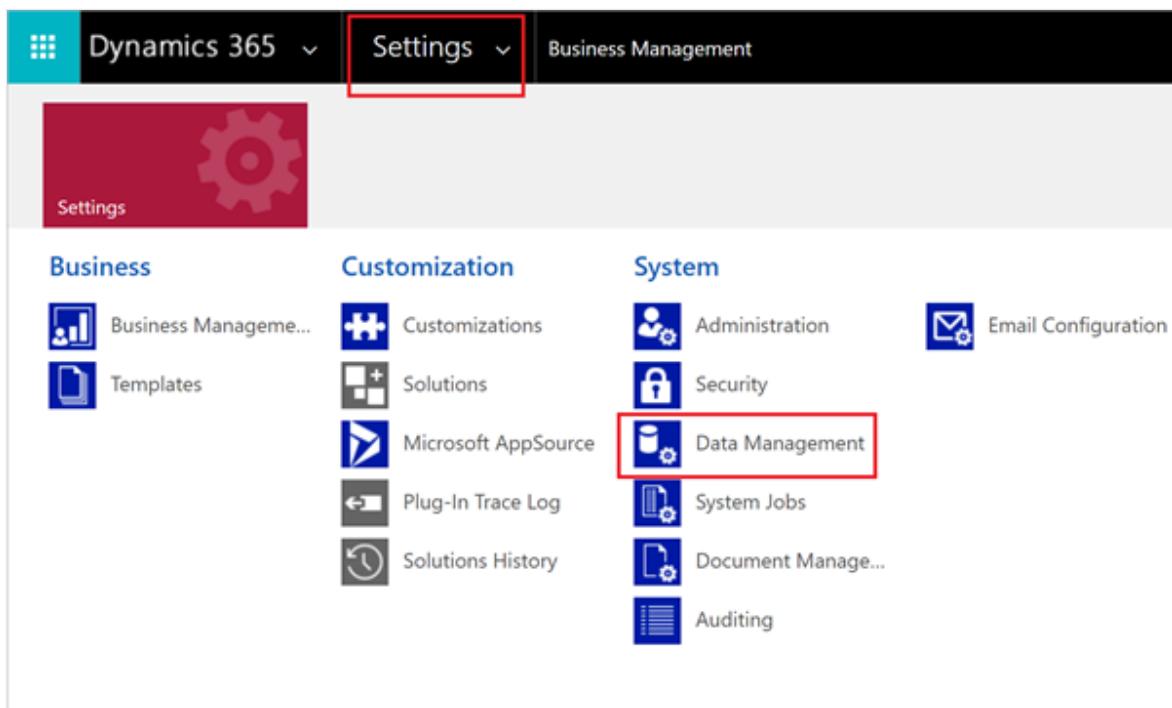
## Cancel the existing bulk delete job

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 Dynamics 365 interface with the 'Data Management' tab selected. In the 'Bulk Record Deletion' section, the 'View' dropdown is set to 'All Bulk Deletion System Jobs'. A red box highlights the 'Recurring Bulk Deletion System Jobs' row in the list, which contains four entries, all of which have a status of 'Succeeded'.

|                                     | Status Reason | Deleted | Failures | Owner (System Job) |
|-------------------------------------|---------------|---------|----------|--------------------|
| s Old...                            | Succeeded     | 0       | 0        | Admin ALMTest      |
| s Old...                            | Succeeded     | 0       | 0        | Admin ALMTest      |
| s Old...                            | Succeeded     | 0       | 0        | Admin ALMTest      |
| Recurring Bulk Deletion System Jobs | Succeeded     | 0       | 0        | Admin ALMTest      |

5. Select the Bulk Delete Conversation Transcript Records Older Than 1 Month job.

The screenshot shows the Dynamics 365 interface with the 'Data Management' tab selected. In the 'Bulk Record Deletion' section, the 'View' dropdown is set to 'Recurring Bulk Deletion System Jobs'. A red box highlights the first row in the list, which is checked and labeled 'Bulk Delete Conversation Transcript Records Older Than 1 Month'.

|                                                                                                    | Status Reaso... | Owner (System Jo... | Next Run           |
|----------------------------------------------------------------------------------------------------|-----------------|---------------------|--------------------|
| <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 Dynamics 365 interface with the 'Data Management' tab selected. In the 'Bulk Record Deletion' section, the 'View' dropdown is set to 'Recurring Bulk Deletion System Jobs'. A red box highlights the 'More Actions' menu for the selected job, which includes options like 'Set View as Default View', 'Modify Recurrence', and 'Cancel'. The 'Cancel' option is also highlighted with a red box.

|                                                    | Status Reaso... | Owner (System Jo... | Next Run           |
|----------------------------------------------------|-----------------|---------------------|--------------------|
| <input checked="" type="checkbox"/> Bulk Delete Co | Waiting         | Admin ALMTest       | 1/17/2020 7:15 PM  |
| Delete comple...                                   | Waiting         | SYSTEM              | 1/17/2020 11:18 AM |
| Delete comple...                                   | 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".

**Define Search Criteria**

Select search criteria to identify records to delete.

**Look for:** ConversationTranscripts    **Use Saved View:** [new]

Clear |  Group AND |  Group OR

|                         |                     |                    |
|-------------------------|---------------------|--------------------|
| ▼ ConversationStartTime | Older Than X Months | 12                 |
| ▼ SchemaType            | Equals              | powervirtualagents |

[Select](#)

[Preview Records](#)

[Back](#) [Next](#) [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**

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@ccbotstest1outlook.onmicrosoft.com) when this job is finished.  
Also notify:

[Back](#) [Next](#) [Cancel](#)

5. Select **Next**, and then select **Submit**.

# 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.

In many places, the conversation transcripts refer to content by its ID. For example, the ID of the topic that's being redirected to by the current topic is only referred to by its ID. To get the name of the topic, look up its name in the agent content.

---

---

## Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

# Understand error codes

Article • 11/19/2024

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 are 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

① Note

The term *dialog* used in some error messages refers to a *topic*.

[ ] Expand table

| Error code                                     | Description                                                                    |
|------------------------------------------------|--------------------------------------------------------------------------------|
| <a href="#">ContentError</a>                   | There's an error in the topic content.                                         |
| <a href="#">DataLossPreventionViolation</a>    | There was a data loss prevention violation.                                    |
| <a href="#">FlowActionException</a>            | An error occurred while executing a <a href="#">cloud flow</a> .               |
| <a href="#">FlowActionBadRequest</a>           | A request made to a <a href="#">cloud flow</a> was malformed.                  |
| <a href="#">FlowActionTimedOut</a>             | A <a href="#">cloud flow</a> took more than 100 seconds to run and timed out.  |
| <a href="#">InvalidContent</a>                 | Invalid content was added to the code editor.                                  |
| <a href="#">InfiniteLoopInBotContent</a>       | A node was executed too many times.                                            |
| <a href="#">LatestPublishedVersionNotFound</a> | Unable to retrieve the published version of the agent.                         |
| <a href="#">RedirectToDisabledDialog</a>       | A topic is <a href="#">redirecting</a> to a disabled topic.                    |
| <a href="#">RedirectToNonExistentDialog</a>    | A topic is <a href="#">redirecting</a> to another topic that no longer exists. |

| Error code  | Description                                |
|-------------|--------------------------------------------|
| SystemError | A system error occurred in Copilot Studio. |

## ContentError

**Error message:** This error produces dynamic messages based on the context of the error.

**Resolution:** This is a catch-all error for problems related to your agent's content. Refer to the error message for 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 to require users to sign in."

**Resolution:**

- Your environment's data loss prevention (DLP) 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 are blocked by the tenant administrator.

## 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 is not 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.

## 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](#).

## 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](#).

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Generative answers pointing to SharePoint sources don't return results

Article • 12/17/2024

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](#).

## Search results are missing

Generative answers for a SharePoint data source rely on making calls to the Microsoft Graph API search endpoint. Only the top three results coming back from Microsoft Graph are used to summarize and generate a response. If no results come back, the generative answers node doesn't provide a response.

To diagnose why Copilot Studio isn't returning results from Microsoft Graph, you can make direct calls to the search endpoint. This call simulates the way Copilot Studio works behind the scenes. Calls to the search endpoint can be generated by using the following template with [Graph Explorer](#). When accessing Graph Explorer, be sure to sign-in using the appropriate credentials for the SharePoint tenant.

The template can be used either by copying the following payload, or using this [deep link](#), which opens Graph Explorer with a prepopulated query.

POST <https://graph.microsoft.com/v1.0/search/query> ↗

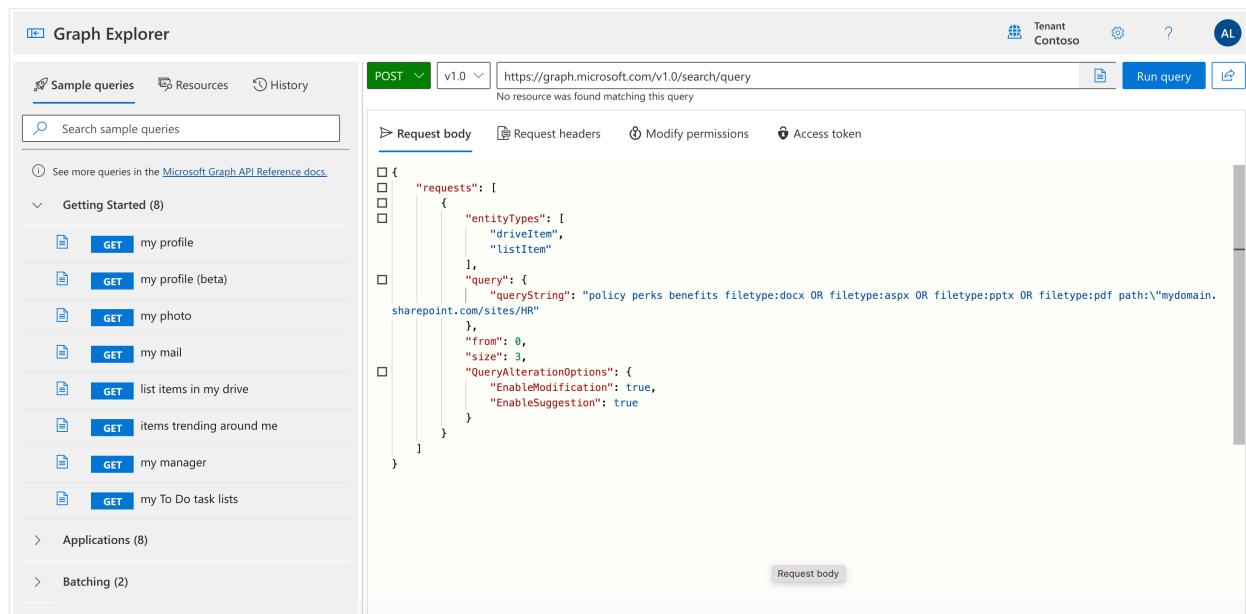
JSON

```
{
 "requests": [
 {
 "entityTypes": ["driveItem", "listItem"],
 "query": {
 "queryString": "SEARCH TERMS filetype:docx OR filetype:aspx
OR filetype:pptx OR filetype:pdf
path:\\"https://DOMAIN.sharepoint.com/sites/SITENAME\\"
 },
 "from": 0,
 "size": 3,
 "QueryAlterationOptions": {
 "EnableModification": true,
 "EnableSuggestion": true
 }
 }
]
}
```

## Missing results

Let's assume that generative answers are configured to provide responses based on content stored in <https://<user-domain>.sharepoint.com/sites/HR>. However, users aren't getting responses when asking, "What is our policy regarding perks & benefits?"

Behind the scenes, users' queries are being rewritten, so only the main keywords are being sent to Microsoft Graph, resulting in a query similar to the following example:



The screenshot shows the Microsoft Graph Explorer interface. The top navigation bar includes 'Graph Explorer', 'Tenant Contoso', and a 'Run query' button. The request URL is set to <https://graph.microsoft.com/v1.0/search/query>. The 'Request body' pane displays the JSON query:

```
{
 "requests": [
 {
 "entityTypes": ["driveItem", "listItem"],
 "query": {
 "queryString": "policy perks benefits filetype:docx OR filetype:aspx OR filetype:pptx OR filetype:pdf path:\\"mydomain.sharepoint.com/sites/HR\\"
 },
 "from": 0,
 "size": 3,
 "QueryAlterationOptions": {
 "EnableModification": true,
 "EnableSuggestion": true
 }
 }
]
}
```

The left sidebar contains a 'Sample queries' section with various pre-defined queries like 'my profile', 'my photo', etc., and a 'Getting Started' section with 8 items.

If no results are returned to the search endpoint, as shown in the following response, generative answers doesn't provide a response, either.

The screenshot shows the Microsoft Graph Explorer interface. In the top navigation bar, it says "Tenant Contoso". The main area has a "POST" dropdown set to "v1.0", a URL field with "https://graph.microsoft.com/v1.0/search/query", and a "Run query" button. Below this, there are tabs for "Request body", "Request headers", "Modify permissions", and "Access token". A green status bar at the top indicates "OK - 200 - 420 ms". The "Response preview" tab is selected, showing a JSON response structure. The response includes "value" with search terms like "policy", "perks", and "benefits", and a "hitsContainers" array with one item having a total of 0 results and "moreResultsAvailable" set to false. The "@odata.context" field points to the Microsoft Graph API metadata.

## 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 making calls to Microsoft Graph. At a minimum, a user must have read permissions on the relevant sites and files, or the call to Microsoft Graph doesn't return any results.

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. <a href="#">Learn more about permissions and consent</a> |           |                                     |             |                                    |        |
| API / Permissions name                                                                                                                                                                                                                                                           |           | Type                                | Description | Admin consent requ...              | Status |
| Files.Read.All                                                                                                                                                                                                                                                                   | Delegated | Read all files that user can access | No          | <span>✓ Granted for Contoso</span> | ...    |
| openid                                                                                                                                                                                                                                                                           | Delegated | Sign users in                       | No          | <span>✓ Granted for Contoso</span> | ...    |
| profile                                                                                                                                                                                                                                                                          | Delegated | View users' basic profile           | No          | <span>✓ Granted for Contoso</span> | ...    |
| Sites.Read.All                                                                                                                                                                                                                                                                   | Delegated | Read items in all site collections  | No          | <span>✓ Granted for Contoso</span> | ...    |
| User.Read                                                                                                                                                                                                                                                                        | Delegated | Sign in and read user profile       | No          | <span>✓ Granted for Contoso</span> | ...    |

The following example shows the required authentication settings in Copilot Studio:

## Authentication

X

Enter the information provided by your Identity Provider (IdP), and then test the connection. For single sign-on with AAD include the token exchange URL. [Learn more](#)

Require users to sign in

### Redirect URL

https://token.botframework.com/.auth/web/redirect

[Copy](#)

### Service provider \*

Azure Active Directory v2

▼

### Client ID \*

[REDACTED]

### Client secret \*

[REDACTED]

### Token exchange URL (required for SSO) [Learn more about SSO](#)

[REDACTED] Read

### Tenant ID

[REDACTED]

### Scopes ⓘ

profile openid Files.Read.All Sites.Read.All

[Save](#)

[Close](#)

## File size support

Currently, for SharePoint sources, generative answers can only process files up to 7 MB in size. 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, [files](#) up to 512 MB in size can be uploaded.

## 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's harmful, malicious, noncompliant, or in breach of copyrights. When content gets moderated, generative answers don'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 Azure 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, and Troubleshooting guides (preview). The Monitoring section is expanded, showing Logs (selected), Workbooks, Usage, Users, Sessions, and Events. In the main area, a query named "pva" is running. The query itself is:

```
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
14
```

The results table has columns: id, topic, message, and result. One row is visible with the message "Conversational boosting". A tooltip indicates it's "Filtered by High Content Mod...". At the bottom, it says "0s 874ms | Display time (UTC+00:00)".

## 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](#).

---

## Feedback

Was this page helpful?

[Yes](#)

[No](#)

[Provide product feedback ↗](#)

# Debug insights when generative answers don't return a response

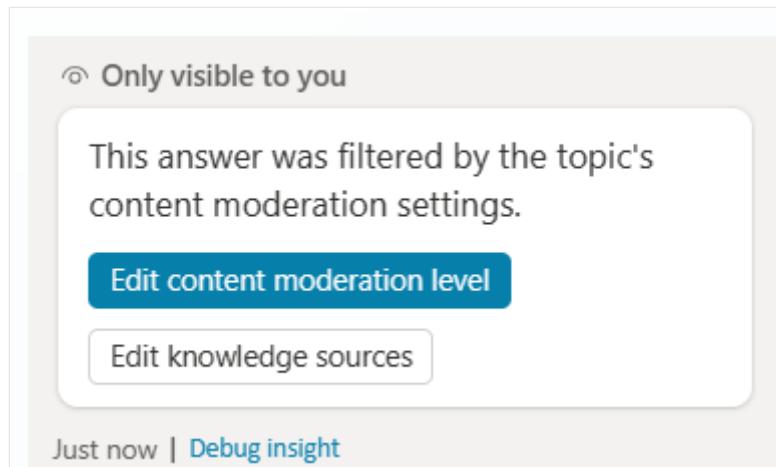
Article • 11/19/2024

Generative answers allow makers to create agents that respond to questions grounded in knowledge sources, like public websites or SharePoint. However, sometimes the agent doesn't provide a response. When an agent doesn't return a result, the test chat's debug mode provides insights on why a response wasn't returned.

## Debug insights in the test chat

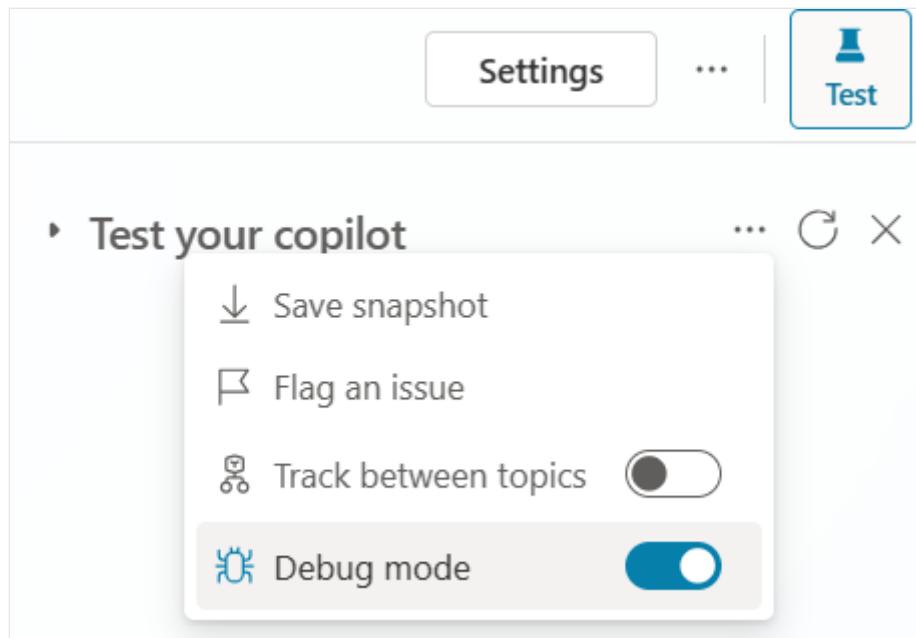
When debug mode is enabled, insights are displayed in the test web chat when generative answers query is made and no results are returned. Debug insights only show in the test chat and aren't shown to users in a published agent.

When there's no response, a debug insight appears in the test chat and provides an indication why no response was provided. If applicable, a suggested action to improve the likelihood that a response is returned is given.



## Disable debug insights

Debug insights are enabled by default. You can temporarily disable the display of debug insights by toggling off the debug mode in the test chat menu, located under the ellipses at the top of the test chat.



## Related content

- Generative answers pointing to SharePoint sources don't return results
- 

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

# Connector request failure

Article • 11/19/2024

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.
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# Troubleshoot voice-enabled agents and diagnose runtime error codes

Article • 11/19/2024

This article covers troubleshooting guidance for possible issues with your voice-enabled agent, along with runtime error codes you might encounter.

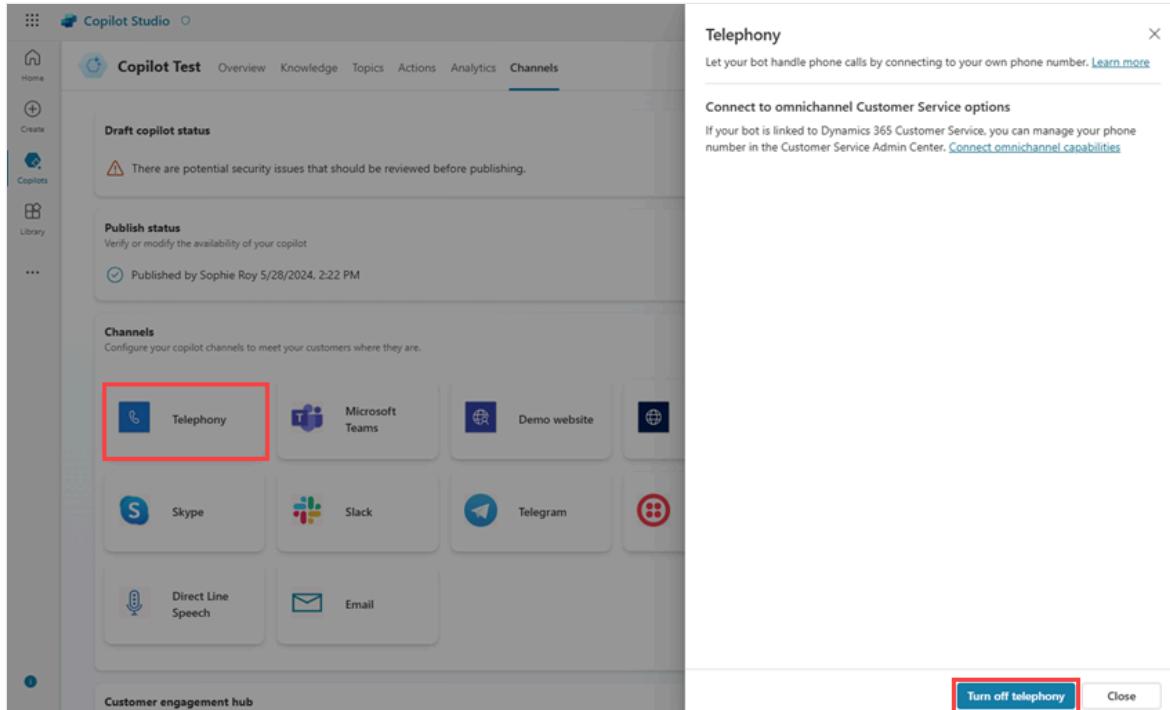
## Callers can't engage with a published agent

**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:**

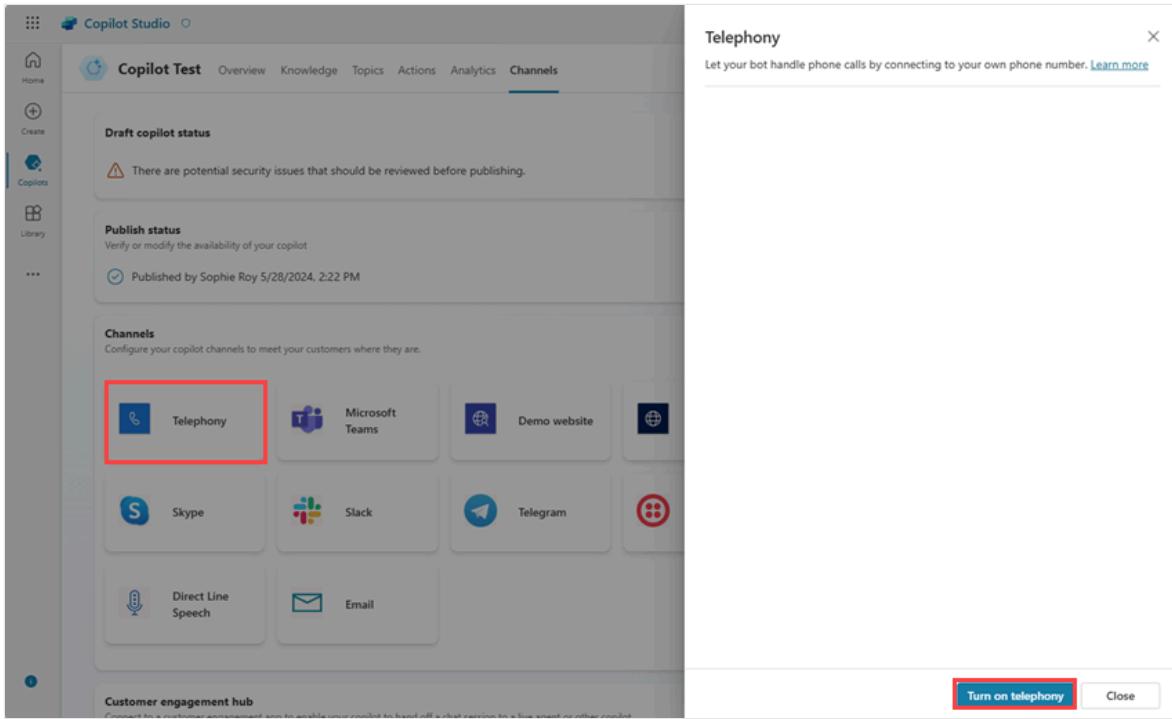
1. In Copilot Studio, go to **Channels > Telephony** and select **Turn off telephony**.



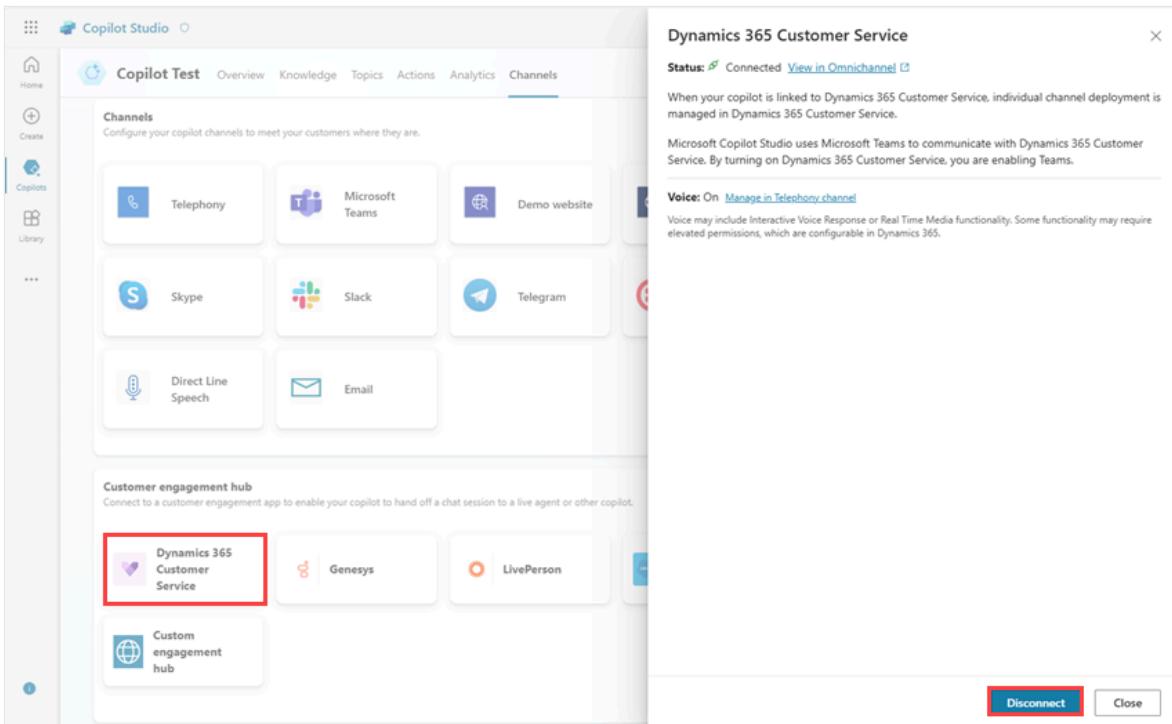
### ⓘ Note

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.



## 3. Go to Channels > Customer engagement hub > Dynamics 365 Customer Service and select Disconnect.



## 4. Select Connect.

**Dynamics 365 Customer Service**

Status: Not connected

When your copilot is linked to Dynamics 365 Customer Service, individual channel deployment is managed in Dynamics 365 Customer Service.

Microsoft Copilot Studio uses Microsoft Teams to communicate with Dynamics 365 Customer Service. By turning on Dynamics 365 Customer Service, you are enabling Teams.

Voice: On [Manage in Telephony channel](#)

Voice may include Interactive Voice Response or Real Time Media functionality. Some functionality may require elevated permissions, which are configurable in Dynamics 365.

**Customer engagement hub**

Connect to a customer engagement app to enable your copilot to hand off a chat session to a live agent or other copilot.

**Dynamics 365 Customer Service** (highlighted with a red box)

Genesys

LivePerson

Custom engagement hub

**Connect** **Close**

## Can't publish an agent or configure the Telephony channel

**Issue:** Maker can't publish an agent or configure the Telephony channel.

Your copilot may not work as intended right now due to your organization's data loss prevention policies. Address affected topics or contact your admin ([power.platform.admin](#))

Some channels aren't available due to your organization's data loss prevention policies. Contact your admin with questions ([power.platform.admins@sandvik.com](#)). [Learn more](#)

Your channels are currently being configured through Dynamics 365 Omnichannel for Customer Service. To manage, [go to Agent transfers](#) or configure in Omnichannel.

### Channels

Configure your copilot channels to meet your customers where they are. [Learn more about channels](#)

Looking for Microsoft Copilot extensions? Personalize your Microsoft Copilot experience. [Learn more](#)

|                                                                                                 |                                                                                  |                                                                   |                        |
|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------|------------------------|
| Microsoft Copilot (preview)<br>Publish plugins to a Microsoft Copilot for a unified experience. | <b>Telephony</b><br>Let your users converse with the copilot via a phone number. | Microsoft Teams<br>Chat with your copilot through a Teams app.    | Den Try o invite the s |
| Skype<br>Expand your copilot's reach to customers on Skype.                                     | Slack<br>Expand your copilot's reach to customers on Slack.                      | Telegram<br>Expand your copilot's reach to customers on Telegram. | Twilio Exp to cu       |
| Email<br>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 loss prevention (DLP) policies in your tenant.

**Steps:** In the Power Platform admin center, the tenant admin can unblock the needed DLPs. See [Data loss prevention 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](#)

---

## Feedback

Was this page helpful?



[Provide product feedback ↗](#)

# Language support

Article • 11/19/2024

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.

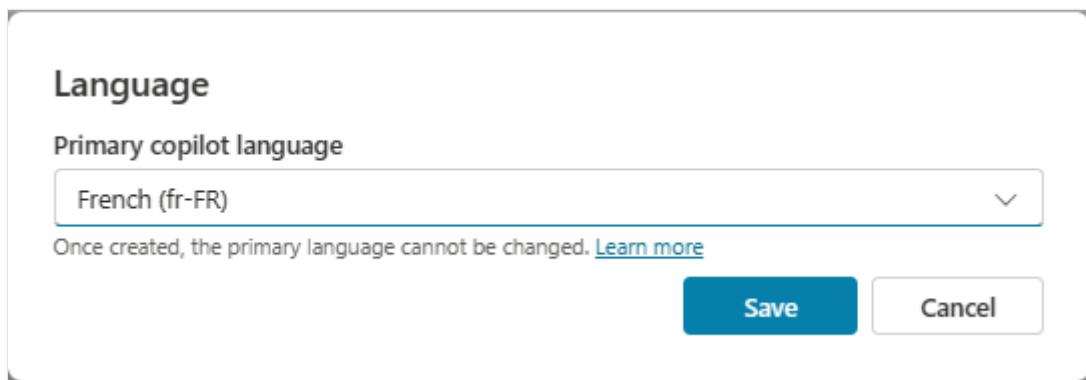
## Specify the primary language for your agent

Specify the target language when you create your agent.

i **Important**

Once you set your agent's primary language, it can't be changed. You can only add secondary languages. For more information on how to configure multilingual agents, see [Configure and create multilingual agents](#).

1. From the **Home** page, select **Create > New agent**.
2. Select **Edit language**, then select a language from **Primary agent language**, and then **Save**.



3. When you're finished configuring your agent, select **Create** at the top of the page.

## Default content in the target language

After your new agent is created, you notice it already has content in the target language, including system topics and template user topics on the **Topics** page.

| Name                             | Trigger | Description                      |
|----------------------------------|---------|----------------------------------|
| Au revoir                        | Phrases | Cette rubrique se déclenche q... |
| Leçon 1 – Rubrique simple        | Phrases |                                  |
| Leçon 2 – Rubrique simple av...  | Phrases |                                  |
| Leçon 3 – Rubrique avec une c... | Phrases |                                  |
| Merci                            | Phrases | Cette rubrique se déclenche l... |
| Recommencer                      | Phrases |                                  |
| Salutations                      | Phrases | Cette rubrique est déclenchée... |

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 chat pane. In chat, you see how well the agent understands user input and how it responds in that language.

## List of supported languages

Copilot Studio supports languages in three stages. The stage indicates how well the agent can understand user input in that language, with general availability being the strongest, followed by preview, and then experimental.

### Note

Support levels include your Microsoft customer service support. For example, with features in preview, you can't yet contact Microsoft for technical support with Copilot Studio. In general availability, you can contact Microsoft for Copilot Studio support.

- General availability: ✓ Language has full support.
- Authoring canvas: The agent maker sees the Copilot Studio UI in the target language.
- User language: The language a user can type in chat to ask questions. Exceptions include agents with generative AI features that aren't yet available in some languages.
- Generative answers: A Copilot Studio feature that autogenerates chat responses.

- Generative orchestration: A Copilot Studio feature that selects the most appropriate combination of topics, actions, and knowledge sources at each turn during a conversation.
- Voice support: A Copilot Studio feature that supports [interactive voice response agents](#).

 **Note**

Generative orchestration is currently only available in English.

 [Expand table](#)

| Language                 | Language code | Authoring canvas | User language | Generative answers | Generative orchestration | Voice support |
|--------------------------|---------------|------------------|---------------|--------------------|--------------------------|---------------|
| Chinese (Simplified)     | zh-CN         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Czech                    | cs-CZ         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Danish                   | da-DK         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Dutch                    | nl-NL         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| English (Australia)      | en-AU         | Not supported    | ✓             | ✓                  | Not supported            | ✓             |
| English (United Kingdom) | en-GB         | Not supported    | ✓             | ✓                  | Not supported            | ✓             |
| English (United States)  | en-US         | ✓                | ✓             | ✓                  | Preview                  | ✓             |
| Finnish                  | fi-FL         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| French (Canada)          | fr-CA         | Not supported    | ✓             | ✓                  | Not supported            | ✓             |
| French (France)          | fr-FR         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| German                   | de-DE         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Greek                    | el-GR         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Hindi                    | hi-IN         | ✓                | ✓             | ✓                  | Not supported            | ✓             |

| Language                | Language code | Authoring canvas | User language | Generative answers | Generative orchestration | Voice support |
|-------------------------|---------------|------------------|---------------|--------------------|--------------------------|---------------|
| Indonesian              | id-ID         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Italian                 | it-IT         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Japanese                | ja-JP         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Korean                  | ko-KR         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Norwegian               | nb-NO         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Polish                  | pl-PL         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Portuguese (Brazil)     | pt-BR         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Russian                 | ru-RU         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Spanish (Spain)         | es-ES         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Spanish (United States) | es-US         | Not supported    | ✓             | ✓                  | Not supported            | ✓             |
| Swedish                 | sv-SE         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Thai                    | th-TH         | ✓                | ✓             | ✓                  | Not supported            | ✓             |
| Turkish                 | tr-TR         | ✓                | ✓             | ✓                  | Not supported            | ✓             |

## Related content

- [Regional settings and data locations for organizations using Copilot Studio](#)
- [Configure and create multilingual agents](#)

---

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

# 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, limits, app registration, certificates, and configuration values for Copilot Studio

Article • 05/21/2024

## ⓘ Important

Power Virtual Agents capabilities and features are [now part of Microsoft Copilot Studio](#) following significant investments in generative AI and enhanced integrations across Microsoft Copilot.

Some articles and screenshots may refer to Power Virtual Agents while we update documentation and training content.

This article contains quotas, limits, app registration usage, certificate rotation, and configuration values for Copilot Studio.

## Quotas

Quotas are default constraints applied to copilots that limit how often messages can be sent to the copilot. 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.

Quotas are applied to your copilots alongside the capacity constraints you gained by purchasing a [Microsoft Copilot Studio plan](#). Also see [Licensing and capacity considerations](#). These quotas are configurable and can be tweaked for your custom copilots on a case-by-case basis. Please open a support ticket with business justification to adjust your copilot's quota.

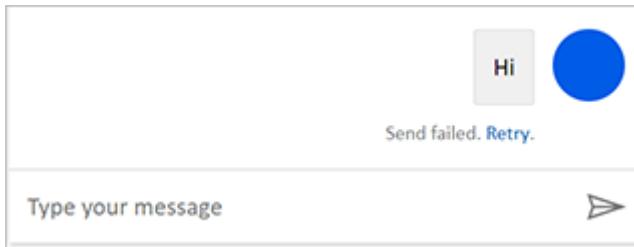
The following quotas, defined as requests per minute (RPM), apply to Copilot Studio copilots' Dataverse environment.

ⓘ [Expand table](#)

| Action                                                                                     | Quota with the paid plan          |
|--------------------------------------------------------------------------------------------|-----------------------------------|
| Messages to a copilot<br>(Includes any message from a user or integrations, such as Azure) | 800 RPM per Dataverse environment |

| Action                                      | Quota with the paid plan |
|---------------------------------------------|--------------------------|
| Bot Framework skills, to a single copilot.) |                          |

If the quota for **messages to a copilot** is met, the user chatting with the copilot sees a failure notice when they try to send a message. They're prompted to retry.



## Limits

These limits apply to the Copilot Studio web app and the Copilot Studio app in Microsoft Teams. These limits are configurable and can be adjusted for your copilots on a case-by-case basis. Please open a support ticket with business justification to adjust your copilot's limits.

### 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 copilot 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, and clear them before making the transfer to avoid going over the limit.

### Copilot Studio web and Teams apps limits

[ ] [Expand table](#)

| Feature  | Web app | Teams app   |
|----------|---------|-------------|
| Copilots | N/A     | 50 per team |

| Feature         | Web app                                    | Teams app                                                                                                                      |
|-----------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| Topics          | 1000 per copilot in Dataverse environments | 250 per copilot in Dataverse for Teams environments <sup>1</sup><br>1000 per copilot in Dataverse environments (after upgrade) |
| Trigger phrases | 200 per topic                              | 200 per topic                                                                                                                  |
| Skills          | 100 per copilot                            | 100 per copilot <sup>2</sup>                                                                                                   |

<sup>1</sup> If you're approaching limits in Dataverse for Teams environments, you can [upgrade Dataverse for Teams](#) to continue adding topics.

<sup>2</sup> You must have a [Copilot Studio standalone subscription](#) to use skills in a Teams app created copilot. Skills aren't available in the [Teams plan](#).

## 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 copilots 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 copilots. 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](#).

## 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 copilot 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                                                                                                                  |

## 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 a copilot, allowing it to communicate with different channels and services. App registration is a mandatory component of Copilot Studio copilots, as it enables the copilot to identify itself to Omnichannel for Customer Service, in Microsoft Teams, and to authenticate with other channels. App registration also allows the copilot 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 copilot. 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 copilot has the necessary credentials and permissions to interact with Omnichannel, selected channels, and skills. The customer can focus on designing and publishing the copilot, 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 copilot 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 copilot. 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 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. 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 copilots 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, 20.49.123.64-20.49.123.127, 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, |

| Region | Outbound IP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | 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, 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, |

| Region                     | Outbound IP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            | 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, 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-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Region | Outbound IP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | 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, 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, |

| <b>Region</b>  | <b>Outbound IP</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 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,                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

| Region       | Outbound IP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 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, 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, |

| Region | Outbound IP                                                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------|
|        | 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.

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# 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

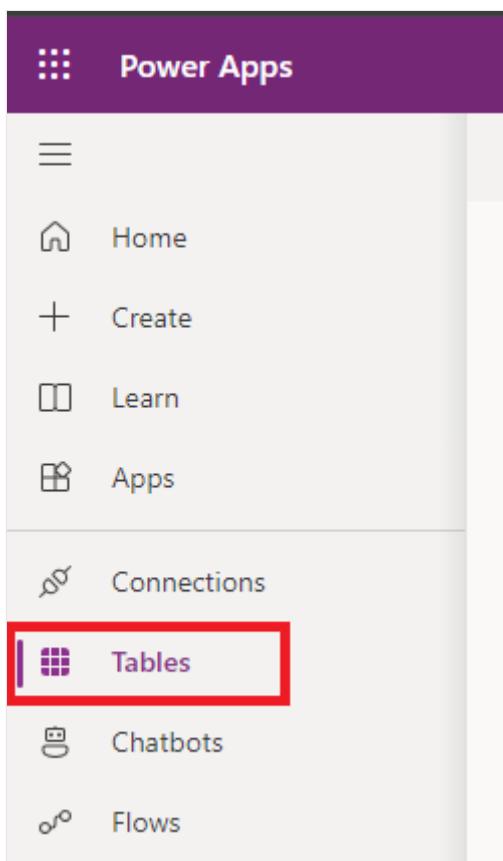
Article • 11/19/2024

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 for the 'Chatbot' row, 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 the 'Tables' section of the Power Apps portal. A success message at the top states: 'Chatbot was exported successfully. Click the link below to download the exported data.' Below the message, there 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 ↗](#)

# 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

Article • 11/19/2024

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 agent creation with generative AI\]](#)

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

# FAQ for generative answers

Article • 11/19/2024

These frequently asked questions (FAQ) describe the AI effect 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 contain 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 images embedded in PDF files. The agent uses generative AI to summarize that information into a response returned to the agent user.

### Note

As of September 2024, agents can also reason over non-text elements in uploaded files, such as images, tabular data, and diagrams.

## What are generative answers intended use?

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, and how can users minimize the impact of limitations when using 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:

- The corpus upon which the model was trained doesn't include data created after 2021.  
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?

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. More information on what data is collected is available in the [preview terms](#).

## 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](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# FAQ for Copilot

Article • 11/19/2024

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

Article • 11/19/2024

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 and/or actions. Generative orchestration enables more natural conversations by filling in inputs, using details from the conversation history. For example, if you ask about the nearest store in Kirkland, and then ask for the weather there, orchestration infers you want to ask for the weather *in Kirkland*. The system can also chain together multiple actions or topics. For example, it can answer "I need to get store hours and find my nearest store." When the agent is unsure about details, it can ask follow-up questions to disambiguate.

## What can generative orchestration do?

With generative orchestration, the system first creates a plan to answer the user query by using the name, description, inputs, and outputs of the topics and actions available. It also references the last 10 turns of conversation history. It then tries to execute the plan by filling in required inputs from the conversation, following up with the user for any missing or ambiguous details. The system checks that it found an answer to the user's question before replying to the user. If not, it goes through the process again. Finally, the system generates a response based on the output of the plan from the topics and/or actions. It also uses any custom instructions for the agent when generating the final response.

## What are the intended uses of generative orchestration?

You can use this mode within your agent to create an agent that can answer user queries based on the conversation history, names and descriptions for topics, and names, descriptions, inputs, and outputs for actions.

## **How is generative orchestration evaluated?**

## **What metrics are used to measure performance?**

Generative orchestration is evaluated for end-to-end quality at each step of the process. Quality is measured in terms of how well the system creates and executes a plan that successfully addresses the user query. Our team manually labels quality scores during fine-tuning. We evaluate quality over various user queries, prompts, and actions. We also evaluate how well the system does at ignoring malicious content from users and authors, and how well the system avoids producing harmful content.

## **What are the limitations of generative orchestration? How can users minimize the impact of generative orchestration limitations when using the system?**

For best results, make sure your topics and actions include high-quality descriptions. We provide guidance on how to write high quality descriptions in the Copilot Studio documentation.

## **What operational factors and settings allow for effective and responsible use of generative orchestration?**

Generative orchestration is currently **English only**. Once you enable generative mode within your agent, you can test the system to see how well it performs using the test panel. You can also add custom instructions for your agent to help generate the final response.

## **What are actions and how does your agent, with generative mode enabled, use them?**

You can add actions to your custom agent to answer user queries. You can use actions developed by Microsoft or third parties, or you can create your own actions. You configure which actions to configure for the custom agent to use. You can also edit the name, description, inputs, and outputs used by the system.

# **What data can Copilot Studio provide to actions? What permissions do Copilot Studio actions have?**

When your agent calls an action, the action receives the input values specified by the action. The input values can include some of the conversation history with the user.

## **What kinds of issues might arise when using Copilot Studio enabled with actions?**

Actions might not always work as intended. Errors might occur when preparing the input for the action or when generating a response based on the action's output. Your agent might also call the wrong action for the user query. To mitigate the risk of such errors when using actions, make sure you have high quality, relevant, and unambiguous descriptions configured for the actions in your custom agent.

## **What protections does Copilot Studio have in place for responsible AI?**

There are many mitigation features in place to protect your agents. You can configure your agent with a set of knowledge, actions, and topics. Agents never take an action that isn't part of their configuration. Admins can disallow actions for agents in your organization. If you're concerned about an action being triggered without confirmation, you can configure an action to only be called when a user agrees to call it.

In addition, we have classifiers that look at input to the system to detect harmful content and jailbreak attacks. According to our tests, these classifiers have a high success rate at blocking harmful content and jailbreak attacks, while also having high success at not blocking content that isn't harmful or a jailbreak attack. However, classifiers can't be perfect so there are risks of an agent producing harmful content or responding to a jailbreak attack. These risks include cross-domain prompt injection attacks, where instructions could be added to the output of an action or a knowledge source that the agent then tries to follow.

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

- [Orchestrate agent behavior with generative AI \(preview\)](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

# FAQ for the conversational first run experience

Article • 11/19/2024

These frequently asked questions (FAQ) describe the AI effect of the the conversational first run experience in Copilot Studio.

## What is the conversational first run experience?

The [conversational first run experience](#) is used to help you create a custom agent or Copilot agent. Through the conversation, it predicts the name, description, instructions, any added web knowledge sources, and starting prompts (for Copilot agents)

## What are the capabilities for the conversational first run experience?

You can get started quickly with custom agent or Copilot agent configuration through a natural language interface. The system will update the configuration based on the maker's inputs during the conversation.

## What is the conversational first run experience's intended use?

You can use this experience to begin your initial agent configuration.

## How was the conversational first run 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 does not produce harmful or malicious content.

# What are the limitations of the conversational first run experience, and how can users minimize the impact of limitations when using it?

- The conversational first run experience supports English, and isn't designed to support other languages.

See [Language support](#) for the latest list of supported languages for this and other generative AI features.

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 experience can't help you test your agent, and it can only be used to configure:
  - The name
  - The description
  - Instructions that create the agent
  - Web knowledge sources
  - Starting prompts for Copilot agents
- See the [Responsible AI FAQ for generative answers](#) for additional 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 first run experience?

You can use natural language to converse with the system over chat, or you can directly edit the configuration manually in the side panel. If you edit manually, the system may update 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

- [Create and delete agents](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)