

Microsoft Copilot Studio documentation

Discover how to build AI-driven copilots 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 copilots with generative AI

Use generative AI to build copilots fast
Tell your copilot what it should do
Make your copilot useful from day one
Build a copilot with generative AI from the ground up

Enhance and improve your copilots

Use entities and slot filling
Use end-user authentication
Use Microsoft Bot Framework Skills

Best practices guidance

[Topics best practices](#)
[Trigger phrases best practices](#)
[Measuring copilot engagement](#)

Administer and extend

Administer Microsoft Copilot Studio

- Assign licenses
- Working with environments
- Export and import copilots using solutions

Publish and integrate

- Configure demo and live sites
- Publish to Microsoft Teams
- Publish to Facebook

Monitor and diagnose

- Use analytics to improve your copilot
- Analyze copilot performance and usage
- Analyze topic usage

Expand copilots with plugins

- Create and configure copilot connectors
- Call a connector action
- Call a plugin action

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

Microsoft Copilot Studio overview

Article • 06/04/2024

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.

Microsoft Copilot Studio is a graphical low-code tool for creating and maintaining copilots.

A copilot is an AI-powered conversational interface based on large language models (LLMs) and additional sources of knowledge. It's a powerful AI companion that can handle a range of requests, from providing simple responses to common questions to resolving issues requiring complex conversations.

Copilots 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.

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

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 copilots for customers. - Familiar with copilot services, you want to trial or test Copilot Studio. - A copilot user who wants to explore advanced copilot concepts, such as entities and variables, and create complex copilots. 	Explore the Copilot Studio demo
Teams app	<ul style="list-style-type: none"> - An employee or member of an organization who wants to use copilots to answer common employee questions. - You want to use advanced concepts, such as entities and variables, and have an internally-available copilot in Teams. - You want to create and distribute a copilot in the shortest time possible. 	Open or add the Microsoft Copilot Studio app in Teams

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 copilot 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 copilot (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 copilot. You are solely

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

Related content

- [AI-based copilot authoring overview.](#)
- [Create and delete copilots.](#)
- [Create and edit topics.](#)
- [Key concepts - Enhanced authoring in Copilot Studio.](#)
- [Key concepts - Publish your copilot.](#)
- [Key concepts – Analytics in Copilot Studio.](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Copilot Studio compliance offerings

Article • 04/12/2024

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 is a Core Online Service, as defined in the [Online Services Terms \(OST\)](#), and 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 table](#)

Standard	Name of report and certificate	Link to standard (www.iso.org)
ISO 9001:2015	Microsoft Azure, Dynamics 365, and Other Online Service - ISO9001 Certificate and Assessment Report	ISO 9001:2015
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

Standard	Name of report and certificate	Link to standard (www.iso.org)
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.](#)

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.

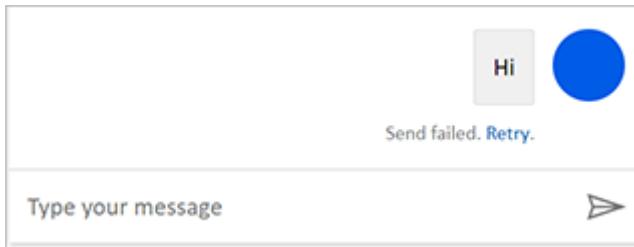
The following quotas, defined as requests per minute (RPM), apply to Copilot Studio copilots' Dataverse environment.

ⓘ [Expand table](#)

Action	Quota with the paid plan
Messages to a copilot (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 ¹ 1000 per copilot in Dataverse environments (after upgrade)
Trigger phrases	200 per topic	200 per topic
Skills	100 per copilot	100 per copilot ²

¹ If you're approaching limits in Dataverse for Teams environments, you can [upgrade Dataverse for Teams](#) to continue adding topics.

² 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. ¹
Power Platform requests ²	250,000 every 24 hours ³	6,000 every 24 hours

¹ Copilot Studio [sessions](#) that are included with [Microsoft 365 subscriptions](#) aren't pooled.

² These requests are consumed by Power Automate flows that are triggered by Copilot Studio copilots. Learn more about [Power Automate limits](#).

³ 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 ¹	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,

Region	Outbound IP
	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

¹ 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 ↗](#)

Considerations for the latest Microsoft Copilot Studio unified authoring release

Article • 04/26/2024

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.

The latest Microsoft 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.

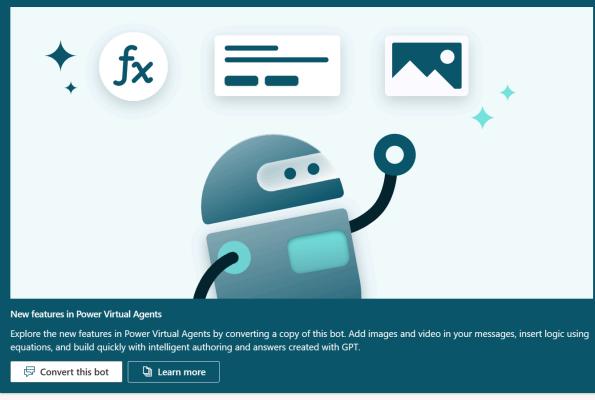
With the release of our latest version of Copilot Studio, now is the time to consider migrating from your classic Power Virtual Agents experience to the latest version of Copilot Studio. This guide aims to help readers understand some of the significant improvements within this release, and areas to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

To help the upgrade experience, at launch, there's a new clone feature that's now available on classic bots built using Power Virtual Agents.



New features in Power Virtual Agents

Explore the new features in Power Virtual Agents by converting a copy of this bot. Add images and video in your messages; insert logic using equations, and build quickly with intelligent authoring and answers created with GPT.

Convert this bot Learn more

Learn more

[Power Virtual Agents documentation](#)

Product videos

- [Create topics and test your bot](#)
- [Use slot filling and built-in entities to save authoring time](#)
- [Sign in directly from within your bot](#)
- [Publish your bot to a website](#)
- [Get topic suggestions from webpages](#)
- [Use analytics to improve the performance of your bot](#)

[View all videos](#)

This feature clones the selected Power Virtual Agents classic bot, and creates a new copilot 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 copilot experience.
- The cloning process only clones topics built using the Microsoft Copilot Studio web canvas (not topics built using Bot Composer).
- The cloning process clones entities and associated synonyms and custom entities.
- A copilot administrator needs to reconfigure authorization, channels, and security settings in the cloned unified authoring copilot.
- Any Bot Framework Skills from the Power Virtual Agents classic bot would need to be reconnected with the cloned unified authoring copilot.
- 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 copilot 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 copilot deployment should be updated. Examples include if the custom data pipeline solution is being used or alternative reporting solutions.

ⓘ Important

This may not be an exhaustive list based on your current implementation and architecture. Ensure you complete the required testing for your copilot before deploying to production, even if you're using the clone feature to migrate your copilot between versions.

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 ↗](#)

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 copilot and, as a result, an improved experience for your customers.

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, check out the [Getting Started Guide](#)

Creating a new copilot

When creating a brand new copilot by selecting **New copilot** under the **Copilots** item in the main navigation, there's a new copilot creation process where copilot makers can name the copilot, specify the copilot's language, and also get started with boosted conversations (public preview feature)

Additionally, you can configure **Advanced Information** including 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.

💡 Tip

For more information on Boosted Conversations, see the following subsections, as well as the feature documentation included in this article.

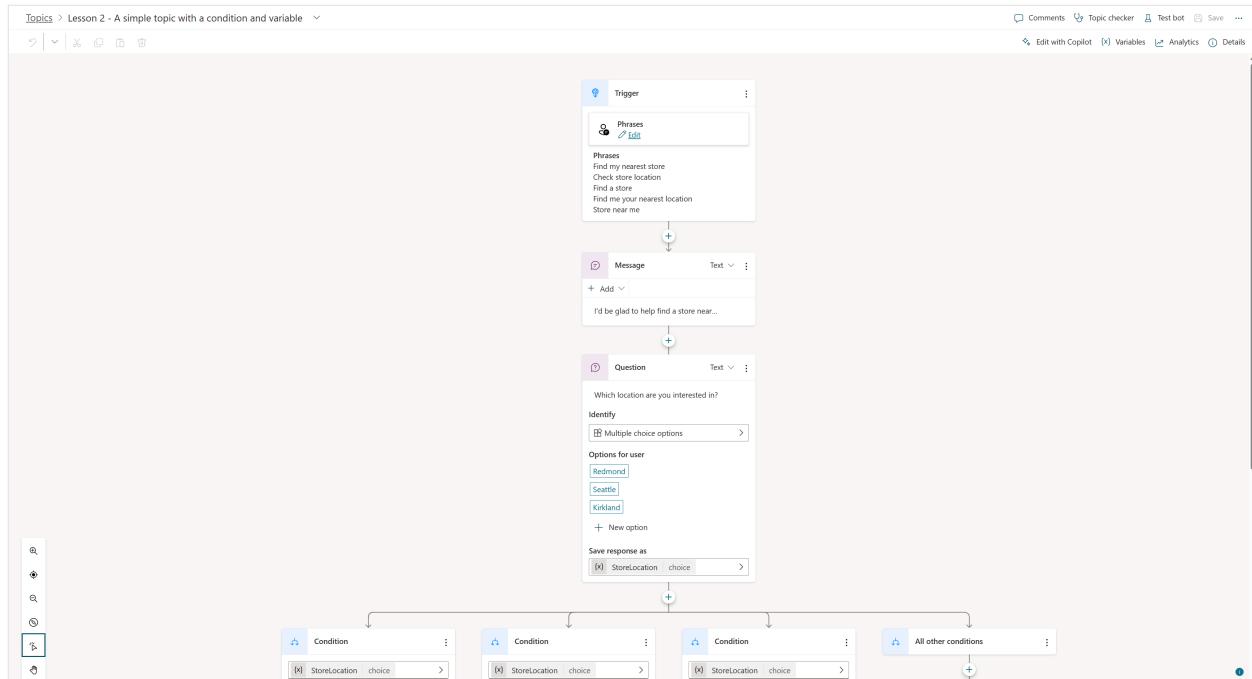
The screenshot shows the 'Create a bot' wizard in the Power Virtual Agents interface. It's Step 1 of 2. On the left, there's a sidebar with 'Configure your bot' options: 'Start fresh with a new bot, and start making it yours.', 'Want to convert a classic bot to the new authoring canvas? [Start here.](#)', and 'Want to enable voice capabilities for your bot? [Start here](#)'. The main area has fields for 'Bot name' (with a required asterisk and a placeholder 'Bot Name') and 'What language do you want your bot to speak?' (with a dropdown menu showing 'English'). A callout box for 'Boost your conversations with GPT (preview)' provides information about letting the bot create responses in real time with GPT and information from a website, with a link to 'learn more'. There's also a note about AI-generated content being reviewed.

User Interface Changes

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

- *Creating a New Topic*: 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 copilot 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:** Microsoft 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 copilot 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 copilot' panel within Copilot Studio when your copilot 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 copilot 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.

Topics ^①

Custom (7) System (8)

System topics are the core of your bot. Without them, your bot will not function or handle errors properly. You can't delete these topics.

Type	Name ↓	Trigger ↑	Status	Errors	Currently editing
Bot	Conversation Start	On Conversation Start	On		
Bot	End of Conversation	On Redirect	On		
Bot	Escalate	On Talk to Agent	On		
Bot	Fallback	On Unknown Intent	On		
Bot	Multiple Topics Matched	On Select Intent	On		
Bot	On Error	On Error	On		
Bot	Reset Conversation	On Redirect	On		
Bot	Sign in	On Sign In	On		

Pass entire Objects back to Microsoft Copilot Studio using Power Automate

Copilot Studio is seamlessly integrated with Power Automate to allow copilot makers to connect to many different connectors to help retrieve and access data in other systems. In the previous version of Microsoft 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.

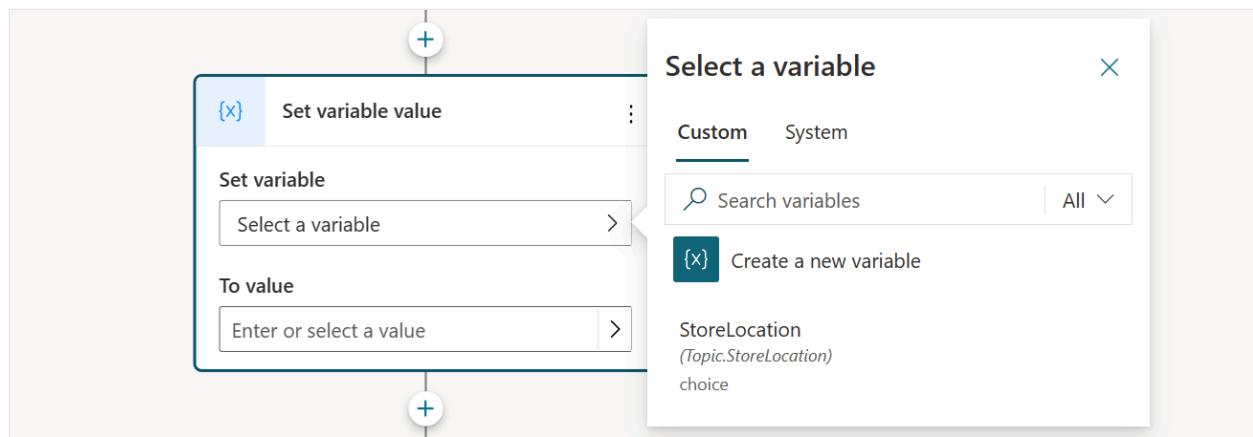
With these new authoring capabilities in Copilot Studio, a copilot author/maker can then retrieve that object from Power Automate. Then, the author/maker uses the parse node feature and the new variable features to extract the information from that object. Once retrieved, they format it as required (for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

[Learn more about the latest features around Variables and PowerFX in Copilot Studio](#)



Write your copilot 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.

```

1 kind: AdaptiveDialog
2 beginDialog:
3   kind: RecognizedIntent
4   id: main
5   intent:
6     displayName: Lesson 2 - A simple topic with a condition and variable
7     triggerPhrases:
8       - Find my nearest store
9       - Check store location
10      - Find a store
11      - Show me the nearest location
12      - Store near me
13
14   actions:
15     - kind: SendActivity
16       id: qvduia
17       activity: I'd be glad to help find a store near you.
18
19   + kind: SetVariable
20     id: setvariable_M7UjBz
21     variable:
22
23   + kind: Question
24     id: LbMwpD
25     alwaysPrompt: true
26     variable: init:Topic.Storelocation
27     prompt: Which location are you interested in?
28     entity:
29       kind: EntityWithIdentity
30       id: location
31       kind: ClosestIdentity
32       smartMatchingEnabled: true
33     items:
34       - id: Redmond
35         displayName: Redmond
36
37       - id: Seattle
38         displayName: Seattle
39
40       - id: Kirkland
41         displayName: Kirkland
42
43   - kind: ConditionGroup
44     id: H63HU
45     conditions:
46       - id: 800nmh
47         condition: Topic.Storelocation = 'cr389_bot2.topic.Lesson2.main.LbMwpD'.Redmond
48         actions:
49           - kind: SendActivity
50             id: WH1TPE
51             activity: "Check out the Company Store at our Visitor Center. The address is: 4567 Main St., Redmond, 26132."
52
53       - id: 9pu
54         condition: Topic.Storelocation = 'cr389_bot2.topic.Lesson2.main.LbMwpD'.Seattle
55         actions:
56           - kind: SendActivity
57             id: X7wlp
58             activity: "Our [Topic.Storelocation] store is located in the U District in the University Village. The address is: 1234 E. University Village St., Seattle, 34567."
59
60       - id: 1qWPF
61         condition: Topic.Storelocation = 'cr389_bot2.topic.Lesson2.main.LbMwpD'.Kirkland
62         actions:
63           - kind: SendActivity
64             id: jTQoR0

```

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

Conversation Booster (Preview) Boosted Conversations: Copilot Makers/Authors are empowered to create a useful copilot in seconds by connecting it to a datastore, such as a public facing website, allowing the copilot to be able to generate answers to questions using that datastore. You can use this feature at copilot creation or within the 'AI Capabilities' section in the copilot's settings.

Copilot (Preview): Creating copilots with AI is simplified by using Copilot in Copilot Studio. In the latest release, Copilot 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.

AI capabilities

Enable intelligent features in your bot. You consent to your data flowing outside your organization's compliance and geo boundaries. By proceeding, you agree to the [supplemental preview terms](#).

AI capabilities are premium features and can be enabled or managed by your administrators. [See pricing tiers](#)

[Learn more about responsible AI at Microsoft](#)

Boost conversational coverage (preview) 

Allow your bot to create responses to users' questions. Information for these responses comes from a website you choose. Your bot will only create responses if it receives a message that doesn't match a topic you've created.

Boosting conversational coverage is intended to help your bot respond to a wider range of information, but it's not intended to replace topic authoring.

Here's what you need to know about this feature:

- This feature is **experimental** and you'll need to speak to your administrator to enable publishing.
- Generating responses from site content **does not guarantee accuracy or relevance**.
- This feature runs on Azure OpenAI service. [Learn more about Copilot service](#)

Boost conversations

Enter your website

How to choose a website

- Pick a publicly accessible website, such as your primary company domain or customer support site.
- Ensure that the site you select is accurate and up to date - the bot will use it as the basis for its conversations.
- Ensure the site is 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, special characters, or the character '*' in your URL.

Bot content moderation

Choose the level of content moderation you'd like your bot to include while boosting conversational coverage. A higher level of moderation means that the bot's answers will be more relevant. A lower level of moderation means that the bot will generate more answers. [See more](#)

High (default)
Bot generates fewer answers, but responses are more relev...

Intelligent authoring support with Copilot (preview)

Write a description of a topic you'd like your bot to cover and Copilot will build it out for you.

Intelligent authoring support is available when you create or edit any preview bot. You can turn it on in the [user settings](#).

Internal documentation support (preview)



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 copilot 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, copilot 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 bots

See [Cloning an existing Power Virtual Agents classic bot to a Copilot Studio copilot](#).

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 copilot 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 copilot testing, even for simple copilots.

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 copilot, found in the testing pane and the metadata of the copilot, found in the **Copilot details** section of the copilot management area, under **Advanced**, such as *Environment ID*, *Tenant ID*, *Bot App ID*.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Key concepts for Bot Framework Composer users

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.

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.

Bot Framework	Copilot Studio	Description
Composer	concept	
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.

Microsoft Copilot Studio plugin architecture

Article • 03/26/2024

ⓘ 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.

A **plugin** is a reusable piece of code that can perform a specific task or provide specific functionality for a copilot. For example, a plugin can help a copilot answer a natural language query, execute a workflow, connect to an external system, or provide topic-specific guidance. A **copilot** is a conversational or UX-based assistant that helps users accomplish their tasks and goals in a specific domain or application.

A plugin, in this context, is created in the plugin authoring experience in the Microsoft Copilot Studio. This feature allows users to create and edit plugins using a graphical user interface and publish them to the plugins registry.

The plugins registry helps you create a plugin once and use it in multiple copilots. The registry provides storage and management for metadata and execution information for plugins. Users can apply the power and flexibility of plugins to enhance the capabilities of copilots without writing code for each copilot separately. The various copilots interact with the plugins registry to discover plugins and execution information available for a user. This capability enables AI plugins to be created once and reused many times.

A plugin is born

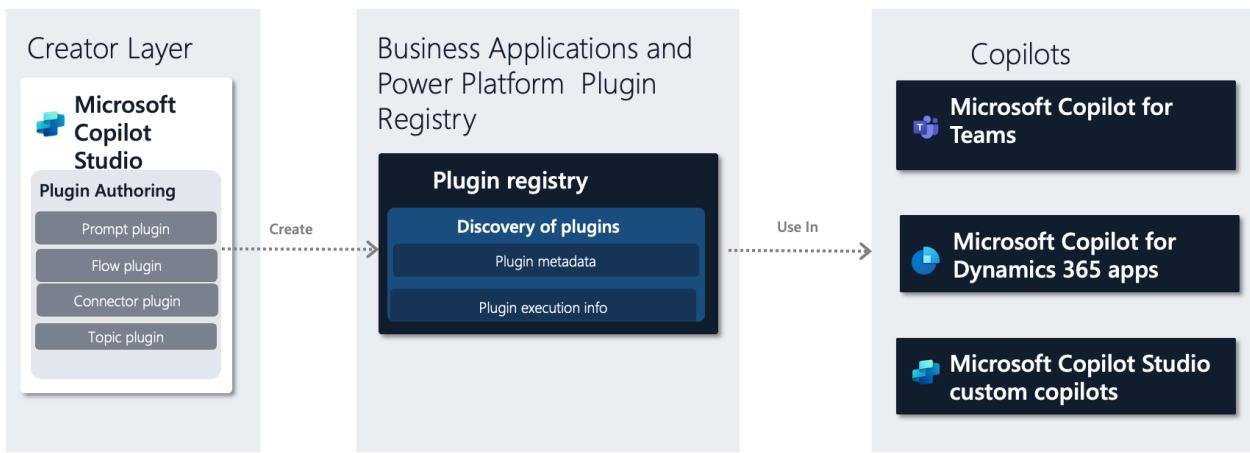


Figure: Plugin Authoring using Microsoft Copilot Studio

Creator Layer

The Creator Layer of Microsoft Copilot Studio is where users can create and edit plugins. This layer supports four types of plugins: Prompt, Flow, Connector, and Topic.

- **Prompt** plugins are plugins that can answer natural language queries from users using natural language processing (NLP) and natural language generation (NLG) techniques. For example, a prompt plugin 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** plugins are plugins that can execute multi-step workflows using Microsoft Power Automate. For example, a flow plugin can create a new contact in Microsoft Dynamics 365 after manager approval or send an email to a customer with a prescribed template.
- **Connector** plugins are plugins that can connect to external systems or data sources using Microsoft Power Platform connectors. For example, a connector plugin can connect to Salesforce or SAP using Power Platform connectors and retrieve or update data. Connector plugins can be certified connectors that are shared across tenants or environment-specific custom connectors.
- **Topic** plugins are single-turn conversational threads between a user and a copilot 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" due to the power of LLMs.

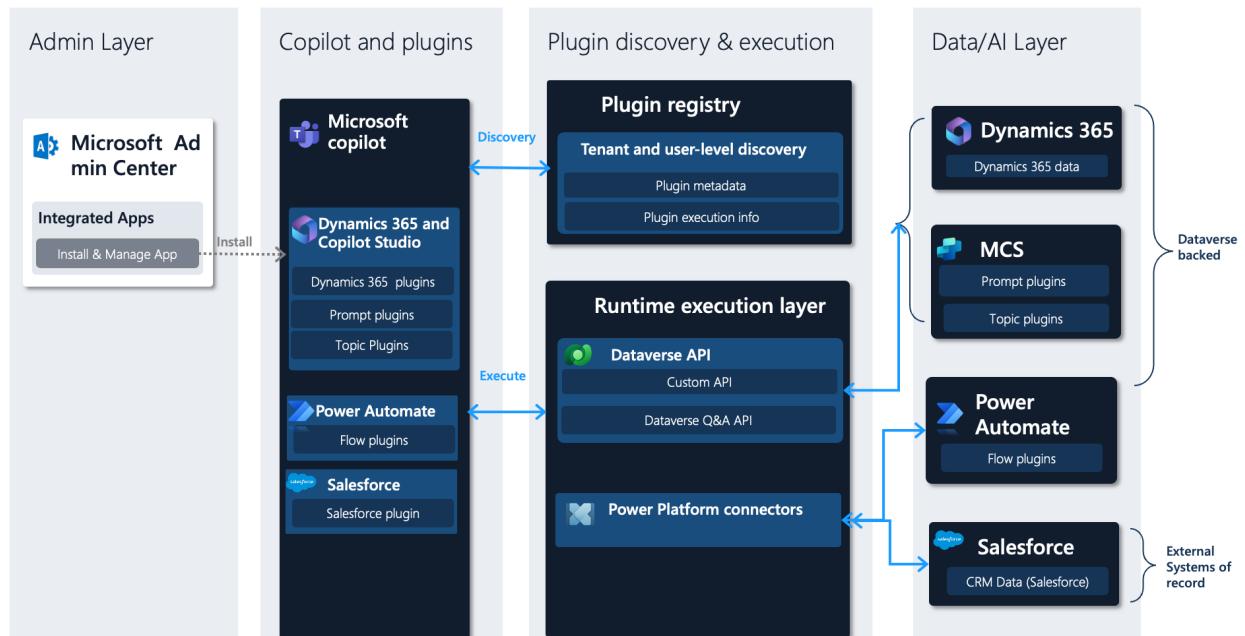
Business applications and Power Platform plugin registry

The plugins registry stores and manages the plugin metadata and execution information. The registry is a single source for discovery of plugins 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 plugins available to a user, or used by copilots in Microsoft Copilot Studio. A plugin has the same security as its underlying artifact, such as a flow and can provide a customized list of plugins for a specific user role.

Copilot samples

This sample set of the various copilots integrates with the plugins registry to consume plugins. These copilots include Microsoft copilots for Dynamics 365 apps, Microsoft copilots for Teams, and custom copilots authored using Microsoft Copilot Studio. The list is expected to grow in the future as more first-party and third-party copilots are developed.

A plugin 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 plugins. The apps available from Microsoft Copilot Studio-created plugins are retrieved from the plugins registry and include:

- **Dynamics 365 and Microsoft Copilot Studio:** Includes Dynamics 365 plugins from teams such as Microsoft Dynamics 365 Sales and Microsoft Dynamics 365 Field Services. You can create custom prompts and topic plugins.
- **Power Automate:** Includes Power Automate flow plugins.
- **Connector specific:** A sample of Salesforce and SAP plugins are called out here. Every certified connector plugin created by Microsoft or ISVs shows up as their own integrated app for fine-grained management.

 **Note**

App names are subject to modification.

Plugin Discovery and Execution

The plugins available to a maker include plugins assigned through deployed integrated apps. The plugins available to a maker are discovered through the plugins registry. Makers are free to further customize the plugins based on how they want to use the plugin flyout experience. The plugin execution information gets retrieved from the plugins registry and includes the runtime path.

There are two runtime execution paths supported today:

1. **Plugins that are Dataverse-based:** Microsoft Dynamics 365 plugins that access or modify Dataverse data, Microsoft Copilot Studio topic plugins, and prompt plugins. This path supports Microsoft Entra ID authentication. The calls are made to the underlying system on behalf of the user.
2. **Plugins 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 plugin for their task, they can navigate to the Microsoft Copilot Studio and create a plugin that addresses their needs.

Data/AI Layer

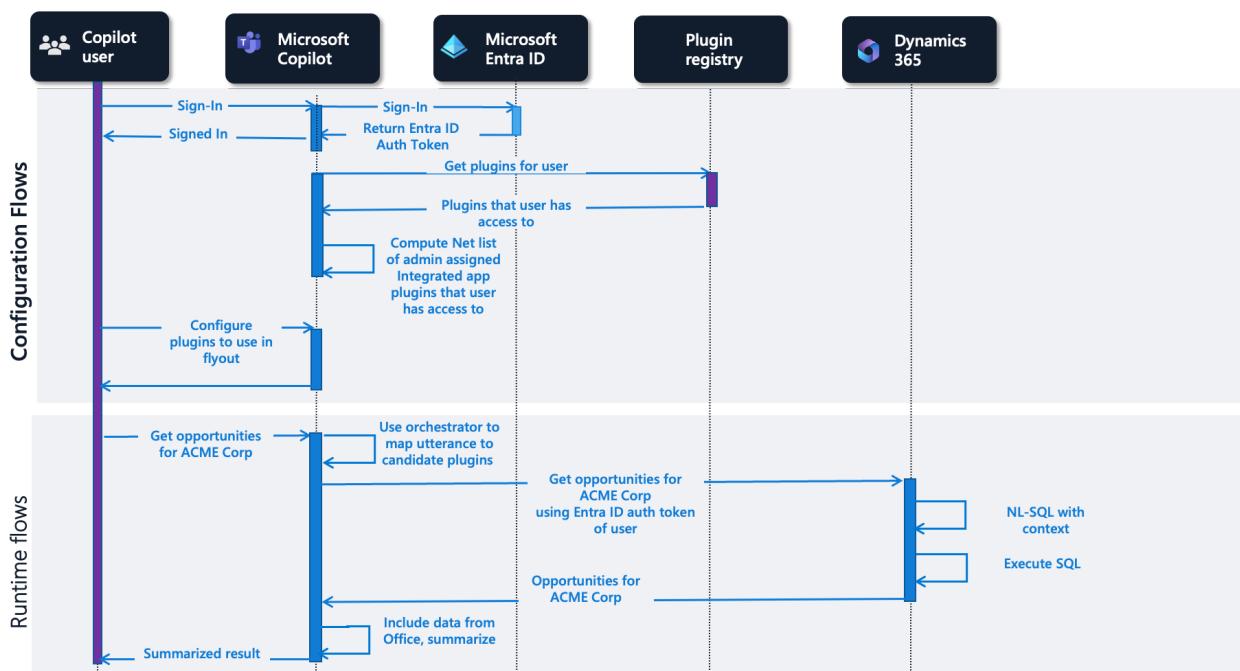
This layer represents the various data stores and systems where the data and business logic reside. These stores and systems include Sales, Field Service and other Dynamics

365 products, Microsoft Copilot Studio, Power Automate, external systems such as Salesforce, SAP, and others.

Data and control flows for some plugin types

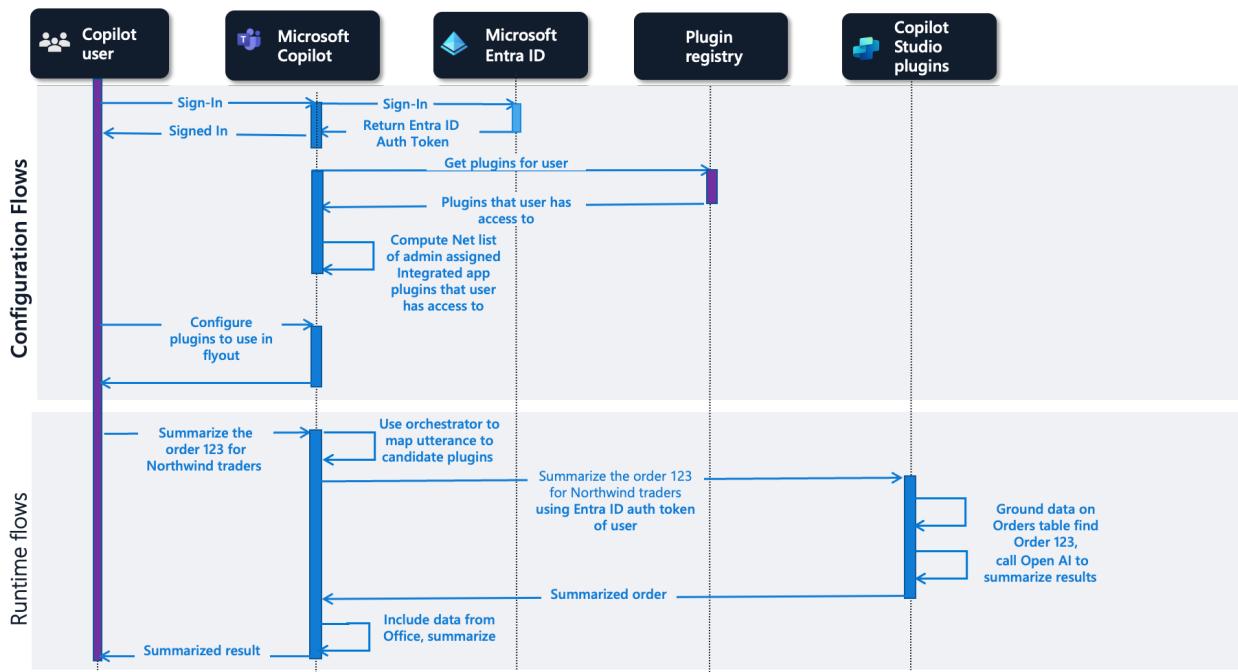
As a prerequisite for all flows, an administrator uses the Microsoft Admin Center to configure the corresponding integrated apps and assigns them to users, including the Copilot user referenced here. In these flows, we first see the **configuration phase** for a copilot user in Copilot for Teams. After the user signs in, the plugins applicable for that user are retrieved from the plugin registry. The plugins in a user's flyout belong to the integrated apps their administrator configures for them and the ones they have access to in the plugin registry. The user can then configure the plugins they want to use in the copilot experience in the flyout in their Copilot for Teams experience.

Dynamics 365 plugins



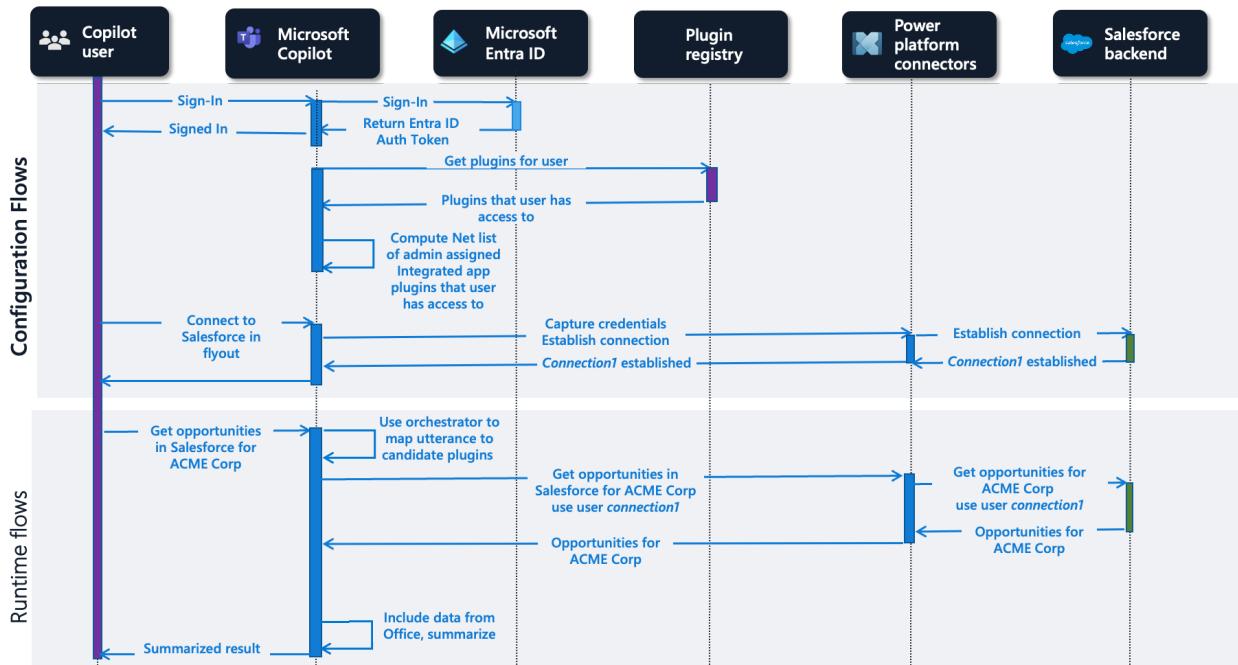
In the runtime flow for the Dynamics 365 plugins, the orchestrator maps a copilot user's utterance of *Get opportunities for ACME corp* to a set of candidate plugins. A plugin from Dynamics 365 is found to be best candidate and then executed, translating the user's utterance to SQL against the dynamic data the user has access to. The results are then returned to the user. Teams copilot can additionally add data from Microsoft 365 and summarize the results.

Microsoft Copilot Studio plugins



In the runtime flow for Copilot Studio prompt plugins, the user's utterance is again mapped to a plugin. In this case, the matching plugin is a prompt authored in Microsoft Copilot Studio. The prompt grounds data in Dataverse table *orders* and calls into Open AI to summarize the results for order 123 and returns the summary to the user.

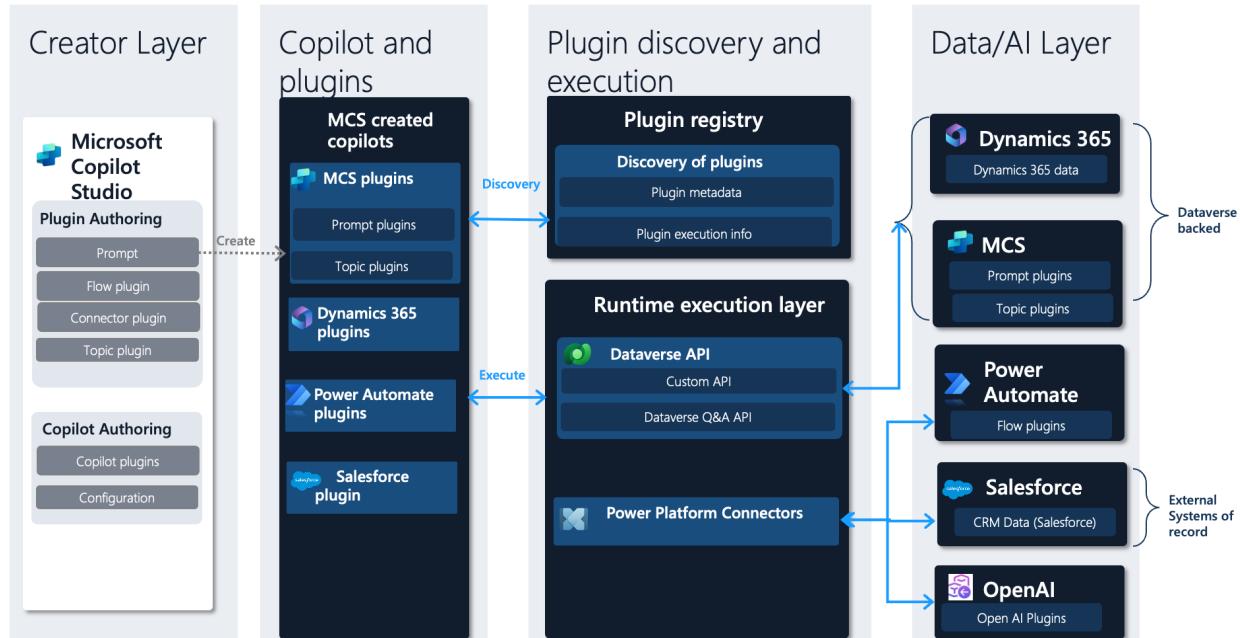
Connector plugins



The configuration flow for Salesforce or other connector-based plugins 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 plugin and this plugin is executed using the Power Platform Connector infrastructure. The results are then returned to the user.

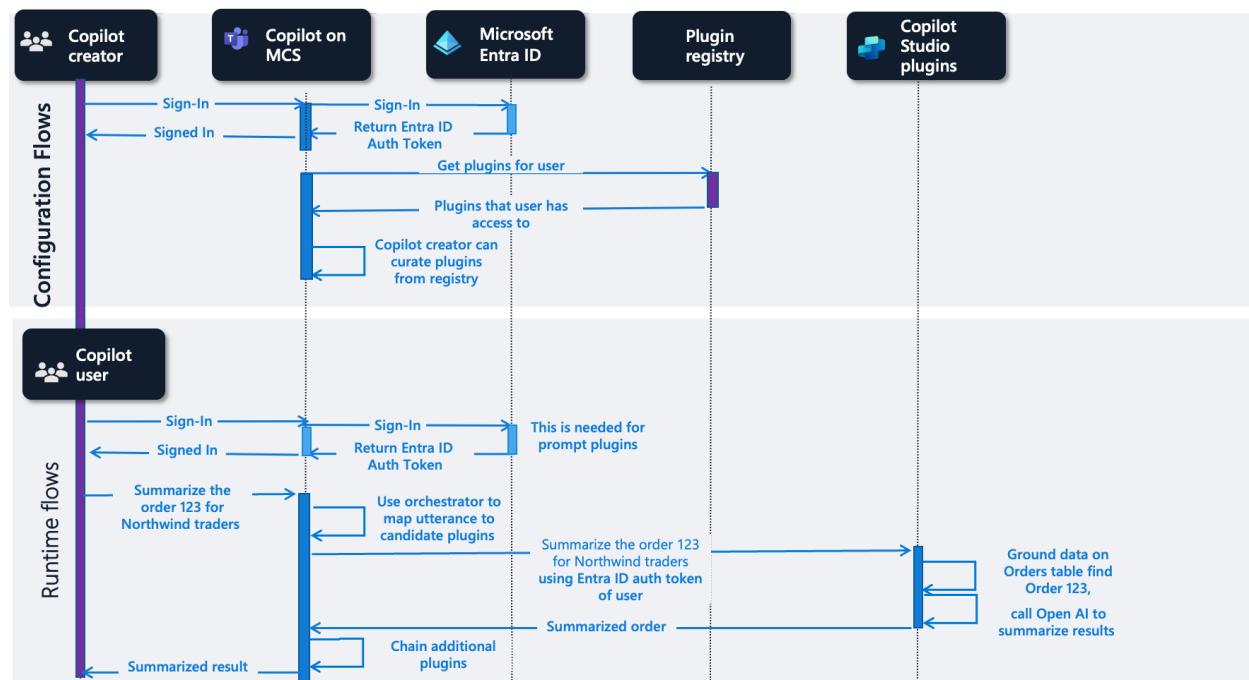
Plugins used in Microsoft Copilot Studio copilots

Microsoft Copilot Studio can be used to create copilots in addition to plugins. Dynamics 365 teams or third parties can build these copilots.



Similar to the creator layer, the plugin discovery, execution, and Data/AI layer look like the figure shown. The difference is in how you manage curation of plugins for the copilot. The copilot creator chooses amongst available plugins from their Dataverse environment.

Data and Control flows for some plugin types



Here's an example of a prompt plugin used from a copilot built in Microsoft Copilot Studio. The configuration flow includes a copilot creator curating the plugins for the copilot from the plugin registry. At runtime, the authenticated copilot user summarizes order 123 for Northwind traders. This summary maps the utterance to candidate plugins. The matching plugin 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 a plugin in Microsoft Copilot Studio, you can further the reach of your plugin across copilots in Microsoft Dynamics 365, Microsoft Copilot for Teams, and other third-party copilots. Using plugins in this way allows the flexibility to create once and deploy across multiple copilots.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Get access to Microsoft 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 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 Microsoft Copilot Studio](#) topic.

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](#).

Microsoft Copilot Studio for Microsoft Teams plan

Copilot Studio for Teams enables customers to build conversational interfaces within Teams. The copilots 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, and configuration values](#) topic for other capacity considerations.

[+] [Expand table](#)

Capability	Select Microsoft 365 subscriptions	Copilot Studio subscription
Gen AI: AI-enabled conversations	Not available	Use Generative Actions in Microsoft Copilot Studio
Deploy copilot 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	Use Power Automate flows in Copilot Studio
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 copilot 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 Copilot Studio copilot as a skill
Integrate Microsoft Bot Framework dialogs	Not available	Develop custom dialogs with Bot Framework Composer
Hand off copilot conversation to a live agent	Not available	Trigger handoff 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 copilot in the same environment. Capabilities that were previously only available in a standalone license are now be available. These capabilities might include [billed sessions that require Copilot Studio capacity](#).

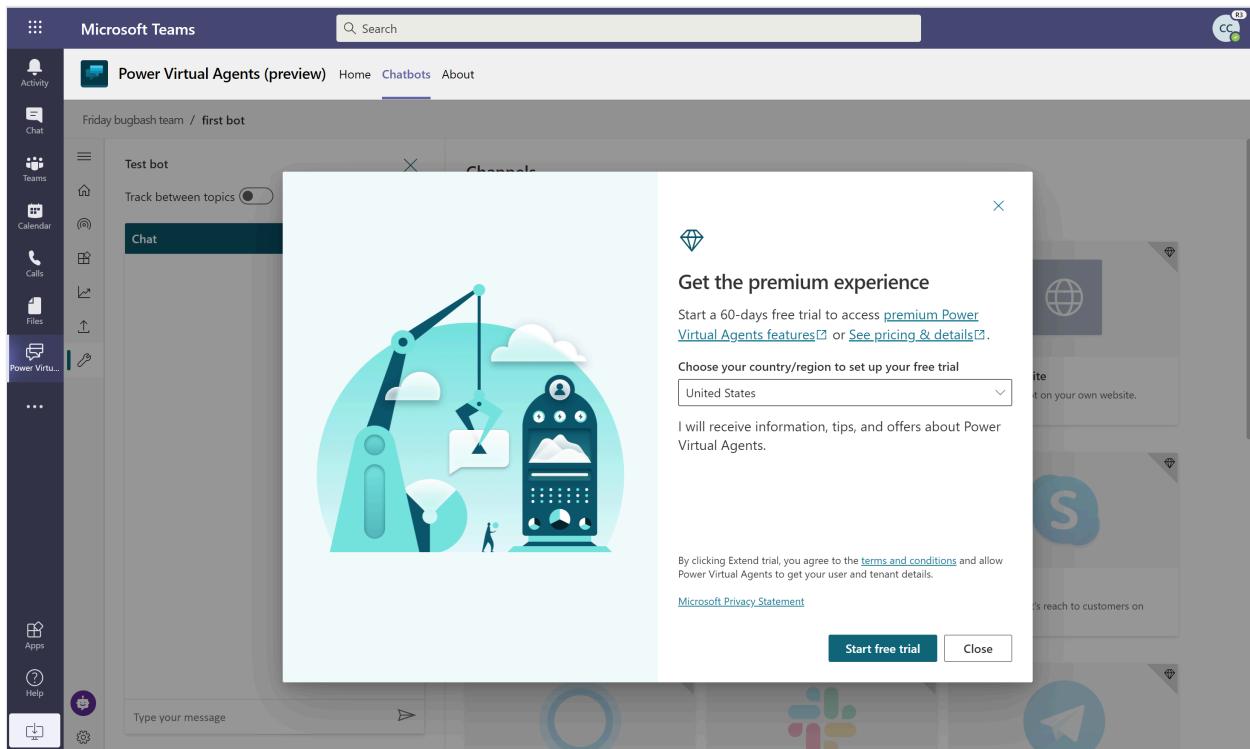
You can upgrade within the Copilot Studio app for Teams from **Settings > Channels**. Select the apps you want to publish to and you're prompted to upgrade.

The screenshot shows the Microsoft Copilot Studio app interface for a team named "Fabrikam manufacturing support". The left sidebar includes options like Overview, Topics, Entities, Generative AI, Analytics, Publish, Settings, AI integration tools, and Channels, with the Channels option highlighted by a red box. The main content area displays a "Channels" section with six configuration cards:

- 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.

At the top of the main content area, there are two status messages: "Your bot may not work as intended right now due to your organization's data loss prevention policies. Address affected topics or contact your admin." and "Some channel settings aren't available due to your organization's data loss prevention policies. Contact your admin with questions." A "Download details" button is also present.

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



Note

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

Standalone Microsoft Copilot Studio subscription

The standalone Copilot Studio subscription allows you to build copilots 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 Microsoft Copilot Studio](#).

Other subscriptions that include Microsoft 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?

 Yes

 No

[Provide product feedback ↗](#)

Quickstart: Create and deploy a Copilot Studio copilot

Article • 06/06/2024

ⓘ 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.

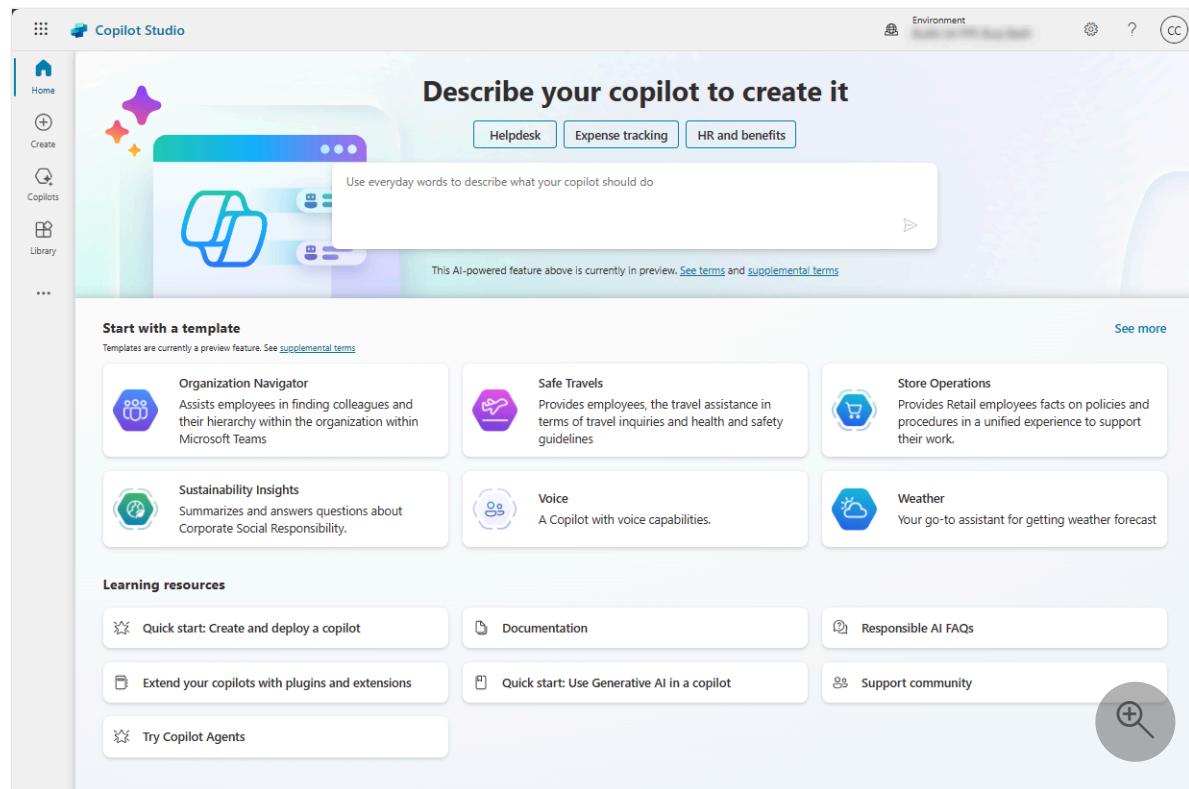
Copilot Studio empowers teams to quickly and easily create powerful copilots using a guided, no-code graphical experience without needing data scientists or developers.

This quickstart article helps you create a copilot for the first time. You learn how to add topics to your copilot, test content changes in real-time, and deploy your copilot to a test page. Once deployed, you can analyze the performance of your copilot.

Web app

Explore the home page

Copilot Studio has an app-level home page, listing your recently accessed copilots. On this page you can create a new copilot, view recent copilots, and access learning resources like documentation and learning paths.



In the navigation menu, select **Copilots** to open the **Copilots** page. All copilots you create in a given environment appear here. In your list of copilots, you can see metadata such as owner, when it was last published, and who last modified the copilot.

Create a copilot

1. Go to the [Copilot Studio introduction website](#). Supported browsers include Microsoft Edge, Chrome, and Firefox.
2. On the website, select **Try free** and sign in with your work email address.

⚠ Note

Personal Microsoft accounts are currently not supported.

3. After you sign up, a default **Power Platform environment** is made for you. For most users, this default is sufficient. However, to specify a custom Power Platform environment, select another one, using the environment picker.

The screenshot shows the Power Platform environment 'Build 24 PPE Bug Bash'. The top navigation bar includes icons for Environment, Build 24 PPE Bug Bash, settings, help, and a copy icon. Below the navigation is a search bar with the placeholder 'Search copilots' and a magnifying glass icon. The main area displays a table with columns: Modified by (sorted), Last published, and Owner. A large pink callout box contains a note about supported environments.

ⓘ Note

Copilot Studio is only supported in these data locations. Outside of these locations, create a custom environment with **Region** set to a supported data location before you create your copilot.

For more information on creating a custom Power Platform environment, see [Working with environments](#).

- Once signed up, you land on the **Home** page. Select **Create** in the left navigation.

The screenshot shows the Copilot Studio 'Create' page. The left sidebar has buttons for Home, Create (which is highlighted with a red box), Copilots, Library, and three dots. The main area features a large 'Describe' input field with a placeholder 'Use everyday words to describe what you want to do'. Below it is a 'Start with a template' section with a note that templates are a preview feature. At the bottom is an 'Organization Navigator'.

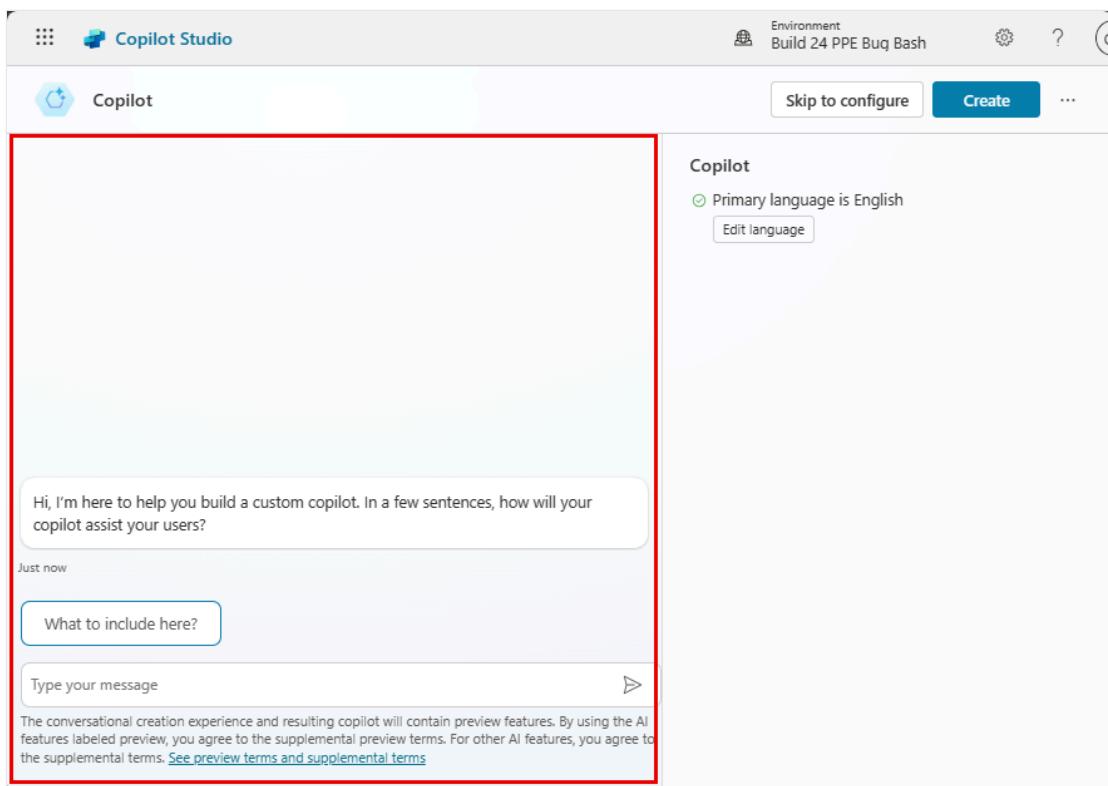
- On the **Create** page, choose to create a new copilot using a *conversational creation* experience or using the **Configuration** page. Alternatively, create a copilot based on an existing template.

Create a new copilot

① Note

While creating a copilot, use of the Configure page is generally available. However, creating a copilot with the conversational builder is presently in preview mode.

1. On the **Create** page, select **New copilot**.
2. On the left side of the pane, engage with the conversational builder in plain language by answering the questions it asks.



① Note

- Avoid single-word responses.
- Use conversational style language. For example:
 - "Provide employees with help on troubleshooting issues with their company issued devices"
 - "Help employees submit expense reimbursement requests"
 - "Create a copilot for topics related to Human Resources and employee benefits"

3. As you respond to the questions, observe the summary in the right pane to know what the LLM (Large Language Model) is interpreting from your inputs. To scope your copilot, continue to refine and give instructions.

The screenshot shows the Copilot Studio interface. At the top, there's a header with the Copilot Studio logo and environment information ('Environment Build 24 PPE Bug Bash'). Below the header, there's a navigation bar with 'Copilot' and buttons for 'Skip to configure', 'Create', and more. The main area has a conversation log on the left and a configuration pane on the right.

Conversation Log:

- Hi, I'm here to help you build a custom copilot. In a few sentences, how will your copilot assist your users?
24 minutes ago
- Provide employees with help on troubleshooting issues with their company issued devices
9 minutes ago
- Great, your copilot will assist employees with troubleshooting issues related to their company-issued devices.
Do you have any instructions for how your copilot should assist, for example a specific tone?
8 minutes ago
- Use a casual tone.
Just now
- Understood, your copilot will use a casual tone when assisting employees with device troubleshooting.
Where should the copilot find important information?
Provide any publicly accessible websites that your copilot will need.
Just now
- Any tips?

Configuration Pane (Right Side):

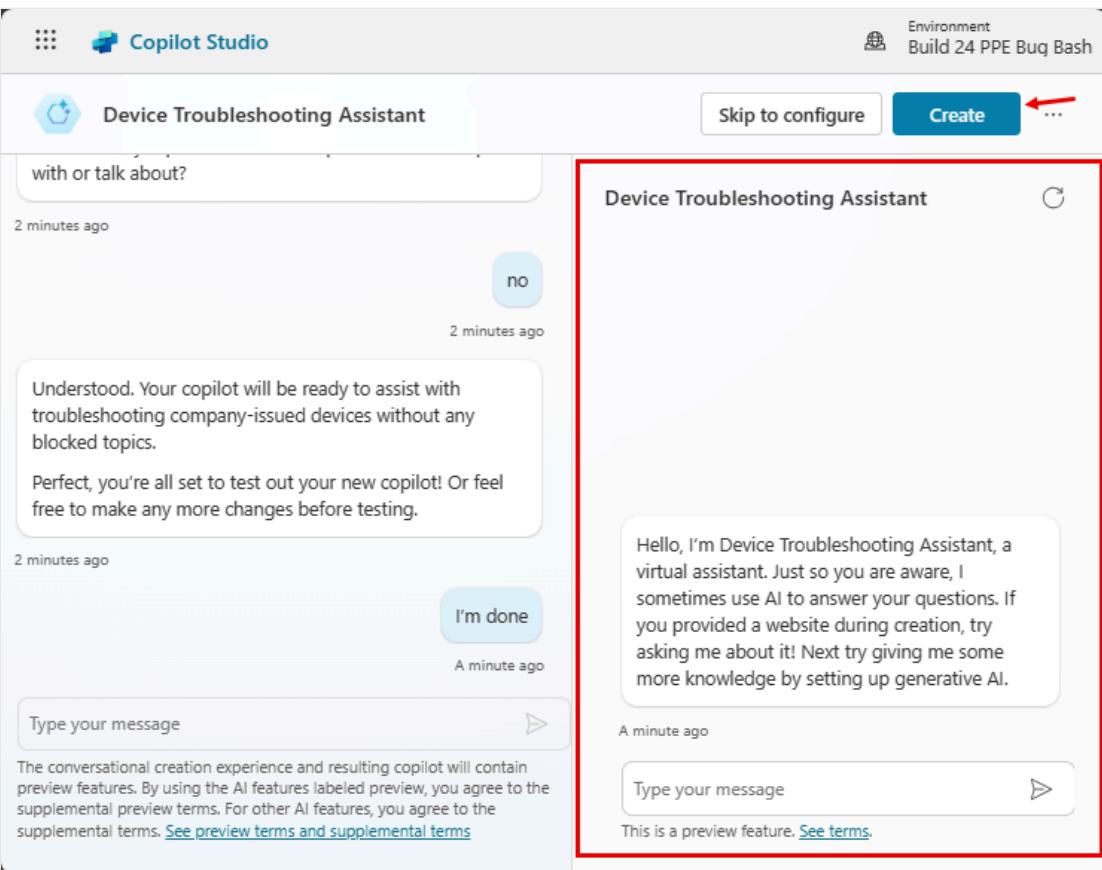
Copilot

- Primary language is English
[Edit language](#)
- Assists employees with troubleshooting issues related to their company-issued devices.
- Uses a casual tone when assisting employees with device troubleshooting.

You can provide one or more knowledge sources to the copilot to use as resources to provide answers from.

4. When you're ready, select **I'm done. Let's try it out..**

The preview pane appears. You can test the copilot's responses and continue to refine in the **Configuration** page based on your testing in the preview pane. The copilot hasn't been created yet.



5. After testing the copilot, select **Create**.

(!) Note

You can select **Create** at any time during the conversation.

Copilot Studio opens the **Configuration** page for your new copilot.

Create a copilot from a template

(!) Note

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

1. On the **Create** page, under **Start with a template**, select a template.
2. On the template configuration page, enter a name, description, and instructions for your new copilot.
3. To change the icon representing your copilot, select **Change icon**.

4. To confirm the template's preset data connection, select the three dots next to the data connection, and then select **Edit**.

Name * 15/100
Weather Copilot

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

Description 49/1000
Your go-to assistant for getting weather forecast

Instructions 66/8000
You are an assistant helping to answer questions about the weather

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

MSN Weather Setup connection ⚠️ ⋮

Language
The primary language your copilot will use

English (en-US) ▼

Edit Delete

5. On the **Connection setup** page, if you're satisfied with the listed connection, select **Confirm**.
6. On the template configuration page, select a primary language.
7. To add extra knowledge sources, select **Add knowledge**.
8. When you're finished configuring your new copilot on the template configuration page, select **Create** at the top of the page.

The screenshot shows the Copilot Studio interface for creating a new copilot. The top navigation bar includes 'Copilot Studio' and 'Environment Build 24 PPE Bug Bash'. A 'Create' button is highlighted with a red box. The main area is titled 'Weather Copilot' with a weather icon. A note suggests reviewing 'template instructions'. The form fields include:

- Name ***: Weather Copilot (15/100 characters)
- Icon**: A sun and cloud icon. A 'Change icon' button is available. A note specifies the icon should be in PNG format and less than 30 KB.
- Description**: Your go-to assistant for getting weather forecast (49/1000 characters)
- Instructions**: You are an assistant helping to answer questions about the weather (66/8000 characters)
- Connect your data**: MSN Weather data source is connected.
- Language**: English (en-US) selected.
- Knowledge (2)**: URLs listed:
 - <https://www.theweathernetwork.com>
 - <https://weather.gc.ca>A '+ Add knowledge' button is present.

At the bottom, a note states: "Review [supplemental terms](#) to learn more about the templates in preview. You are responsible for complying with the terms applicable to the public URLs listed above in Knowledge. See the URLs for terms."

9. To test your new copilot, select **Test** at the top of the page, and then test the responses. Continue to refine your copilot as desired in the **Configuration** page. The copilot hasn't been published yet.

The screenshot shows the Copilot Studio interface. On the left, there's a navigation sidebar with options like Home, Create, and Copilots. The main area is titled 'Weather Copilot' and has a sub-tab 'Configuration' highlighted with a red box. A banner at the top says 'Your free trial expires in 1 day(s). Contact your admin to discover the plan that's right for you. [See pricing](#)'. Below this is a modal window titled 'Your copilot is ready! Here's what's next:' with three items: '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 of the modal is a 'Details' section with fields for Name (Weather Copilot), Description (Your go-to assistant for getting weather forecast), and Instructions (You are an assistant helping to answer questions about the weather). Below these are sections for Knowledge (with a '+ Add knowledge' button) and Actions (with a '+ Add action' button). On the far right, there's a 'Test your copilot' pane showing a conversation: 'Hello, I'm Weather Copilot, 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.' (7 minutes ago), 'What can I ask?', 'Ask a question or describe what you need', and '0/2000' (with a send arrow icon). At the bottom of the test pane, it says 'Make sure AI-generated content is accurate and appropriate before using. [See terms](#)'.

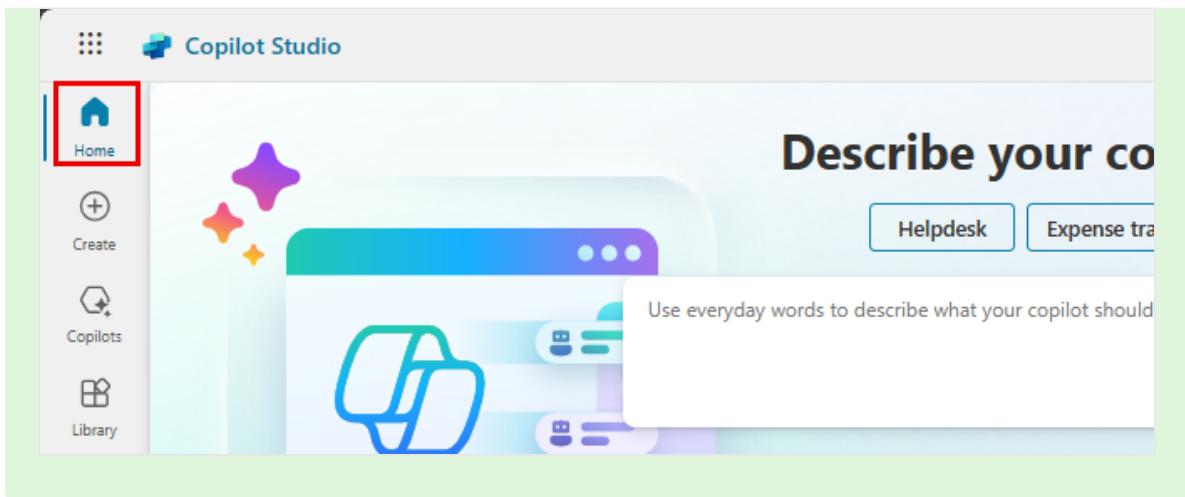
Familiarize yourself with Copilot Studio

While waiting for processing, you might want to:

- Explore the overall user interface for Copilot Studio.
- Explore topics and start creating and editing topics.
- Experiment with the preloaded user topics and system topics.
- Interact with your copilot using the **Test copilot** chat pane.
- Review [Key concepts - Authoring Copilot Studio copilots](#) to better understand authoring.

💡 Tip

Select **Home**, in the navigation menu, to go back to the app-level home page from your copilot.



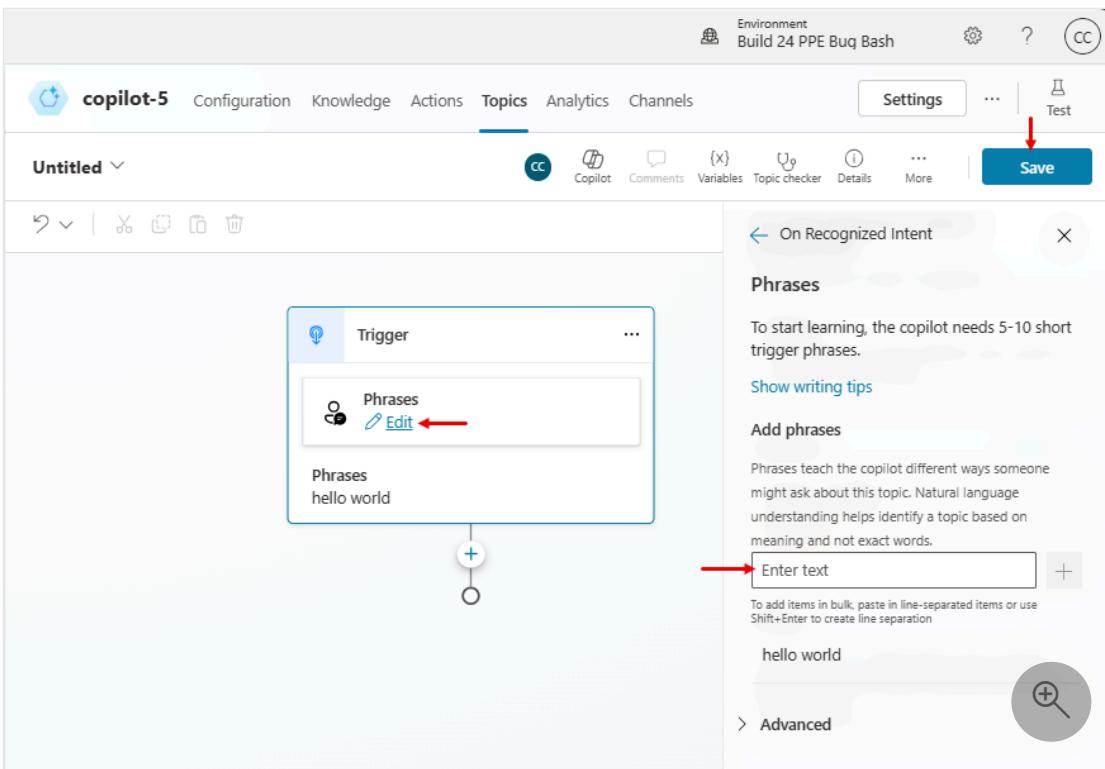
Create a topic

With a new copilot, you can start creating topics. A topic is a dialog tree that describes how your copilot responds to a user's question.

1. Open your copilot in Copilot Studio, select **Topics** at the top of the page, and then select **Add a topic > From blank**.

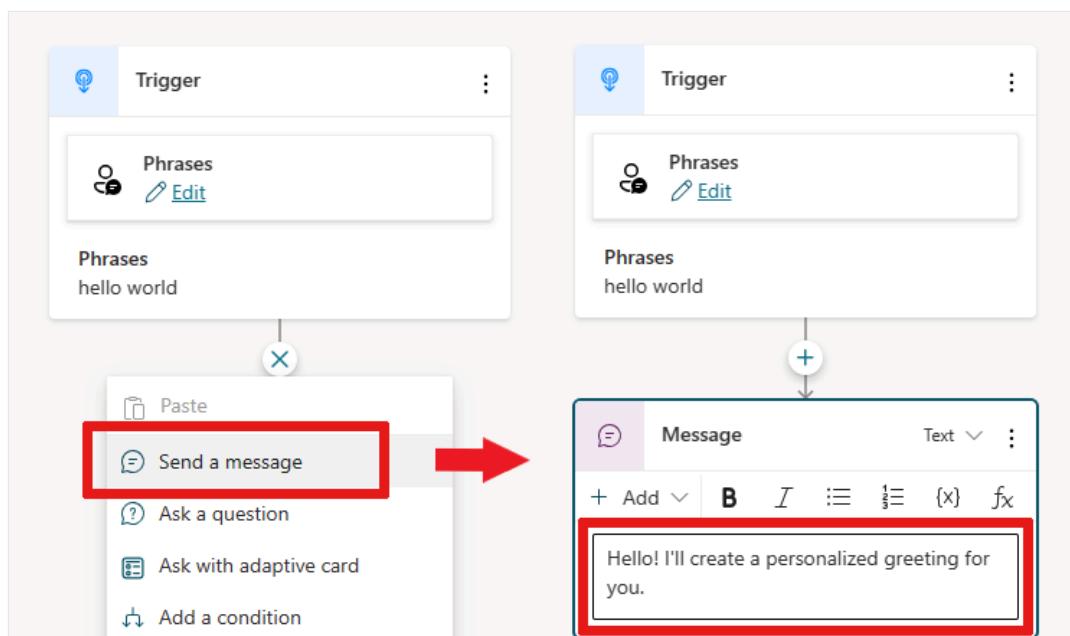
A screenshot of the Copilot Studio 'Topics' page. At the top, there's a navigation bar with tabs: Configuration, Knowledge, Actions, **Topics** (which is underlined and highlighted with a red box), Analytics, and Channels. Below the tabs, there's a button '+ Add a topic' with a dropdown arrow pointing down. The dropdown menu shows two options: 'From blank' (highlighted with a red arrow) and 'Create from description with Copilot'. Below the dropdown, there are several topic cards. One card for 'Greeting' has a magnifying glass icon and the text 'Phrases'. Another card for 'Lesson 1 - A simple topic' also has a magnifying glass icon and the text 'Phrases'. To the right of these cards, there are sections for 'Description' and 'This topic triggers when'.

2. Here you build a topic by giving it a name and configuring a trigger. Trigger phrases are examples of user questions or utterances that teach the copilot to respond when the questions or utterances are detected.
 - a. Name your topic 'Personal Hello World' at the top.
 - b. In the **Trigger** box under **Phrases**, select **Edit**. The **On Recognized Intent** pane appears.
 - c. Add 'hello world' text, also called an utterance, under **Add phrases**.
 - d. Select the + icon next to the phrase box to add it to your trigger.
 - e. Select the **Save** icon at the top to save your trigger.



3. Select the (+) icon below your **Trigger** node and select **Send a message**.

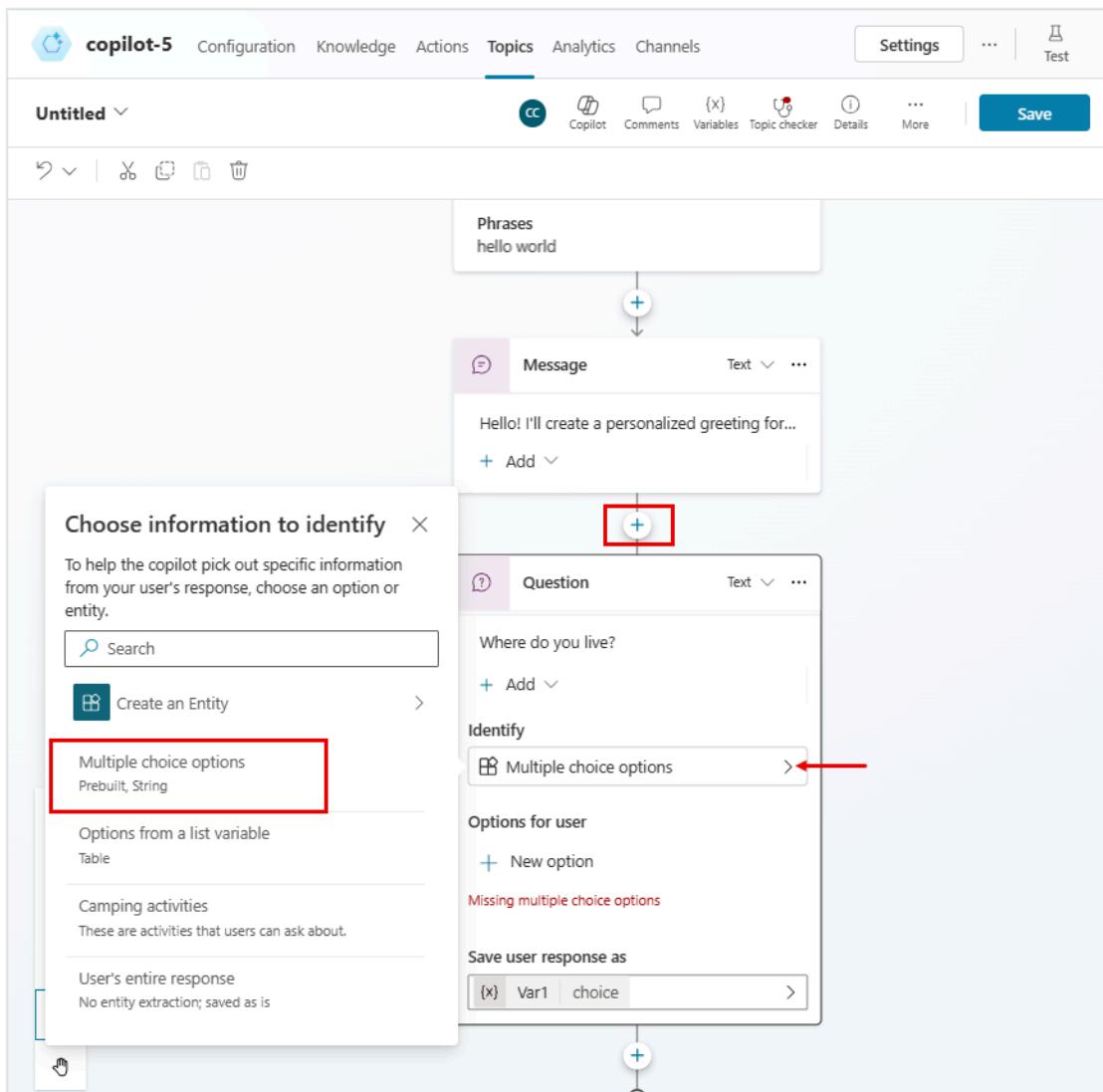
- Enter 'Hello! I'll create a personalized greeting for you.' into the newly added **Message** node.



- Select the **Save** icon at the top to save your trigger.

4. Select the (+) icon below your new message node and add an **Ask a question** node.

Enter the question text, 'Where do you live?', in the **Ask a message** box. Select **Multiple choice options** under **Identify** to give the customer a choice in responses.



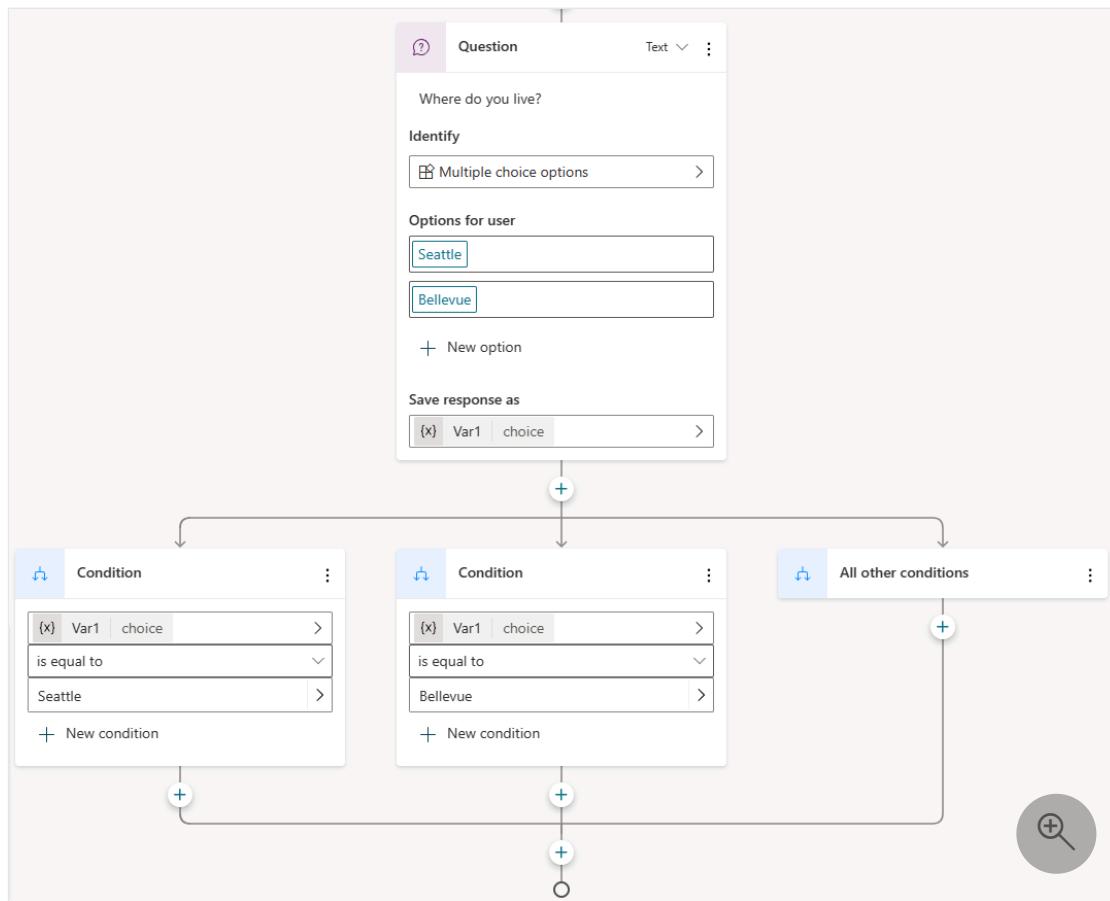
With **Multiple choice options** selected, you need to add more options.

5. Add two options for the user, by selecting **+ New option** under **Options for user** in the **Question** node, then enter 'Seattle' as the text. Add another option and enter 'Bellevue' as the text.

Each option is presented in the copilot chat as a multiple choice button to the user.

6. To help understand the conversation flow, view the whole dialog tree in the authoring canvas.

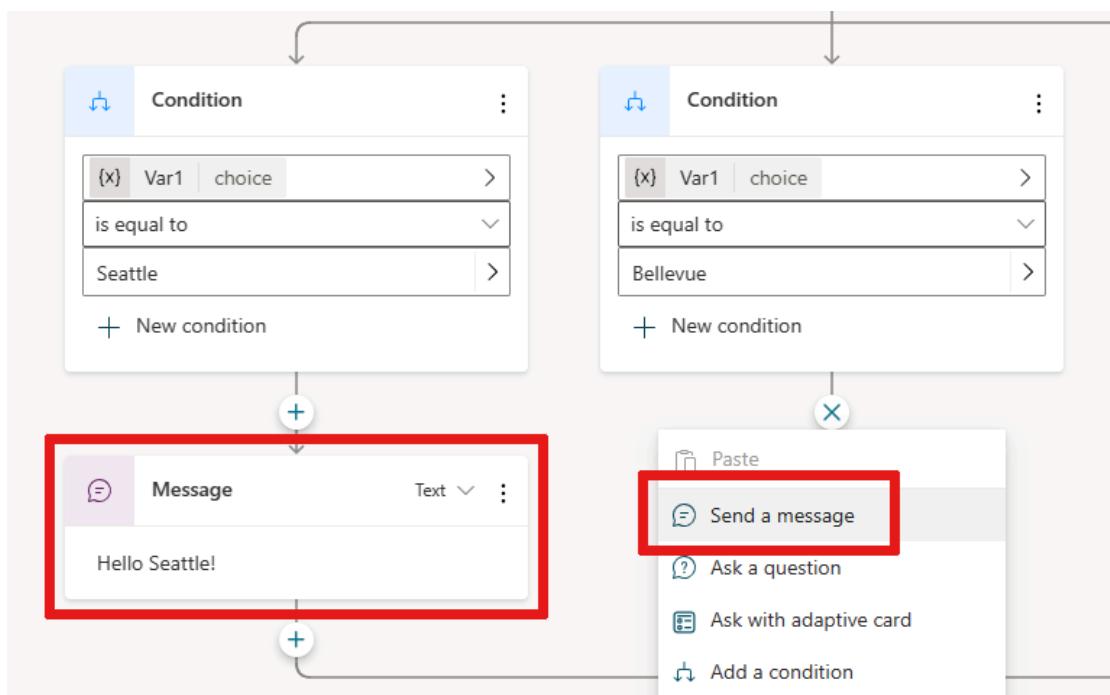
The dialog tree creates separate paths in the conversation, depending on the customer's response. This path leads the customer to the best resolution for each user response.



In the forked conversation path, the nodes automatically check for 'Seattle' in one path and 'Bellevue' in the other path to take the best next step.

- Finally, select the (+) icon below each of the **Condition** nodes to add a **Message** node in each branch.

Add a small message like 'Hello Seattle!' in the Seattle branch and 'Hello Bellevue!' in the Bellevue branch.



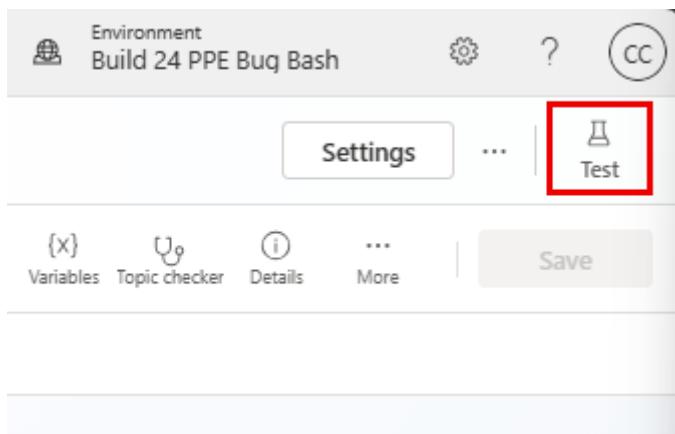
8. Select the **Save** icon at the top to save your trigger.

You now have a basic branching dialog tree, congratulations! You can begin to create more complex versions of this tree by incorporating [variables](#), [entities](#), and [Power Automate flows](#).

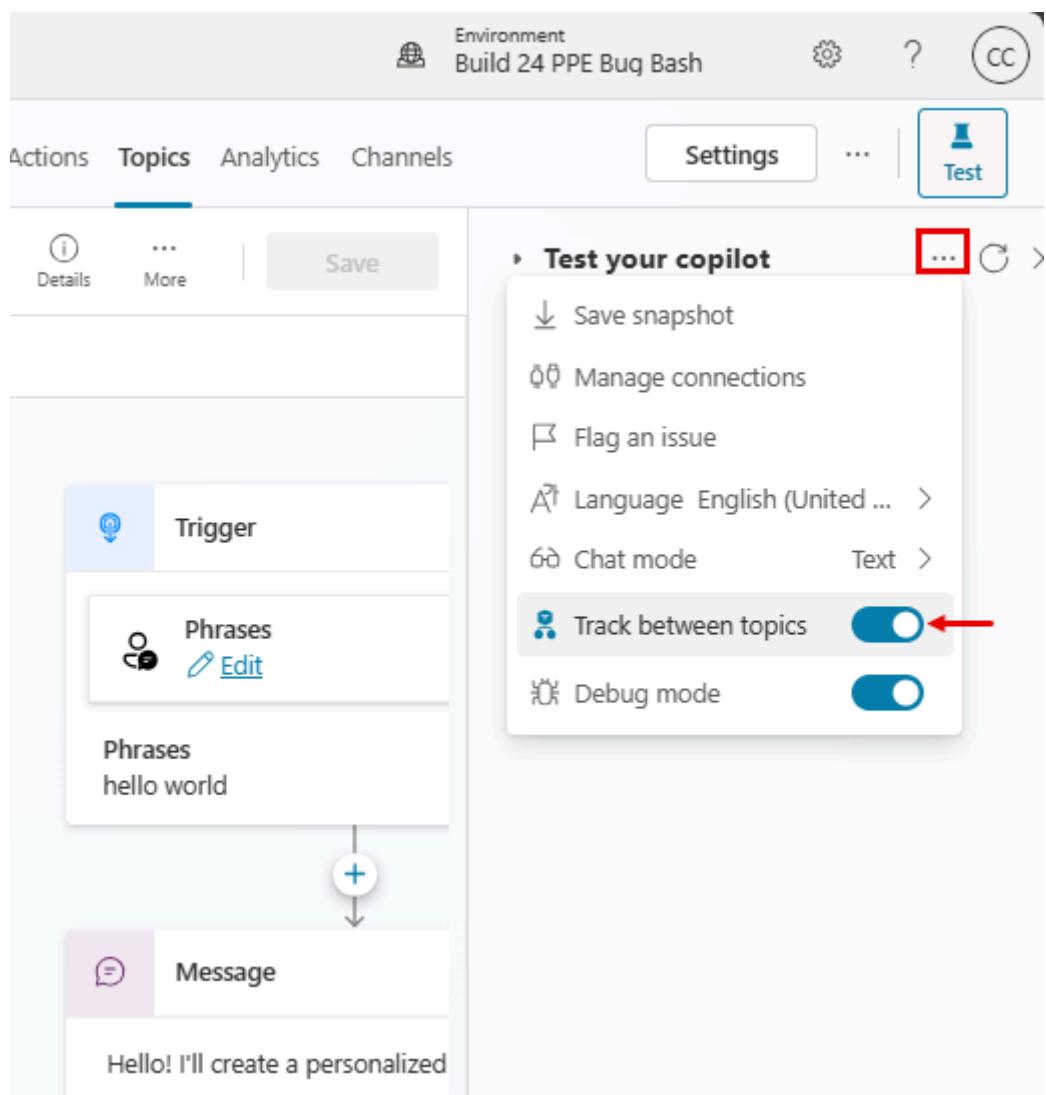
Test your content in real time

With content authored into a dialog tree, test the conversation in real time to see if it's working as you expected. You can test, using the **Test your copilot pane**.

1. If the test copilot pane isn't showing on your screen, select **Test** at the top of the page.



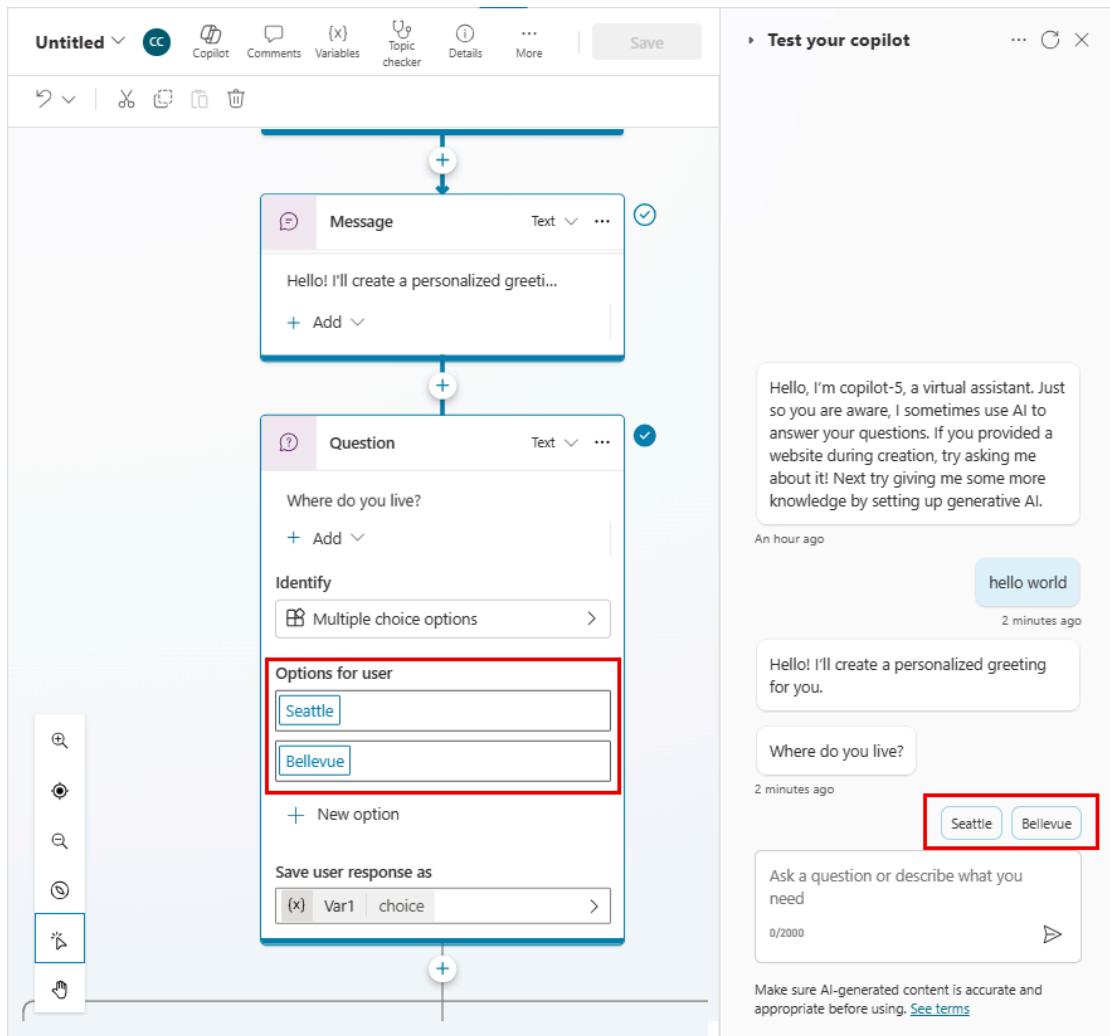
2. To follow along on a node-by-node basis as the copilot executes your dialog, select the **More** icon (...) in the **Test your copilot pane** and turn on **Track between topics**. You see parts of your dialog tree highlighted as the copilot gets to that portion of the dialog.



3. Type "hello world" in the chat window, and send the message to the copilot. You see the top portion of your dialog tree highlighted in green, and **Seattle** and **Bellevue** presented as user options in the test copilot pane.

The copilot is now waiting for you to respond, displaying suggestions on how to respond. These suggestion buttons reflect what you authored within your dialog tree in the **Ask a question** node.

In the test copilot, you can either select these suggestion buttons to continue, or you can enter your response into the chat window.



4. Continue the dialog by selecting the Seattle branch.

You see the chat stop once you reach the bottom of this branch. If you author more content, the dialog continues, but since we only created a small dialog tree, we reach the end of the content quickly.

This test experience empowers you to quickly create and test a conversation to ensure that the conversation flows as anticipated. If the dialog doesn't reflect your intention, you can change the dialog, and save it. The latest content is pushed into the test copilot, and you can try it out again. The published version of the copilot doesn't change, so feel free to play around with your content until you're happy with it.

Publish your copilot - web app

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

1. If you are on the page for any one topic, select any of the menu items at the top of the page.

The screenshot shows the Copilot Studio interface with the 'Topics' tab selected. The main content area displays a 'Trigger' card with a 'Phrases' section containing the phrase 'I want to do an outdoor activity.'

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

The screenshot shows the Copilot Studio interface with the 'Copilots' tab selected. A green success message is displayed: 'Your copilot was published at 2:44 PM on 5/9/2024! Your users will see the new content soon.'

3. At the top of the page, select **Channels** and then **Demo website** under **Channels**.

The screenshot shows the Microsoft Copilot Studio interface. On the left is a vertical sidebar with icons for Home, Create, Copilots (which is currently selected), Library, and three dots. The main area has a header with the title 'copilot-5' and navigation links for Configuration, Knowledge, Actions, Topics, Analytics, and Channels (which is underlined and highlighted with a red box). Below this is a message about personalized extensions with a 'Learn more' link. A 'Publish status' section indicates the copilot was published by Christopher Chew on 5/9/2024 at 3:28 PM. The 'Channels' section, also highlighted with a red box, allows users to configure channels to meet customers where they are. It lists several options: Telephony, Microsoft Teams, Demo website (which is highlighted with a red box), Skype, Slack, Telegram, Direct Line Speech, and Email.

4. On the **Demo Website** pane, enter a welcome message and some common trigger phrases.

The screenshot shows the Microsoft Copilot Studio interface. At the top, there's a navigation bar with 'Actions', 'Topics', 'Analytics', and 'Channels'. Below it, a message says 'Microsoft Copilot experience. [Learn more](#)'. On the left, there's a sidebar with a tree view showing 'soft' and 'Demo website' under 'Demo website'. The main area has sections for 'Welcome message', 'Conversation starters', and 'Share your website'. In the 'Share your website' section, a URL is displayed: <https://copilotstudio.preprod.microsoft.com/environments/dd0b075f-ca89-ec28-a5>. A red arrow points to the 'Copy' button next to the URL. At the bottom right are 'Save' and 'Cancel' buttons.

5. Copy the website URL and select **Save**.

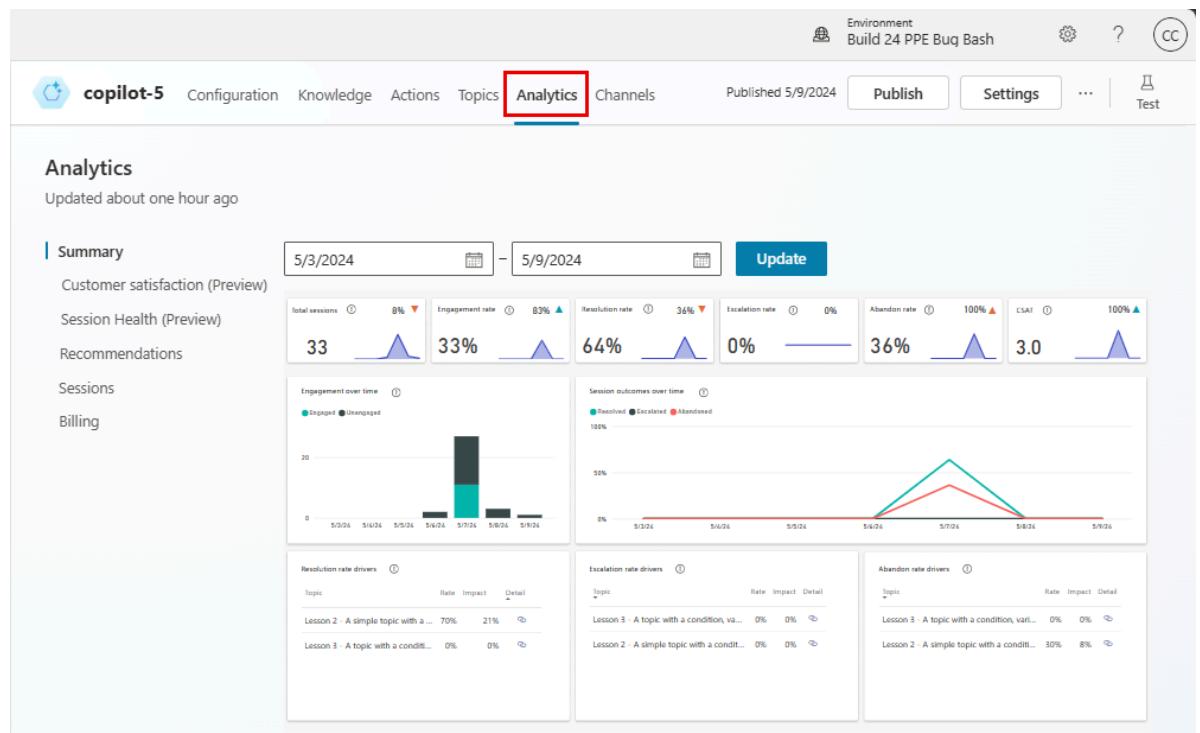
6. Send the URL to others to demonstrate it.

The webpage demonstrates what your copilot looks like to an end-user who comes to your webpage. The copilot canvas is at the bottom. You can interact with it by typing into the window or by selecting a starter phrase from the provided options.

For more information on publishing your copilot to other channels, see the documentation under [Publication key concepts](#).

Analyze the performance of your copilot

When your copilot completes interactions with users, the statistics are recorded in your copilot's analytics. To view analytics, select **Analytics** at the top of the page.



Here, you can find key performance indicators (KPIs) showing:

- The volume of sessions your copilot handled.
- How effectively your copilot was able to engage end-users and resolve issues.
- Escalation rates to human agents.
- Abandonment rates during conversations.

You find customer satisfaction information at the KPI level and in the **Customer Satisfaction** tab.

(!) Note

There is up to a 1-hour delay between when the conversations occur and when the statistics for those conversations appear in the analytics views. Also, all interactions with the copilot are logged in analytics, including interactions from your demo website, custom website, or test copilot.

You can also view detailed session history and transcripts by selecting **Analytics > Sessions**. Here, you can download a CSV file with the full session transcript. You can tune the performance of your copilot and change the content in your topics to improve your copilot's efficiency.

The screenshot shows the Copilot Studio interface with the 'Analytics' tab selected. On the left, there's a sidebar with icons for Home, Create, Copilots, Library, and three dots. The main content area has a title 'Analytics' and a subtitle 'Updated about one hour ago'. It includes sections for 'Summary', 'Customer satisfaction (Preview)', 'Session Health (Preview)', 'Recommendations', and 'Sessions'. The 'Sessions' section is highlighted with a red box. It features a date range selector from '5/2/2024' to '5/9/2024' with an 'Update' button. Below this, it says 'Download sessions from the past 28 days, up to 7 days at a time (starting at midnight UTC)'. A list of session download links is provided: '5/5, 8:00 PM EDT - 5/6, 7:59 PM EDT', '5/6, 8:00 PM EDT - 5/7, 7:59 PM EDT', '5/7, 8:00 PM EDT - 5/8, 7:59 PM EDT', and '5/8, 8:00 PM EDT - 5/9, 2:53 PM EDT'.

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

Conclusion

You created a copilot, created your own topic, tested it out, published it to a demo website, and learned how to analyze your copilot's flow and performance. Congratulations! Your copilot 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?

Yes

No

[Provide product feedback](#)

Community help and support for Microsoft Copilot Studio

Article • 06/06/2024

ⓘ 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.

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 [Copilot Studio training paths and modules](#), which include the [Create copilots 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 admin center interface. On the left, there's a navigation sidebar with several sections: Home, Environments, Advisor (Preview), Analytics, Billing, Resources, Help + support (which is selected and highlighted with a red box), Data integration, Data (preview), Policies, and Admin centers. The main content area has a header with 'Get Help + support', 'Export to Excel', and a search bar. Below the header, it says 'Help + support' and lists 'Support requests' and 'Known issues (Preview)'. A central message reads 'Welcome to the Power Platform support portal' and 'If you have questions or need support to resolve an issue, create a new support request. [Learn more](#)'. A prominent purple button labeled 'New support request' is centered below the message, also highlighted with a red box. In the bottom right corner of the main area, there's a 'Feedback' button with a plus sign icon.

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 ↗](#)

Copilot Studio security and governance

Article • 06/26/2024

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 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](#).

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](#).

The [Microsoft Trust Center](#) is the primary resource for Power Platform compliance information. Learn more at [Copilot Studio Compliance Offerings](#).

Furthermore, Power Platform has an extensive set of [Data Loss Prevention](#) features to help you manage the security of your data. Learn how to [configure Data Loss Prevention policies for copilots](#) in your organization.

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

- Disable copilot publishing:
 - Your admin can use the Power Platform admin center to turn off the ability to publish copilots with generative answers and actions for your tenant.

The screenshot shows the Power Platform admin center interface. On the left is a navigation sidebar with icons for Home, Environments, Environment groups, Advisor, Security, Analytics, Billing, Settings (selected), and Copilot. The main area is titled "Tenant settings" with the sub-instruction "These settings are applicable across your environments". A modal window titled "Publish bots with AI features" is open on the right. It contains the text "Allow Copilot authors to publish from Copilot Studio when AI features are enabled." with a link "Learn more", the instruction "You agree to the [supplemental terms](#) when this feature is enabled.", and a toggle switch labeled "Enabled" which is turned on. There is also a close button "X" at the top right of the modal.

- Disable data movement across geographic locations for Copilot Studio generative AI features outside the United States.
- Enable Copilot Studio conversational plugins.

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 • 06/21/2024

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.

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 multi-tenant Azure app registration on custom copilot creation. Does a multi-tenant Azure app registration for custom copilots pose any security risk?

No, multi-tenant Azure app registration does not pose any security risk.

Copilot Studio creates one app registration per custom copilot to identify it and enable secure communication with the channels and skills it may use. This app registration does not access or expose any customer data, resources, or any copilot 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 has [existed in Bot Framework and Azure Bot Service](#) since 2016.

- We are investigating moving to a single tenant Azure 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 that

allow authorized Microsoft Entra 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 does not 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 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 here](#)

Microsoft Copilot Studio creates service principal and certificates in customer's Azure Active Direct (AAD) tenant every time a custom copilot is created. What is the purpose of this and how are they governed?

To let custom copilots communicate with your data sources and services, Copilot Studio creates an application in your Azure Active Directory (AAD) 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, Microsoft Copilot Studio rotates the certificates on

a regular cadence. This means that the service principal gets a new certificate, and the old one is revoked. This process is automatic and does not require any action from you.

- Learn more about certificate rotation in Copilot Studio

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

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

- Learn more about Copilot Studio DLP policies
- 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 copilot security and privacy related to data used by copilots, data used by underlying models in copilots, 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.

- To learn more 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 capability—for example, who built a

custom copilot, are there any co-owners, are public endpoints for Generative Answers available, who configured or changed a copilot, and so on.

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 copilots in your organization. Use the Power Platform admin center to:

- [Make publishing of custom copilots available or unavailable](#), at the tenant level.
- [Control whether custom copilots 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 copilots, [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 copilots. For example, you want makers in your default environment to only upload files or use specific sites as knowledge sources when building custom copilots.

Copilot Studio offers granular control to enable or disable specific knowledge sources using data policies in Power Platform admin center. You can configure separate built-in data policies to control usage of SharePoint, OneDrive, 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 copilots 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?



[Provide product feedback ↗](#)

Configure data loss prevention for copilots

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.

✖ Caution

On May 21, 2024, we released new Data Loss Prevention (DLP) policy built-in connectors, with enforcement 30 days after the release.

Copilot makers might see DLP error banners and might not be able to publish copilots that use generative answers, depending on how DLP is configured in your organization.

These DLP policies cover controls for:

1. Knowledge sources with SharePoint and OneDrive in Copilot Studio
2. Knowledge sources with public websites and data in Copilot Studio
3. Knowledge sources with documents in Copilot Studio
4. Application Insights in Copilot Studio

This doesn't affect your customer's conversations with copilots.

What do I need to do to prepare?

Admins can change how these new connectors are governed by updating DLP policy to include the new connector in either the Business or Non-business data group. If the policy default group is set to Blocked, you can advise your copilot makers to either abstain from using these connectors or relocate them to the appropriate data group.

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 copilots for your end users. You can connect your copilots with many data sources and services. Some of these sources and services might be external, third-party services, and might even include social networks.

It is 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 copilots 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 copilot makers.

Important

By default, DLP enforcement for copilots is disabled in all tenants. Learn about [enabling enforcement](#).

The connectors need to be in a single data group as data can't be shared among connectors that are in different groups.

The following Copilot Studio connectors are available in the Power Platform admin center.

Name ↑ ↓
 Application Insights in Copilot Studio
 Chat without Microsoft Entra ID authentication in Copilot Studio
 Direct Line channels in Copilot Studio
 Facebook channel in Copilot Studio
 Knowledge source with SharePoint and OneDrive in Copilot Studio
 Knowledge source with documents in Copilot Studio
 Knowledge source with public websites and data in Copilot Studio
 Microsoft Teams channel in Copilot Studio
 Omnichannel in Copilot Studio
 Skills in Copilot Studio

These connectors can be configured for DLP as follows:

[] Expand table

Connector name	Description
Application Insights in Copilot Studio	Block copilot makers from connecting copilot with Application Insights .

Connector name	Description
Chat without Microsoft Entra ID authentication in Copilot Studio	<p>Block copilot makers from publishing copilots that aren't configured for authentication.</p> <p>Copilot users will require authentication to chat with the copilot. See Example: Require end-user authentication for copilots for more details.</p>
Direct Line channels in Copilot Studio	<p>Block copilot 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.</p>
Facebook channel in Copilot Studio	<p>Block copilot makers from enabling or using the Facebook channel.</p>
Knowledge source with SharePoint and OneDrive in Copilot Studio	<p>Block copilot makers from publishing copilots configured with SharePoint and OneDrive as a knowledge source. Supports DLP connector endpoint filtering to allow or deny endpoints.</p>
Knowledge source with public websites and data in Copilot Studio	<p>Block copilot makers from publishing copilots configured with public websites as a knowledge source. Supports DLP connector endpoint filtering to allow or deny endpoints.</p>
Knowledge source with documents in Copilot Studio	<p>Block copilot makers from publishing copilots configured with documents as a knowledge source.</p>
Microsoft Teams channel in Copilot Studio	<p>Block copilot makers from enabling or using the Teams channel.</p>
Omnichannel in Copilot Studio	<p>Block copilot makers from enabling or using the Omnichannel channel.</p>
Skills with Copilot Studio	<p>Block copilot makers from using skills in Copilot Studio copilots. See Example: Use DLP to block skills in Copilot Studio copilots and Example: Use DLP to block HTTP requests from Copilot Studio copilots for more details.</p>

Example DLP policy configurations

To help you get started with Copilot Studio copilot governance, we created the following examples that detail different scenarios:

- [Example: Use DLP to require end-user authentication for copilots](#)
- [Example: Use DLP to block SharePoint and OneDrive knowledge source in copilots](#)
- [Example: Use DLP to block Power Platform Connectors in copilots](#)
- [Example: Use DLP to block HTTP requests in copilots](#)
- [Example: Use DLP to block skills in copilots](#)
- [Example: Use DLP to block copilot publish to Channels](#)

Use PowerShell to enable and administer DLP enforcement for copilots in your organization

You can configure whether DLP policies should be applied to your copilots with the [PowerAppDlpErrorSettings](#) and [PowerVirtualAgentsDlpEnforcement](#) PowerShell cmdlets.

You can:

- Confirm if DLP is enabled for copilots in your tenant.
- Enable or disable DLP in an auditing mode (`-Mode SoftEnabled`) so copilot 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, which will show DLP enforcement errors, and prevent copilot makers from publishing DLP-affected bots or configuring DLP-related settings.
- Exempt specific copilots from DLP enforcement.
- Add and update the learn-more and contact email links that are shown to copilot 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:

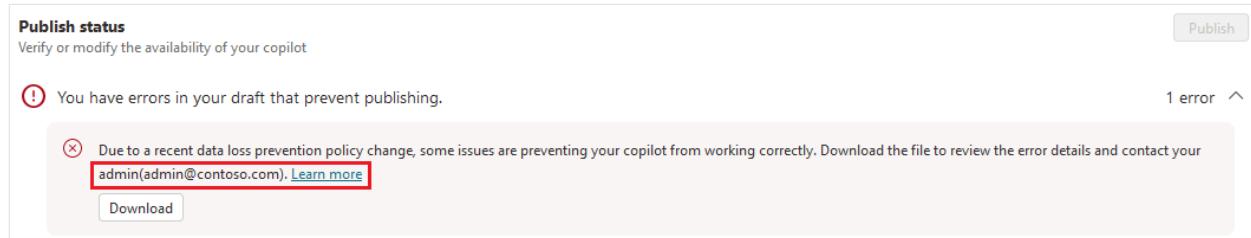
1. Add or update the learn-more and admin contact email links that are shown in DLP errors for copilot makers.
2. Determine which (if any) copilots currently have DLP policy enforcement enabled.
3. Use auditing or "soft" mode so makers can see DLP errors in the Copilot Studio web and Teams apps.
4. Mitigate risk by contacting makers and informing them about the best course of action for their app or flow.

5. Enable DLP enforcement for copilots to prevent DLP-affected tasks and features.

You may also decide to exempt one or more copilots from DLP policy enforcement, depending on the copilot'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 copilot makers will see this information when they experience DLP errors.



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 copilots

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 copilots 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 copilots

By default, DLP enforcement for copilots 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

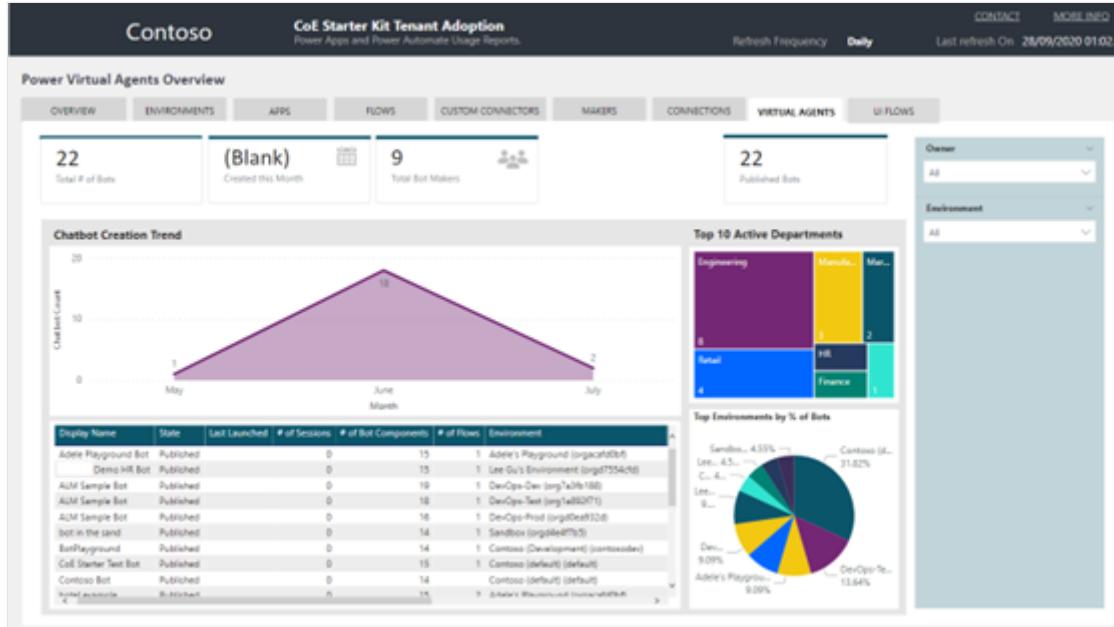
Run the following PowerShell script to enable DLP policies in auditing mode. Copilot makers will see DLP-related errors when configuring copilots in the Copilot Studio web and Teams apps, but they won't be blocked from performing DLP-related actions. They can also publish copilots as usual.

PowerShell

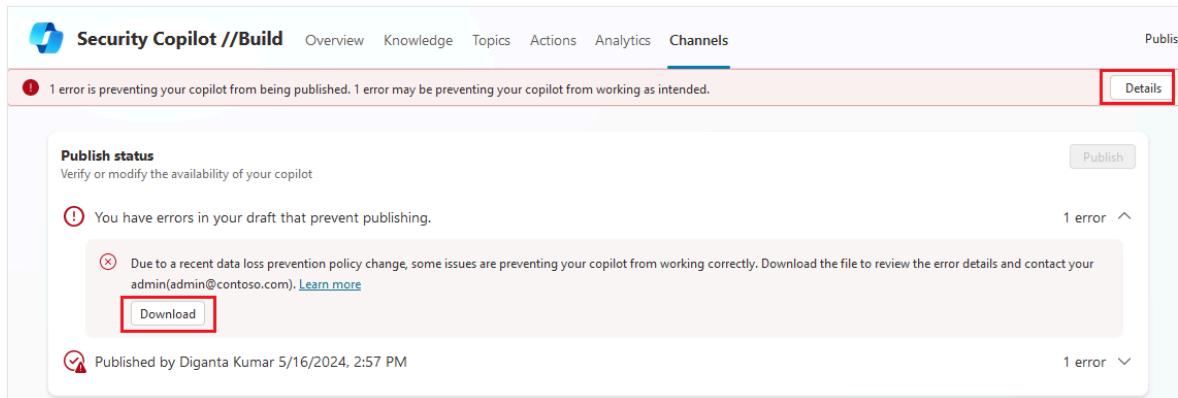
```
Set-PowerVirtualAgentsDlpEnforcement -TenantId <tenant ID> -Mode SoftEnabled
```

To find copilots 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 copilots in your organization. Go to the Copilot Studio overview page on the CoE Dashboard to see the copilots and environment names in your organization.



2. Run a campaign with the copilot makers in your organization to address DLP errors or updated DLP policies. You can download all copilot DLP errors by selecting **Details** in the error notification banner and selecting **Download** from the error message details.



Enable DLP enforcement for copilots

Important

Before enabling DLP enforcement, ensure you know which copilots will show errors to your copilot users due to DLP policy violations.

If you run into issues, you can exempt a copilot from DLP policies or disable DLP enforcement while your makers fix the copilot to comply with DLP policies.

You can run the following PowerShell command to enforce DLP policies in Copilot Studio. Copilot makers will be prevented from performing DLP-impacted actions, and end 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 a copilot from showing DLP errors to makers and end-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 copilot you want to exempt.

Tip

You can find the `<environment ID>` and `<bot ID>` from the copilot'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 copilots

The following command will disable DLP enforcement in copilots.

PowerShell

```
Set-PowerVirtualAgentsDlpEnforcement -TenantId <tenant ID> -Mode Disabled
```

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Data loss prevention example - Require end-user authentication in copilots

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.

When you create a new copilot the **Authenticate with Microsoft** authentication option is turned on by default. The copilot automatically uses Microsoft Entra ID authentication for Authenticate with Microsoft without requiring any manual setup and only lets you chat with your copilot on Teams. However, copilot makers in your organization can select the **No authentication** authentication option to allow anyone with the link to chat with your copilot.

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 copilot makers from configuring and publishing copilots that aren't configured for authentication to help

prevent data exfiltration.

Copilot makers will need to [configure end-user authentication with Authenticate with Microsoft or Authenticate manually](#) in Microsoft Copilot Studio if you use this connector in an enforced DLP policy.

See the [Configure data loss prevention for Microsoft Copilot Studio copilots](#) topic for information about other DLP-related connectors.

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:
 - a. If you want to create a new policy, select **New policy**.
 - b. 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 Microsoft Copilot Studio connector.

The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with various options like Home, Environments, Analytics, Billing, Settings, Resources, Capacity, and Dynamics 365 apps. The main area is titled 'DLP Policies > Edit Policy'. Under 'Prebuilt connectors', a connector named 'chat without Microsoft Entra ID authentication in Copilot Studio' is listed. This connector is in the 'Blocked' group, which has 5 items. The 'More actions' menu for this connector is open, showing options: 'Move to Business', 'Block' (which is highlighted with a red box), and 'Configure connector'.

2. Select the connector's **More actions** menu (⋮), and then select **Block**.

This screenshot shows the 'More actions' menu for a connector named 'Chat without Microsoft Entra ID authentication in Copilot Studio'. The 'Block' option is highlighted with a red box. Other options in the menu include 'Move to Business' and 'Configure connector'.

3. Select **Next**.

4. If you're a tenant admin, or an environment admin for multiple environments, you'll see the **Scope** step. Choose one or more environments that your DLP policy will apply to.

The screenshot shows the 'Edit Policy' page in the Power Platform admin center. The 'Scope' step is currently selected. On the right, there's a 'Define scope' section with instructions: 'Choose the environments to add to this policy.' It includes a 'Learn more' link and three radio button options under 'I want to': 'Add all environments' (unchecked), 'Add multiple environments' (checked), and 'Exclude certain environments' (unchecked).

! Note

If the policy has a tenant scope, DLP policy will apply to all copilots.

5. Review your policy, then select **Update policy** to apply the DLP changes.

The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with various options like Home, Environments, Analytics, Billing (Preview), Resources, Help + support, Data integration, Data (preview), Policies, Data policies (which is selected), Billing policies, and Ask virtual agent. The main area is titled 'DLP Policies > Edit Policy' and shows a tree structure for a policy named 'My Policy'. The nodes are: Policy name (checked), Environments (checked), Connectors (checked), and Review (checked). To the right of the tree, there's a 'Review and update policy' section with details: Policy name (My Policy), Environments (Single environment), Connectors (0 Business, 966 Non-business, 1 Blocked), and Scope (Single environment). At the bottom right, there are 'Back' and 'Update policy' buttons, with 'Update policy' being highlighted by a red box.

Confirm policy enforcement in Microsoft Copilot Studio

You can confirm that this connector is being used in the DLP policy from the Microsoft Copilot Studio web app.

First, open your copilot from the environment where the DLP policy is applied.

If the policy is enforced, you'll see an error banner with a **Details** button. On the **Channels** page, expand **error** link and select the **Download** button to see details.

The screenshot shows the Microsoft Security Copilot //Build interface. At the top, there are navigation links: Overview, Knowledge, Topics, Actions, Analytics, and Channels. The Channels link is underlined, indicating it is the active page. In the top right corner, the word "Publis" is visible. A red box highlights the "Details" button in the top right corner of a red-bordered bar. The bar contains the text "1 error is preventing your copilot from being published. 1 error may be preventing your copilot from working as intended." Below this, a "Publish status" section is shown with a "Verify or modify the availability of your copilot" link and a "Publish" button (also highlighted with a red box). A detailed error message follows: "You have errors in your draft that prevent publishing." It includes a warning icon, the text "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](#)", a "Download" button (highlighted with a red box), and a "1 error" link. At the bottom, it shows "Published by Diganta Kumar 5/16/2024, 2:57 PM" and another "1 error" link.

In the details file, a row will appear describing the violation.

Content	Topic name	Subcomponent	Subcomponent type	DLP policy name	Policy id	DLP error type	Connector (data group)
Published	-	Chat without Microsoft Entra ID authentication	Copilot authentication	Contoso policy	251eefd3-5031-4d91-b67c-af281673e0f0	Connector blocked	Chat without Microsoft Entra ID authentication in Microsoft Copilot Studio

A copilot makers can contact their admins with the DLP download spreadsheet details to make appropriate updates to the DLP policy. Alternatively, the copilot maker can update the copilot 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 Microsoft Copilot Studio](#).

Authentication options that do not use Microsoft Entra ID authentication will not be selectable.

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 and OneDrive knowledge source in copilots

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.

You can use data loss prevention (DLP) policies to prevent copilot authors from connecting to data. Doing so can help prevent data exfiltration.

See [Configure data loss prevention for copilots](#) for information about other DLP policy configurations.

Configure DLP to block or configure endpoints using Knowledge source with SharePoint and OneDrive in Copilot Studio connector in Power Platform admin center

Select or create a policy

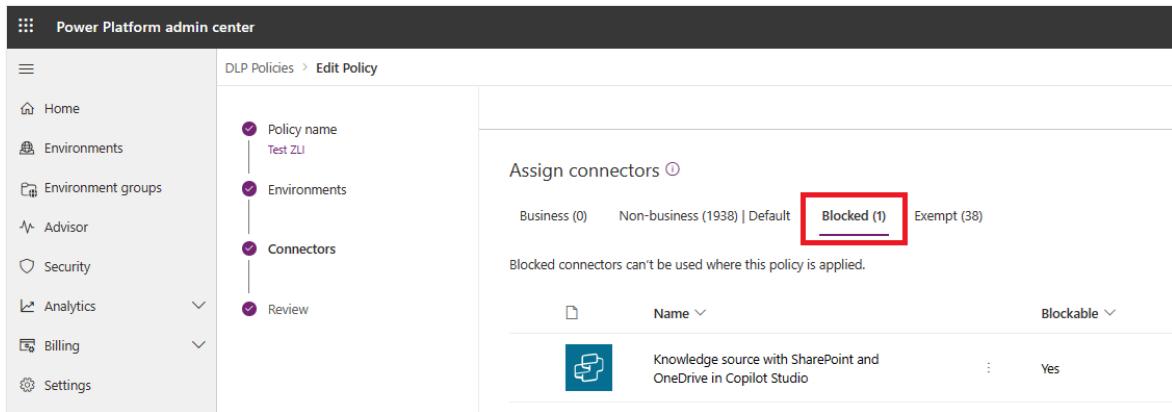
1. In the [Power Platform admin center](#), under Policies, select Data policies.
2. Create a new policy, or choose an existing policy to edit:
 - a. If you want to create a new policy, select New policy.
 - b. If you want to choose an existing policy to edit, select the policy and select Edit policy.
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.



The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with options like Home, Environments, Environment groups, Advisor, Security, Analytics, Billing, and Settings. The main area is titled 'DLP Policies > Edit Policy' and shows a tree structure with 'Policy name' (Test ZLI), 'Environments', 'Connectors', and 'Review'. Under 'Connectors', there's a table titled 'Assign connectors'. At the top of this table, there are four tabs: 'Business (0)', 'Non-business (1938) | Default', 'Blocked (1)' (which is highlighted with a red box), and 'Exempt (38)'. Below the tabs, it says 'Blocked connectors can't be used where this policy is applied.' The table has columns for 'Name' and 'Blockable'. There is one item listed: 'Knowledge source with SharePoint and OneDrive in Copilot Studio' with a 'Blockable' status of 'Yes'.

2. Select the connector's **More actions** menu (⋮), 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 Copilo Studio, they can use [DLP connector endpoint filtering](#) instead of blocking it.

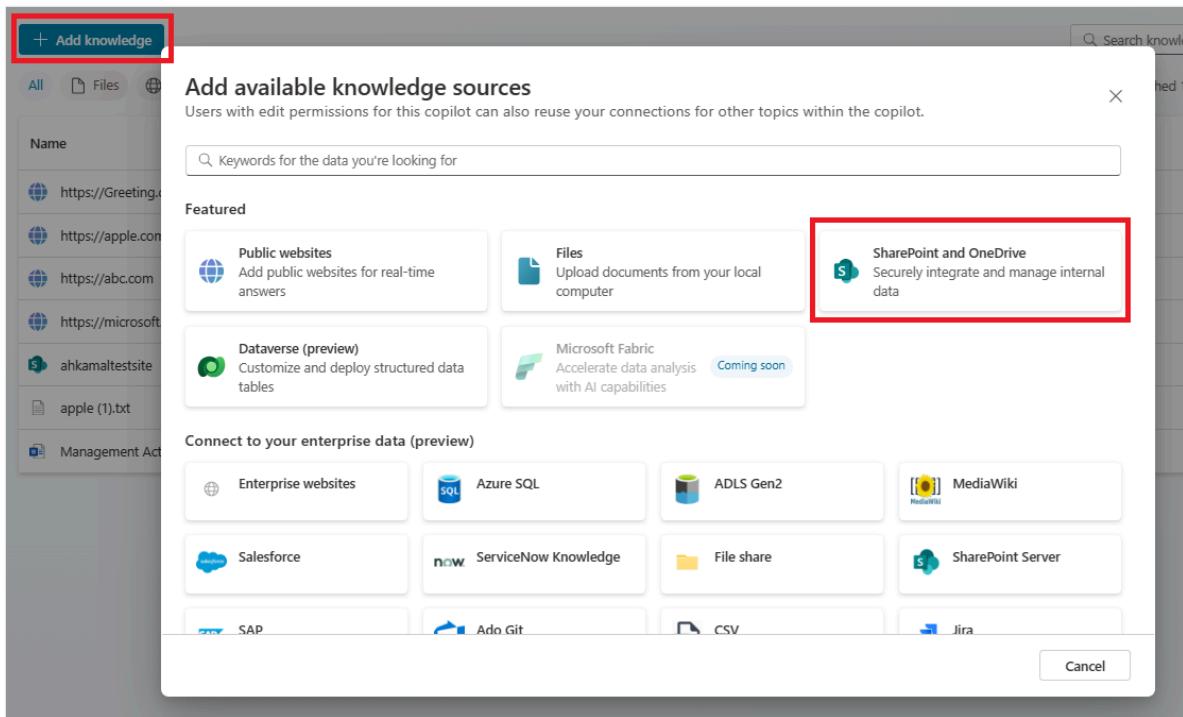
Order	Action	Endpoint	Move up	Move down
1	Allow	https://pvabugash.sharepoint.com...		
2	Deny	https://pvabugash.sharepoint.com...		
3	Deny			

Confirm policy enforcement

Web App

You can confirm that this connector is being used in the DLP policy from the Microsoft Copilot Studio.

First, open your copilot from the environment where the DLP policy is applied. Go to the **Knowledge** tab, select **Add knowledge**, add a **SharePoint and OneDrive** knowledge source.



If the policy is enforced, you'll see an error banner with a **Details** button after the knowledge is added. On the **Channels** page, expand **error** link and select the **Download** button to see details. **Published** button is disabled if there is a DLP violation.

The screenshot shows the 'Security Copilot //Build' interface with the 'Channels' tab selected. A red box highlights the 'Details' button in the error banner. The 'Publish status' section shows a red box around the 'Download' button in the error message. The message states: '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'. The 'Published by' section shows 'Published by Diganta Kumar 5/16/2024, 2:57 PM' with a red box around the '1 error' link.

In the details file, a row will appear for each violation. If a knowledge source has a DLP violation, a row will appear for the knowledge page and for each generative answers node that uses that knowledge source.

Content	Topic name	Subcomponent	Subcomponent type	DLP policy name	Policy id	DLP error type	Connector (data group)
Published	-	https://pyabugash.sharepoint.com/sites/ahkamaltestsite	Knowledge source endpoint	Test ZLU	251eef03-5031-4d91-b67c-a2f261673e0f0	Connector blocked	Knowledge source with SharePoint and OneDrive in Copilot Studio
Latest	-	https://pyabugash.sharepoint.com/sites/ahkamaltestsite	Knowledge source endpoint	Test ZLU	251eef03-5031-4d91-b67c-a2f261673e0f0	Connector blocked	Knowledge source with SharePoint and OneDrive in Copilot Studio
Latest	Greeting	https://pyabugash.sharepoint.com/sites/ahkamaltestsite	Knowledge source endpoint	Test ZLU	251eef03-5031-4d91-b67c-a2f261673e0f0	Connector blocked	Knowledge source with SharePoint and OneDrive in Copilot Studio

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Data loss prevention example - Block Power Platform Connectors in copilots

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.

You can use data loss prevention (DLP) policies to prevent copilot authors from configuring connectors. Doing so can help prevent data exfiltration.

See the [Configure data loss prevention for Microsoft Copilot Studio](#) article for information about other DLP policy configurations.

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:
 - a. If you want to create a new policy, select New policy.
 - b. 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.

The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with various options like Home, Environments, Advisor (Preview), Analytics, Billing, Settings, Resources, Help + support, Data integration, Data (preview), Policies, Data policies (which is selected and highlighted in purple), and Tenant isolation. The main area is titled 'DLP Policies > Edit Policy'. Under 'Policy name', it says 'Di'. Under 'Prebuilt connectors', there are four items: 'Custom connectors', 'Scope', 'Environments', and 'Review'. On the right, there's a section titled 'Assign connectors' with tabs for 'Business (0)', 'Non-business (1145) | Default', and 'Blocked (4)'. A note below says 'Blocked connectors can't be used where this policy is applied.' Below this is a table with columns 'Name' and 'Blockable'. It lists four entries:

Name	Blockable
Salesforce	Yes
MSN Weather	Yes
ServiceNow	Yes
Zendesk	Yes

2. Select the connector's **More actions** menu (⋮), and then select **Block**.

3. Select **Next**.

4. Review your policy, then select **Update policy** to apply the DLP changes.

DLP Policies > Edit Policy

Policy name
Diganta(do not delete)

Prebuilt connectors

Custom connectors

Scope

Environments

Review

Desktop flows in non-managed environments are not governed by DLP policies. [Learn](#)

Review and update policy

Policy name
Diganta(do not delete)
[Edit](#)

Prebuilt connectors
(0) Business, (1145) Non-business, (4) Blocked
[Edit](#)

Custom connector patterns
1 pattern(s) added
[Edit](#)

Scope
Add multiple environments
[Edit](#)

Environments
1 environment(s) selected
[Edit](#)

Back Update policy

Confirm policy enforcement

Web App

You can confirm that this connector is being used in the DLP policy from Microsoft Copilot Studio:

1. Open your copilot 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 **error** link and select the **Download** button to see details. **Published** button is disabled if there is a DLP violation.



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

! You have errors in your draft that prevent publishing.

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 ▼

In the details file, a row will appear for each violation. If a connector has a DLP violation, a row will appear for each connector.

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Data loss prevention example - Block HTTP requests in copilots

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.

Copilot 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 copilot makers from configuring HTTP requests to help prevent data exfiltration.

See the [Configure data loss prevention for Microsoft Copilot Studio copilots](#) topic for information about other DLP-related connectors.

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:
 - a. If you want to create a new policy, select **New policy**.
 - b. 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.

The screenshot shows the 'Edit Policy' page in the Power Platform admin center. The left sidebar shows 'Data policies' selected. In the main area, 'Connectors' is selected under 'Environments'. A search bar at the top right contains 'HTTP'. Below it, a table lists connectors:

Name	Blockable
http garden (Independent Publisher)	Yes
HTTP with Azure AD	Yes
HTTP	Yes
When a HTTP request is received	Yes

At the bottom are 'Back', 'Next', and 'Cancel' buttons.

2. Select the connector's **More actions** menu (⋮), and then select **Block**.

The screenshot shows the 'HTTP' connector details page. A context menu is open over the 'HTTP' connector, listing:

- Move to Business
- Block** (highlighted with a red box)
 - Block all requests to this connector
- Configure connector

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.

The screenshot shows the Power Platform admin center interface. On the left, there's a navigation sidebar with various options like Home, Environments, Analytics, Billing (Preview), Resources, Help + support, Data integration, Data (preview), Policies, Data policies (which is selected and highlighted in purple), Billing policies, and Ask virtual agent. The main area is titled 'DLP Policies > Edit Policy' and shows a tree structure for a policy named 'My Policy'. The 'Review' node is currently selected. To the right of the tree, there's a 'Review and update policy' section with fields for Policy name (set to 'My Policy'), Environments (selected), Connectors (0 Business, 966 Non-business, 1 Blocked), and Scope (Single environment). At the bottom right of this section is a 'Back' button and a prominent 'Update policy' button, which is highlighted with a red border.

Confirm policy enforcement

Web App

You can confirm that this connector is being used in the DLP policy from the Microsoft Copilot Studio.

First, open your copilot 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 then Save your Topic.

If the policy is enforced, you'll see an error banner with a **Details** button. On the **Channels** page, expand **error** link and select the **Download** button to see details.

 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.

Publish status
Verify or modify the availability of your copilot

! You have errors in your draft that prevent publishing.

(X) 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

Publish

1 error ^

1 error ↴

In the details file, a row will appear describing the violation. A violation will occur 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 is not blocked but an endpoint is denied.

Content	Topic name	Subcomponent	Subcomponent type	DLP policy name	Policy id	DLP error type	Connector (data group)
Published	Http Node	https://www.contoso.com/	Http request	ContosoDLP	e1ad689c-9e7c-4325-b608-ca45ccc65a961	Connector blocked	Http
Latest	Http Node	https://www.contoso.com/	Http request	ContosoDLP	e1ad689c-9e7c-4325-b608-ca45ccc65a961	Connector blocked	Http

Feedback

Was this page helpful?

 Yes  No

Provide product feedback 

Data loss prevention example - Block skills in Microsoft Copilot Studio copilots

Article • 03/26/2024

ⓘ 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.

Bot makers in your organization can [extend their bots with skills](#). Skills can be a useful way of extending the functionality of your bots, however you may want to restrict their use to help prevent data exfiltration.

You can use the Skills with Microsoft Copilot Studio connector in Power Platform data loss prevention (DLP) policies to stop bot makers from adding skills to their bot.

See the [Configure data loss prevention for Microsoft Copilot Studio chatbots](#) 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:
 - a. If you want to create a new policy, select **New policy**.
 - b. 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

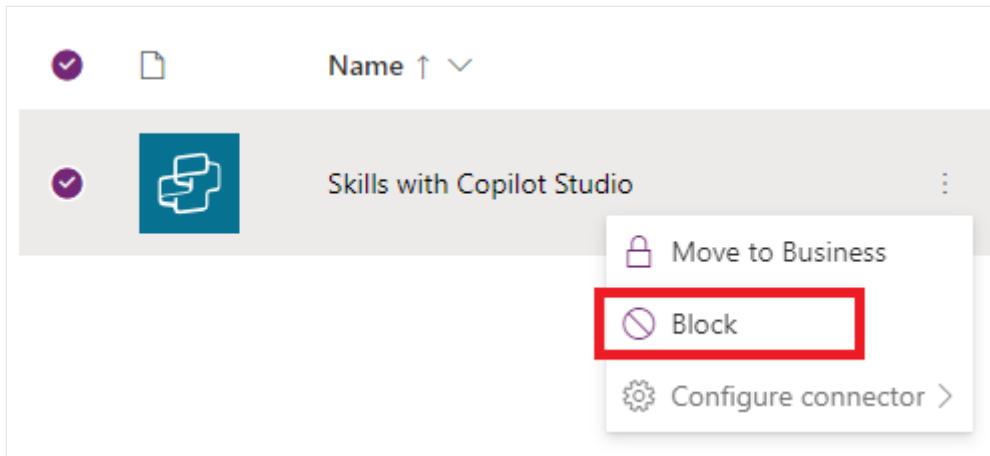
1. Use the search box to find the Skills with Microsoft Copilot Studio connector.

The screenshot shows the 'Power Platform admin center' interface. On the left, the navigation menu includes 'Home', 'Environments', 'Analytics', 'Billing (Preview)', 'Resources', 'Help + support', 'Data integration', 'Data (preview)', 'Policies', 'Data policies', 'Billing policies', and 'Ask virtual agent'. The 'Policies' section is currently selected. In the center, under 'DLP Policies > Edit Policy', the 'Connectors' section is highlighted. A search bar at the top right contains the text 'Skills with Powe...'. Below it, a table lists a single connector: 'Skills with Power Virtual Agents (preview)'. The entire row for this connector is highlighted with a red box. At the bottom, there are 'Back', 'Next', and 'Cancel' buttons.

2. Select the connector's More actions menu (⋮), and then select Block.

The screenshot shows the 'Power Platform admin center' interface. The navigation menu is similar to the previous screenshot. The 'Policies' section is selected. In the center, under 'DLP Policies > Edit Policy', the 'Prebuilt connectors' section is highlighted. A table lists a single connector: 'Skills with Copilot Studio'. The entire row for this connector is highlighted with a red box. At the bottom, there are 'Back', 'Next', and 'Cancel' buttons.

3. Select Next.
4. Review your policy, then select Update policy to apply the DLP changes.



Confirm policy enforcement in Microsoft Copilot Studio

You can confirm that this connector is being used in the DLP policy from the Microsoft Copilot Studio web app.

First, open your bot from the environment where the DLP policy is applied, and then try to [add a skill to the bot](#).

If the policy is enforced, you'll see an error in the **Add a skill** pane. The error says "Skill not allowed by data loss prevention policy" and suggests you contact an admin to add the skill to the allowlist. The **Skill manifest URL** field will also be highlighted with the message "This skill couldn't be validated. Review details and try again".

The screenshot shows the "Add a skill" interface. On the left, there's a decorative graphic of a robot and various icons. Below it, text explains what skills are and how to add them. On the right, there's a form with a "Skill manifest URL" field containing a URL. Below the field, an error message states: "This skill couldn't be validated. Review details and try again". Underneath, a pink box contains an error message: "Skill not allowed by data loss prevention policy" followed by a note: "Contact your admin (newadmin@contoso.org) to add your skill to the allow list." There's also a link "Learn more".

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Data loss prevention example - Block Channels to disable copilot publish

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.

You can use data loss prevention (DLP) policies to block copilot from publishing to the channels. Copilot Studio supports four channels - Direct Line, Team, Omnichannel, and Facebook.

Block all channels to disable copilot 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 main area is titled 'DLP Policies > Edit Policy'. A tree view on the left shows 'Policy name: Security Build Demo (do not delete)', 'Prebuilt connectors', 'Custom connectors', 'Scope', 'Environments', and 'Review'. Under 'Scope', 'Environments' is selected. To the right, there's a section titled 'Assign connectors' with tabs for Business (0), Non-business (1657) | Default, Blocked (4), and Exempt (38). It says 'Blocked connectors can't be used where this policy is applied.' Below this is a table with columns for Name and Blockable. Four rows are shown, each with a connector icon and a 'Yes' under 'Blockable': Microsoft Teams channel in Copilot Studio, Direct Line channels in Copilot Studio, Facebook channel in Copilot Studio, and Omnichannel in Copilot Studio. The last three rows are grouped together by a red box.

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 copilots will see disabled Publish button and the 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 copilot 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 copilot cannot be published.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Export and import copilots using solutions

Article • 05/25/2024

ⓘ 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.

You can export and import copilots using [solutions](#) so you can move your copilots across multiple [environments](#).

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 copilots, another environment to test copilots for only a subset of users, and a production environment where you share copilots with customers.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4CsHI?postJs||Msg=true>

ⓘ Note

You can't export [topic-level or node-level comments](#) when you export a copilot.

Prerequisites

- A maker requires at least the System Customizer security roles to use this feature.
Learn more about [configuring user security in an environment](#).

Web app

ⓘ 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 copilot, see [Microsoft Copilot Studio Overview](#).

Things to know

- You can't export [topic-level or node-level comments](#) when you export a copilot.
- Removing a copilot from a solution doesn't remove its components from the solution. You should remove the components separately.

Warning

Don't remove unmanaged copilot components, such as copilot topics, directly from the Power Apps portal unless you've removed the copilot from the solution.

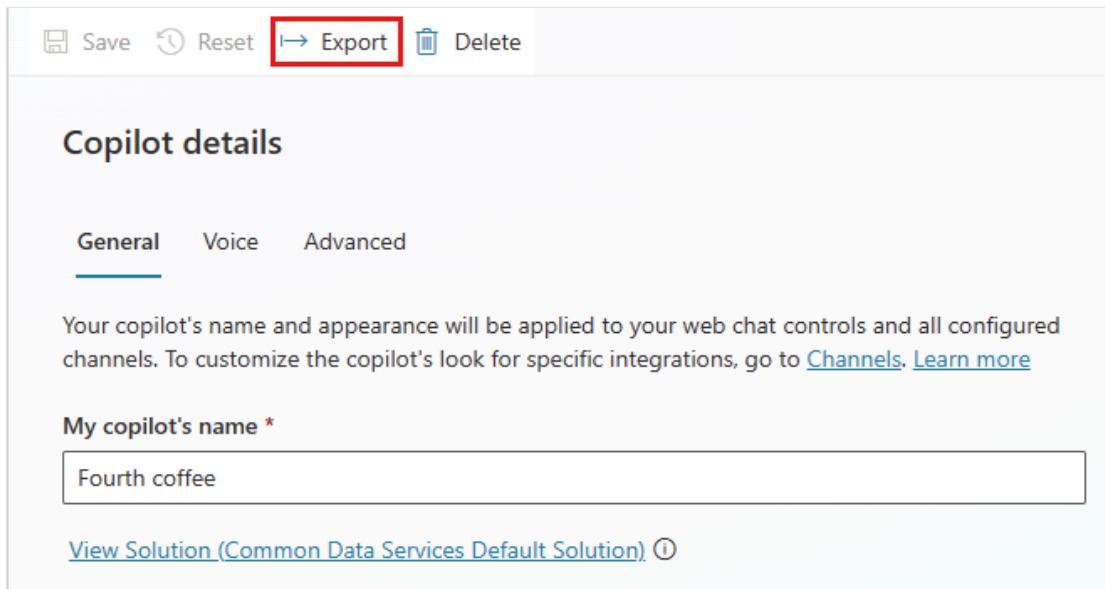
You should only make changes to topics from Copilot Studio. Removing or changing a copilot's components from Power Apps will cause the export and import to fail.

- 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 a copilot with periods (.) in the name of any of its topics.
- If your copilot has a large number of components (for example, more than 250 topics or more than 100 entities), export it using the classic Power Apps portal instead.

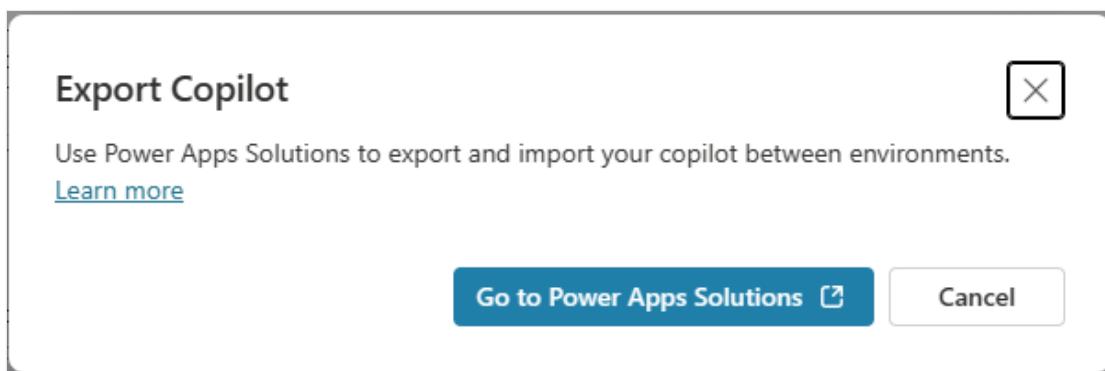
Create a solution to manage export and import

You use solutions to export copilots from one environment and import them into another. The solution acts as a "carrier" for the copilots. You can export and import multiple copilots in one solution.

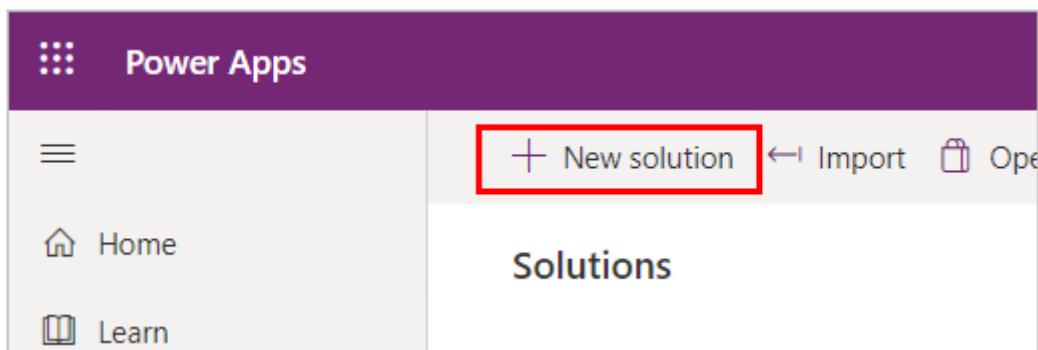
1. In Copilot Studio, open the copilot you want to export.
2. Go to Settings, select **Copilot details**, and then select **Export**.



3. Select **Go to Power Apps Solutions**.



4. In Power Apps, on the **Solutions** tab, select **New solution**.



5. Enter the following information for the new solution:

Field	Description
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 <i>before</i> 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.

6. Select **Create**.

Add your copilot to the solution

1. The solution you created should be open in Power Apps. If it isn't, select it in the list of solutions.
2. Select **Add existing**, and then select **Chatbot**.
3. In the **Add existing chatbots** list, select the copilot or copilots you want to export, and then select **Add**.

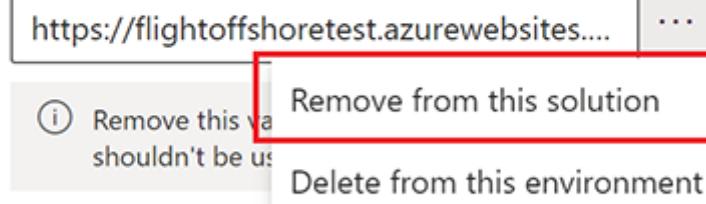
① Note

For a **Classic** chatbot:

If you're using skills, remove the skill's environment variable **Current Value** by selecting **Remove from this solution**.

Current Value

Override the default value by setting the current value for your environment.



Removing a copilot from a solution doesn't remove its components from the solution. You must remove the components separately.

⚠ Warning

Don't remove any unmanaged copilot subcomponents (such as copilot topics) directly from the Power Apps portal, unless you have removed the copilot itself from the solution.

You should only make changes to topics from within the Copilot Studio portal.

Removing or changing the copilot subcomponents from within Power Apps will cause the export and import to fail.

Export the solution with your copilot

You export and import copilots by exporting and importing the solutions that contain them from one environment to another.

ⓘ Important

If your copilot has a large number of components (for example, more than 250 topics or more than 100 entities), see [Export using the classic experience](#).

1. In the Power Apps **Solutions** page, select the solution that contains your copilot, and then select **Export solution**.

The screenshot shows the Power Apps Solutions page. On the left, there's a navigation bar with options like Learn, Apps, Create, Dataverse, Flows, Chatbots, AI Builder, Cards (preview), and Solutions. The 'Solutions' option is highlighted with a red box. On the right, there's a table of solutions. The first row, 'Contoso Bots', has its 'Select' checkbox checked and is also highlighted with a red box. The table columns include Display name, Name, and Created. There are five rows in total.

Display name	Name	Created
Contoso Bots	Contoso Bots	5 min
Publisher Publishers	Publisher Publishers	11 hours
Publisher Publishers	Publisher Publishers	1 week
Publisher Publishers	Publisher Publishers	1 month
Publisher Publishers	Publisher Publishers	1 month

2. In the **Before you export** pane, select **Next**.

3. Enter or select the following options:

[Expand table](#)

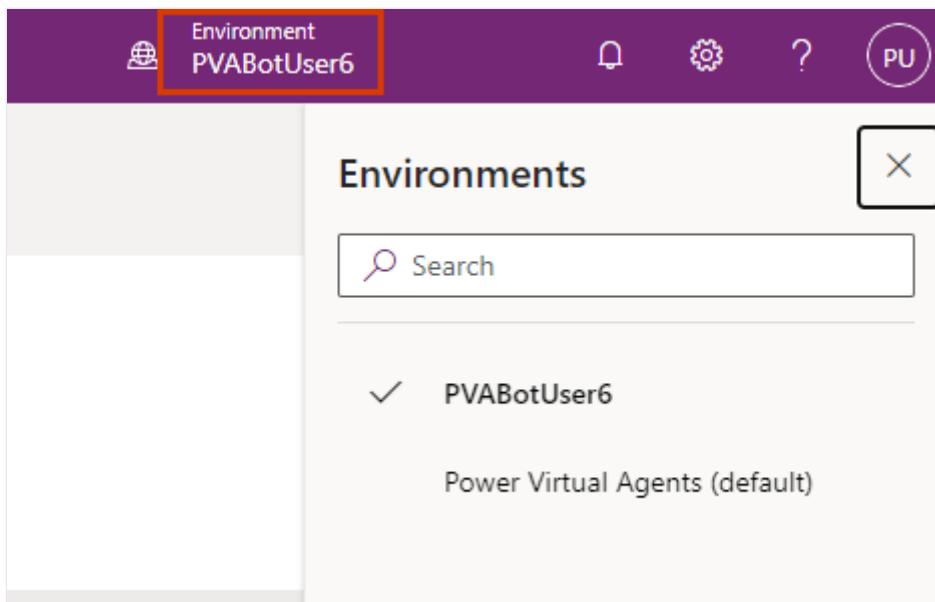
Option	Description
Version number	Copilot Studio automatically increments your solution version. You can accept the default or enter your own.
Export as	Select the package type, either Managed or Unmanaged . Learn more about managed and unmanaged solutions .

4. Select **Export**.

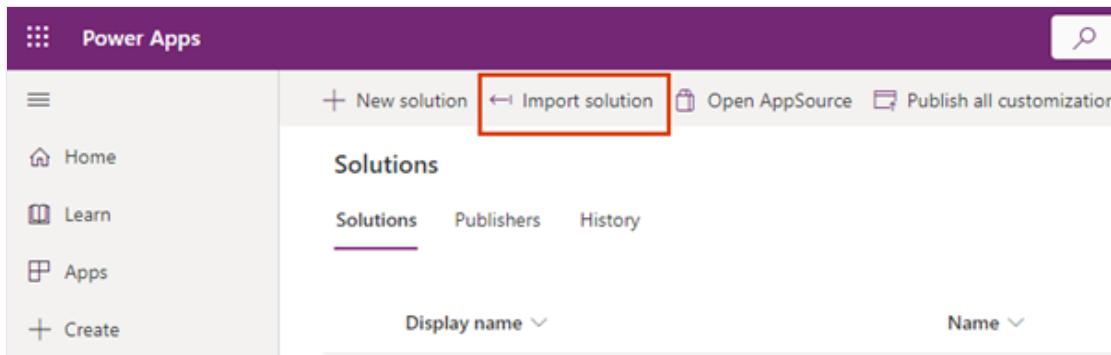
The export can take several minutes to complete. Watch the status message at the top of the **Solutions** page. When the export is finished, your browser downloads a .zip file. The file name is in the format `SolutionName_Version_ManagementType.zip`; for example, `MyBotPublisher_1.0.0.0.1_Unmanaged.zip`.

Import the solution with your copilot

1. In Power Apps, select the environment where you want to import your copilot.



2. In the **Solutions** page, make sure that no solutions are selected, and select **Import solution**.

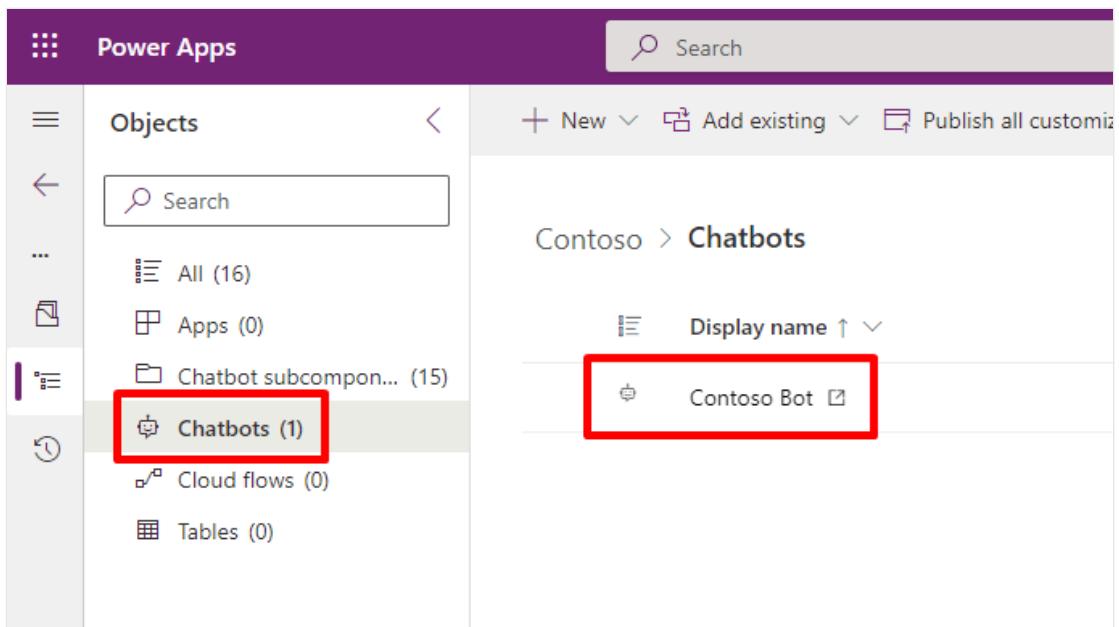


3. In the **Import a solution** pane, select **Browse**, select the .zip file that contains the solution with your copilot, and then select **Next**.
4. Select **Import**.

5. Wait a few moments while the import completes. View the results, and then select **Close**.

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 didn't contain some required components.

6. If your copilot has **user authentication** enabled, you'll need to configure it again.
7. In the **Objects** pane, select **Chatbots**. Then select the copilot's name to open the copilot in the Copilot Studio portal.



You can also open Copilot Studio directly and then open your copilot in the environment you imported it to.

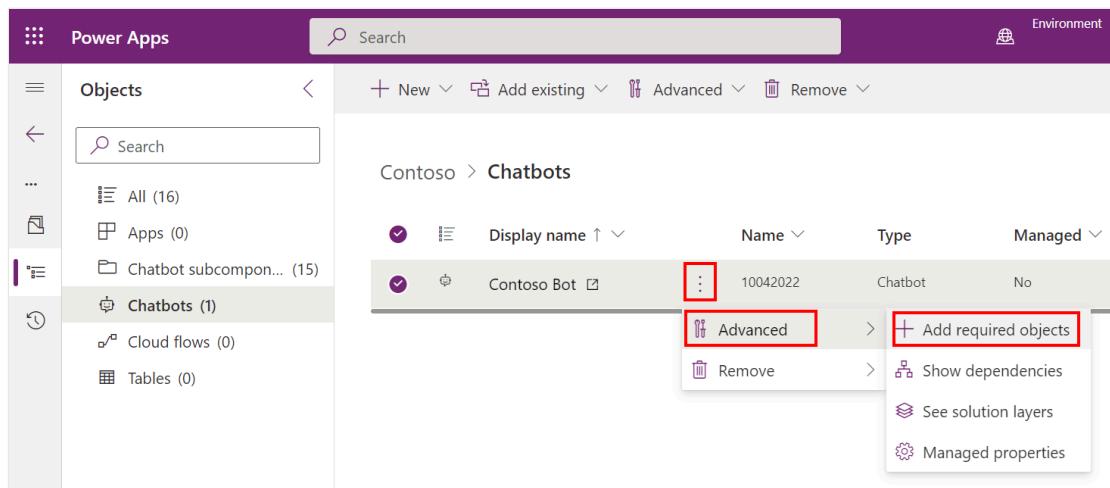
ⓘ Important

- You must [publish your imported copilot](#) before it can be shared.
- It may take up to 24 hours for your copilot's icon to appear everywhere.

Add components to a copilot in a custom solution

If you add components, such as topics or flows, to your copilot in Copilot Studio, you must also add them to the copilot in your unmanaged solution.

1. Open your solution in the Power Apps portal.
2. Select **Chatbots**, and then select your copilot in the list.
3. Select **Commands (:**), select **Advanced**, and then select **+Add required objects**.



Upgrade or update a solution with a copilot

Learn how to [upgrade or update a solution](#).

Remove an unmanaged layer from a managed copilot

Managed and unmanaged solutions exist at different levels in a Microsoft Dataverse environment. To learn more, go to [Solution layers](#).

When you edit a managed component, such as a topic or a flow, it gets an unmanaged "Active" layer. To show the latest updates after you deploy the solution, you'll need to remove that layer.

Use the **See solution layers** option to see all solutions that a component is a part of. You can also see one "Active" solution on top of all other solutions if you've changed something directly.

1. Open your solution in the Power Apps portal.
2. Select **Chatbots**, and then select your copilot in the list.
3. Select **Commands (:**), select **Advanced**, and then select **See solution layers**.

Display name	Name	Type	Managed...
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	Topic	🔓
CSAT Rating	Add required components	Flow	🔓
End of Conversation	Managed properties	Bot entity	🔓
Escalate	Show dependencies	Topic	🔓
Fallback	See solution layers	Topic	🔓
Get user name	Escalate	Topic	🔓
	Fallback	Topic	🔓
	Get user name	Flow	🔓

4. In the **Solution Layers** page, select the unmanaged layer, and then select **Remove unmanaged layer**.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#)

Regional settings and data locations for organizations using Copilot Studio

Article • 06/04/2024

ⓘ 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 supports multiple globalization scenarios, and lets you choose where you want your data to live.

Supported locales and formats

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 copilot 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**.

Copilot Studio can be deployed into Microsoft Azure datacenters (also referred to as "regions"). You can create a copilot in your tenant's location by default, or you can choose the datacenter you want to use when [setting up your environment](#).

Data locations

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

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.

Data is stored in the United States if a copilot author's tenant location isn't listed under the following **Data locations** table.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4mPmc?postJsIMsg=true>

 Expand table

Azure geographic locations	Azure datacenters (regions)
Asia Pacific	Southeast Asia (Singapore), East Asia (Hong Kong Special Administrative Region)
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)

Azure geographic locations	Azure datacenters (regions)
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 doesn't transfer customer data outside the selected Azure geographic location for Copilot Studio except when:

- The geographic location is South America.
- Microsoft must provide customer support, troubleshoot the service, or comply with legal requirements.
- Customers use services that are designed to operate globally, including:
 - Email used to send marketing messaging globally, as configured by the customer.
 - The Dynamics 365 home page, which stores application names, descriptions, and logos globally for performance.
 - [Microsoft Entra ID](#) (Microsoft Entra ID), which might store Microsoft Entra ID data globally.
 - [Azure Multi-Factor Authentication](#), which might store multi-factor authentication data globally.
 - Customer data collected during the onboarding process by the [Microsoft 365 admin center](#).
 - Services that provide global routing functions and don't process or store customer data. Services providing routing include Azure DNS, which provides domain name services that route to different regions. Other such services include preview, beta, or other pre-release services, which typically store customer data in the United States but might store it globally.

- 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.
- Customers configure external services to extend Copilot Studio. Such customer configurations might cause customer data to be transferred outside of the selected geographic location. Examples of customer configurable external services include:
 - [Customer Service Hand-off](#), Configurable external services that hand-off copilot escalations to a human agent.
 - Multi-channel, configurable to external channels such as Facebook and internal, non Copilot Studio services, such as Microsoft Teams.
 - [Facebook documentation](#)
 - [Microsoft Teams documentation](#)
 - [Dynamics 365 Customer Service Insights Topic suggestion](#), Topic suggestions from Customer Service Insights are automated into Copilot Studio with a single click. Data might go to a copilot in a different geographic region than the workspace region in Customer Service Insights.

Related content

- [Language support](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

View Copilot Studio audit logs

Article • 04/18/2024

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.

Changes to the content and settings of a copilot can affect security as well as its behavior. It's important to audit such actions that 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

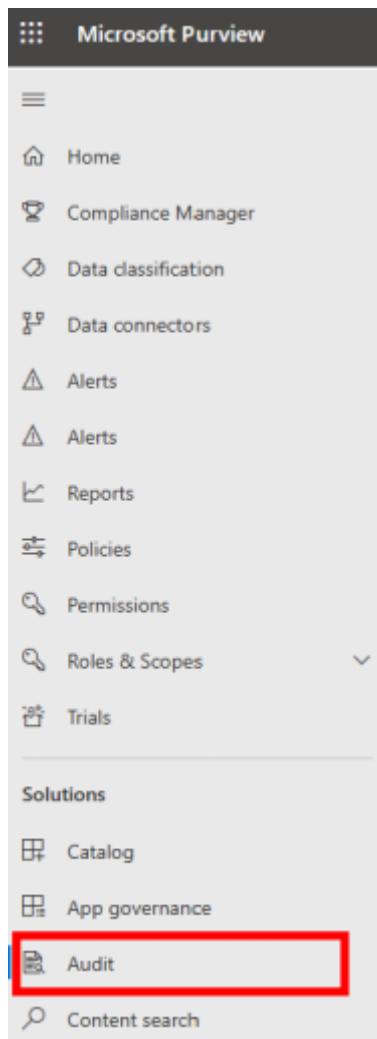
You can find logs of Copilot Studio activities in the [Microsoft Purview compliance portal](#). These logs are also accessible to developers via the [Office 365 Management API](#).

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 via the **Activities** list. You'll find the Copilot Studio activities in this list.

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
Copilots	BotCreate	The creation of a new copilot in Copilot Studio
Copilots	BotDelete	The deletion of a copilot in Copilot Studio

Category	Event	Description
Copilots	BotAuthUpdate	Updating the authentication settings of a copilot in Copilot Studio
Copilots	BotIconUpdate	Updating the copilot icon in Copilot Studio
Copilots	BotPublish	Publishing of a copilot in Copilot Studio
Copilots	BotShare	Sharing of a copilot to other users in Copilot Studio
Copilots	BotAppInsightsUpdate	Updating the App Insights logging configuration of a copilot in Copilot Studio
Copilot Component	BotComponentCreate	The creation of a component (such as a topic or skill) for a copilot in Copilot Studio
Copilot Component	BotComponentUpdate	The update of a component (e.g. topic, skill, etc.) for a copilot in Copilot Studio
Copilot Component	BotComponentDelete	The deletion of a component (e.g. topic, skill, etc.) for a copilot in Copilot Studio
AI Plugin	AIPluginOperationCreate	Creating an AI Plugin for a copilot in Copilot Studio
AI Plugin	AIPluginOperationUpdate	Updating an AI Plugin for a copilot in Copilot Studio
AI Plugin	AIPluginOperationDelete	Removing an AI Plugin for a copilot in Copilot Studio
Environment Variable	EnvironmentVariableCreate	Creating an environment variable for a copilot in Copilot Studio
Environment Variable	EnvironmentVariableUpdate	Updating an environment variable for a copilot in Copilot Studio
Environment Variable	EnvironmentVariableDelete	Deleting an environment variable for a copilot in Copilot Studio

Get details on 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 Copilot 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. Below 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 row logged.
Result Status	ResultStatus	Edm.String	No	Status of the row logged.
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 fields listed below.

[\[+\] Expand table](#)

Name	Type	Mandatory	Description
BotId	Edm.String	No	A unique identifier of the copilot
BotSchemaName	Edm.String	No	A unique string identifying the

Name	Type	Mandatory	Description
			copilot
BotUpdateDetails	Edm.ComplexType (Collection)	No	Details of the properties updated on the copilot
BotComponentId	Edm.String	No	A unique identifier of the copilot components such as topics, entities etc.
BotComponentSchemaName	Edm.String	No	A unique string identifying the components of copilot such as topics, entities etc.
BotComponentType	Edm.String	No	Type of the copilot component such as topics, entities etc.
BotComponentUpdateDetails	Edm.ComplexType (Collection)	No	Details of the properties updated of the copilot 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

Name	Type	Mandatory	Description
			associated with the copilot
EnvironmentVariableDefinitionId	Edm.String	No	A unique identifier for the environment variable definition associated with the copilot

See also

- [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 ↗](#)

Copilot connectors and actions overview (preview)

Article • 06/21/2024

ⓘ 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 is prerelease documentation and is subject to change.]

How does it work?

When you sign in to Copilot Studio, it determines which copilots you're allowed to access, based on the license associated with your credentials. The list of copilots available to you depends on these permissions. If you don't see the copilot you wish to extend, verify your credentials. Also try connecting to the copilot itself, or try to trigger any jobs that must occur to prepare the copilot for use. See the sections detailing the authoring experience for examples.

A maker starts by selecting the copilot they wish to extend in:

- Custom copilots
- Copilots created by Microsoft
- Other copilots that they have maker access to in their organization

Select a copilot to extend by selecting **Copilots** in the main menu and then choosing from the list of custom copilots and Microsoft Copilots.

View existing extensions for a copilot

After selecting a copilot, you can view the extensions currently available for the copilot within the configuration/authoring panel for the copilot.

The steps to view existing extensions for a copilot depends on the type of copilot:

- Custom copilot: Select **Actions**.
- Microsoft Copilot: Select **Extensions** or **Actions**, depending on which is shown for the copilot.

The list of available extensions appears.

Creating your own new extension

You can also create your own new extensions.

The steps depend on the type of copilot:

- Custom copilot: Select **Add an action**.
- Microsoft Copilot: Depending on what is shown for the copilot, select **Add extension** and then **New Action**, or select **Add Action**.

You can then select a type of action from a list of available options to start the process of extending the copilot with more capabilities.

Copilot Studio guides you through the configuration steps. The details of the steps depend on whether you're extending a custom copilot or Microsoft Copilot.

Detailed configuration steps are covered later in the documentation. After completion, the user can publish the action for the copilot. 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.

- **Microsoft Copilot action creation:** [Create conversational actions for Microsoft Copilot](#) and [Create AI actions for Microsoft Copilot](#)
- **Custom copilot action creation:** [Use plugin actions in Copilot Studio](#)

Copilot Studio offers multiple ways to make the extension process easy and comprehensive, including:

- Authoring: Author the extension for the copilot, which results in an action being created.
- Testing: Validate that the action works with your copilot 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.

- Provide behaviors: 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 copilot.
- Branding: Provide icons and branding information for your action for the purposes of distribution.

What happens after you publish your action?

Once an action is published, your copilot administrator has to review and enable the plugin for use. When an administrator enables a copilot, they must also determine which groups of users have permission to use it. The location for managing permissions might differ depending on the copilot. More information is available in the administration documentation. Once activated, a plugin appears in the environment where it was created. The plugin can also be exported for use in other copilots. Exporting creates a package that can be deployed in other environments, removing the 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 copilots as well.

For more information on certification, see the following documents:

- [Verified publisher certification process](#)
- [Submit your connector for Microsoft certification](#)

Extend a copilot by authoring an action

Copilot Studio provides a quick and easy way to author a copilot action using a step-by-step wizard to guide you through the process.

It can be used to extend Microsoft-provided copilots or custom copilots. Once you pick the copilot 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 actions you want to include in the plugin
- Define the input and output parameters for each 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 the 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 extension

Extensions created in Copilot Studio have a few components that are brought together to create the action for use in copilots.

- Handlers and content sources
- Authentication
- Actions
- Inputs and outputs
- Enhanced features (user consent, conversation starters, adaptive cards, branding, solution management)

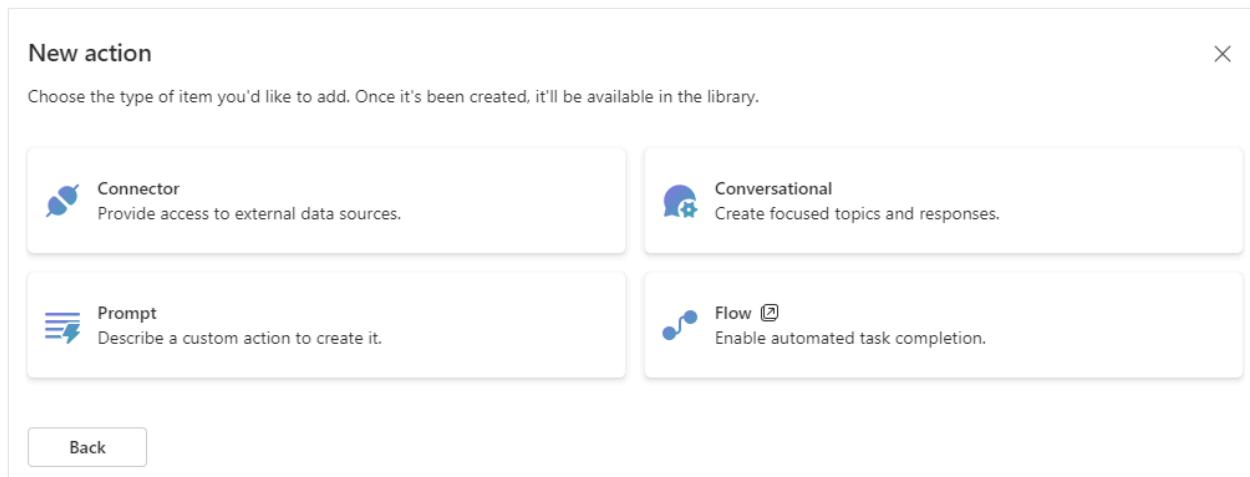
Handlers and content sources - action types

There are multiple input handlers and content sources available to serve as a basis for plugins. 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 copilot's behavior. You can create new abilities that aren't natively in Microsoft Copilot, like calling a specific API, applying sophisticated business logic, or accessing data sources that require configuration outside of Microsoft Copilot's access. These are configured similarly to a regular copilot topic built in the Copilot Studio canvas.
- **Connectors:** Connector actions are used within Microsoft Dataverse to connect to other data sources. When created, the connector includes specific actions that can

be used with the connector. It is possible to use Microsoft Certified connectors or custom connectors created within an organization or tenant.

- **Flows:** Microsoft Power Automate cloud flows can be called from a copilot to perform actions, retrieve information, and automate processes across other apps and data sources using connectors.
- **Prompts:** These 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.{{



Not every action type is available for every copilot:

[\[...\] Expand table](#)

Feature	Copilot for Microsoft 365	Copilot for Sales	Copilot in Dynamics 365 Customer Service
Conversational	Y	N	N
Connector	Y	Y	Y
Prompt	Y	N	Y
Flow	Y	N	N

In the documentation that follows, these action types are presented in two categories:

- Conversational actions: conversational action
- AI actions: connector, prompt, flow

Conversational actions

These actions are similar to [standard topics in Copilot Studio](#) that you use to create a functional copilot.

You create, configure, and publish conversational actions in a similar manner to topics. Afterwards, they're added to the action registry and available for use in Microsoft Copilot.

See [Create conversational actions for Microsoft Copilot \(preview\)](#) for details on how these plugins work and how to create and use them.

AI actions

These actions let you connect your copilot to data or perform activities. During this preview, you can use actions in Microsoft Copilot. You can't use actions in custom copilots that you build with Microsoft Copilot Studio.

You create and configure AI actions in Copilot Studio. You can also create and edit some types of AI actions in Power Apps.

The following table describes each type of AI plugin, with links to articles explaining where and how to create them:

[+] [Expand table](#)

AI plugin type	Description	Creation in Copilot Studio	Creation in Power Platform
Prompts (AI builder)	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.	Generate content or extract insights with AI Builder prompts	Create a custom prompt (preview) in Power Automate or Power Apps
Power Automate flows	Flows can be called from within a Microsoft Copilot chat that can perform actions or retrieve information across the end user's environment.	Create custom automation with Power Automate flows	Not available
Power Platform custom connectors	Custom connectors let your plugin 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, and are	Update or get answers about external data with connectors	

AI plugin type	Description	Creation in Copilot Studio	Creation in Power Platform
	routinely used by makers in their Power Apps and flows.		

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:

- Non-authenticated: 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 your data source, it has its authentication method built in to the connector. You will not configure it at setup but will be asked to supply 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 after it is created.
- 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 actions already defined, but when configuring, you can select which of the actions you wish to use. You can select from any number of actions to use with the source. For example, with multiple actions, you could perform the following:

- Retrieve a list of users
- Add a new user
- Update a user's details
- Delete a user

As long as the actions are present in the source, the handler can carry out any of the actions.

ⓘ Note

Every action is available to users of the extension. 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 plugin.

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 Copilot types.

Enhanced features include:

- Solutions
- User consent
- Conversation starters
- Adaptive Cards
- Testing

Not all enhanced features are available in every copilot 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

Select connector

Action details

Select Actions

Configure action

Select action parameters

Configure adaptive card

Review and test

Publish

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

User consent is a toggle provided on each action. It determines if the copilot will prompt users to continue when executing an action. Turning on this option means that the copilot will ask users if they're sure they want to take the action. This helps prevent unintentional actions that could affect data.

User consent required * ⓘ

No

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 plugin 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 plugin 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 plugin. Adaptive cards provide an alternative way of displaying results from a copilot query.

Update or get answer about external data X

✓ - ✓ - 3

Configure your Adaptive card (optional)

Upload a template or configure manually your Adaptive card

Get related activities

Root path ⓘ
\$.data.invoices

Adaptive card details

Upload custom Adaptive card template

Title
\$.id

Subtitle

→ Skip advanced configuration Back Next

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 a plugin. 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.

Root path

\$.MemberResponseModel

Select root path

✓ \$.MemberResponseModel

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 is how the references used in the query are displayed for more information.

It allows for inclusion of a title, URL, and subtitle. These are also dropdowns that can be populated based on the root path selection.

The screenshot shows a user interface for configuring adaptive cards. At the top, there is a dropdown menu labeled "Root path" containing the value ".DataSetsMetadataV2". Below it is another dropdown menu labeled "Title" with the placeholder "Select title". A scrollable list of options is displayed, including "Select title", ".\$tabular.source", ".\$tabular.displayName" (which is highlighted with a gray background), ".\$tabular.urlEncoding", ".\$tabular.tableDisplayName", and ".\$tabular.tablePluralName".

ⓘ 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 copilot. Any field left blank in the Default template doesn't appear on the adaptive card within the copilot.

Testing

For connector actions, a maker can test the new plugin inside of Microsoft 365 by sideloading the plugin created. This is an option on the review screen when the action has been authored. It will prompt the user to select or create a connection, and then create a side loaded test version of the plugin that is available in the M365 environment. This is not visible externally and is not published for admin approval. It is only available for the author so that it can be used only to validate that the plugin is working as expected.

Test your action

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.

Selected connection

Select a connection

+ New connection

 Test action

Use actions in Microsoft Copilot

End users in your tenant can use conversational and AI actions in their chats with Microsoft Copilot if:

- Your Microsoft 365 tenant admin deploys the Copilot Studio app in the Microsoft 365 admin center.
- The end user enables the connection from within their chat with Microsoft Copilot.

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 Microsoft Copilot.

Tenant admins can enable or disable:

- showing Conversational AI/topic actions and Prompt actions in the M365 Chat app using the "Copilot Studio" app.
- showing Flow actions in the M365 Chat app using the "Power Automate" app.
- Connector actions and API actions in the M365 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.

(i) Important

In this release, there are some limitations:

- Integration with Copilot Studio and Power Automate is enabled by default. That means, 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 end users. This might change in 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

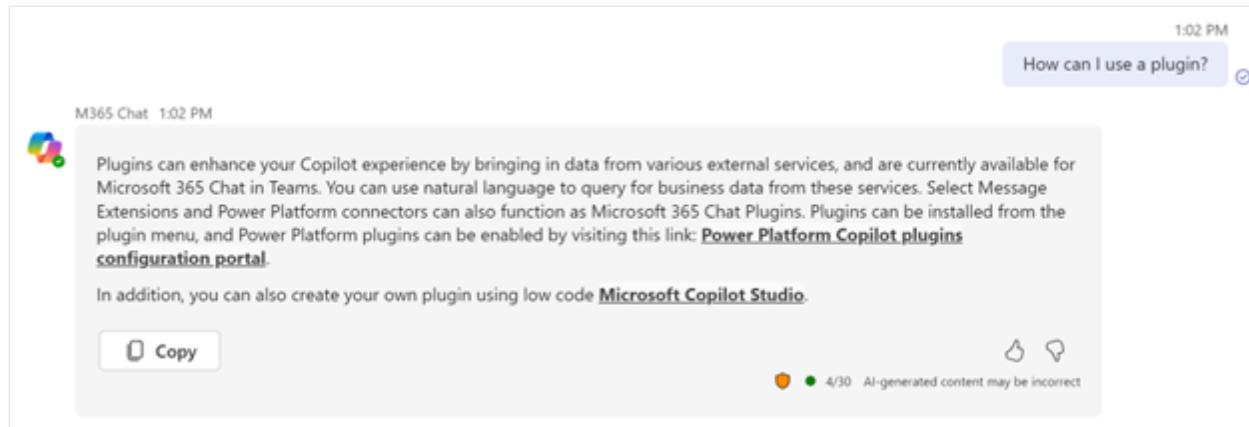
of Power platform environment (>100) and want specific Power platform environment to be enabled.

In the future, customers can use the Power Platform admin center to enable or disable plugins in an environment for Microsoft Copilot.

Enable the connection in Microsoft Copilot

Microsoft Copilot end 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.

They can get to the portal by asking about plugins in their chat with Microsoft Copilot. They receive a short summary about plugins 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 a plugin?*
- *Tell me about Power platform Copilot plugins*
- *How can I get data from an external system?*
- *How can I get data from Salesforce?*

Share AI actions

By default, actions are only visible and usable in Microsoft Copilot by the person who authored them.

However, the action author can share their plugins 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 Copilot	Extend a <i>Microsoft Copilot</i> by creating actions that can respond to a user's question with extra data.
Create AI actions for Microsoft Copilot (preview)	Extend a <i>Microsoft Copilot</i> by creating actions that perform tasks and query data across a user's environment with flows, AI Builder prompt templates, and connectors.
Use actions with custom copilots in Copilot Studio (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 ↗](#)

Copilot connectors in Copilot Studio

Article • 06/04/2024

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 is prerelease documentation and is subject to change.]

Copilot Studio makes it easy to extend the capabilities of copilots with actions, copilot connectors, and sources of data. You can take an existing copilot, for example 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 copilot 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 Graph connectors and Power Platform connectors, to expand the capabilities of copilots and copilot extensions. Using Copilot connectors, users can add knowledge from their enterprise systems to copilots through a no-code experience, and perform actions on these enterprise systems.

Copilot 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 Copilot connectors, you can now ground your copilot responses with multiple enterprise data sources for more relevant and reliable responses from copilots and copilot extensions.

- For details on all the supported knowledge sources via copilot connectors, see [Knowledge in Copilot Studio](#).

- For details on copilot connectors that have enterprise data added using Microsoft Graph connectors, see [Use graph connections with knowledge sources](#).

Requirements

- Maker access to the copilot 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 copilot to enable the extension once it's published

Related content

[] [Expand table](#)

Article	Description
Create conversational actions for Microsoft Copilot (preview)	Extend a <i>Microsoft Copilot</i> by creating actions that can respond to a user's question with extra data.
Create AI actions for Microsoft Copilot (preview)	Extend a <i>Microsoft Copilot</i> by creating actions that perform tasks and query data across a user's environment with flows, AI Builder prompt templates, and connectors.
Use actions with custom copilots (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 ↗](#)

Create conversational actions for Microsoft Copilot

Article • 06/22/2024

ⓘ 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.

ⓘ Note

Actions were previously known as Plugins. Some articles and screenshots may refer to Actions while we update documentation and training content.

Microsoft Copilot provides out of the box experiences to engage with content and resources from across your organization. In some situations, answers and interaction with external systems are required. With Microsoft Copilot Studio, you can author a conversational topic that can be published as a Copilot Action. Once your Tenant Admin approves the Action, it can be added to your organization's M365 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 M365 Chat.

ⓘ Important

You can't use conversational actions that you've created in Microsoft Copilot Studio as a [plugin action](#).

What is a copilot action and how does it work?

Microsoft Copilot's conversational system uses orchestration logic and large language models to help the user achieve their goals. This system interprets the user's request,

determining the best course of action, then activates the appropriate system, action, component, or series of components. Finally, a generated result is returned to the user.

A conversational action built in Microsoft Copilot Studio enables you to extend your app's behavior. You can create new abilities that aren't natively in Microsoft Copilot, like calling a specific API, applying sophisticated business logic, or accessing data sources that require configuration outside of Microsoft Copilot's access.

Unlike a traditional component model that creates a hard-coded integration point with the copilot, a maker in Microsoft Copilot Studio creates a conversational action. This maker gives descriptive text to the Microsoft Copilot's orchestrator about the action's abilities. The orchestrator can then make its own decision about when the action would be most useful and ask the user appropriate questions to fill action parameters.

Prerequisites

- End users must have a license to use Microsoft Copilot for Microsoft 365.
- Your Power Platform tenant must be configured to enable publishing actions, as described in the Copilot for Microsoft 365 article [Build plugins using Microsoft business applications](#).
- Microsoft Copilot Studio application must be a Microsoft Copilot Studio copilot, not a Power Virtual Agents "Classic" bot.
- Before end-to-end works, 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 and OneDrive sources for generative answers won't be functional.
- This release is limited to English language copilots.
- All users of Microsoft 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 Copilot is enabled by default. That means, 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 end users. This might change in 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.

Creating a basic conversational action to publish to Microsoft Copilot

Your topic used within Microsoft Copilot chat uses the System variable **Activity.Text** as an input that represents the user's question in Microsoft Copilot. The chat processes it, such as passing it to **Generative Answers**, and returns the result as one or more messages. These messages could include the response that comes from **Generative Answers** or **Send a 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 Microsoft Copilot Studio. You author basic capabilities and publish them to the Microsoft Admin Center. You can choose and enable the action in the UI.

1. From the left navigation pane in Copilot Studio, choose the **Copilots**.
2. In the Copilots list, select the Microsoft Copilot you wish to extend. The information page for the specific Copilot appears.
3. Select the **Actions/Extensions** tab.
4. The details here will depend on the specific Microsoft Copilot you want to extend.
 - If you see an **+ Add extension** button, select **+ Add extension** and then select **New action**.
 - If you see **+ Add action**, select **+ Add action**.
5. Select **Conversational** to create a conversational action. A **Create a conversational action** window appears letting you set some basic configurations for the conversational action.

6. Choose a name and primary language. Optionally, you can also select a solution and schema name to use.
7. Select **Create** to proceed. Your new conversational action will take a few moments to be created. When it is done, you are dropped into the conversational authoring canvas.
8. Select the **Topics** tab to edit your conversational action.
9. 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 Microsoft Copilot and Microsoft Admin Center respectively.
10. In the **Trigger** node's description, provide a clear description of how the conversational action can help the user and what it can do. Unlike other descriptions, this one has functional purpose and is used by the Microsoft Copilot to determine whether to invoke your action or not. Use clear sentences to describe what your action does for the user and what it returns.
11. In the conversational action, create any type of logic that doesn't require user interaction. There should be no buttons, questions, cards, or other prompts. The authoring canvas menu for conversational actions is limited to remove options that could require user interaction.
12. For this example, let's use the **Generative Answers** node to search through web content in a specified web site and provide an answer to the user in the Microsoft Copilot.
13. To use 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.
14. For the **Input** field, invoke the variable picker by selecting the arrow in the **Input** box.
15. 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 M365 Chat which caused Microsoft Copilot to invoke the action.
16. Select the **Data Sources** editor in the center of the **Create generative Answers** tile. A **Create generative answers properties** configuration window appears.
17. 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 bot. 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 a copilot](#).

 **Note**

Don't choose content that's not indexed by Bing, for example 301 redirects.

18. Once the link is added (don't forget to hit "+" once entering the URL), the properties panel can be closed
19. If you wish, you can send one or more messages to the user as a response. You can use a **Send a Message** node, or another **Generative Answer** node call, or multiple of these actions as your conversational action needs.
20. If your action needs to receive input parameters from a Connector and/or pass output parameters to a connector, you can [Add input and output variables](#)
21. Finally, select **Save** to save your conversational action.

Publishing your conversational action to Microsoft 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 in the Microsoft Copilot actions catalog.

In Microsoft Copilot, your conversational action is labeled as *Conversational action name* in the add actions UI. As such, specify your conversational action name and copilot name for readability.

To publish your conversational action:

1. First, create and save your conversational action as described in the prior section.
2. Press the **Publish** above the canvas to publish the conversational action.
3. Select **Publish**. A **Publish latest content?** dialog opens.
4. Finally select **Publish** again. The publish should complete quickly. The actual availability in the Microsoft 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 Admin Center](#) under **Settings**, then **Integrated Apps**. In this preview, Deploy/Block is at this category level only.

The screenshot shows the Microsoft 365 Admin Center interface. On the left, there's a navigation sidebar with various categories like Home, Users, Devices, etc. Under the 'Integrated apps' section, it says: 'Discover, purchase, acquire, manage, and deploy Microsoft 365 Apps developed by Microsoft partners. You can also deploy and manage Line-of-business add-ins developed within your organization.' Below this, tabs for 'Deployed apps', 'Available apps' (which is selected), and 'Blocked apps' are shown. A note states: 'All apps in this list are available for users in your organization to install according to the status shown in Availability Status.' Another note says: 'Any changes you make here will affect how these apps show up in Outlook, Microsoft 365 (formerly known as Office.com) and Microsoft Copilot. Apps installed for a user may be automatically used by Microsoft 365 Copilot to answer that user's prompt. Each app is provided under its own terms of use. Learn more.' To manage how these apps show up in Teams, it points to the Teams Admin Center. To view or edit policies for Outlook add-ins, it points to the Exchange Admin Center. To view or edit status for Word, Excel, and PowerPoint add-ins, it points to the Microsoft 365 Admin Center > Org Settings.

Integrated apps

Discover, purchase, acquire, manage, and deploy Microsoft 365 Apps developed by Microsoft partners. You can also deploy and manage Line-of-business add-ins developed within your organization.

For advanced management of these apps go to the respective admin center or page : Azure Active Directory | SharePoint | Teams | Add-ins

Deployed apps Available apps Blocked apps

All apps in this list are available for users in your organization to install according to the status shown in Availability Status.

Any changes you make here will affect how these apps show up in Outlook, Microsoft 365 (formerly known as Office.com) and Microsoft Copilot. Apps installed for a user may be automatically used by Microsoft 365 Copilot to answer that user's prompt. Each app is provided under its own terms of use. [Learn more](#)

To manage how these apps show up in Teams, go to the Teams Admin Center. To view or edit policies for Outlook add-ins, go to the Exchange Admin Center. To view or edit status for Word, Excel, and PowerPoint add-ins, go to Microsoft 365 Admin Center > Org Settings.

Refresh 2341 items Filter Search

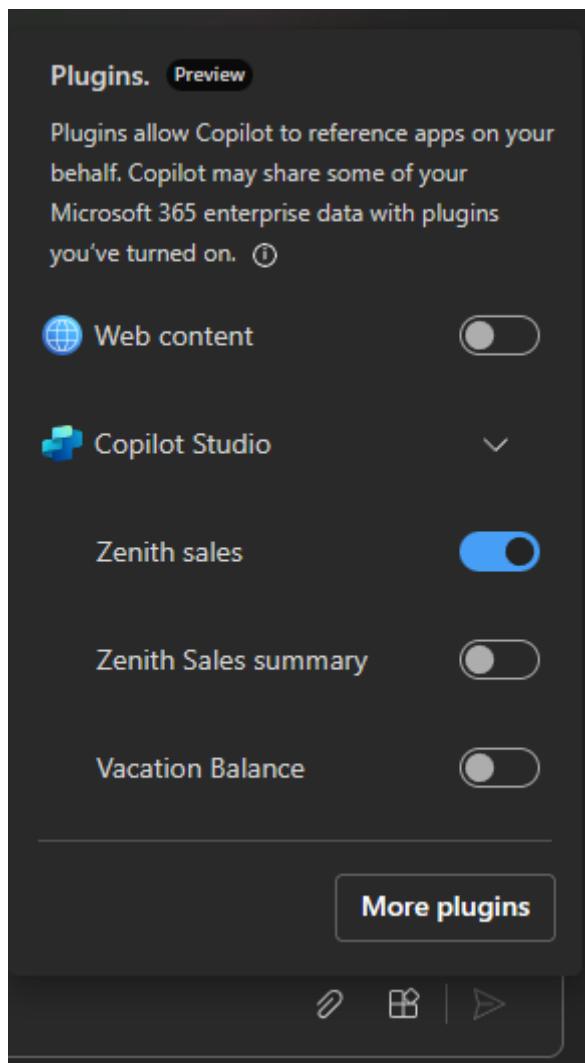
Name	Host products	Availability Status
1-on-1 Hub	Teams	Some or all users in the organization can install
1-1 Business Communicator	Teams	Some or all users in the organization can install
1&1 Business Phone	Teams	Some or all users in the organization can install
15Flow	Teams	Some or all users in the organization can install
1Page	Microsoft 365, Outlook, Teams	All users in the organization can install
1st Reporting	Teams	Some or all users in the organization can install
22Miles	Teams	Some or all users in the organization can install
260Learning	Teams	All users in the organization can install
365-QA	Teams	Some or all users in the organization can install

ⓘ Important

In this release, Copilot Studio integration with Microsoft Copilot is enabled by default. That means, 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 end users. This might change in 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.

- Once your Tenant admin approves the integrated app, it should appear in the user's list of plugins in their Microsoft Copilot UI.



Once enabled, your user can ask questions to Microsoft Copilot and see the plugin invoked and return 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 Copilot app catalog.

💡 Tip

Interacting with the Microsoft Copilot orchestrator is likely to be a different experience than you might have had integrating with previous systems.

- Microsoft 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 end user.

- 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 ↗](#)

Create AI plugin actions for a Microsoft Copilot (preview)

Article • 06/04/2024

ⓘ 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.

AI plugin actions can be used to extend a Microsoft Copilot. Alternatively, they can be used within a custom copilot as a plugin action.

You can create multiple types of AI plugin actions in Microsoft Copilot Studio:

- Prompt actions, which use AI Builder and natural language understanding to target the specific scenarios and workflows within your business.
- Flow actions, which use Power Automate flows to perform actions, and retrieve and work with data.
- Connector actions, which use Power Platform connectors to access data from other systems, such as popular enterprise products like Salesforce, Zendesk, MailChimp, and GitHub.

This article provides instructions for creating, sharing, and publishing plugin actions.

After you create plugin actions, [enable them for use in Microsoft Copilot](#), or [add them to a custom copilot](#).

The remainder of this article describes how to create each type of action.

Generate content or extract insights with AI Builder prompts

Note: Extending with prompt actions is only available for some types of Copilots.

To create prompt actions:

1. Sign in to [Copilot Studio](#) and select **Library** on the side navigation pane.
2. Select **+ Add an item**. You're prompted to select a Microsoft Copilot to extend.
3. Select one of the Copilots to extend.
4. For Copilot for Microsoft 365, you're prompted to choose between extending the Copilot with a **New copilot** or a **New action**. Choose **New action**.
5. A **New action** menu appears. Select **Prompt**.
6. Enter a **Name** for your prompt under **Prompt details**. Use natural language to describe what you want to achieve. For example, *Summarize this text as a paragraph with fewer than 50 words*. You can also select from several preset prompt templates to help you get started with a prompt. You can then customize the template for your scenario.
7. Under **Input**, add variables for any information you want to give your prompt as input. You can define multiple inputs if needed. You can also add sample data for each input variable. For example, if you want a prompt template to summarize a given text, define a *text* input variable and add a sample text to summarize as a test.
8. With the necessary variables defined, add an input variable under **Prompt** for each input that the copilot should provide when invoking the prompt action. You can add more than one variable to your prompt, if you intend to receive more than one piece of context data at runtime from the copilot.
9. You can also add dataverse table and columns as grounding data for your prompt response. By adding a table and its related tables, you can inform the prompt to use it as reference data to generate grounded responses.
10. To try specific sample data for your input variables, and click on **Test prompt** to evaluate how well the prompt works.
11. Select **Finalize prompt** to review the details you defined for the prompt.
12. Finally, select **Create prompt action** to save your prompt action.

After you create your action, you need to [enable it for use in Microsoft Copilot](#).

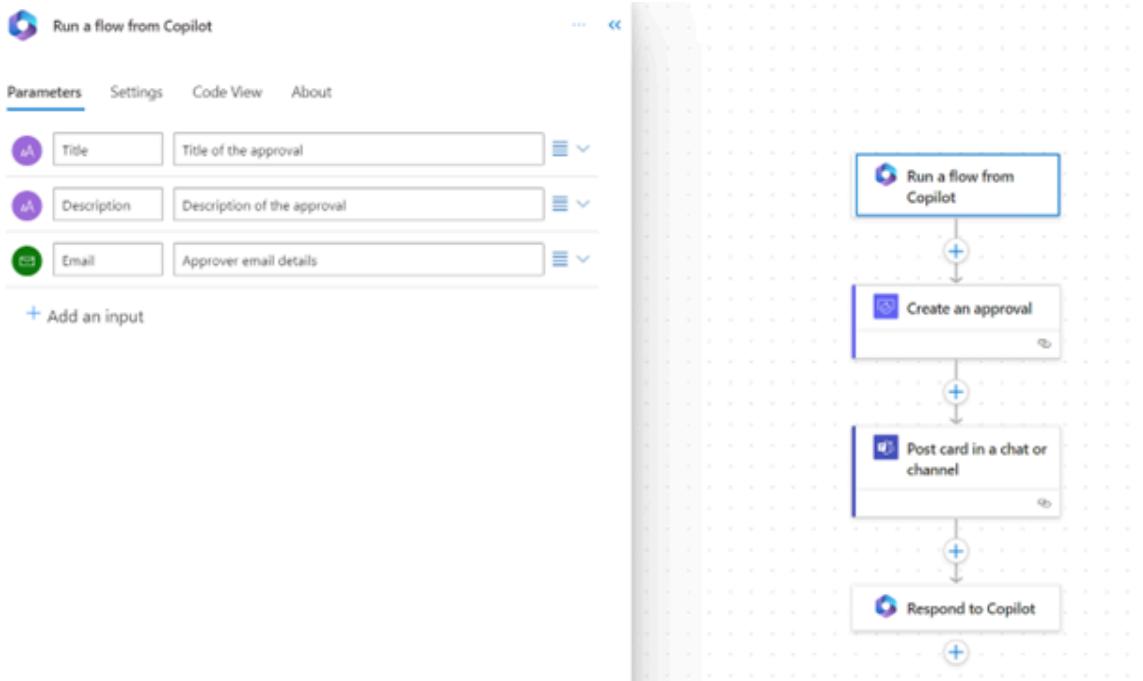
For more information about building and using prompt plugins, see [AI Builder prompts](#).

Custom automation with Power Automate flows

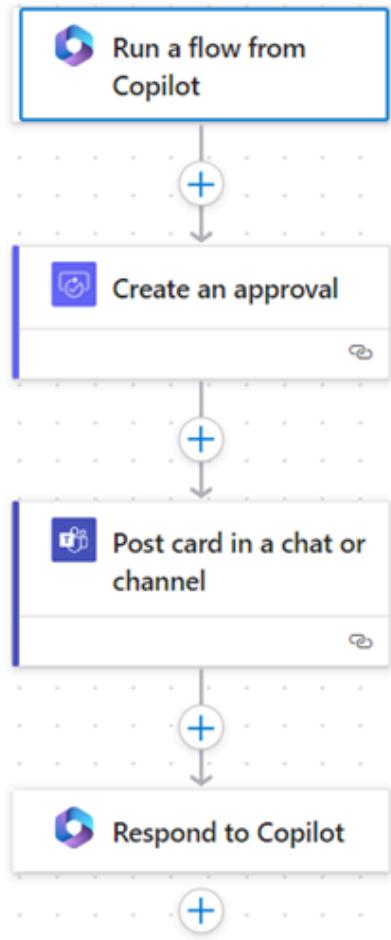
Power Automate flow plugins let you define flows that can be called from AI surfaces in Power Platform. Flow plugins use the new **Run from Copilot** trigger and **Respond to Copilot** action to define custom processes that can be invoked with natural language.

To create automation plugins:

1. Sign in to [Copilot Studio](#) and select Plugins (preview) on the side navigation pane.
2. Select Add a Power Platform component as an AI plugin.
3. Select Custom Automation.
4. The flow editor automatically opens with the Run 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 plugins. Ensure your flows run correctly, as only tested flows show up as available plugins in Microsoft Copilot.

After you create your action, [enable it for use in Microsoft Copilot](#).

Author a Connector action

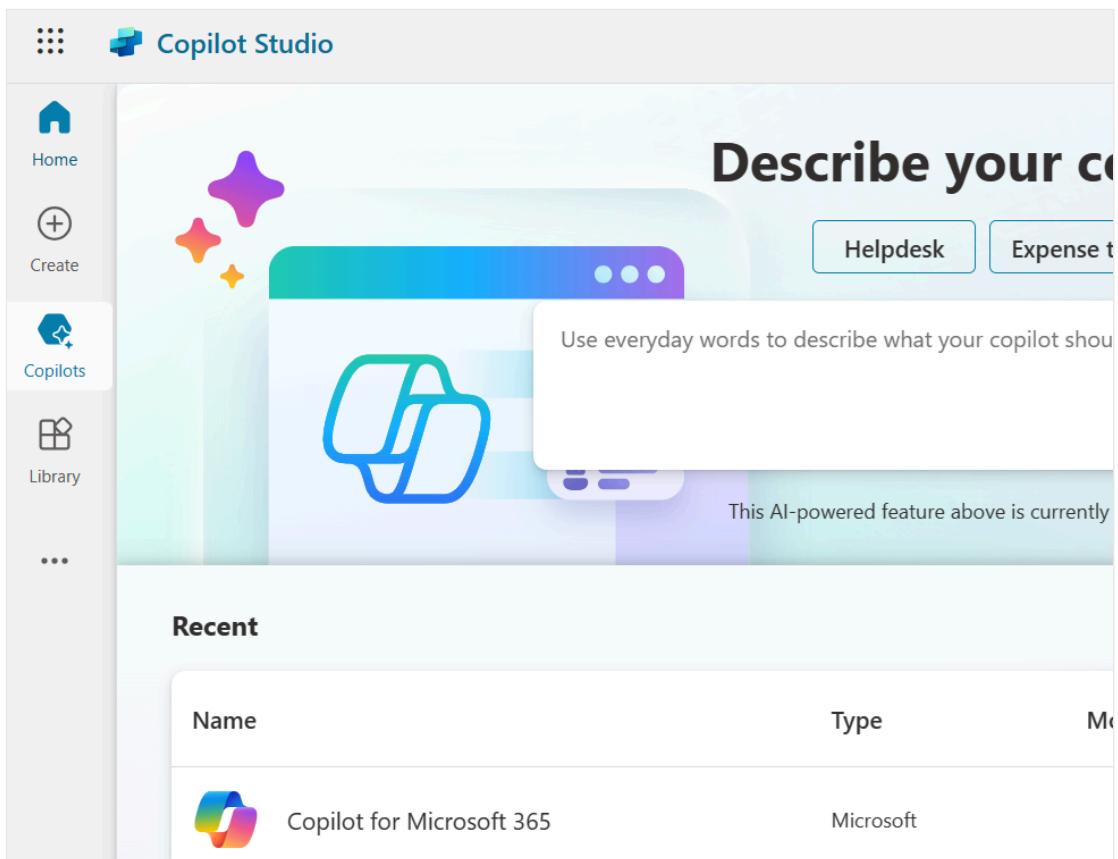
Connector actions let you define connectors that can be invoked from AI surfaces in Power Platform. For connector actions, you need to identify the actions enabled for use in Microsoft Copilot. You can also capture the information that large language models require to effectively identify and utilize the plugin, such as summary and description fields.

Adding a connector action for Copilot for Microsoft 365

You must have maker level credentials and be assigned a copilot license and a Microsoft Teams license in order to author a plugin action for Microsoft 365.

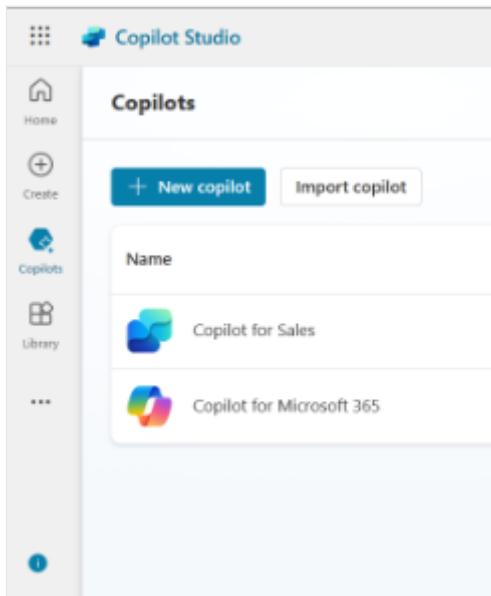
To add a connector action to Copilot for Microsoft 365:

1. On the home page, select **Copilots**.



The screenshot shows the Copilot Studio interface. On the left, there's a vertical navigation bar with icons for Home, Create, Copilots (which is selected and highlighted in blue), Library, and three dots. The main area has a large 'Describe your co...' input field with a placeholder 'Use everyday words to describe what your copilot should do'. Below it, a message says 'This AI-powered feature above is currently...'. A 'Recent' section lists a single item: 'Copilot for Microsoft 365' (Type: Microsoft). The top right of the screen shows 'Helpdesk' and 'Expense t' buttons.

2. On the Copilots list in the left hand navigation, select **Copilot for Microsoft 365**.



The Copilot for Microsoft 365 main page is displayed on the configure tab.

The image shows the 'Copilot for Microsoft 365' configuration page. The top navigation bar includes 'Copilots', 'Copilot for Mic...', 'Overview' (which is underlined), and 'Extensions'. On the left, there's a sidebar with sections for 'Custom copilots' (expanded to show 'Microsoft' which contains 'Copilot for Microsoft 365' and 'Copilot for Sales'), and 'Coming soon'. The main content area has a heading 'Copilot for Microsoft 365 is ready to use or customize' with three bullet points: 'See how adding extensions can add value to your service team', 'Add extensions to extend your copilot', and 'Enable extensions to make them available for copilot users to use'. At the bottom, there's a section for 'Extensions' with a '+ Add' button.

3. Select the **Extensions** tab. You can see a list of the actions already present within the Copilot.
4. To create a new action, select **Add Extension**, and then select **New action**.

The image shows a dialog box titled 'Extend Copilot for Microsoft 365 (preview)'. It asks 'How would you like to customize the way Copilot for Microsoft 365 responds?'. There are two main sections: 'New copilot' (with an icon of a speech bubble and a plus sign) and 'New action' (with an icon of a lightning bolt and a star). Both sections have descriptive text below them.

You're provided a list of all supported action types.

The screenshot shows a 'New action' dialog box with the following content:

- Connector**: Provide access to external data sources.
- Conversational**: Create focused topics and responses.
- Prompt**: Describe a custom action to create it.
- Flow**: Enable automated task completion.

A 'Back' button is located at the bottom left of the dialog.

5. Select **Connector**. You're provided a list of available connectors to select from. Connectors connect copilots to different services. The list includes all certified connectors offered by Microsoft and any custom connectors that are present on the current environment being used. A search bar is available to help find the connector that you wish to use.

6. Find your connector type using the search bar and select it.

The screenshot shows the 'Add a connector action' dialog box with the following interface:

- Left sidebar (Workflow steps):** Select connector, Action details, Select Actions, Configure action, Select action parameters, Review and test, Publish.
- Main area (Connector selection):**
 - Select your connector:** Select the connector you want to use to get information from external sources.
 - Search bar:** Search
 - List of connectors:** SharePoint, OneDrive for Business, Dynamics 365 (deprecated).

The **Action details** page is displayed.

7. 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

Action details

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 may be provided if present in the connector you selected. The description is used by the Copilot 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 the Mail Service example above, you may 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, set priority for messages..."

The more specific and detailed the better. The description lets the copilot know to use the plugin without being explicitly instructed, allowing for a more natural language experience.

- 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 auto-selected. You can either select a solution, or leave it blank. If you leave the solution blank, a solution is created for you with

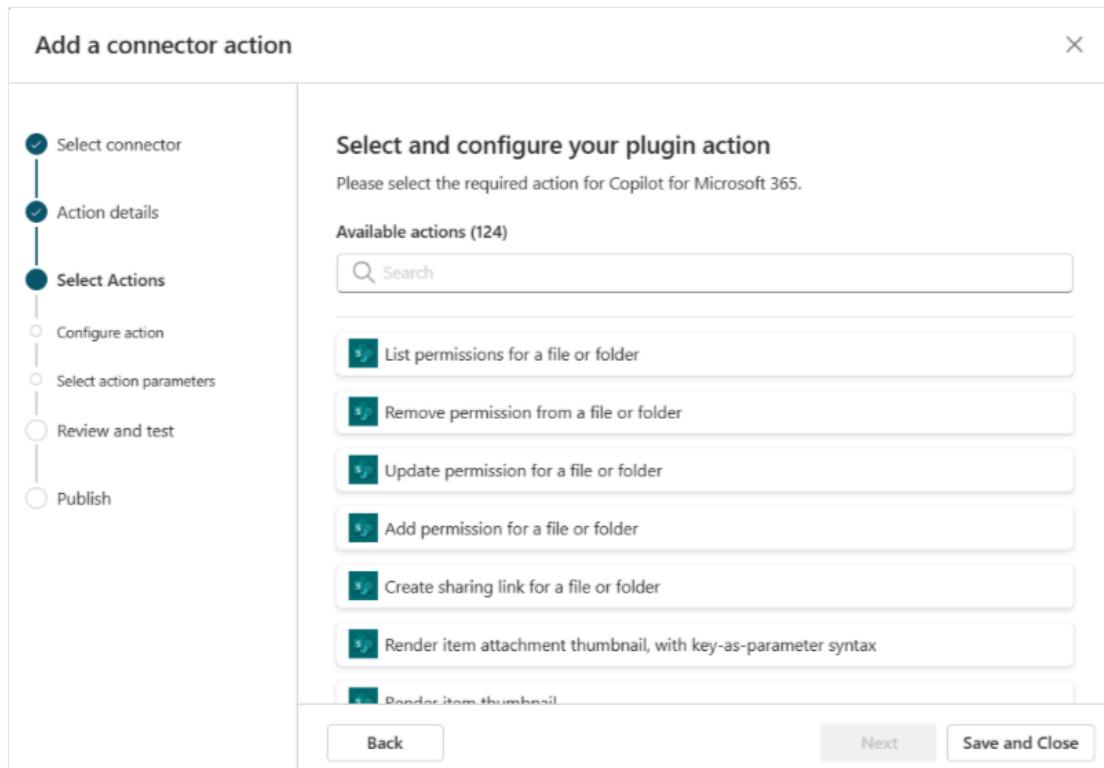
action name and default publisher. Storing your action in a solution lets you move it very easily across environments.

ⓘ 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](#).

9. Once the solution is chosen, select **Next**.

The **Select Actions** page is displayed. This page shows all the actions that are available to add for your connector. You can select and configure the actions you want your copilot to support. A search bar is included to help you find the action you wish to select. There is no limit to the number of actions you can select for the connector.



ⓘ Note

Connectors only execute the actions you specifically select and configure here. If an action isn't listed, then it is not available. It's not possible to add or remove actions from a connector from within the Copilot Studio actions wizard.

10. Using the search bar as needed, find and select an action to proceed. The **Configure Action** page is displayed.

The screenshot shows the 'Add a connector action' interface. On the left, a vertical navigation path is displayed with steps: 'Select connector', 'Action details', 'Select Actions', 'Configure action' (which is the current step, indicated by a solid blue dot), 'Select action parameters', 'Review and test', and 'Publish'. The right side of the screen is titled 'Configure your plugin action' and contains a description: 'Select the connector action you wish to leverage and enter details to configure your plugin action.' Below this, there is a section for 'Get lists' with fields for 'Action name *' (containing 'Get lists') and 'Action description *' (containing 'Gets SharePoint lists from a site.'). At the bottom, there are three buttons: 'Back', 'Next' (which is highlighted in blue), and 'Save and Close'.

11. 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 specific description so that the copilot 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.

12. When you're done, select **Next**. The **Select action parameters** page is displayed. All the inputs and outputs for the action are listed. The inputs and outputs list is provided by the connector and can't be changed. However, the descriptions can be modified. These descriptions have no impact on the copilot. They simply provide clarity for future makers who manage the action.

Add a connector action

Set your action's parameters
Define and configure your action's input and output

Get lists

Input name	Input description *
dataset	Example: https://contoso.sharepoint.com/sites/sitename

Output name	Output description *
Name	The name of the table. The name is used at runtime.
DisplayName	The display name of the table.
DynamicProperties	Additional table properties provided by the connector to the table.

Back **Next** **Save and Close**

13. Modify the descriptions as you like.

14. Select **Next** to proceed.

The **Select Actions** page is displayed showing the action you configured.

Add a connector action

Select and configure your plugin action
Please select the required action for Copilot for Microsoft 365.

Selected actions (1)

Get lists

Available actions (123)

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

Back **Next** **Save and Close**

15. 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

Review, test and publish your action

Almost done! Make sure you test your action and review its details before creating it.

Test your action

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.

Selected connection

Select a connection + New connection

Action details

Action name
SharePoint-Contoso Marketing

Action description

The **Review, test, and publish your action** screen displays the details of all of the configuration steps up to this point. You can review the configurations, and then test your action before publishing.

16. 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 such as username and password to build the connection.

17. When you are done, select **Test action**.

18. Once you see a message that the action is ready to test, go to **Copilot for M365 chat**, open the flyout to see the action, and toggle it on.

The action appears with a beaker icon under the chosen connector with format *Test-Action name*.

19. 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 Copilot Studio.

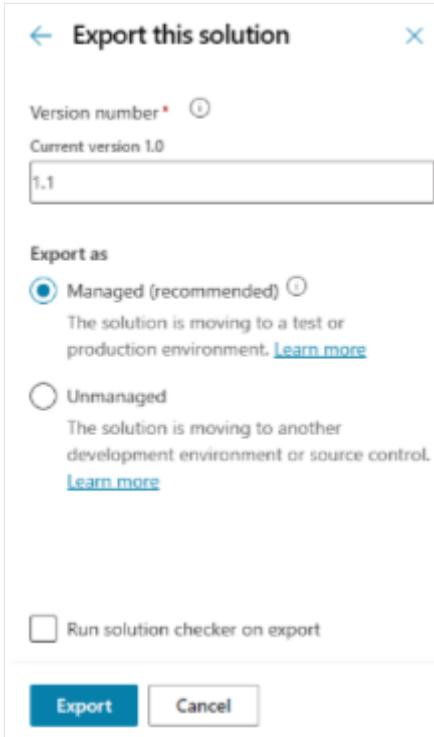
20. Once you are done testing, select **Publish**.

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.

21. Select **Go To Details Page** to see the details of the published plugin.

Now that your connector is published an administrator must enable it before it can be used in the copilot.

22. You can also easily move a connector from one environment to another using a solution. To do so select **Export this solution** on the details page. Exporting the solution lets you export the connector.



23. 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.

24. 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.

25. Select **Import Solution** in the Copilot main page in the target environment.

Import a solution

X

Environment

Build Sandbox testing

Select a file

Browse for the solution file to import.

[Browse](#)

No file chosen

26. Select the .zip file and proceed. You can see the connector appearing in the **Extensions** list.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Key concepts - Use Power Platform connectors (preview) in 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.

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

copilot makers to create more dynamic, responsive, and useful copilots, tailored to specific business needs and processes.

You can call connectors as [connector actions](#) in your copilot, 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 Copilot authoring canvas.
2. In the node selection window, select **Call an action**, select **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 end-user credentials. For more information about the supported authentication modes, see [Configure end-user authentication for actions](#). To change this behavior, see the following section.

Use connectors with copilot author's credentials

Connector actions require a valid set of credentials. By default, connector actions are configured to ask end users (users of your copilot) to provide their credentials for the associated service, when the action is invoked. To have your copilot use the author's credentials or credentials for a proxy account, perform the following steps:

1. Configure your copilot to use an [authenticated channel](#).
2. Add a connector action to your copilot as a plugin action, and configure it.
3. Go to the connector action properties.
4. Under **End user authentication**, select **Copilot author authentication**.
5. Publish and test the experience in the copilot test pane, or in a desired channel.

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

Use actions with custom copilots in Copilot Studio (preview)

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.

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

[This article is prerelease documentation and is subject to change.]

Core action types

If you turn on [generative mode](#), your copilot can automatically select the most appropriate action or topic, to respond to a user at runtime.

In classic mode, a copilot can only use topics to respond to the user. However, you can still design your copilot 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
- Bot Framework skill

ⓘ Important

You can't use [conversational actions created in Copilot Studio](#) as actions with custom copilots.

Each core action has additional information that describes its purpose, allowing the copilot 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 the inputs on a flow. Inputs are handled for you during runtime.

Actions can generate a contextual response to a user's query, using the results of the action. Alternatively, you can explicitly author a response for the action.

💡 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 custom copilot by choosing **Copilots** in the left hand navigation pane and selecting your copilot from the list.
2. Select **Actions** to view actions for the copilot.
3. Select **+ Add an action**.
4. In the **Choose an action (preview)** 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 copilot.

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](#).

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

Connector

Connectors let data move from a system or service to Microsoft Copilot Studio. Create a link to a connector by signing in.

Connect to



MSN Weather

...

End user authentication

User authentication

Control which user authentication type will be used

Usage Description

Enter a description

Describes the purpose of the connector action

0/2000 maximum characters

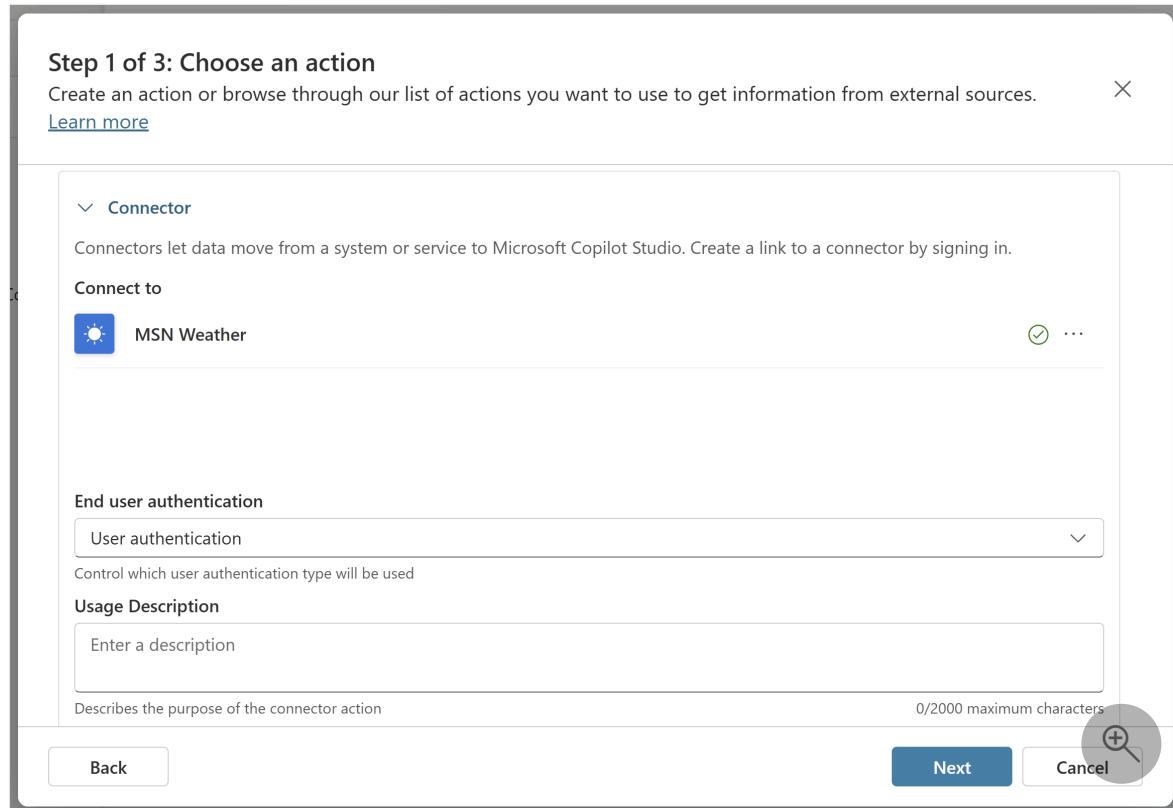
[Back](#)

[Next](#)

[Cancel](#)



Depending on the action, your connection is automatically configured, as shown in the example. Alternatively, you're required to configure the connection if instructions are given in the wizard. Once your connection is successfully configured, select **Next**.



5. Review and [make changes to your plugin actions configuration](#) if needed. Select **Finish** to add the action to your copilot.

Select **Next**.

Call an action

If you enabled [generative actions](#), your action is called automatically in response to a relevant user query.

Alternatively, you can [call an action explicitly from within a topic](#).

Test your action

Calling an action

Your copilot selects and runs your action based on the name and description. You can author high quality descriptions, using the dedicated tracing mode to see which actions your copilot selected from the [generative actions](#) page.

Alternatively, if you added an [topic](#), you can test it by entering a message similar to one of the trigger phrases for that topic.

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

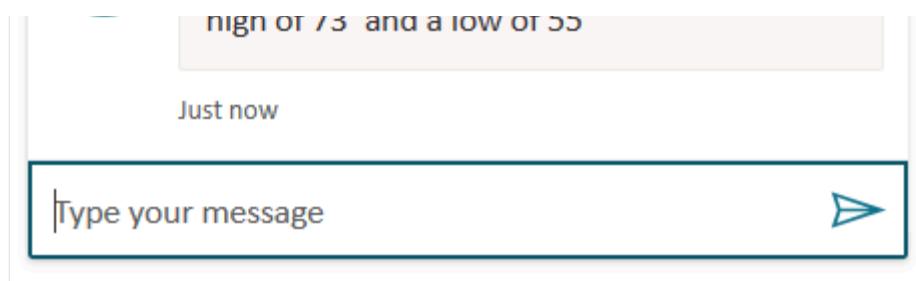


Just now



The weather in Seattle is forecast to have a

High of 63° and a Low of 55°



Alternatively, if you added an action to a topic, you can test by triggering your topic with a suitable trigger phrase.

Configuring inputs and outputs for an action

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.

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 copilot 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 asking the user for a location, the copilot generates a question similar to, *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 copilot accurately extract a value from a user's response.

By default, each input is set to **Dynamically fill with the best option**. The copilot tries to populate the value from available context, such as extracting a value 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 override, change the **How will the copilot fill this input?** dropdown to **Set as a value**, then enter a value. This value can be a literal value, an existing variable, or a Power Fx formula.

Step 2 of 3: Review inputs and outputs
You can edit each input and output for your action

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
Location

Description
The location search query. Valid inputs are City, Region, State, Country, Landmark, Postal Code, latitude and longitude

[Additional settings](#)

Units

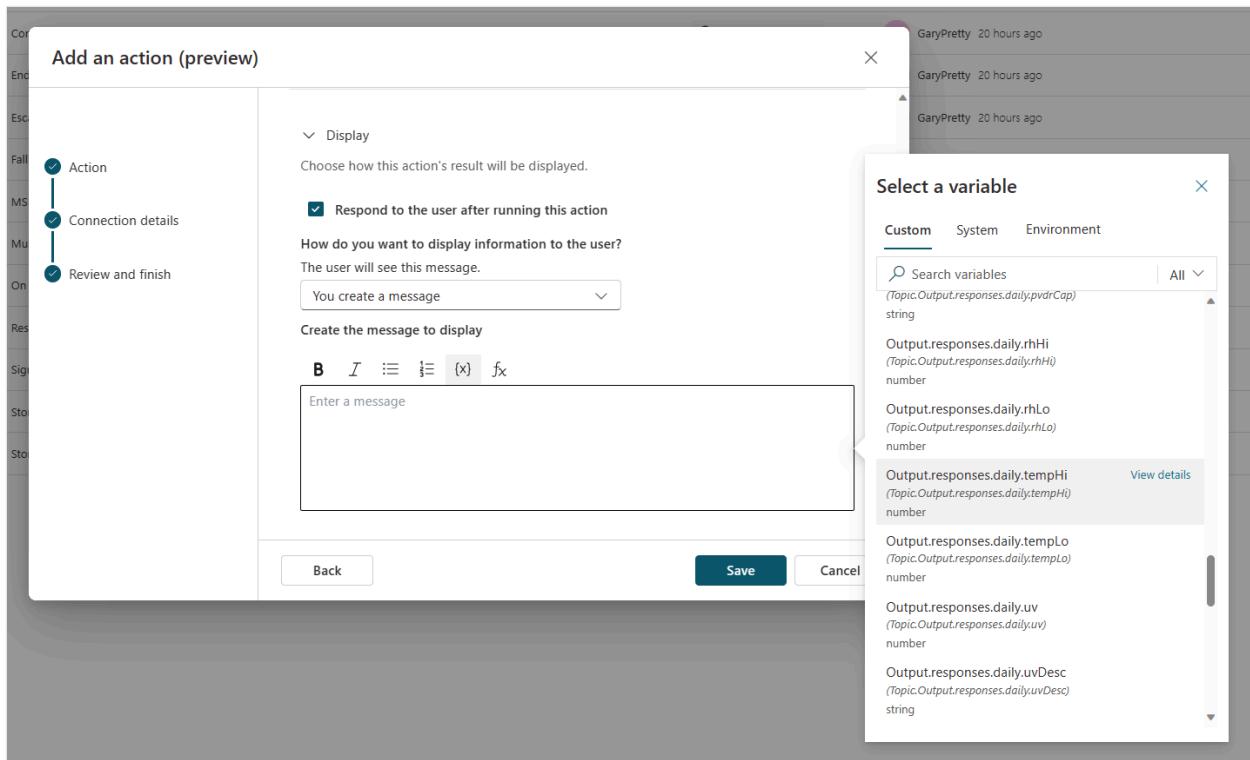
[Back](#) [Save](#) [Cancel](#)

To control your copilot's behavior when asking for a value or validating a user's answer, you can configure [additional properties](#) 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 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:

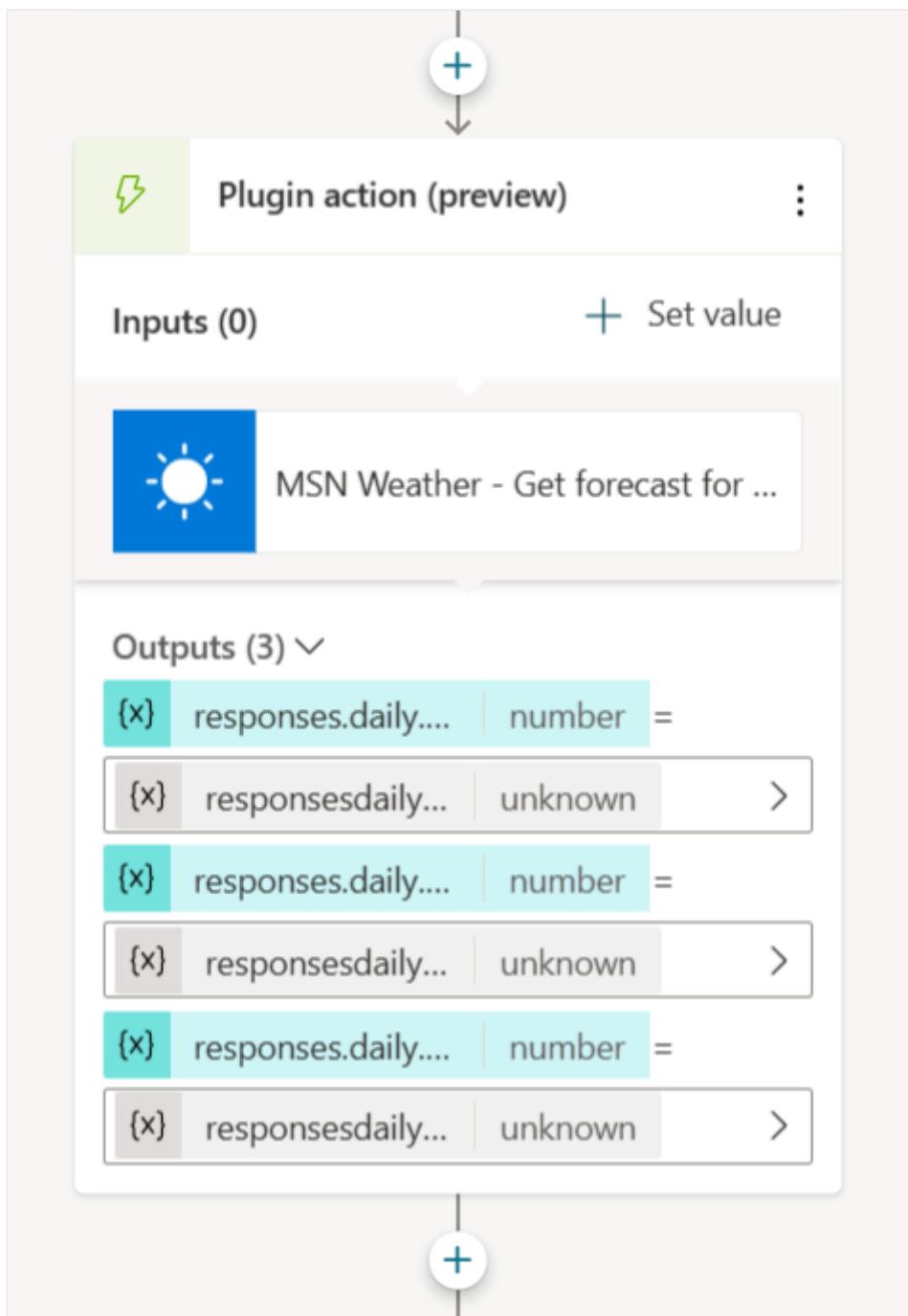
1. In Copilot Studio, go to the **Topics** for the copilot you want to edit.
2. Create a new topic, and name it **Get weather**.
3. Add the following **Trigger phrases**:
 - *will it rain*
 - *today's forecast*
 - *get weather*
 - *what's the weather*

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 highlighted, showing its description: 'Get the forecast for the current day in the specified location.'

5. Your **Plugin action (preview)** node is now added to your topic.

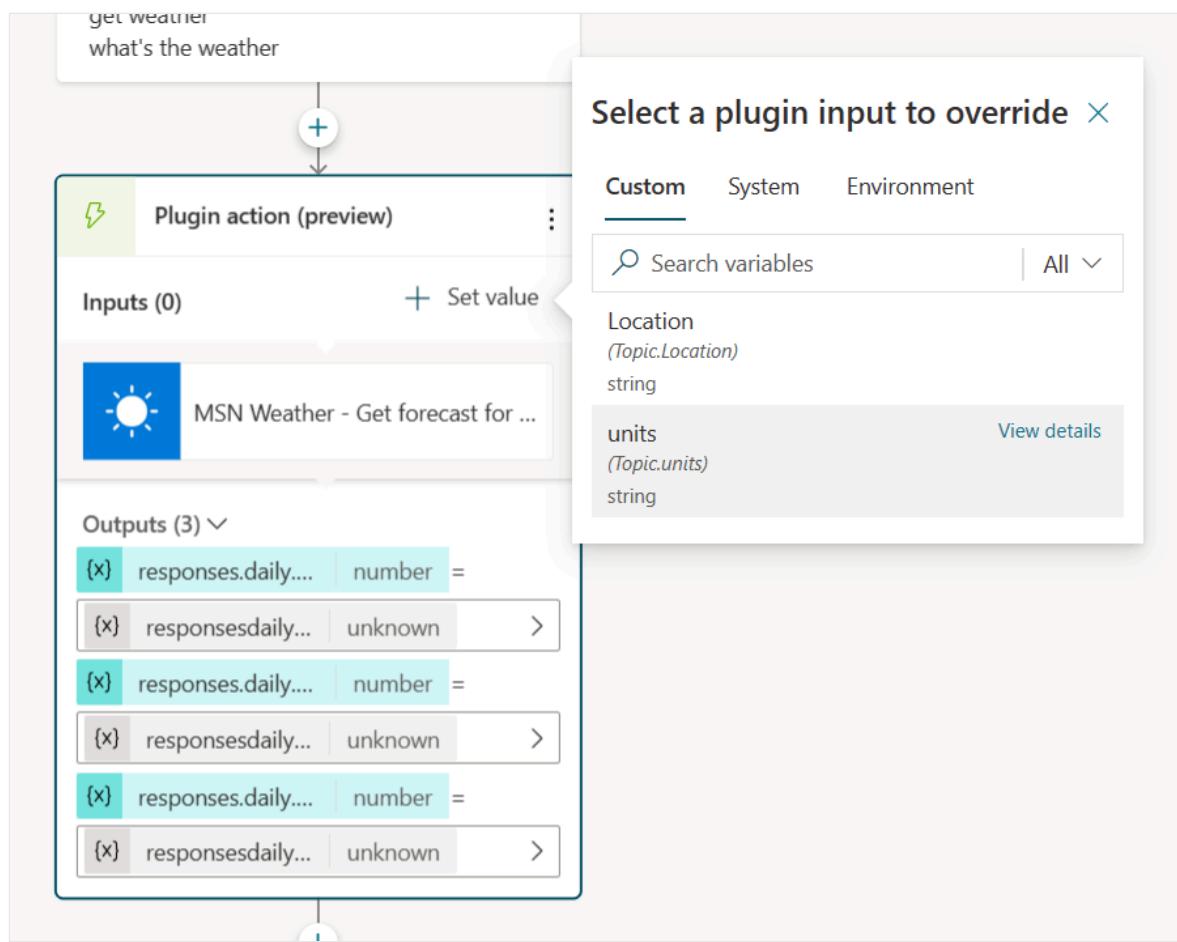


6. Select Save.

Override inputs when calling an action from a topic

By default, when your action runs, your copilot 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.

The screenshot shows the Microsoft Copilot AI Actions interface. At the top, there's a green button with a lightning bolt icon and the text "Plugin action (preview)". Below it, there's a "Set value" button with a plus sign. The main area is divided into "Inputs (1)" and "Outputs (3)".

Inputs (1)

- {x} units string = Metric >

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 copilot 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 Microsoft Copilot runtime in the end-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 Copilot author authentication.
3. Publish and test the experience in the copilot test pane, or in a desired channel.

The screenshot shows a configuration dialog for a copilot action. At the top, there's a field labeled "Description for the copilot to know when to use this action *". Below it is a text input box containing the placeholder "Get the forecast for the current day in the specified location.". Underneath this is a checkbox labeled "Ask the user before running this action.", with a note explaining that user confirmation is recommended for sensitive actions. A dropdown menu titled "End user authentication" is open, showing two options: "User authentication" (which is currently selected) and "Copilot author authentication" (which has a checkmark). At the bottom right of the dialog are "Next" and "Cancel" buttons.

Use flow actions with end-user credentials

Flow actions can be [configured to use end-user credentials](#). If you use the flow in a supported channel, this requires no special configuration in your copilot dialog, and end 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 end-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 copilot to a Teams channel. This method also works for copilots embedded in any Power Platform app that uses tokens built with the PPAPI standard.
- **Manual authentication (third-party copilots):** 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 Copilot Studio copilot.

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

`https://token.botframework.com/.auth/web/redirect`

[Copy](#)

Service provider *

Azure Active Directory v2

Client ID *

Client secret *

Token exchange URL (required for SSO) [Learn more about SSO](#)

Tenant ID

91bee3d9-0c15-4f17-8624-c92bb8b36ead

Scopes ⓘ

profile openid

[Save](#)

[Close](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Key concepts - Use Power Automate flows in 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.

Extend the capabilities of your copilot 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 copilots 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 copilot. Existing flows can be [added to a solution](#) to be available when adding actions to your copilot.
- [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 copilots.

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 a copilot topic using an action node.
Add input and output variables	Pass variables between a Power Automate flow and a Copilot Studio copilot.
Return a list of results	Return a list of results from a Power Automate flow to a Copilot Studio copilot.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create a flow

Article • 05/21/2024

ⓘ Important

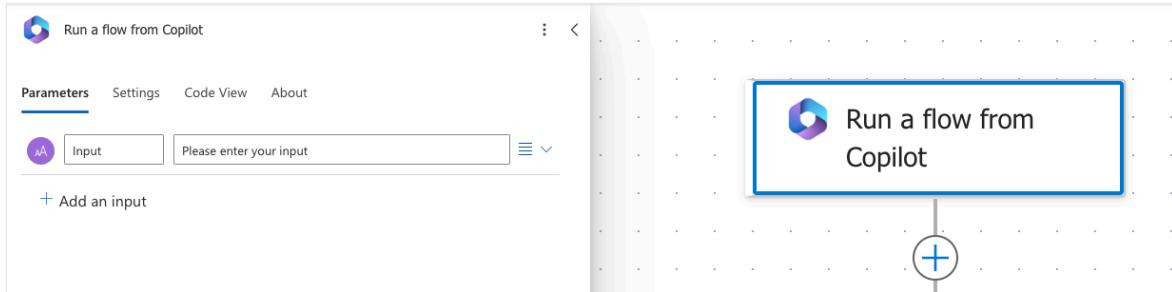
Power Virtual Agents capabilities and features are [now part of Microsoft Copilot Studio](#) following significant investments in generative AI and enhanced integrations across Microsoft Copilot.

Some articles and screenshots may refer to Power Virtual Agents while we update documentation and training content.

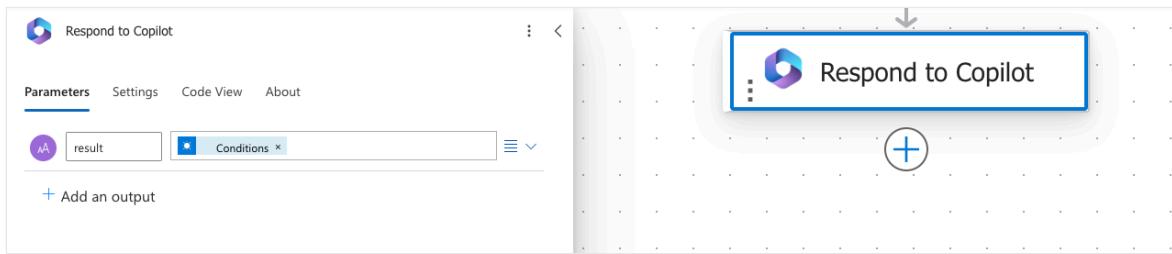
Learn how to use a cloud flow in Copilot Studio by giving a copilot the ability to produce a weather forecast on request. In this first example, you'll create a flow that fetches a weather forecast. In the next example, you'll [call the flow as an action from a topic](#) so the copilot can respond with the forecast when a customer asks about the weather.

To work with copilots, 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 copilots by changing it to use these actions.

Important

To use cloud flows in custom copilots in Copilot Studio, the flows must:

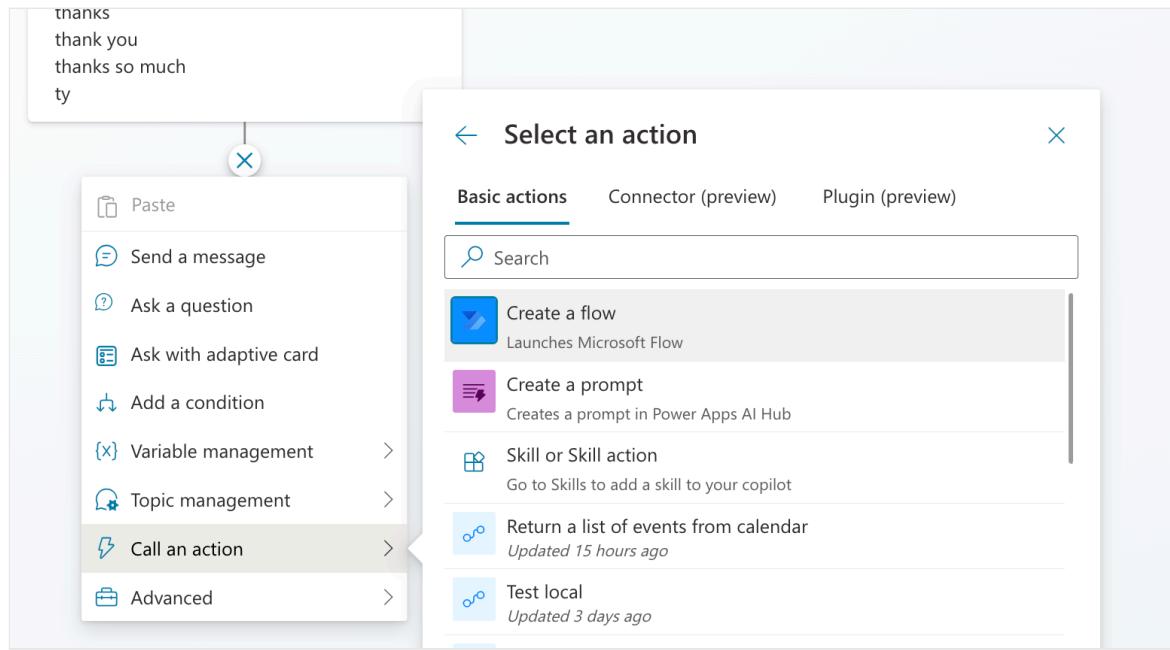
- Have the **Run a flow from Copilot** trigger.
- Be located in a solution within the same environment as the copilot. Existing flows can be [added to a solution](#) to be available when adding actions to your copilot.
- [**Return a response synchronously**](#) with the **Respond to Copilot** action, 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

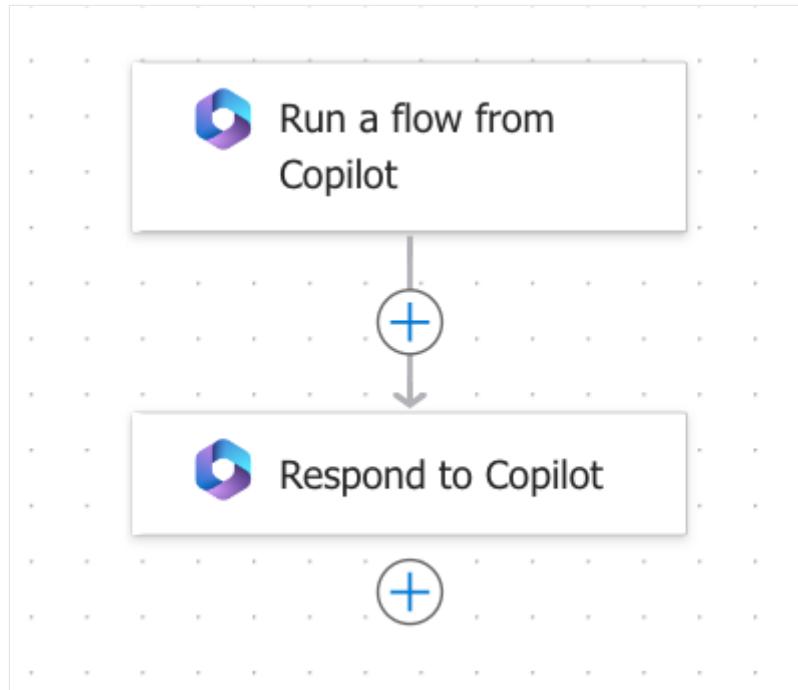
- [Access to Copilot Studio](#) with a license that includes the use of flows within the context of Copilot Studio creations.

Create a flow to use with a copilot

1. Go to the [Topics page](#) for your copilot.
2. Open the authoring canvas for 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**.

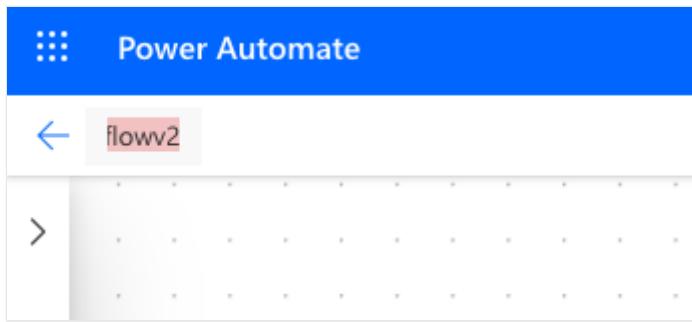


The [Power Automate portal](#) opens in a new tab, with a starter flow template open for editing. The portal opens in the same environment as the copilot and uses the same user credentials.



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**.

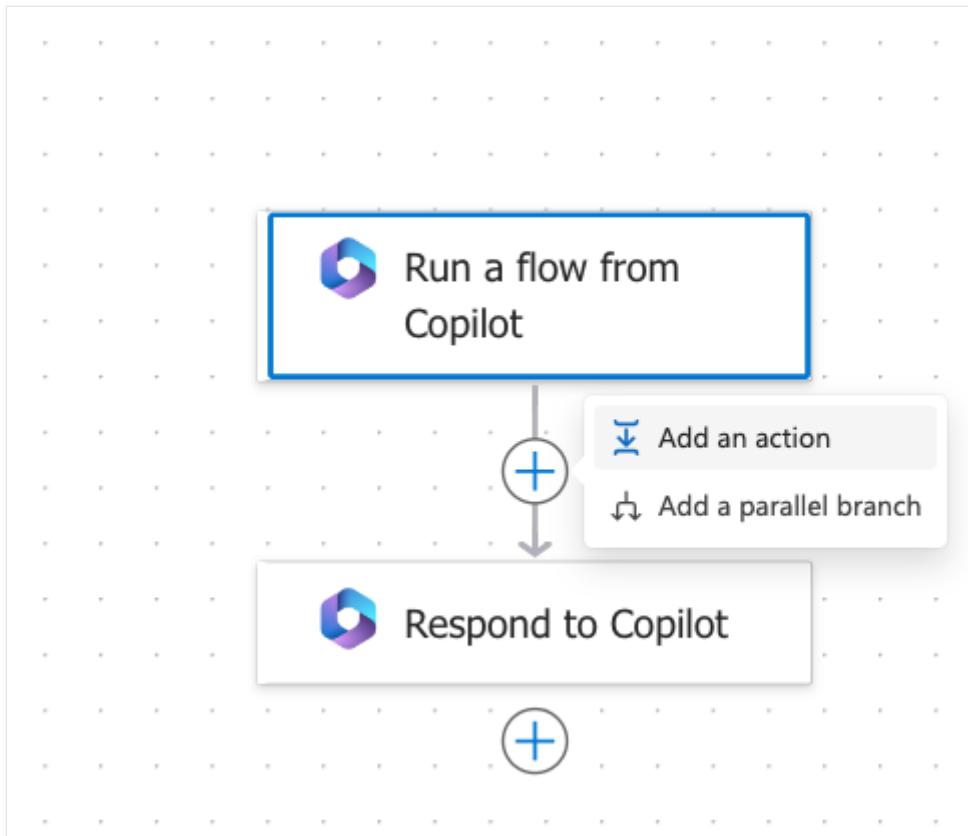


2. Add the following input parameters to the **Run a flow from Copilot** trigger:

- City (Text)
- Zipcode (Number)

The screenshot shows the configuration page for the "Run a flow from Copilot" trigger. At the top, there's a logo and the trigger name. Below it, a navigation bar has tabs for "Parameters", "Settings", "Code View", and "About", with "Parameters" being the active tab. Under the "Parameters" tab, there are two input fields: one for "City" (text type) with placeholder "Please enter your input" and one for "NumZipcode" (number type) with placeholder "Please enter a number". Both fields have a "More options" icon (three horizontal lines with a downward arrow). At the bottom, there's a button labeled "+ 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.

Add an action	
msn	X
Runtime	Action Type
Select a runtime	Actions
<input checked="" type="checkbox"/> Group by Connector	
 MSN Weather	See more
Get forecast for today	(i)
Get current weather	(i)
Get the forecast for tomorrow	(i)

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 Name	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

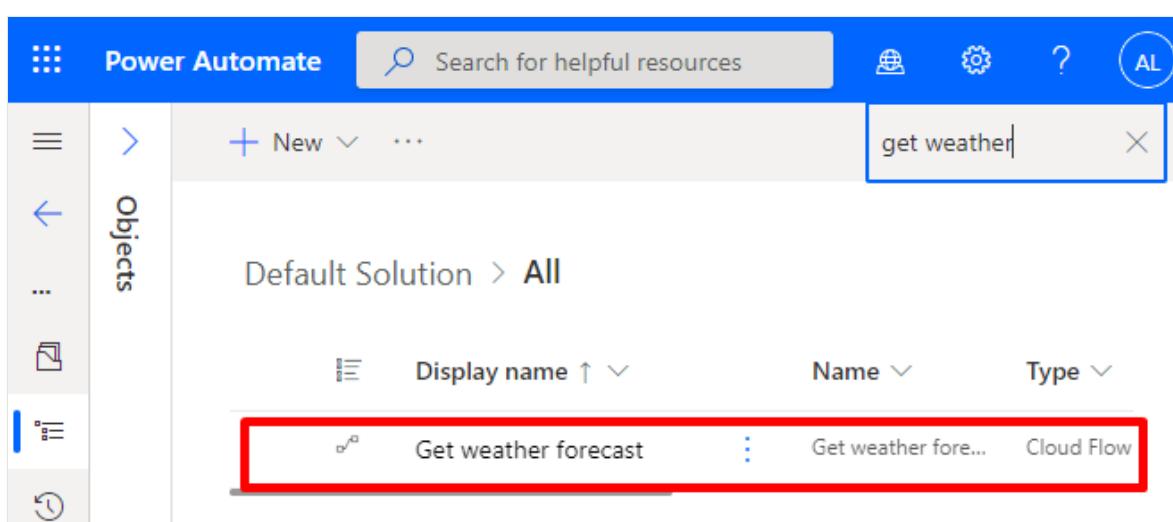
In a supported authenticated copilot, cloud flows can be [configured to use end user credentials](#) when they're run as a generative action or from a topic. If you are using the flow in a supported channel, this requires no special configuration in your copilot dialog, and end-users will be able to [manage their connections](#) while using the copilot.

Cloud flows in environments using [customer-managed keys \(CMK\)](#) cannot yet be run with end user credentials as generative actions 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 a copilot

If you have a flow with a different trigger that you wish to use with your copilot, 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 copilot.

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.



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 copilot.

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 a copilot must return values in real time, or [synchronously](#). Flows that run in the background, or asynchronously, may cause an error when your copilot tries to run them. Instead of running the flow, the copilot 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**.

[Parameters](#) [Settings](#) [Code View](#) [About](#)∨ **Networking**

Asynchronous Response

 Off∨ **Run After**

+ Select Actions ▾

> Get forecast for today

● ⋮

∨ **Security**

Secure Inputs

Enabling secure inputs will automatically secure outputs. Off∨ **Tracking**

Tracked properties

Key	Value
-----	-------

Add the flow to a solution

To be available to copilots, 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 copilots, 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 PowerApps portal. On the left, there's a navigation menu with options like Home, Action items, My flows, Create, Templates, Connectors, and Solutions. The 'Solutions' tab is selected. At the top, there are buttons for '+ New solution', 'Import', 'Open AppSource', 'Publish all customizations', and 'Switch to classic'. A table below lists existing solutions with columns for Display name, Name, Created, and Version. One solution named 'Contextual Help Base' is highlighted.

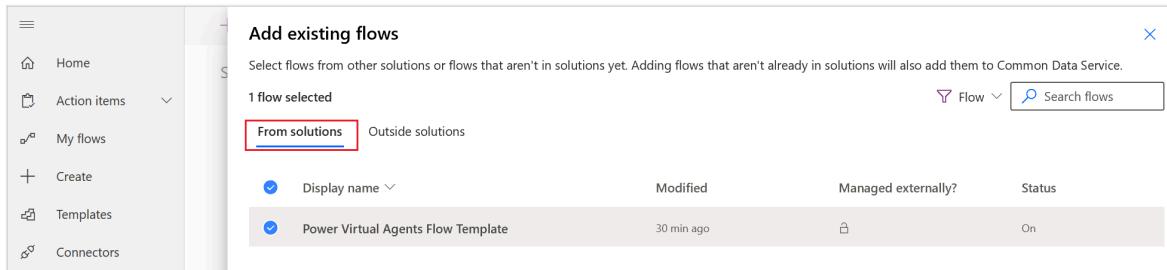
Give your new solution a name, select **CDS Default Publisher** in the Publisher field, enter a **Version** number, and then select **Create**.

The screenshot shows the 'New solution' dialog box. It has fields for 'Display name *' (filled with 'Demo Bot Flows'), 'Name *' (filled with 'DemoBotFlows'), 'Publisher *' (set to 'CDS Default Publisher'), 'Version *' (set to '1.0'), and a 'More options' section. 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.

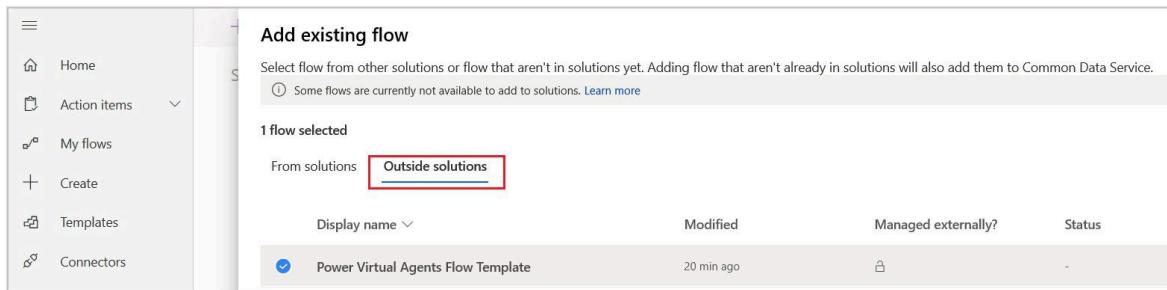
The screenshot shows the 'Demo Bot Flows' solution page. The navigation bar includes '+ New', '+ Add existing' (which is highlighted with a red box), 'Delete', 'Export', 'Publish all customizations', and more. Below the navigation, it says 'Solutions > Demo Bot Flows'. There's a large circular icon with a document symbol and the text 'No components found.' A note at the bottom states 'There are no components to show in this view. Create one or more components to get started.' with a 'Learn more' link.

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 red box highlights the 'From solutions' tab.

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 red box highlights the 'Outside solutions' tab.

Related topics

- [Call the flow as an action](#)
- [Learn more about input and output variables](#)
- [Configure End-user authentication](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Call a flow as an action

Article • 04/28/2024

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.

Learn how to use a cloud flow in Copilot Studio by giving a copilot 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 a copilot topic to call the flow when a customer asks about the weather.

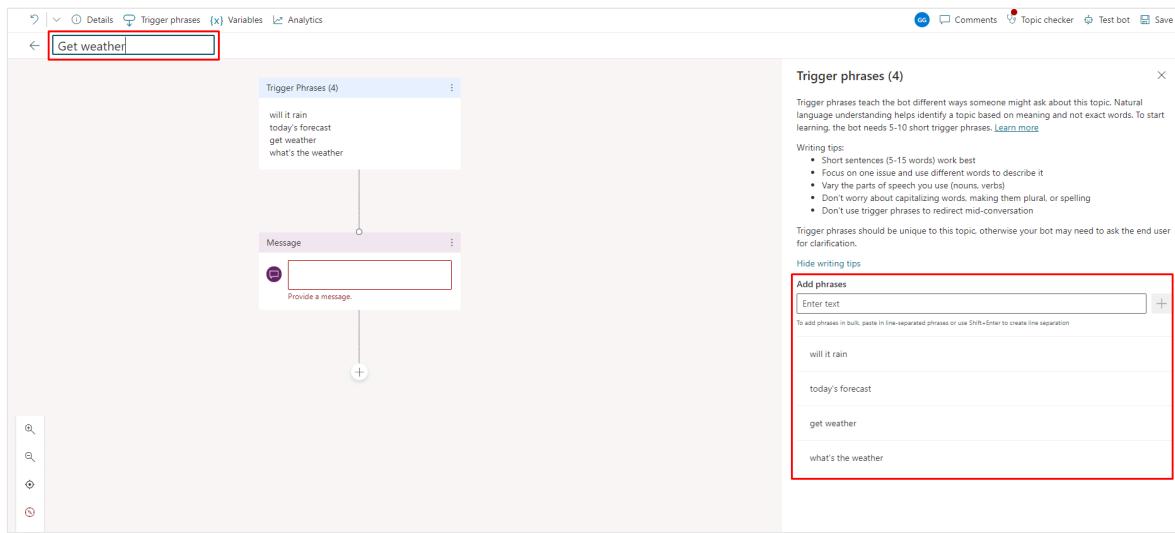
If you haven't already [created the weather forecast flow](#), do that now.

Prerequisites

- [Create a flow](#)

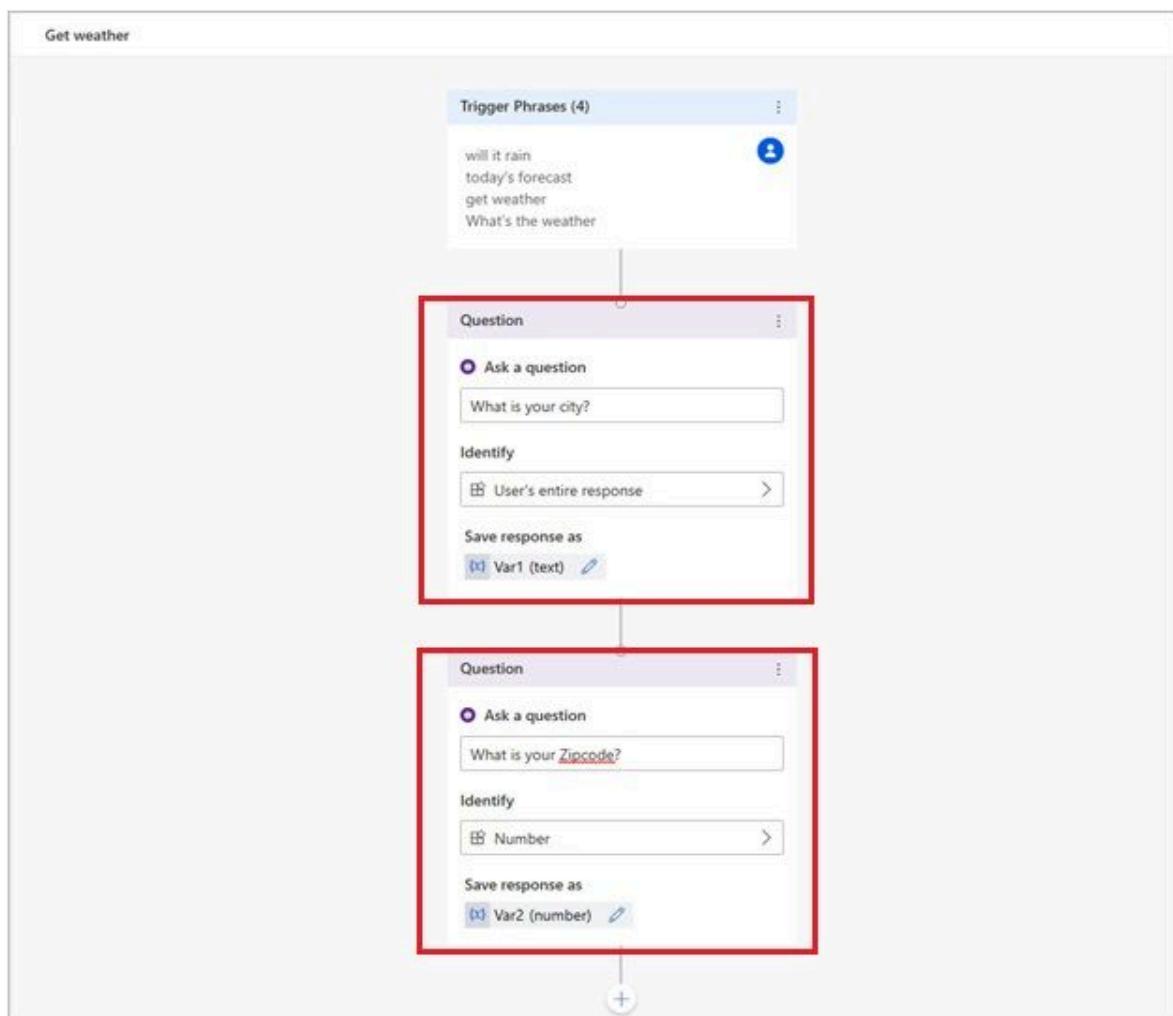
Call a flow from a topic

1. In Copilot Studio, go to the [Topics page](#) for the copilot 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.

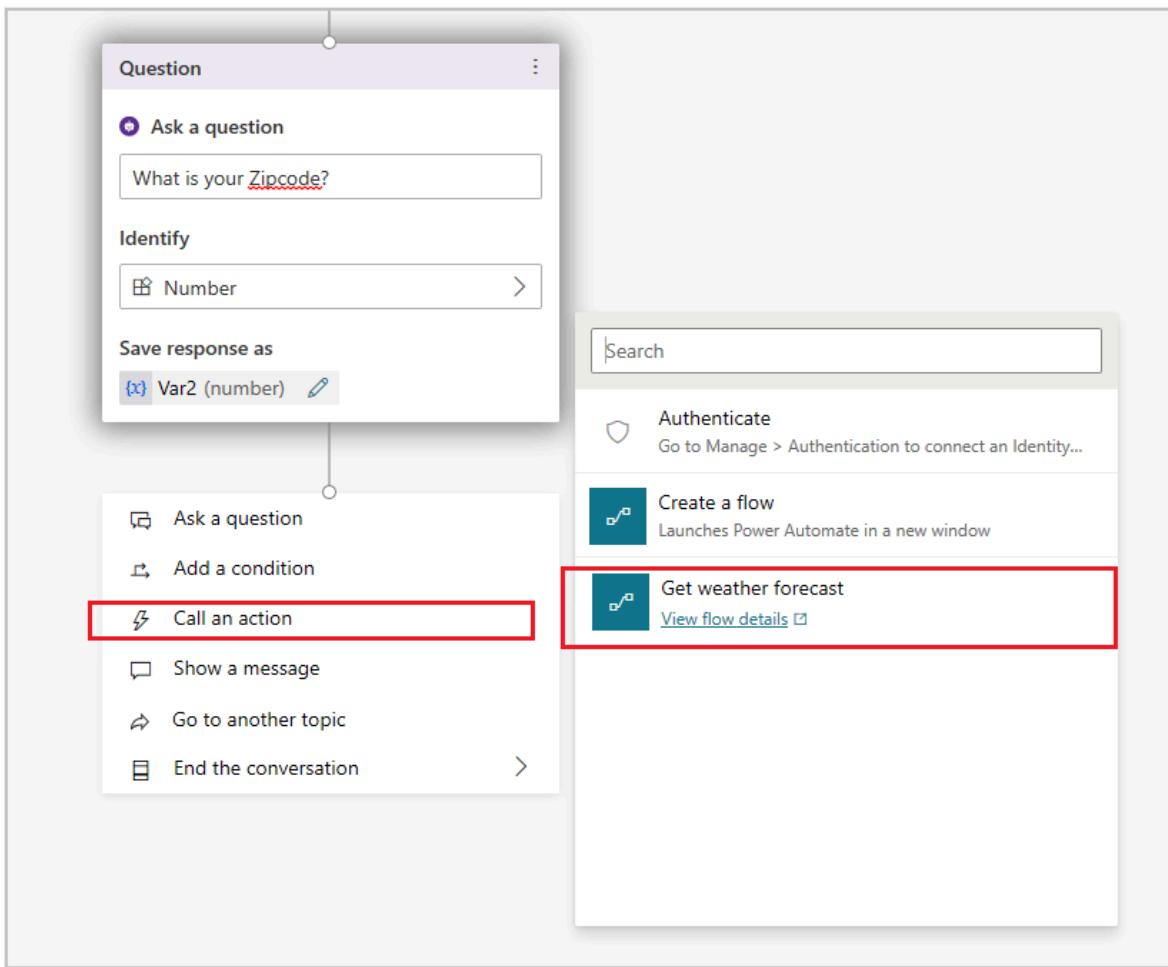
4. Enter **I can help you with that** in the message node, and then select **Add node (+)** under it to add a new node.
5. Add two **Ask a question** nodes for the **City** and **Zipcode** inputs the flow needs, as shown in the following example.



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

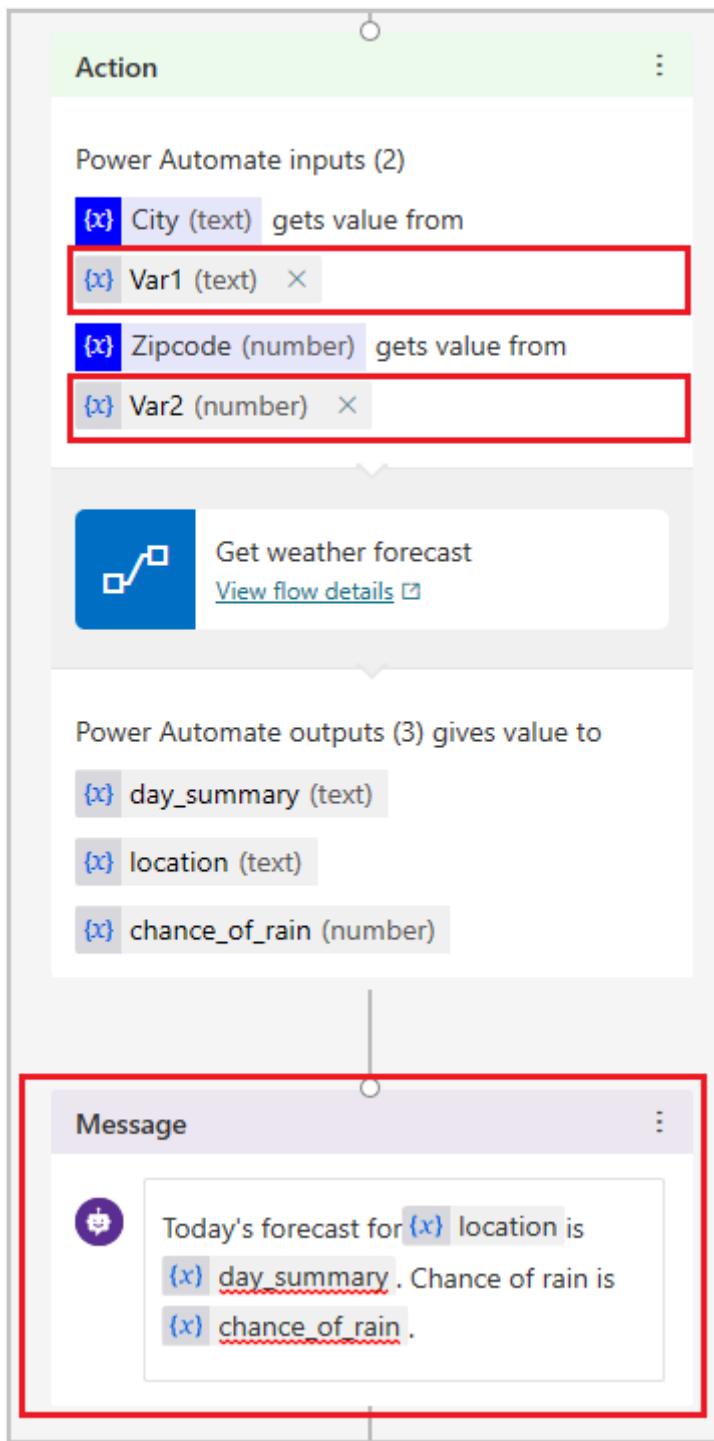
a more meaningful name such as "City" and "Zipcode."

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%`



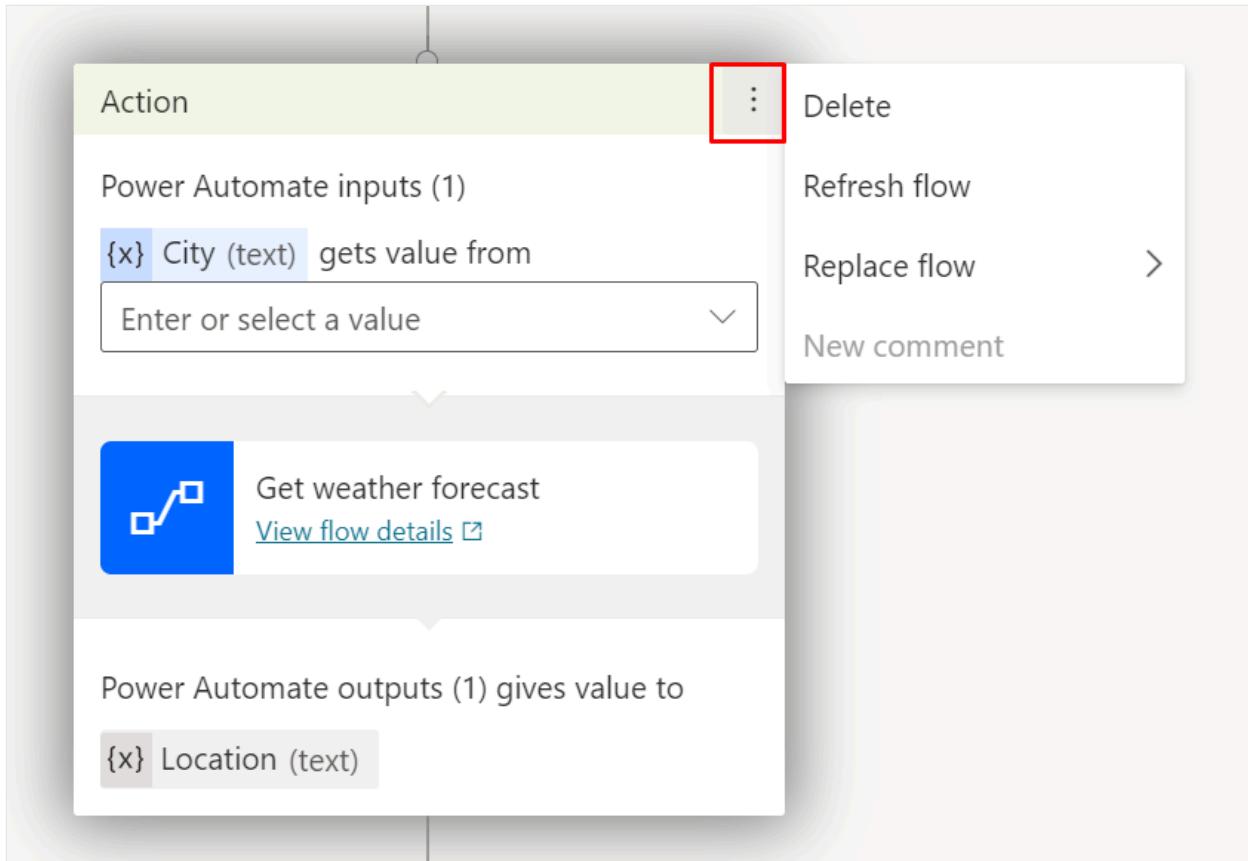
10. Select Save.

Manage a flow used in a copilot topic

After you've called a flow from an action node in a copilot 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.

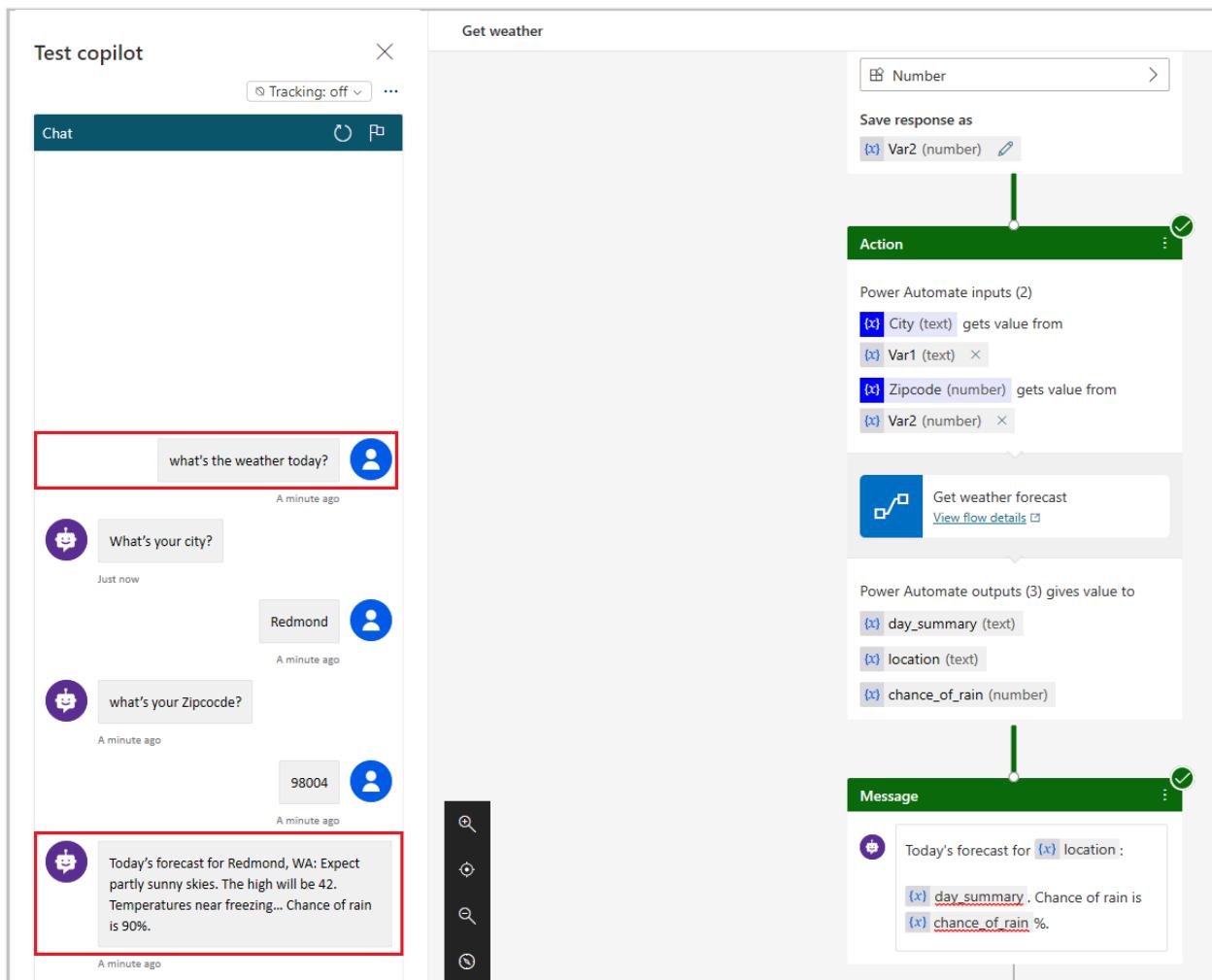
- **Replace flow:** Select to open the action picker, where you can select or create a different flow.



Test your flow and topic

It's a good idea to test the flow and the topic that calls it.

In the **Test copilot** pane, type one of the topic's trigger phrases to start a conversation with the copilot. Enter your city and zip code when the copilot asks for them. Make sure the copilot returns today's weather forecast and your message looks the way you expect.



Troubleshoot your copilot

Test your copilot when you make changes to topics and flows to make sure everything is working. If the copilot encounters a problem during a conversation, it will respond with an error message.

You'll find most flow-related issues in the [Flow Checker](#). Any issues on the authoring canvas will appear in the [topic checker](#).

Feedback

Was this page helpful?

[Yes](#)

[No](#)

Provide product feedback ↗

Add input and output variables

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.

Use variables as input and output parameters to pass information between Power Automate flows and Microsoft Copilot Studio copilots.

ⓘ Important

A copilot 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 a copilot can send or how many variables a copilot can pass.

In this example, you'll create a flow with an input parameter and return it to a copilot 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

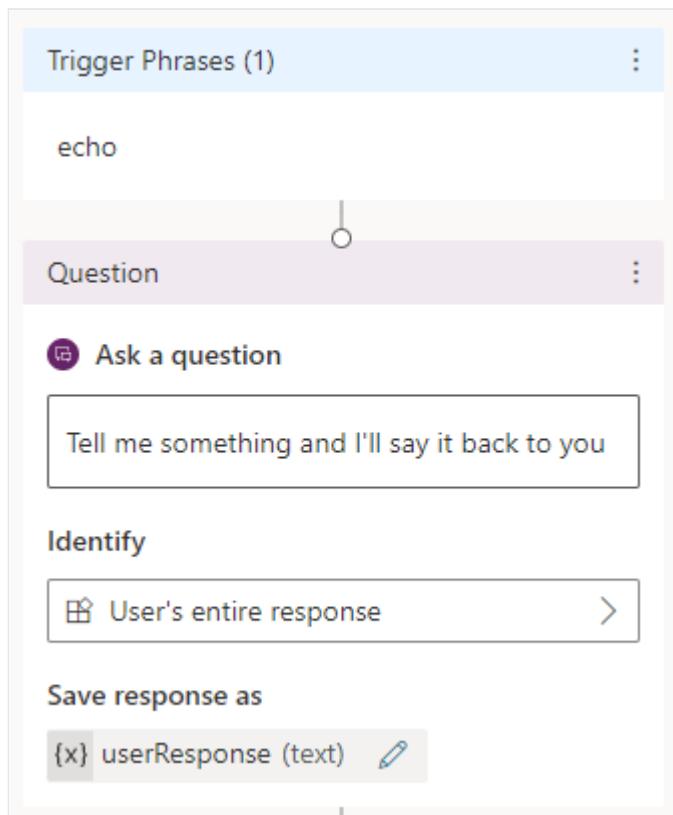
The following types aren't supported:

- 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 copilot.
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**.

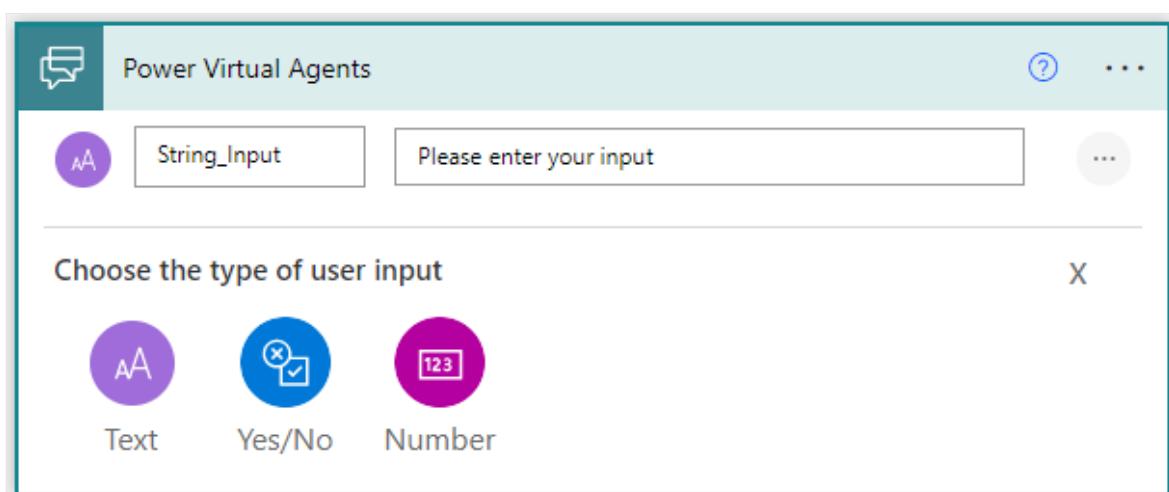
8. In the Power Automate portal, name the flow Echo parameter.

Add input parameters to the flow

1. In the Microsoft Copilot Studio flow trigger, select Add an input.

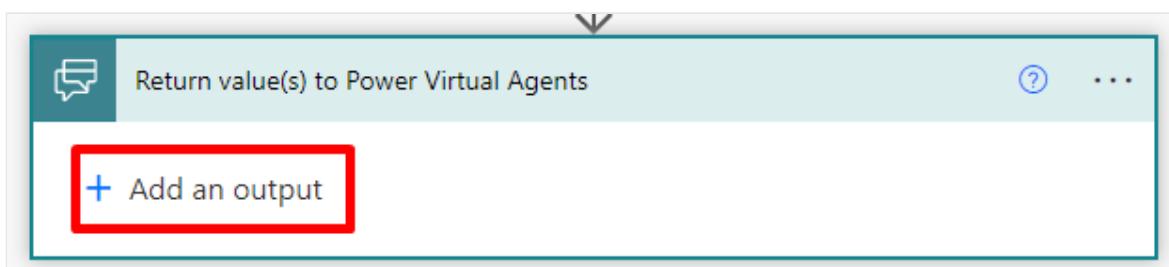


2. Select Text and enter the name String_Input.

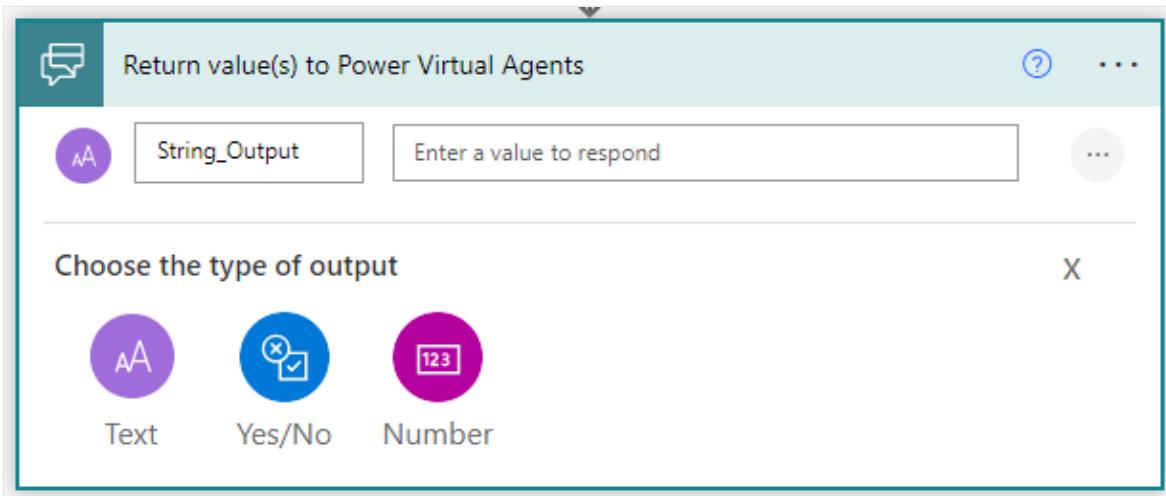


Add output parameters to the flow

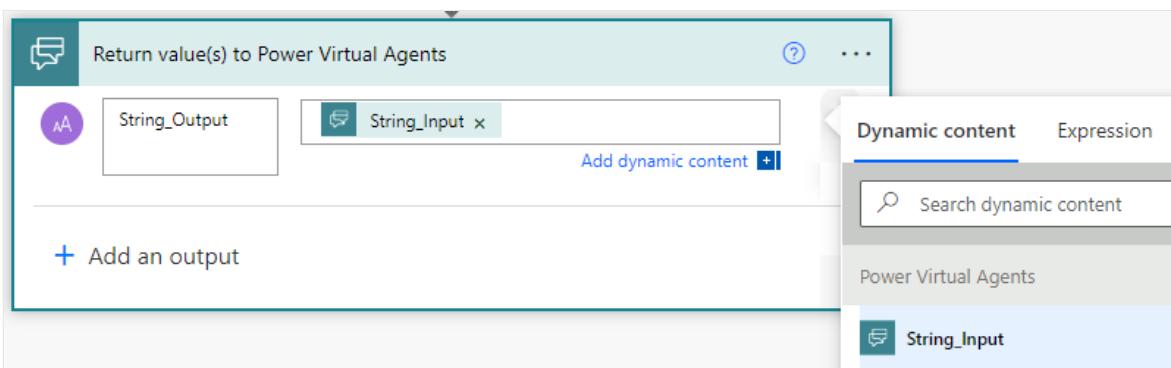
1. In the Return value(s) to Microsoft Copilot Studio response action, select Add an output.



2. Select Text and enter the name String_Output.



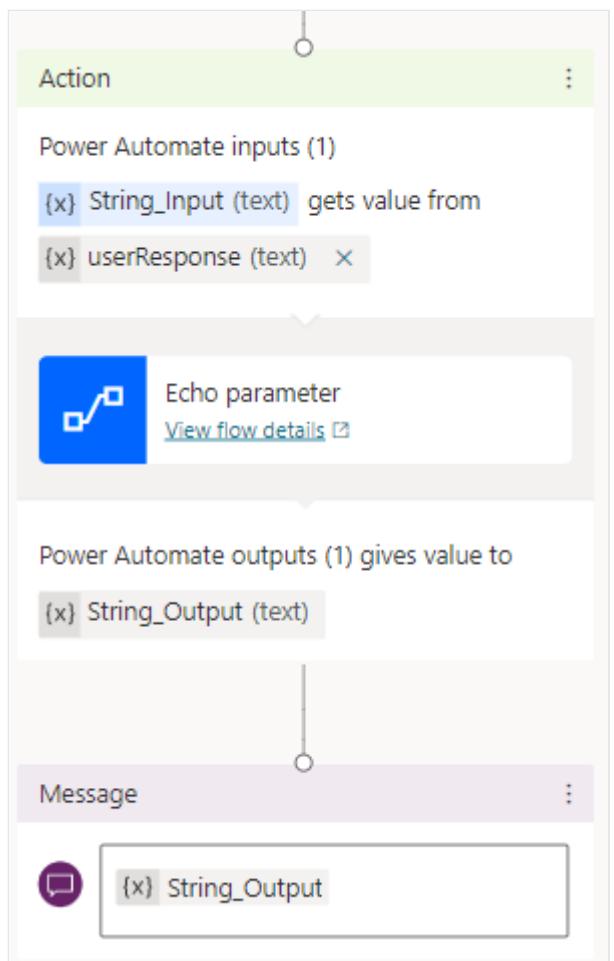
3. For the value of **String_Output**, select the **String_Input** variable.



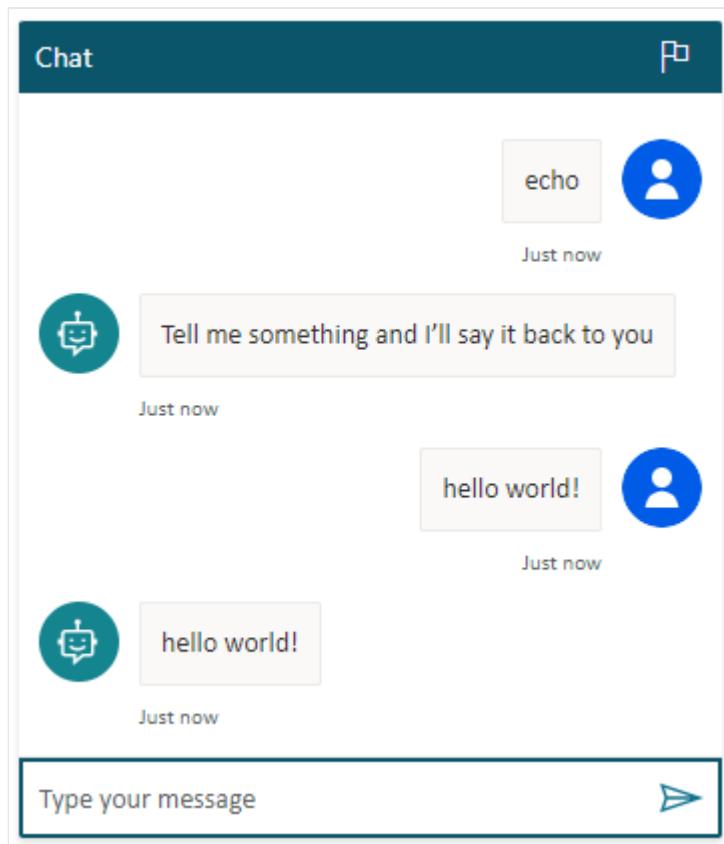
4. Select Save.

Select variables in the copilot 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**.

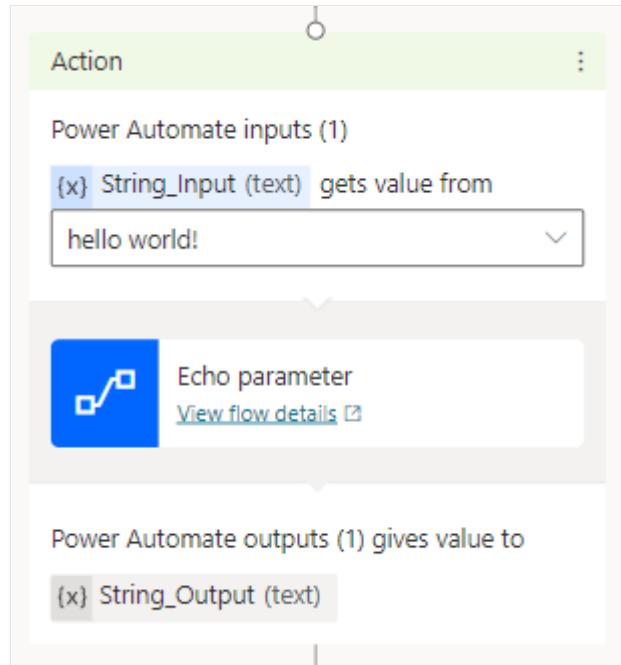


5. Test your topic in the Test copilot pane.



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 topics

[Use variables](#)

Return a list of results

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.

It's common to ask a copilot 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 copilot

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 Microsoft Copilot Studio authoring canvas
- [Add input and output variables](#)
- [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 copilot 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 copilot 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 copilot.
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 copilot's language understanding capabilities to extract the organization name from the response.

The screenshot shows a configuration interface for a Power Virtual Agents conversation. 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**.

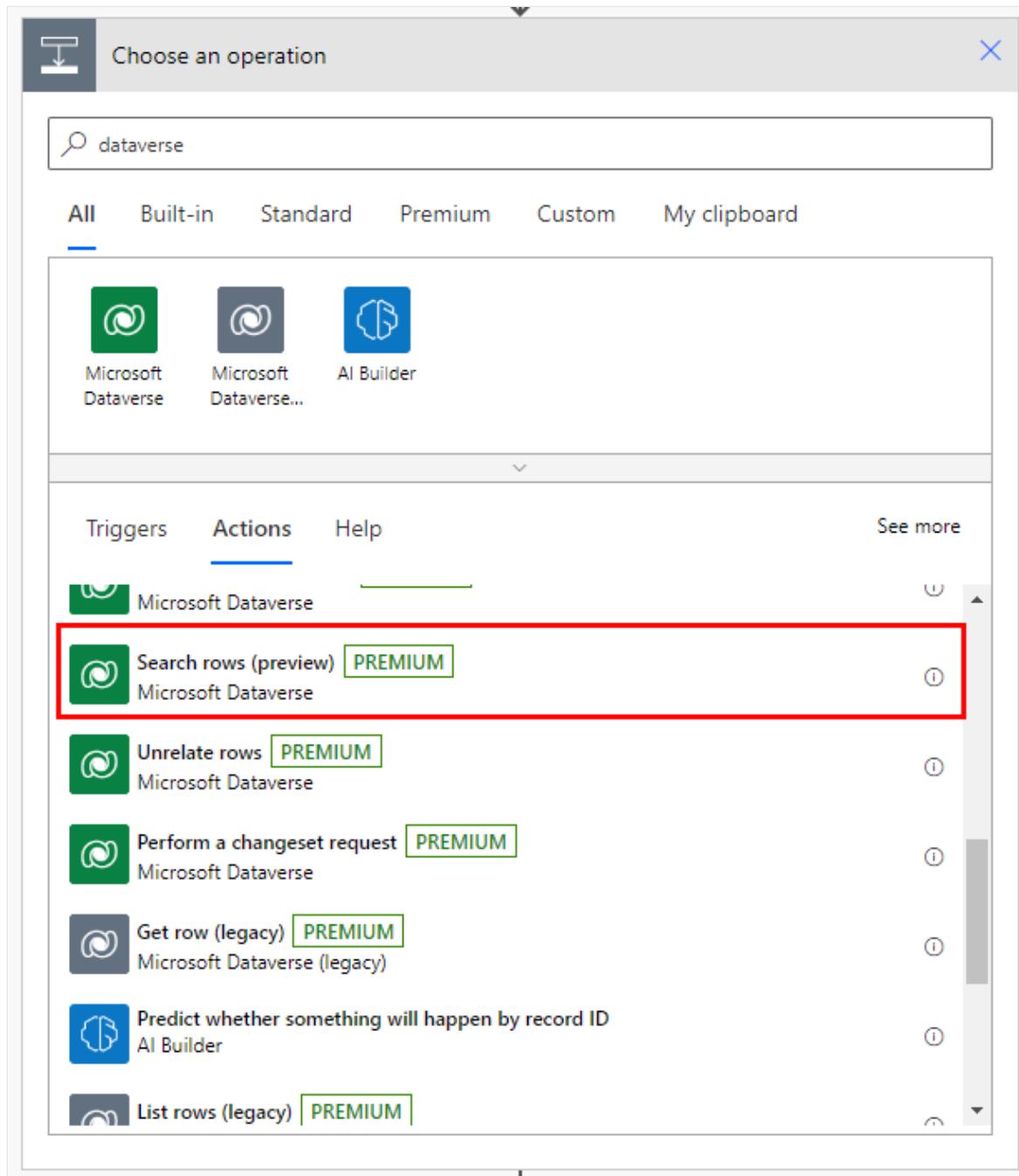
The screenshot shows a dialog for choosing the type of user input. It includes:

- When Power Virtual Agents calls a flow (V2)**
- Organization** (selected)
- Please enter your input**
- Choose the type of user input**
- Text** (selected)
- 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"  
        ]  
    }  
}
```

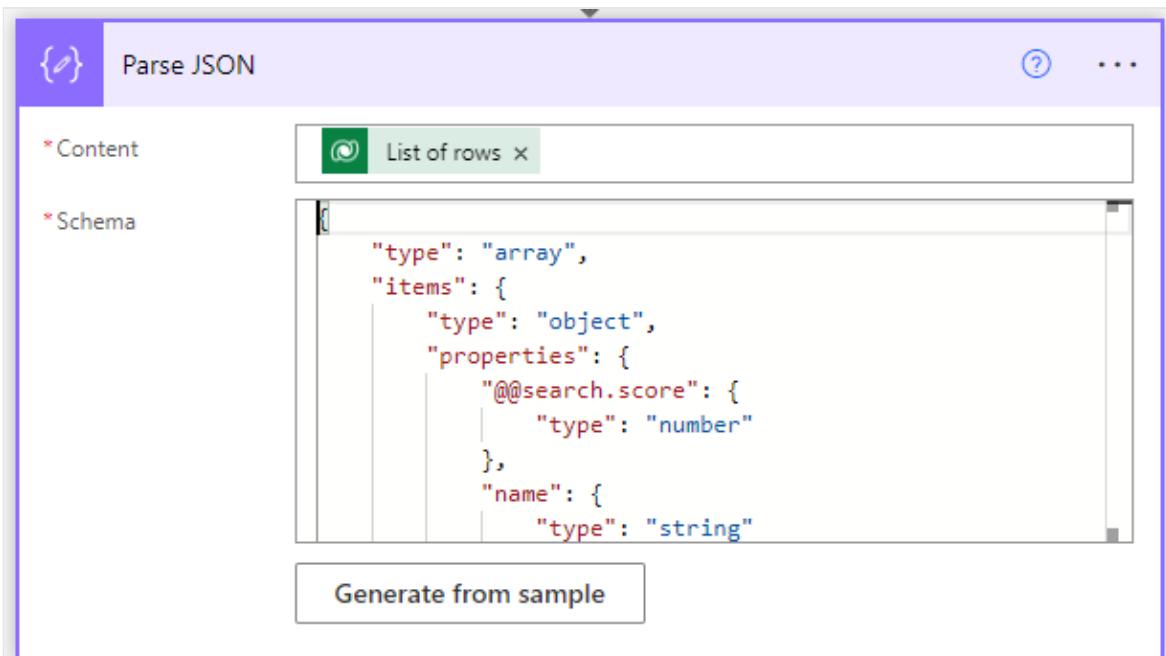
Parse JSON

* Content List of rows

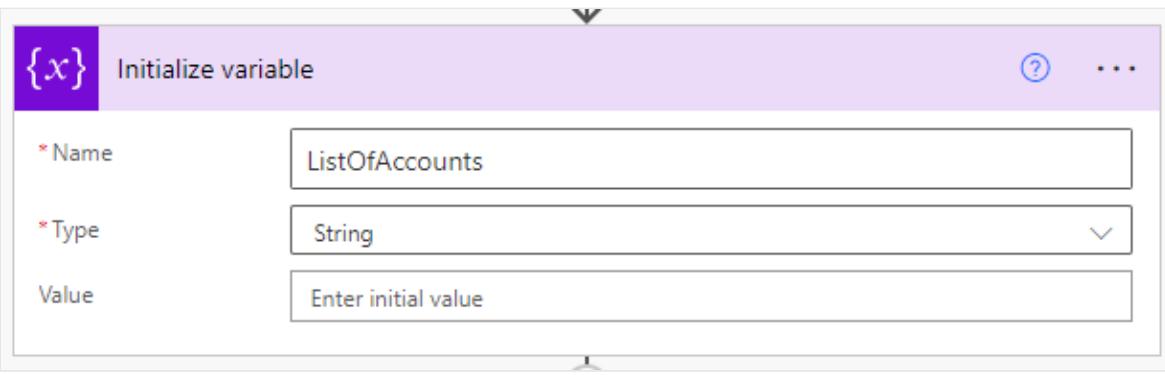
* Schema

```
[  
    "type": "array",  
    "items": {  
        "type": "object",  
        "properties": {  
            "@@search.score": {  
                "type": "number"  
            },  
            "name": {  
                "type": "string"  
            }  
        },  
        "required": [  
            "name"  
        ]  
    }  
]
```

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:

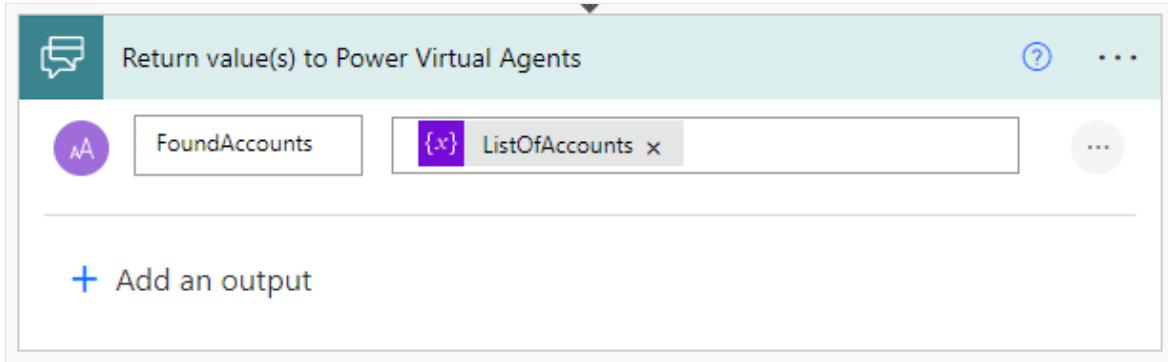
```
PowerApps Formula

- @{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 PowerApps formula editor with an 'Apply to each' step. The 'Select an output from previous steps' dropdown is set to 'Body'. Below it is an 'Append to string variable' step. The 'Name' is 'ListOfAccounts' and the 'Value' is a formula: '- {accountnumber} : {name} - {address1_city}'.

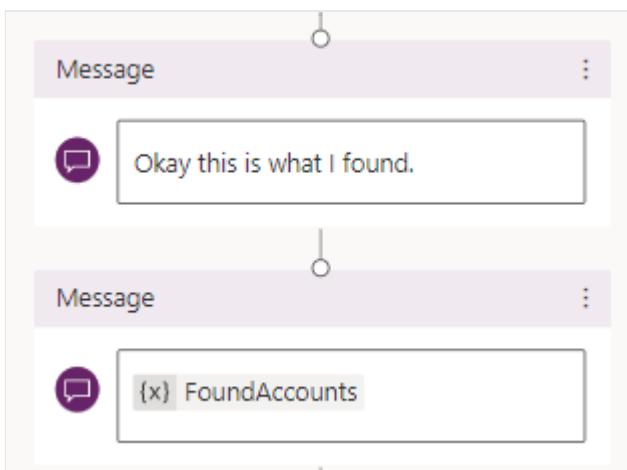
12. In the **Return value(s) to Microsoft Copilot Studio** action, add a **Text** output. For **Name**, enter **FoundAccounts**. For **Value**, select **ListOfAccounts**.



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 copilot in the **Test copilot** pane.

Chat



search accounts



Just now



Sounds like you're looking for an account. I can look that up for you.

What's the name of the customer you're looking for?

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



Orchestrate copilot topics and actions with generative AI (preview)

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 is prerelease documentation and is subject to change.]

By default, a copilot responds to users by triggering the topic whose trigger phrases match most closely the user's query, and it fills topic inputs from the conversation context.

You can [configure your copilot to use generative AI](#) to choose not only from [topics](#) you created, but also from [actions](#) you added to extend the copilot.

In generative mode, a copilot can fill topic inputs, not only from the conversation context but also by generating questions to prompt the user for values. To learn more about this behavior and how to manage it, see [Manage topic inputs and outputs](#).

How does generative mode work?

Using generative AI to determine how your copilot responds can make the conversation more natural and fluid for the users.

When a user sends a message, your copilot selects one or more actions or topics to prepare its response. Multiple factors determine the selection. The most important factor is the description of the topics and actions. Other factors include the name of a topic or actions, any input or output parameters, and their names and descriptions. Descriptions make it possible for your copilot to be more accurate when it associates a user's intent with actions and topics.

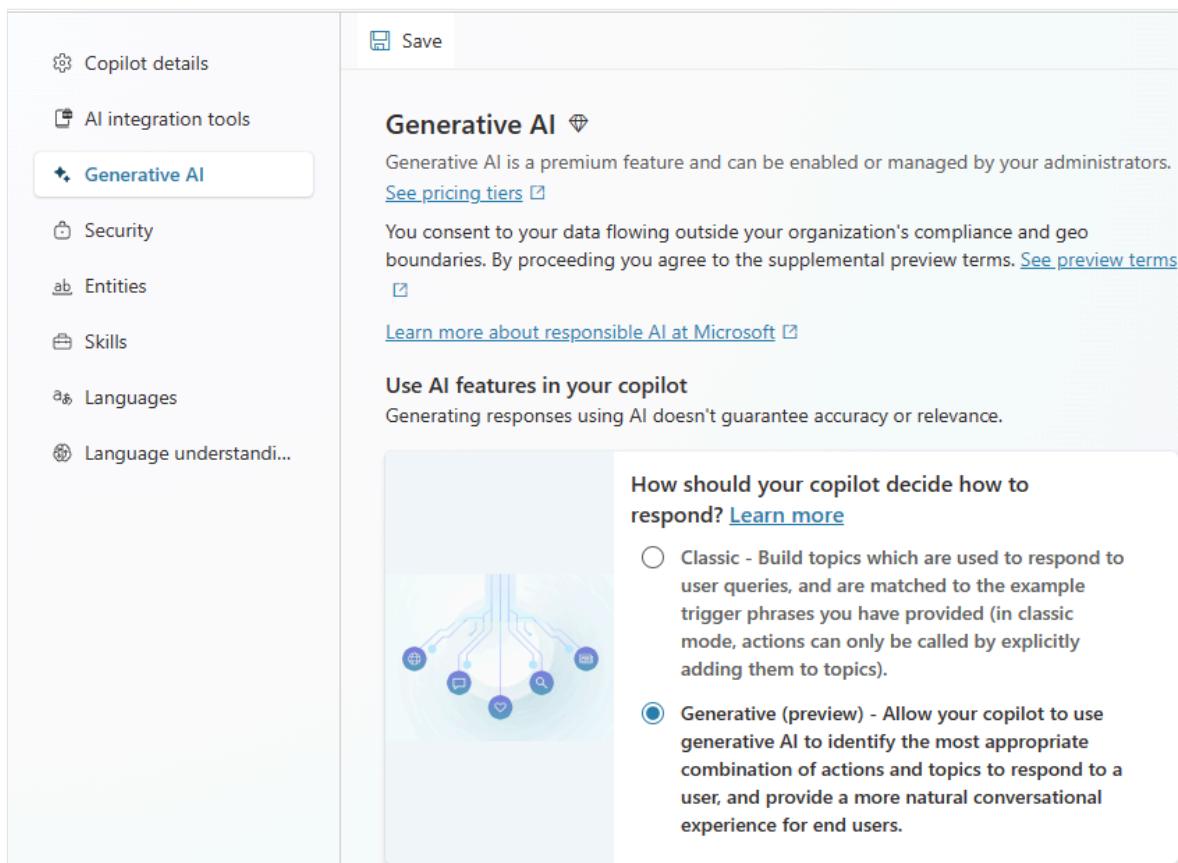
In generative mode, a copilot can select multiple actions or topics at once, to handle multi-intent queries. Once actions and topics are selected, the copilot generates a plan

that determines their execution order.

When you test a copilot that uses generative mode in Copilot Studio, you can [open the conversation map](#) to follow the execution of the plan.

Turn on generative mode for a copilot

1. Open your copilot in Copilot Studio.
2. On the top menu bar, select **Settings**.
3. Select the **Generative AI** tab.
4. Under **How should your copilot decide how to respond?** select **Generative (Preview)**.
5. Select **Save** on the toolbar.



ⓘ Important

Using generative AI to determine how to respond to users 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

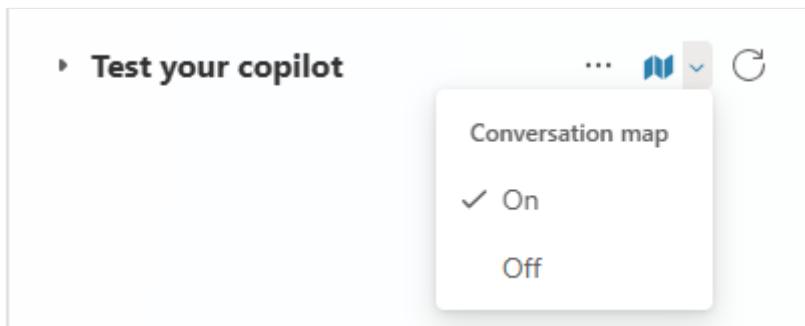
When using AI-generated content, such as the generated questions when running an action action, it's a good idea to inform your users that some of the conversation may be generated by AI.

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 bot.

Open the conversation map

The conversation map is available for copilots configured to respond to users in generative mode.

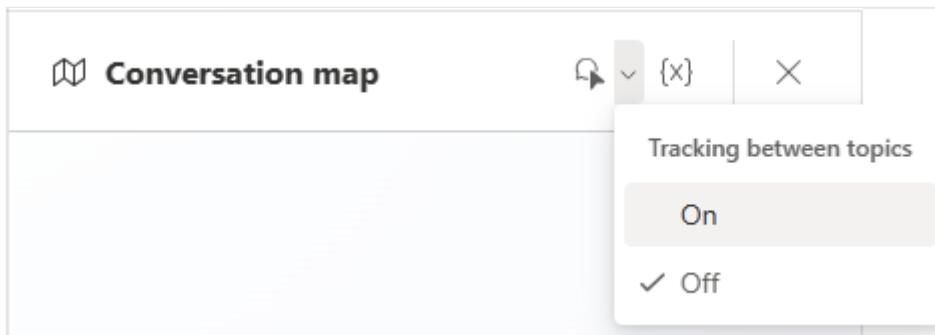
- To open the conversation map, select the map icon  at the top of the test pane.
- To make the conversation map automatically appear when you send a query to your copilot in the test pane, use the down arrow and select **On**.



When you ask your copilot a question in the test pane, the conversation map shows a visual representation of the plan that was generated. Here, the user asked, "what are the current weather conditions?", and you can see the action that was selected to respond and the input that still needs to be collected, along with annotations from the copilot about its decision making.

The screenshot shows the Microsoft Copilot Studio interface. On the left, there's a 'Conversation map' section with a 'Get current weather' connector node highlighted. The node has two input fields ('Units' and 'Location') and 22 output fields ('Latitude', 'Longitude', etc.). One input field for 'Units' is marked as 'Missing required information'. On the right, there's a 'Copilot context' panel with a test chat window. The user asks 'what are the current weather conditions?' and the copilot responds with 'Hello, I'm Contoso Copilot, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI.' Below the chat, there are buttons for 'Ask a question or describe what you need' and a search icon.

When you use the conversation map, the option to track between topics is available at the top of the conversation map. With tracking enabled, when a topic is triggered as part of a plan, the nodes within the topic appear on the conversation map as they are executed, allowing you to keep track of where you are in the conversation.



Authoring descriptions

If you selected **Generative** for how your copilot should decide to respond, it's important to provide a high-quality description for each of the actions and topics in your copilot, to ensure your copilot selects the right topics and actions to respond to users.

For actions, authoring a description is part of the wizard used to add them to the copilot. The description is often prepopulated for you, but you can make changes as appropriate. To learn more about adding and managing actions, see [Create actions for custom copilots](#).

For topics, once **Generative** is selected, **Triggered by copilot** appears on **Trigger** nodes (instead of **Phrases**), which allows you to add or edit the description for the topic.

The screenshot shows the Copilot Studio interface. At the top, there's a header with a blue icon and the word "Trigger". Below it is a section titled "Triggered by copilot (preview)" with a brain icon and an "Edit" button. A large text box contains the placeholder text: "This tool can handle queries like these: weather for tomorrow, what's the forecast for tomorrow, tell me about tomorrow's weather, what will the weather be like tomorrow, can you give me the weather update for tomorrow".

💡 Tip

When you turn on generative orchestration of topics and actions, 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 for topic and action names and descriptions

Here are some best practices for naming topics and actions, 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."

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 is 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 disambiguate between similar topics and actions.

For example, if your copilot already has an action that provides information about current weather conditions, but you want to add another topic that provides the weather forecast for tomorrow, make sure the names and descriptions are specific enough to avoid ambiguity. To disambiguate the action and the topic, you should give each 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 in this case.

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 a copilot using generative AI to handle user requests

The following examples are based on a copilot 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

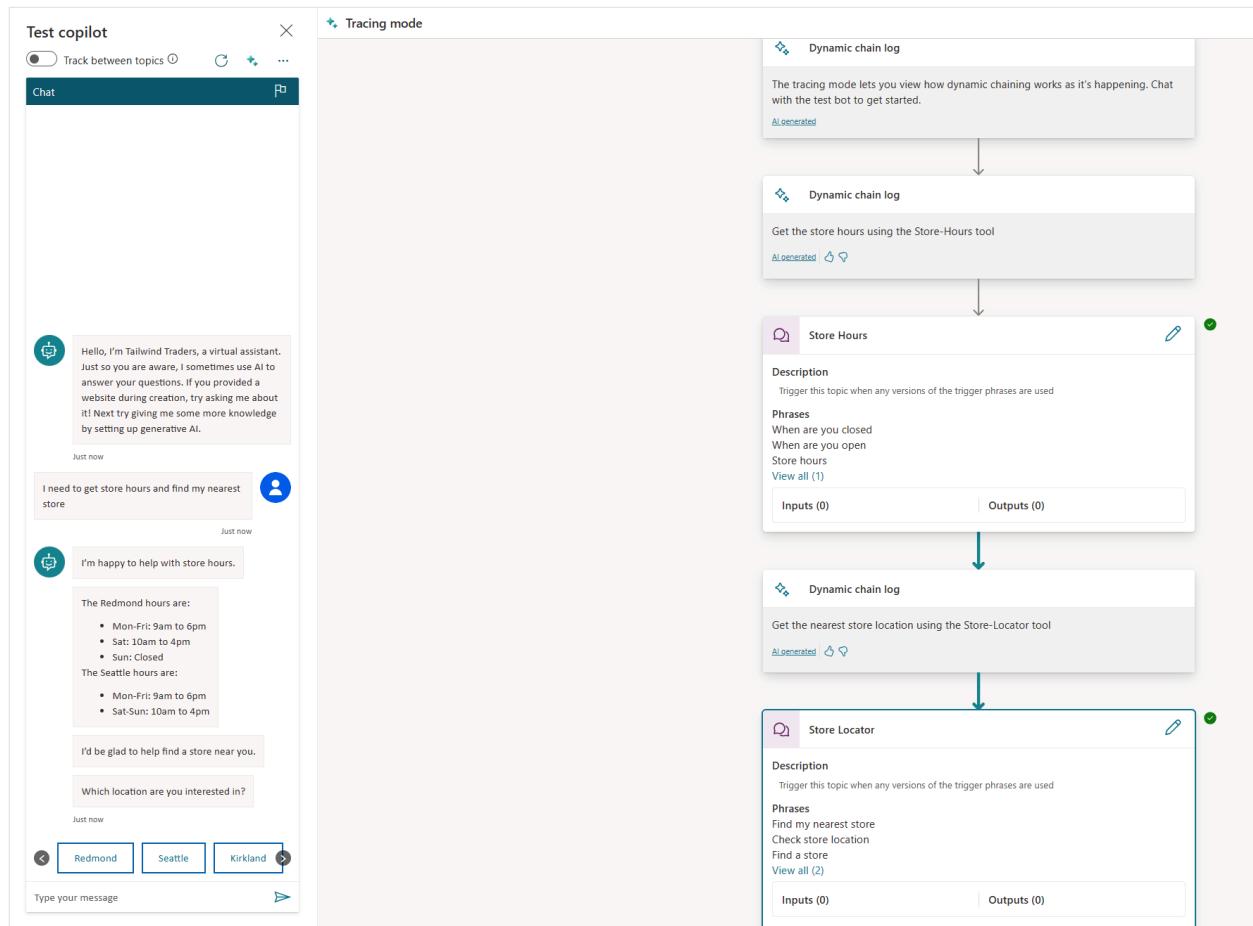
In this example, the user asks, "What is the weather like in Seattle?". The copilot selects the current weather plugin action and also prepopulates the Location input with "Seattle," which it recognized from the user's question.

The screenshot shows the Microsoft Copilot interface with the following components:

- Conversation map:** A sidebar on the left containing two "Copilot context" cards. The top card says "The conversation map enables you to see what's happening in the conversation and why it's happening. To get started, ask your copilot a question." and "AI-generated content may be incorrect". The bottom card says "Get the weather forecast for Seattle using Get-forecast-for-today (Location has value from _query). **no disambiguation**" and "AI-generated content may be incorrect".
- Test your copilot:** A main window on the right with a message history:
 - A copilot message: "Hello, I'm Your copilot, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI." (3 minutes ago)
 - A user message: "What is the weather like in Seattle?" (2 minutes ago)
 - A copilot response: "The weather in Seattle is mostly cloudy during the day with a wind speed of 10.0 and a 13% chance of precipitation at night." (2 minutes ago)
 - A user message placeholder: "Ask a question or describe what you need" with a character count of "0/2000".
- Bottom right corner:** A circular button with a plus sign and a magnifying glass icon, with the text "Make sure AI-generated content is accurate and appropriate before using. [See terms](#)".

Example 2

In this example, the user asks, "I need to get store hours and find my nearest store." Here, the copilot 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.



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 copilot selects the current weather action, but prepopulates the location with "Kirkland," based on the recent conversation context.

The screenshot shows the Tailwind Traders AI interface. On the left, a sidebar titled "Test copilot" has a "Track between topics" toggle and a "Chat" button. The main area displays a conversation:

- User: Hello, I'm Tailwind Traders, a virtual assistant.
- AI: 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.
- User: Just now
- User: find my nearest store
- AI: Just now
- AI: I'd be glad to help find a store near you.
- User: Which location are you interested in?
- AI: Just now
- User: Kirkland
- AI: Just now
- AI: Our Kirkland is located in downtown Kirkland. The address is: 1234 Central Way, Kirkland, 98321.
- User: Just now
- User: what's the weather like there?
- AI: Just now
- AI: The weather in Kirkland, WA is currently foggy with a temperature of 34°F. The visibility is 2.4 miles and the wind speed is 2.0 mph.
- User: Just now

At the bottom, there is a text input field: "Type your message" with a send button. On the right, a "Tracing mode" section titled "Dynamic chain log" shows the flow of the AI's responses:

- Step 1: "Hello, I'm Tailwind Traders, a virtual assistant." (AI generated)
- Step 2: "I'd be glad to help find a store near you." (AI generated)
- Step 3: "Our Kirkland is located in downtown Kirkland. The address is: 1234 Central Way, Kirkland, 98321." (AI generated)
- Step 4: "The weather in Kirkland, WA is currently foggy with a temperature of 34°F. The visibility is 2.4 miles and the wind speed is 2.0 mph." (AI generated)

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Manage topic inputs and outputs

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.

[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 copilot uses [generative AI to decide how to respond to users](#), it can automatically fill inputs, before running the topic, by using conversation context or generate questions to collect values from the user's response. This behavior is similar to how generative slot filling for actions works.

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 copilot uses generative AI to decide how to respond to users, more properties are available for input variables. Specifically **How will the copilot fill this input?** determines how the copilot fills this variable with a value before running the topic. By default, each input is set to **Dynamically fill with the best option**. The copilot 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 copilot 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 copilot captures the correct type of value for the input, you must also specify an entity type under **Identify as**.

The screenshot shows the 'Topic details' page in the Microsoft Copilot interface. The 'Input' tab is selected. Key visible elements include:

- Title:** A field containing 'Title' with a trash icon.
- Variable name:** A field containing 'Title'.
- How will the copilot fill this input?**: A dropdown menu set to 'Dynamically fill with best option (default)'.
- Variable data type:** A dropdown menu set to 'String'.
- Display name:** A field containing 'Title'.
- Identify as:** A dropdown menu set to 'User's entire response'.
- Description:** A text area containing 'The title of the ticket to be created.'
- Additional settings:** A section with a link to 'Additional settings'.
- Create a new variable:** A button at the bottom left.

There are a number of [additional settings](#) that can be configured for inputs, that can control the copilot's behavior when asking for a value or validating a user's answer.

One of these properties, **Should prompt user**, is only available for topic inputs. It determines if the copilot 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 • 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.

When you configure [inputs for topics](#) or [inputs for actions](#), common settings allow you to determine how your copilot gathers input.

💡 Tip

The additional settings are always available for action inputs. However, for topic inputs, these settings are only available if your copilot is configured to use generative AI to determine how to respond to a user.

The following additional settings are available.

Copilot prompts user for input

By default, the copilot will use the name and description for the input to generate an appropriate question to prompt the user for a value. However, you can use the **Copilot prompts user for input** setting to override the question.

To override and provide your own question, tick **Customize**. You can then author a question, including adding references to variables or Power Fx formulas.

② Copilot prompts user for input

Configure how the copilot 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?

Copilot doesn't identify entity

Retry prompt 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](#).

- **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 question used to ask a user again for input, where the basic entity validation has failed, select **Customize**, and then enter the new question. e.g. If you were expecting a number, you might enter a question like
`What is the ID? This 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. e.g. If you received a number, but it was above 10, breaking the validation rule, you might enter a question like `What is the ID? This needs to be less than 10..`

No valid entity found

No valid entity found determines what happens when your copilot stops trying to get a valid response from the user, because the maximum retry count has been reached. You can escalate to a human agent 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. e.g. Your message might say `Looks like you're struggling. I am redirecting you to one of our team.`

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Knowledge in Copilot Studio

Article • 05/21/2024

Knowledge in Microsoft Copilot Studio allows you to add enterprise data from Power Platform, Dynamics 365 data, and external systems, so your copilots provide relevant information and insights for your end users. In addition, knowledge can be incorporated with [Generative answers](#) in copilots. Published copilots that contain knowledge use the configured knowledge sources to ground the published copilot.

Supported knowledge sources

[+] [Expand table](#)

Name	Source	Description	Number of inputs supported in general answers	Authentication
Public Website	External	Searches the query input on Bing, only returns results from provided websites	4 public URLs (for example, <i>microsoft.com</i>)	None
Documents	Internal	Searches documents uploaded to Dataverse, returns results from the document contents	Limited by Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	4 URLs	Copilot user's Microsoft Entra ID authentication
OneDrive	Internal	Connects to a OneDrive URL, uses GraphSearch to return results	4 URLs	Copilot user's Microsoft Entra ID authentication
Dataverse	Internal	Connects to the connected Dataverse environment and uses retrieval-augmented generative technique in Dataverse to return results	Two Dataverse knowledge sources (and up to 15 tables per knowledge source)	Copilot user's Microsoft Entra ID authentication
Enterprise data via graph	Internal	Connects to the connected Dataverse	Two per custom copilot	Copilot user's Microsoft Entra ID

Name	Source	Description	Number of inputs supported in general answers	Authentication
connections		environment and uses retrieval-augmented generative technique in Dataverse to return results		authentication

ⓘ Note

Copilot user authentication for information sources means that when a specific user asks a question of the copilot, the copilot only surfaces content for that specific user to access.

ⓘ Note

Knowledge sources in generative answers don't support external resources, such as [Bing Custom Search](#) or internal resources, such as custom data.

Interact with knowledge sources

Knowledge sources can be incorporated into copilots during their initial creation, added after the copilot is created, or added to a generative answers topic node. The following tables describe how you can interact with knowledge sources during these processes.

Knowledge entry points

[+] [Expand table](#)

Knowledge source	Conversational creation	Creation configuration	Post-Creation configuration	Knowledge page	Topics node
Public website	Yes	Yes	Yes	Yes	Yes
SharePoint / OneDrive	Yes	Yes	Yes	Yes	Yes
File Upload	No	Yes	Yes	Yes	Yes
Dataverse	No	Yes	Yes	Yes	Yes

Knowledge source	Conversational creation	Creation configuration	Post-Creation configuration	Knowledge page	Topics node
Graph Connectors	No	Yes	Yes	Yes	Yes

Knowledge experiences

[\[\] Expand table](#)

Knowledge source	Create	Open/Download	Delete	Edit	Generative answers
Public website	Yes	Yes	Yes	Yes: All - such as Name, Description, Web page link	Yes
SharePoint/OneDrive	Yes	Yes	Yes	Yes: All - such as Name, Description, Web page link	Yes
File Upload	Yes	Yes	Yes	Yes	Yes
Dataverse	Yes: Name, Description, Table Selection, Synonyms, Glossary	No	Yes	Yes	Yes
Graph Connectors	Yes	No	Yes	No	Yes

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Add knowledge to your new copilot

Article • 05/21/2024

You can add knowledge to a copilot while you're creating it. While responding to the prompts to create your copilot, you can specify the following types of knowledge sources:

- A public website
- SharePoint and OneDrive

After you enter the URL to the knowledge source, your copilot automatically adds the knowledge source. After the copilot is created, the knowledge source is displayed in the **Configuration** and **Knowledge** pages.

The screenshot shows the Microsoft Copilot interface for creating a new copilot named "Gold Coast Travel Buddy".

Top Navigation: Configuration, Knowledge, Actions, Topics, Analytics, Channels

Modal Message: Your copilot is ready! Here's what's next:

- Add actions so your copilot can do things for you
- Build topics to focus and guide how your copilot answers
- Publish your copilot so others can use it

Details Section:

- Name:** Gold Coast Travel Buddy
- Description:** Assists travelers by providing guidance, information, and support throughout their travel journey.
- Instructions:** Your name is Gold Coast Travel Buddy. You are a custom Copilot that guides, informs, and supports travelers throughout their journey to the Gold Coast. Maintain a friendly and playful tone while providing travel information about the Gold Coast. There are no blocked topics for discussion.

Knowledge Section:

- Add knowledge:** + Add knowledge
- Allow the AI to use its own general knowledge (preview):** Enabled
- Knowledge Sources:** https://www.destinationgoldcoast.com/ (highlighted with a red box)
- See all:** See all

[+ Add knowledge](#) Search knowledge

All

[Public website](#)

Last refreshed now

Name	Type	Last modified	Status
https://www.destinationgoldcoast.com/	Public website	Eric Kinser...	Ready

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

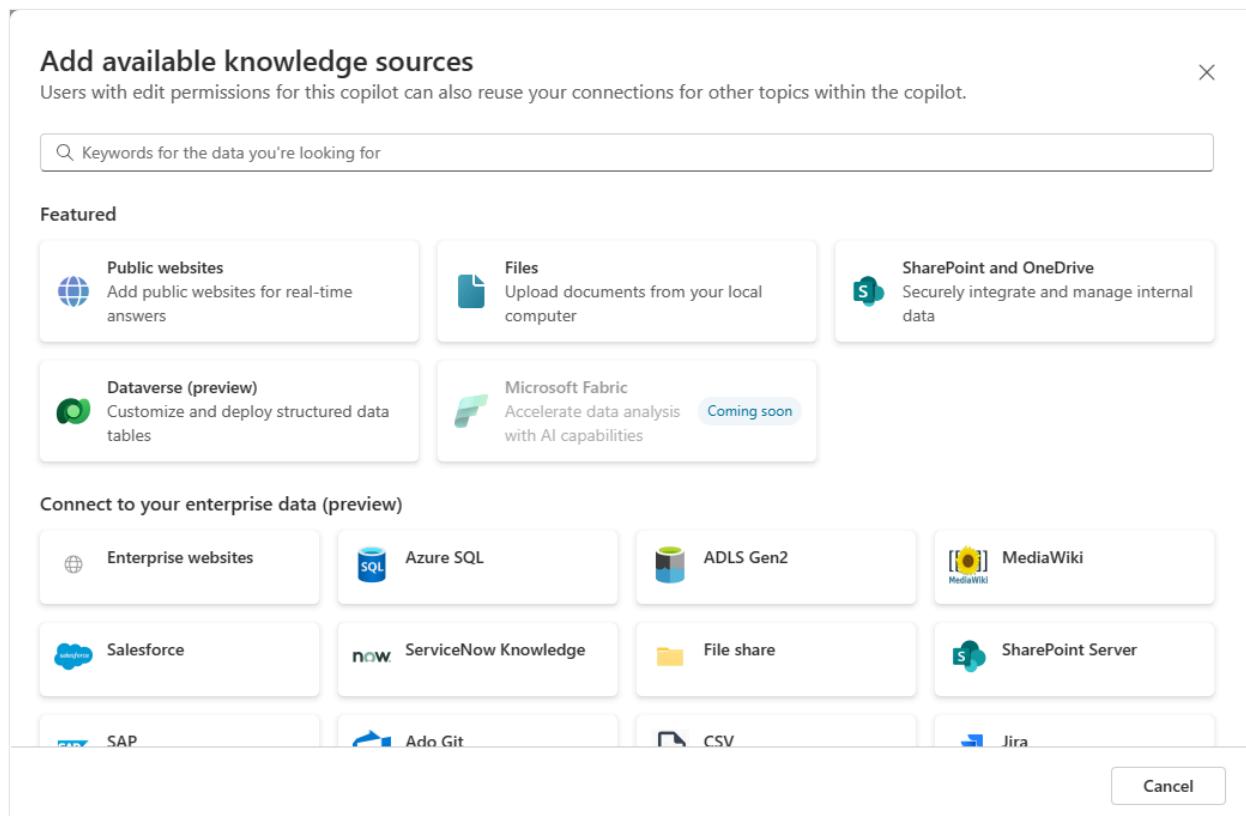
Add knowledge to a copilot

Article • 05/21/2024

You can add a knowledge source to a copilot from the following areas in Copilot Studio:

- The **Configuration** page
- The **Knowledge** page
- A generative answers topic

Select the **Add knowledge** button to launch the **Add available knowledge sources** dialog.



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).

- Public website
- File upload
- SharePoint and OneDrive
- Dataverse

ⓘ Note

Graph connection knowledge sources obtain their knowledge name and description from the graph connection name and description. These are provided

by the tenant admin, and are configured in Microsoft 365.

Public website

ⓘ Note

For more information on how to add a public website with a generative answers topic, see [Search public data or use a Bing Custom Search for generative answers](#).

1. Select **Public websites**.
2. Provide the URL to the public website.
3. Add a name (by default, the website's URL is used).
4. Add a description.
5. Select **Add** to finish adding the knowledge source.

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.

Webpage link

Link	Name	Description
https://www.australia.com/en	official Australia travel site	This knowledge source searches information ...

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](#)

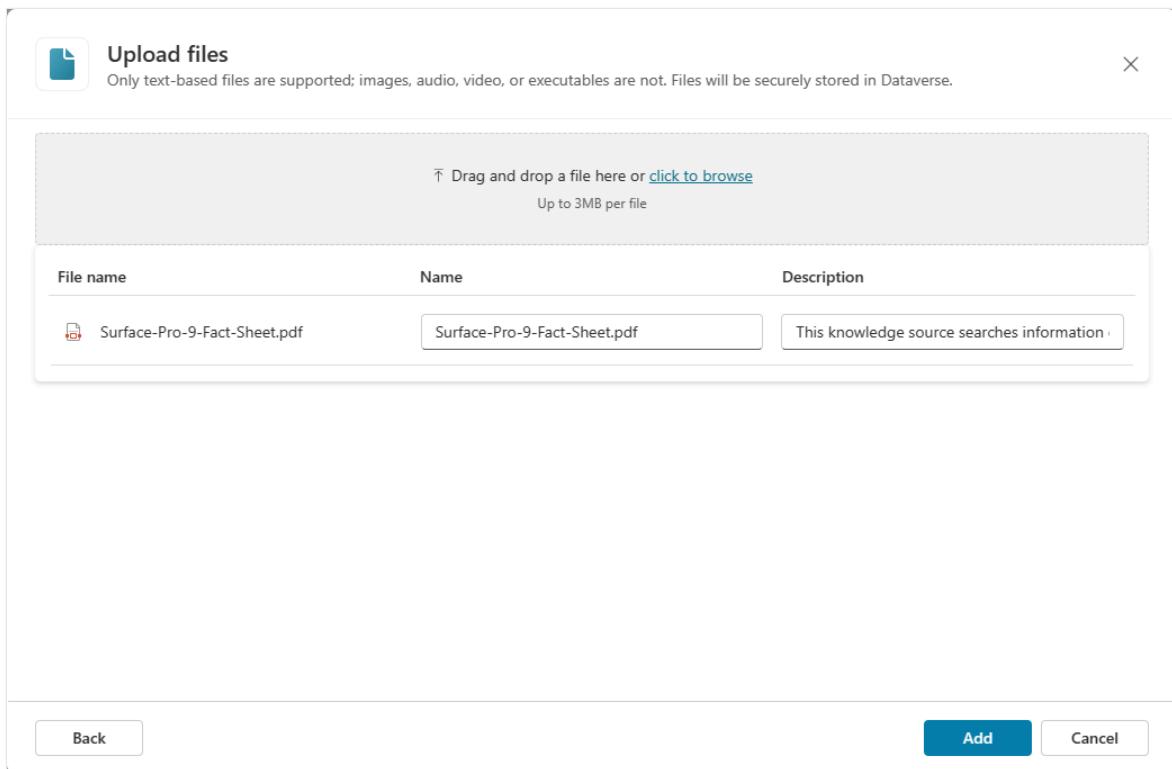
[Back](#) [Add](#) [Cancel](#)

File Upload

ⓘ Note

For more information on how to upload a file for use with a generative answers topic, see [Use uploaded documents for generative answers](#).

1. Select Files.
2. Upload the file by either drag and drop, or browsing to the file location.
3. Add a name (by default, the file's name is used).
4. Add a description.
5. 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)

File types, such as images, video, executable files, and audio, can't be used as uploaded documents.

SharePoint and OneDrive

ⓘ Note

For more information on how to use SharePoint or OneDrive content with a generative answers topic, see [Use content on SharePoint or OneDrive for Business for generative answers](#).

1. Select **SharePoint and OneDrive**.
2. Provide the SharePoint or OneDrive URL.
3. Add a name (by default, the file's name is used).
4. Add a description.
5. Select **Add** to finish adding the knowledge source.

The screenshot shows a modal dialog titled "Add SharePoint and OneDrive". It includes a note that "SharePoint and OneDrive will only be available to authenticated end-users." A "Learn more" link is provided. The "Webpage link" section has a placeholder "Enter a link" and an "Add" button. Below it is a table with columns "Link", "Name", and "Description". A single row is shown with the Link "https://pvabugbash.sharepoint.com/" and the Name "pvabugbash.sharepoint.com". The Description field contains the text "This knowledge source provides information!". At the bottom are "Back", "Add", and "Cancel" buttons.

Link	Name	Description
https://pvabugbash.sharepoint.com/	pvabugbash.sharepoint.com	This knowledge source provides information!

Dataverse

1. Select Dataverse (preview).
2. Locate one or more Dataverse tables to add. To narrow your selections, use the search field.

Step 1 of 3: Select Dataverse tables
Select up to 15 tables.

All Selected (1)

Keywords for the data you're looking for

Recommended by Copilot

AI-generated content may be incorrect

AI Copilot

Available tables

- AI Builder Dataset
- msdyn_AIBDataset
- AI Builder Dataset File
- msdyn_AIBDatasetFile
- AI Builder Dataset Record
- msdyn_AIBDatasetRecord
- AI Builder Datasets Container

Back Next Cancel

⚠ Note

The table recommendation is based on the name of your copilot.

3. Preview the tables to ensure the appropriate tables were added. The preview only displays 20 rows and a set of columns.

 Step 2 of 3: Preview data

Preview the first 20 available rows and columns from your selected tables.

Select a table to review

AICopilot

Created On	Description	Display Name	Is Managed	Modified On
2024-04-18T08:12:48Z	Copilot For Microsoft 365	Copilot For Microsoft 365	true	2024-05-10T09:39:24Z
2024-04-18T08:12:48Z	Copilot For Sales	Copilot For Sales	true	2024-05-10T09:39:25Z

Back **Next** **Cancel**

4. Review the knowledge name and description.
 - a. Optionally, to help improve the quality of the answers, add synonyms to columns of the tables that you selected. Select the **Back** button to accept changes.

 **Synonyms**

Improve data discovery with synonyms
Increase your copilot's data retrieval accuracy by adding synonyms for complex or unclear column names in your tables.

Select table

AICopilot

Column name	Synonyms	Description
Created By	+ Add synonyms	Enter description
Created On	+ Add synonyms	Enter description
Created By (Delegate)	+ Add synonyms	Enter description

Back **Cancel**

- b. Optionally, to help improve the quality of the answers, add a glossary to knowledge sources. Select the **Back** button to accept changes.

The screenshot shows a 'Glossary' section within a larger setup wizard. At the top, there's a header with a green icon and the word 'Glossary'. Below it, a sub-header says 'Build your data glossary' with the sub-instruction 'Enhance your copilot's understanding and responses by defining domain-specific terms and acronyms.' There are two input fields: 'Enter term' and 'Enter description', followed by a 'Next' button. A message at the bottom states 'No glossary terms yet. Add a term to improve data discovery.' At the bottom of the screen are 'Back' and 'Cancel' buttons.

5. Select **Add** to finish adding the knowledge source.

The screenshot shows the final review step of the setup wizard. It's titled 'Step 3 of 3: Review and finish' and includes instructions to 'Review your data selections and provide a name and description to help your copilot access the data.' Under 'Knowledge details', there's a 'Knowledge name' field containing 'AICopilot' and a 'Knowledge description' field containing 'This knowledge source answers questions found in the following dataverse tables: AICopilot'. Below these are sections for 'Synonyms' (0 synonym(s) in 0 column(s) in 0 tables(s)) and 'Glossary' (0). Both have 'Edit' buttons. At the bottom, under 'Selected knowledge sources', there's a list with one item: 'AICopilot' (with an '@' icon). At the very bottom are 'Back', 'Add' (highlighted in blue), and 'Cancel' buttons.

Enterprise data

ⓘ Note

For more information about enterprise data, see [Use graph connections with knowledge sources](#).

1. Choose from one of the supported data sources:

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

① Note

- Graph connection(s) that have been setup by your tenant admin are displayed here.
- If you want to create a new connection, or your graph connection hasn't been setup yet, select **Create Actions** to use Actions instead.

2. Add a name (by default, the connection's name is used).

3. Add a description.

4. Select **Add** to finish adding the knowledge source.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Use graph connections with knowledge sources

Article • 06/12/2024

With knowledge in Microsoft Copilot Studio, makers can ground their copilot responses with enterprise data available through Microsoft graph connectors. With graph connectors, customers can use the same data sources used to augment Microsoft Search to ground copilot responses.

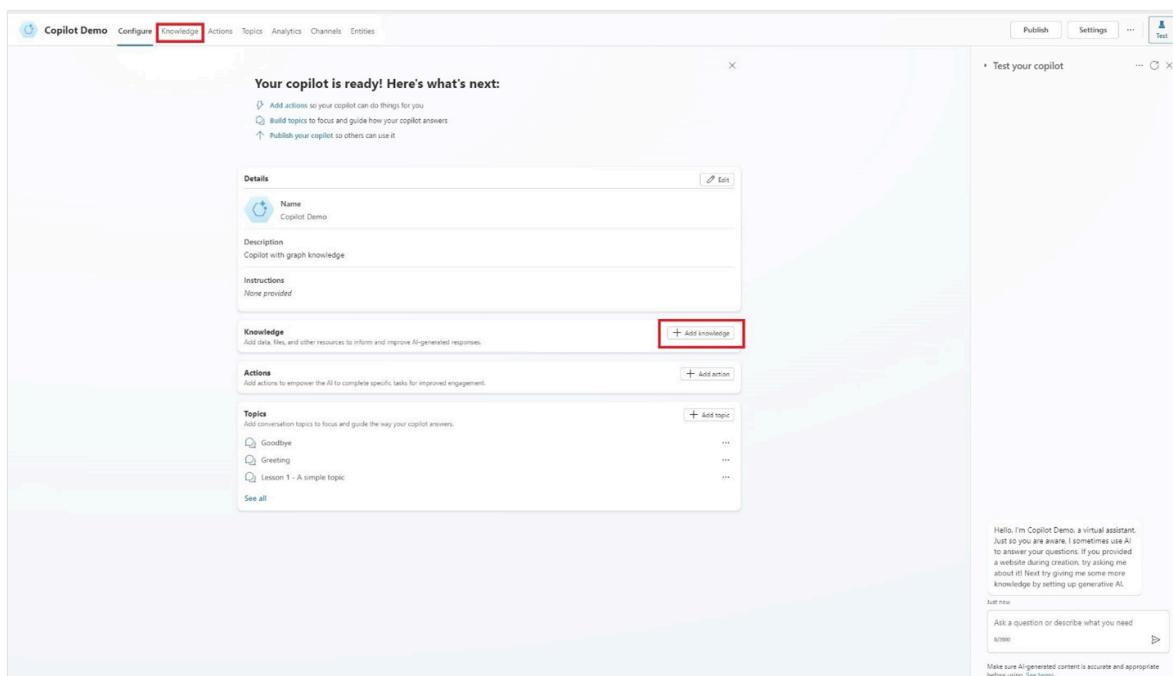
ⓘ Note

- 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.

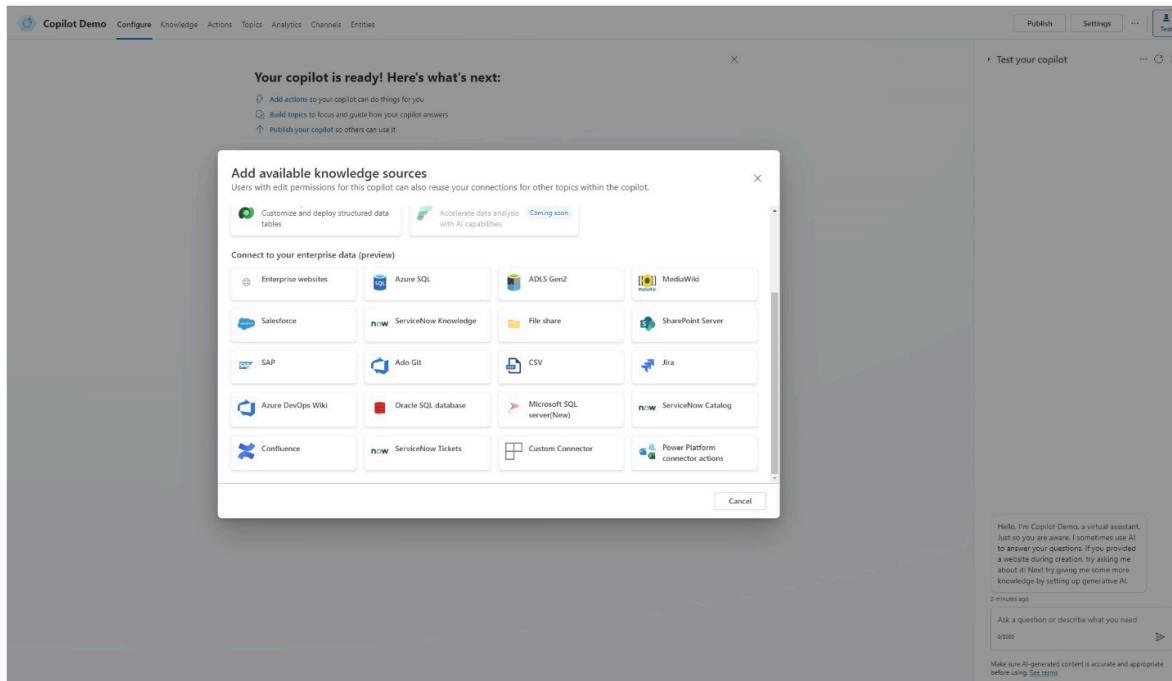
With Copilot Studio, users can select any graph connection previously configured in the tenant by the admin, and add them as knowledge to their copilot.

Add graph connections to your copilot

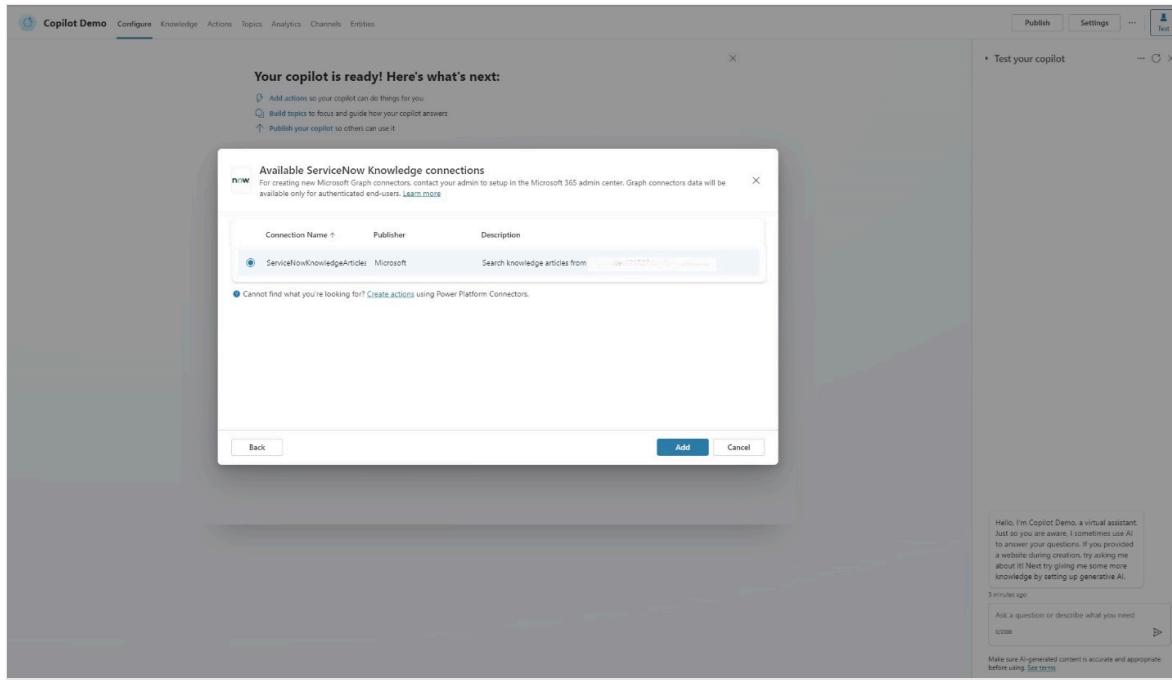
- Open the copilot.
- Select **Add knowledge** from either the **Configure** or **Knowledge** pages.



- From the Add available knowledge sources dialog, select the enterprise connection.



- Select the graph connection available within your tenant. In this example, the copilot maker added a ServiceNow knowledge connection, previously configured by the tenant admin.



- Press **Add** to complete the connection.

! Note

Copilots that use graph connections 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 Microsoft Copilot Studio](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Edit or view details of knowledge

Article • 05/21/2024

1. Select your copilot.
2. Open the Knowledge page.
3. Select the ellipsis of the desired knowledge source, and select **Edit** from the dropdown list.

The screenshot shows the 'Knowledge' section of the PC Guide interface. A file named 'Surface-Pro-9-Fact-Sheet.pdf' is listed. An ellipsis menu is open over the file, with the 'Edit' option highlighted by a red box. The menu also includes 'Download' and 'Delete' options. Other columns in the table include 'Name', 'Type', 'Last modified', and 'Status'.

Name	Type	Last modified	Status
Surface-Pro-9-Fact-Sheet.pdf	Files	Eric Kinser 18 h...	Ready

4. Modify the knowledge source details as needed.

The screenshot shows the 'Edit' page for the knowledge source 'Surface-Pro-9-Fact-Sheet.pdf'. It includes fields for 'Knowledge name' (Surface-Pro-9-Fact-Sheet.pdf), 'Knowledge description' (This knowledge source searches information contained in Surface-Pro-9-Fact-Sheet.pdf), and 'File detail' (Surface-Pro-9-Fact-Sheet.pdf). To the right, there are read-only details: Type (Files, PDF File), Modified by (Eric Kinser 18 hours ago), Status (Ready).

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 and OneDrive	Yes	Yes	URL is editable
Dataverse	Yes	Yes	Tables, Synonyms, and Glossary are editable
Enterprise Data via Graph Connections	-	-	-

 **Note**

A published copilot might be using the latest unpublished updates of a Dataverse knowledge source.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Test your copilot's knowledge sources

Article • 05/21/2024

Testing a copilot with knowledge sources automatically grounds that copilot with those knowledge sources.

After adding knowledge sources, use **Test your copilot** to see if it's working in copilot. Only knowledge sources with a **Ready** status can be used in **Test your copilot**.

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 copilot .
Preparing	This knowledge source isn't ready for you to use in Test your copilot . 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 copilot .
Error	This knowledge source had issues while preparing.

Known issues

- A published copilot might be using the latest unpublished updates of a Dataverse knowledge source.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

AI-based copilot authoring overview

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.

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

Generative AI is an artificial intelligence technology that uses large language models (LLMs) to generate original content and provide natural language understanding and responses. For details, see Technology guidance > [Generative AI](#).

In Copilot Studio, you can use the following generative AI features to retrieve and create content, either individually or all together. For scenarios covering all these features, see the generative AI [Quickstart guide](#).

- **Create an instantly useful copilot.** With no manual authoring of topics required, an "empty" copilot can generate answers based on knowledge sources you specify such as websites and files. See [Generative answers](#).
- **Harness AI general knowledge.** When this option is enabled, the copilot 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 copilot 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 with Copilot Studio](#).
- **Enable generative actions.** Let the copilot select the most appropriate topics and actions (formerly known as plugins) at runtime. See [Generative actions](#).

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

Prerequisites

- An account for Microsoft Copilot Studio. If you don't have an account, see the [Microsoft Copilot Studio introduction website](#) and select Try free.
- The [current version of Copilot Studio](#). The copilot type shouldn't be Classic. Classic copilots have (classic) added to their name, for example "Contoso store hours (classic)."
- AI-based authoring might be subject to usage limits or capacity throttling.

Next steps

- Get up and running with the [generative AI quickstart guide](#).
 - Dive in and start using [generative answers](#) capability in your copilots.
 - Have a conversation with the AI to [author topics](#).
 - Use [generative actions](#) to call your actions automatically at runtime.
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Quickstart guide for building copilots with generative AI

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.

It's even simpler to build a copilot using AI-powered capabilities in Microsoft Copilot Studio. 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.

This quickstart helps you get started quickly to create a copilot with generative AI capabilities. For a summary of the available features, see the generative AI [overview](#).

Prerequisites

- An account for Copilot Studio. If you don't have an account, see the [Microsoft Copilot Studio introduction website](#) and select Try free.

ⓘ Note

Personal Microsoft accounts aren't supported. Use a work account instead.

- Supported browsers include Microsoft Edge, Chrome, and Firefox.
- Use the [current version of Copilot Studio](#).
- Your copilot must be created in the US region. View the list of [Supported languages](#).
- Generative AI might be subject to [usage limits](#) or capacity throttling.

How copilot conversations work

Copilot Studio uses [customized NLU model and AI capabilities](#) to understand what a user types, then responds with the best copilot topic. A copilot topic is a portion of a conversational thread between a user and the copilot. Topics are linked together to form nodes. For more information, see [Create and edit topics](#).

For example, you might create a copilot for your customers to ask common questions about your business. Your support overhead is reduced by deflecting support calls. In the copilot, 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 copilot uses natural language understanding (NLU) to understand the *intent* behind the question. The copilot matches that intent to the best topic, the **Store hours** topic.

The copilot 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 copilot linked to a public or Bing-indexed website, your copilot automatically generates responses. These responses are conversational, plain language, and don't depend on the copilot builder to create topics for every eventuality.

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

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

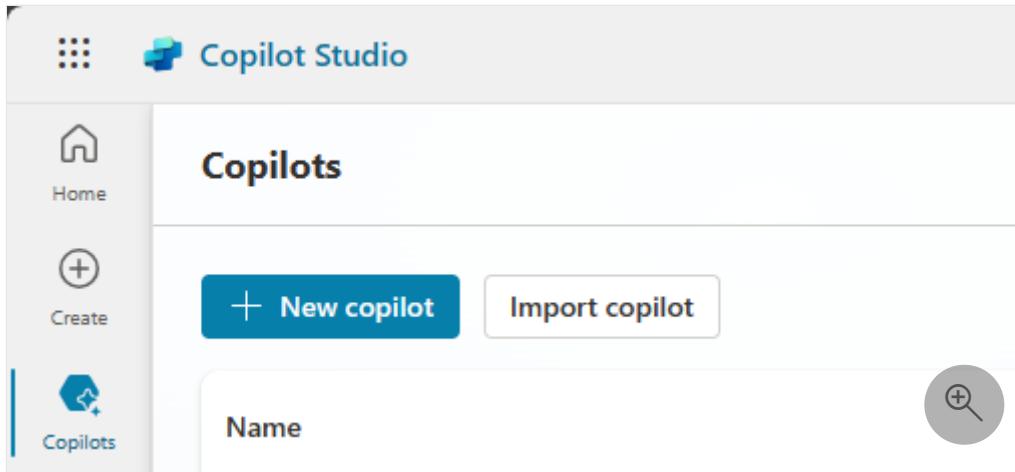
Ready to get started? The first step is to create your copilot.

Create a copilot with instant knowledge

For any user-sent message that can't be matched to an existing topic, your copilot can look for an answer on a fallback website or other knowledge source. The copilot turns the answer into a simple message that it sends to the user.

1. Go to the [Microsoft Copilot Studio home page](#).

2. Create a copilot. There are many ways to create a copilot but, for this quickstart, you might open the **Copilots** page, select **+ New copilot**, and then select **Skip to configure**.



s1. In the configuration page, enter a name for your copilot or accept the default, **Copilot** or **Copilot n**. Optionally add a description.

1. In the **Knowledge** section, select **+ Add knowledge** and then **Public websites**.
2. In **Add public websites**, enter the URL of a public or Bing-indexed website. This website is where the copilot can search for answers if it can't find the right topic for a user question. Select **Add** to save the URL and **Add** again to exit the window.

💡 Tip

Any URL with text works, for example <https://www.microsoft.com/en-us/store/>.

To add a URL or other knowledge sources after creating a website, go to the **Knowledge** page.

Copilot 2

Name * Copilot 2

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

Description 0/1000
Describe your copilot

Instructions 0/8000
Give instructions for your copilot

Knowledge (1) Add knowledge sources + Add knowledge https://www.microsoft.com/en-us/store/ ...

Create

3. Select **Create** to finish creating your copilot.

Copilot Studio displays an overview of the copilot you created, with ideas for what to do next.

Notice the **Knowledge** section, which shows the website you added. It also includes the option **Allow the AI to use its own general knowledge**, enabled by default, which lets the copilot answer questions outside the specific knowledge sources.

Details Edit

Name Copilot 2

Description None provided

Instructions None provided

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

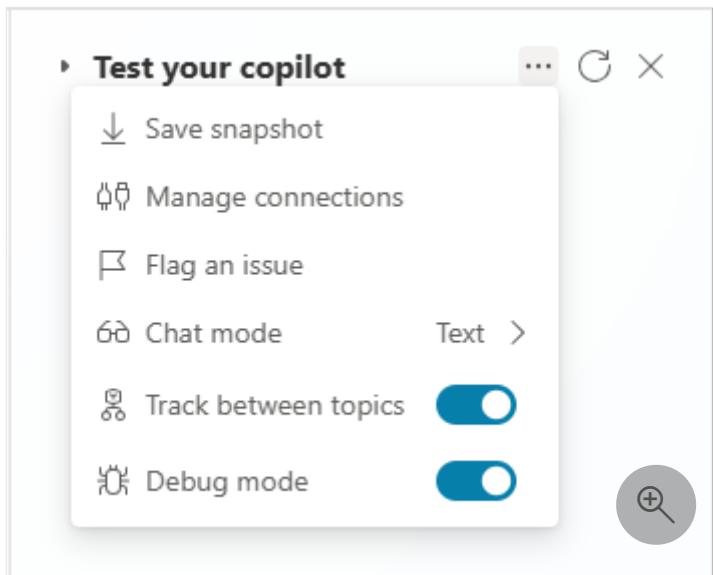
https://www.microsoft.com/en-us/store/ ...

See all

Test your copilot for knowledge

Your copilot is now ready to answer questions based on its knowledge sources. Use the **Test** pane to try it out:

1. Select the **Test** button at the top right of the Copilot Studio page.
2. To view the underlying action of the copilot while it's answering your questions, select ... in the **Test** pane and choose **Track between topics**.



3. In the **Ask a question** prompt, ask the copilot a question related to the website. For example, type *What is your return policy?* or *Do you give military discounts?*

The copilot retrieves information from the website and returns a response. The response provides one or more links to where it found that information.

In the authoring canvas, you can see the conversation flow and the nodes that the copilot used to answer your question. In this case, the copilot used the **Conversational boosting** system topic to find the information and respond to your question.

4. Now ask a general question, for example *Why is the sky blue?* or even a nonsense question such as *What color is the sky in your world?* If **Allow AI general Knowledge** is enabled, the copilot can answer questions like these.

It might take a few minutes for the AI general knowledge to be available. If you don't get an answer immediately, try again.

Why is the sky blue?

Just now

The sky appears blue due to a phenomenon called Rayleigh scattering. When sunlight passes through the Earth's atmosphere, it interacts with the molecules and particles in the air. The shorter blue wavelengths of light are scattered more than the longer red wavelengths. As a result, our eyes perceive the scattered blue light, making the sky appear blue. This effect is more pronounced when the sun is higher in the sky, which is why the sky often appears bluest around midday.

AI-generated content may be incorrect

What color is the sky in your world?

A minute ago

In my virtual world, the sky doesn't have a specific color as it can be customized based on personal preferences or the context of the virtual environment. It can be blue, gray, purple, or any other color you can imagine. Is there anything else I can assist you with?

AI-generated content may be incorrect



For AI general knowledge answers, the copilot reminds users that the information might not be accurate.

If the copilot can't find an answer, it prompts you to rephrase the question. After two prompts, if it still can't determine your intent, it escalates to a live agent through the **Escalate** system topic.

You notice that even general knowledge has its limits. Although the copilot can answer many questions, it doesn't have access to live information such as news or weather reports. You add this capability using generative actions later in this quickstart.

For more information, see [Generative answers](#) and [AI general knowledge](#). You can also find instructions for enabling generative AI in copilots you already created.

ⓘ Note

After you test your copilot, select the reset icon at the top of the **Test copilot** pane to clear previous conversations. Resetting makes it easier to follow the flow of the current topic without getting confused by previous conversations.

› **Test your copilot**



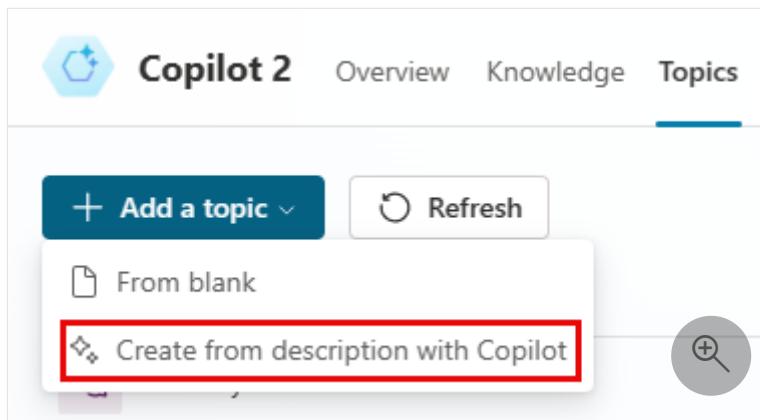
You can return to the authoring canvas at any time to revise the conversation path. The **Test copilot** pane automatically refreshes when you select **Save** after editing your topics.

If error codes appear in the test pane, see [Understand error codes](#).

Auto-create a new topic

You can create a new topic by describing what you want the topic to do. Copilot Studio uses your description to generate a topic with conversational responses and multiple types of nodes.

1. With your copilot open on the **Topics** page, select **+ Add a topic > Create from description with Copilot**.



ⓘ Note

If you don't see **Create from description**, you may need to enable it:

- a. On the top menu bar, select the **Settings** icon then **General settings**.
- b. Turn on **Build and iterate on your topics by describing what you want**.

2. In the window that appears, enter a name in **Name your topic**.
3. In the **Create a topic to...** field, describe your topic in simple, plain English. See the examples to understand the structure of a description and select **View more examples** for more ideas.

A screenshot of a modal dialog box titled 'Create from description with Copilot'. The dialog has a light gray background. At the top left is the title, and at the top right is a small 'X' button. Below the title is a instruction message: 'Write a description of what you'd like your copilot to cover, and Copilot will create your topic. [Learn more](#)'. There are two main input fields: 'Name your topic *' with a placeholder 'Find a job' and 'Create a topic to... *' with a placeholder 'Let the user ask about job opportunities for positions such as sales clerk, manager, packer, cashier, greeter.'. In the bottom right corner of the 'Create a topic to...' input field, there is a circular button containing a magnifying glass icon.

4. Select **Create** and then **Save**.

Once your topic is created, you can test it. In the **Test** pane, ask a question that matches the intent of the topic you created. For example, if you created a topic about job opportunities, enter *I'm looking for a job*.

You can explore ways to extend and enhance your topic in [Create and edit with Copilot](#).

Use generative actions

When generative actions are enabled, the copilot automatically chooses the topics and actions to respond to the user. Instead of relying on trigger phrases for each topic, or calling actions manually from within a topic, the copilot can use its own logic to select the most appropriate actions at runtime.

ⓘ Note

Generative actions 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.

1. Enable generative actions: select the **Settings** button then **Generative AI**. Choose the **Generative** option. (The default is **Classic**.)



2. Select **Save** at the top of the page.
3. Create an action: go to the **Actions** page and select **+ Add action**. Follow the instructions in [Use actions](#) to add a weather website.

**Copilot 2**

Overview Knowledge Topics Actions Analytics

+ Add an action**Refresh**

Name

Trigger

[MSN Weather - Get current weather](#) ...Triggered by copilot
(preview)

When you add this action and enable generative actions, the copilot has access to live information, as well as specific topics, specific knowledge sources, and AI general knowledge.

To see the action of the copilot as it answers your questions, select the **Conversation map** icon in the **Test** pane and turn it On. In the conversation map, select the **Tracking between topics** icon and turn it On.

The screenshot shows the Microsoft Copilot interface. At the top, there's a navigation bar with 'Copilot 2' and tabs for 'Overview', 'Knowledge', 'Topics', 'Actions' (which is underlined in blue), and 'Analytics'. Below this is a button to '+ Add an action' and a 'Refresh' button. The main area shows a table with one row: 'Name' (MSN Weather - Get current weather) and 'Trigger' (Triggered by copilot (preview)). To the right of the table is a search icon. Below the table, there's a text block explaining that adding this action and enabling generative actions gives the copilot access to live information and AI general knowledge. Further down, instructions say to turn on 'Conversation map' and 'Tracking between topics' in the 'Test' pane.

For example, ask the copilot *What's the weather in Montreal in fahrenheit?* The copilot uses the action you created to find the answer.

The screenshot shows a conversation history. The user asks 'What's the weather in Montreal in fahrenheit?' at 'Just now'. The copilot responds with 'The weather in Montreal is partly sunny with a high of 71°F and partly cloudy with a low of 54°F.' There is a magnifying glass icon next to the copilot's response.

Add features to further develop your copilot

You can enhance your copilot to make it unique and enjoyable. Try adding images and video clips, adaptive cards, entities, and variable expressions.

For more information, see [Key concepts - Enhanced authoring](#) or continue your copilot-building journey by exploring the rest of the [Microsoft Copilot Studio documentation](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Generative answers

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.

Generative answers in Microsoft Copilot Studio allow your copilot 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 copilot. You don't need to manually author multiple topics that might not address all customer questions.

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

Today, before involving a live agent, the copilot 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 and OneDrive for Business.
- Summarize search results into plain language delivered to the copilot user.

This article helps you get started, using generative answers as a fallback topic, when a user's intent can't be addressed by existing copilot topics.

ⓘ Note

[Generative answers with search and summarize](#) discuss how your copilot can query information sources, using generative answers in a single topic node.

Your workflow might look like:

1. You create a copilot and enable generative answers. You test it thoroughly.
2. After testing, you publish your copilot to instantly provide answers, help, and guidance to your copilot users.
3. You create individual topics for frequently asked questions. These topics might develop from [analytics from previous copilots](#) or existing support issues.

Generative answers as a fallback

When a copilot can't find a matching intent (topic) for the user's query, it uses generative answers and tries to answer the question. This behavior is known as generative answers for 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 copilot.

Generative answers aren't limited to fallback scenarios. Generative answers aren't limited to fallback scenarios. Your copilot can also use other web sites, external or internal web sources, AI general knowledge, and [knowledge sources](#) such as SharePoint or OneDrive.

Generative answers can use these sources:

- External resources:
 - AI general knowledge
 - [Bing Web Search](#) (doesn't require external configuration)
 - [Bing Custom Search](#) ↗ (requires external configuration)
- Internal resources:
 - SharePoint (only [specific file formats](#) are supported)
 - OneDrive
 - Documents uploaded to Dataverse
 - Custom data (internal or external): supply your own source, such as a Power Automate Flow or from Skill.

Note

You can expand your copilot's use of generative answers in [Generative Answers with Search and Summarize](#).

Source authentication

In addition to [URL considerations](#), you might need to authentication for your sources. For example, if you use an internal SharePoint site or OneNote as a source for **generative answers**.

For more information, see [Information sources](#).

Prerequisites

- An account for Microsoft Copilot Studio.

 **Note**

If you don't have a Microsoft Copilot Studio account, or you haven't created copilots before, see the [Quickstart guide for building copilots with generative AI](#).

- If you already have a copilot created, enable the **generative answers** option in the [Generative AI page](#).
- Review AI response generation training, model, and usage in the [FAQ for generative answers](#) and [Learn more about Azure OpenAI](#).
- Generative answers might be subject to usage limits or capacity throttling.

Increasing your copilot's reach

1. Go to the [Microsoft Copilot Studio home page](#).
2. Select **Create a copilot** on the **Home** page or **New copilot** from the **Copilots** page.
3. Enter a name for your copilot.
4. Select a language for your copilot.
5. Provide a website you'd like the copilot to use for generating answers.

See the [URL considerations](#) section for the types of URLs you can use.

Copilot 2

Name * **Copilot 2**

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

Description
Describe your copilot

Instructions
Give instructions for your copilot

Knowledge (1)
Add knowledge sources

<https://www.microsoft.com/en-us/store/>

+ Add knowledge

⋮

6. Select Create.

You now see your copilot's **Overview** page.

Change your URL or toggle generative answers

You can customize further, after your copilot is created, from the **Generative AI** page.

- To enable or disable generative answers, use the toggle in the **Boost conversational coverage with generative answers** section.
- To change the URL, add more websites or remove them in the **Websites** section.

Boost conversational coverage with generative answers
Customize your conversations with a node to get information in real time from a data source you choose. [Learn more](#) [Go to Conversational Boosting topic](#)

Websites

Enter your website	Add
https://www.microsoft.com/store/	...

Websites Data source

https://www.microsoft.com/store/ Public website

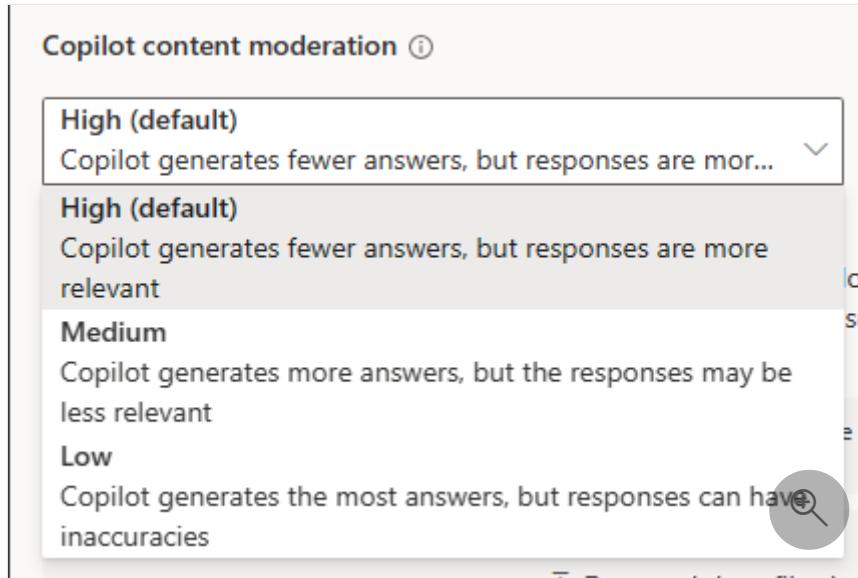
Content moderation

Adjust the content moderation settings from the **Generative AI** page.

1. Under **Copilot content moderation**, select the level you want for your copilot.

[+] Expand table

High	Medium	Low
Copilot's answers are more relevant. (default)	Copilot generates more answers, but might be irrelevant or undesirable.	Copilot generates the most answers, but might be inaccurate.



2. Select **Save** at the top of the page.

Test your copilot 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.

URL considerations

The URL used in your copilot 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
www.contoso.com	www.fabrikam.com/engines/rotary/dual-shaft
www.fabrikam.com/engines/rotary	
www.fabrikam.com/engines/rotary/	

- 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 copilot 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 copilot.

Examples of useful or unuseful domains:

- If you use www.fabrikam.com/engines/rotary, the content on www.fabrikam.com/engines/rotary/dual-shaft is also used by the copilot 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 network & forum URLs

Your copilot 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.

Test your copilot's generative answers reach

1. Select **Test your copilot** at the bottom of the navigation pane.
2. In the **Test copilot** pane, ask your copilot 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.

Forming questions

- Your copilot has difficulty answering questions that require calculations, comparisons, or form submissions. Your copilot might not understand comparative and superlative terms such as better or best, latest, or cheapest in a question.
- If the copilot can't generate an answer to a question, it prompts you to rephrase the question. After two of these prompts, the copilot initiates the [system Escalate topic](#). System topics are topics automatically included with each copilot.
- 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 copilot. They're used before any URLs are accessed.

Select the  next to a sample topic on your **Topics** page and use the **Status** toggle to enable or disable the topic.

Topics

The screenshot shows the 'Topics' section of Microsoft Copilot Studio. There are 8 topics listed: 'Goodbye', 'Greeting', 'Lesson 1 - A simple topic', 'Lesson 2 - A simple', 'Lesson 3 - A topic w...', 'Microsoft store top...', and 'Start Over'. A context menu is open over the fourth item, 'Lesson 2 - A simple', with options: 'Details', 'Make a Copy', 'Status' (which is highlighted with a red box), and 'Delete'.

Icon	Name	Trigger	Description
Q	Goodbye	Phrases	This topic trigger
Q	Greeting	Phrases	This topic is triggered by a greeting
Q	Lesson 1 - A simple topic	Phrases	
Q	Lesson 2 - A simple		
Q	Lesson 3 - A topic w...		
Q	Microsoft store top...		
Q	Start Over	Phrases	

For more information, see [Use lesson topics in Microsoft Copilot Studio](#).

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'll use this feedback to improve the quality of the AI.

What's supported

Quotas

Quotas are default constraints applied to copilots that limit how often messages can be sent to a 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.

Copilots 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 copilot topics follow the [usual quotas and limitations](#).

Pricing

The use of the boosted conversations capability isn't billable and follows the [usual quotas and limitations](#).

Languages

See the [Supported languages](#) section for the list of supported languages by this feature and their respective stage.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

AI general knowledge

Article • 06/04/2024

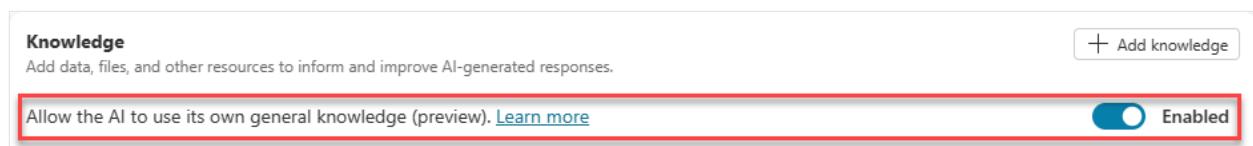
In addition to generative answers, you can use AI general knowledge to allow your copilot to find and present information in response to your customer's questions.

General knowledge saves you from needing to manually author multiple topics, which might not even address all your customer's questions.

This article helps you get started using AI general knowledge, which helps when a user's intent can't be addressed by existing copilot 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.



Why use it

- **Accessibility:** The copilot 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 a variety of 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.

Use AI general knowledge as a part of generative answers for fallback

When a copilot can't find a matching intent (topic) for the user's query, it uses generative answers and tries to answer the question. This behavior is known as

generative answers for 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 copilot.

Generative answers aren't limited to fallback scenarios. Your copilot can also use other web sites, external or internal web sources, AI general knowledge, and [knowledge sources](#) such as SharePoint or OneDrive.

Generative answers can use these sources:

- External resources:
 - AI general knowledge
 - [Bing Web Search](#) (doesn't require external configuration)
 - [Bing Custom Search](#) ↗ (requires external configuration)
- Internal resources:
 - SharePoint (only [specific file formats](#) are supported)
 - OneDrive
 - Documents uploaded to Dataverse
 - Custom data (internal or external): supply your own source, such as a Power Automate Flow or from Skill.

Prerequisites

- An account for Microsoft Copilot Studio.

ⓘ Note

If you don't have a Copilot Studio account, or you haven't created copilots before, see the [Quickstart guide for building copilots with generative AI](#).

- Review AI response generation training, model, and usage in the [FAQ for generative answers](#) and [Learn more about Azure OpenAI](#).
- Generative answers might be subject to usage limits or capacity throttling.

Supported features

Quotas

Quotas are default constraints applied to copilots that limit how often messages can be sent to a 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.

Copilots 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 copilot topics follow the [usual quotas and limitations](#).

Languages

See [Supported languages](#) for the list of supported languages by this feature and their respective stage.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

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

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.

ⓘ Note

This article describes the classic method to add public data sources. For the updated method, see [Add knowledge to a copilot](#).

You can search publicly available data by using the Bing search engine API.

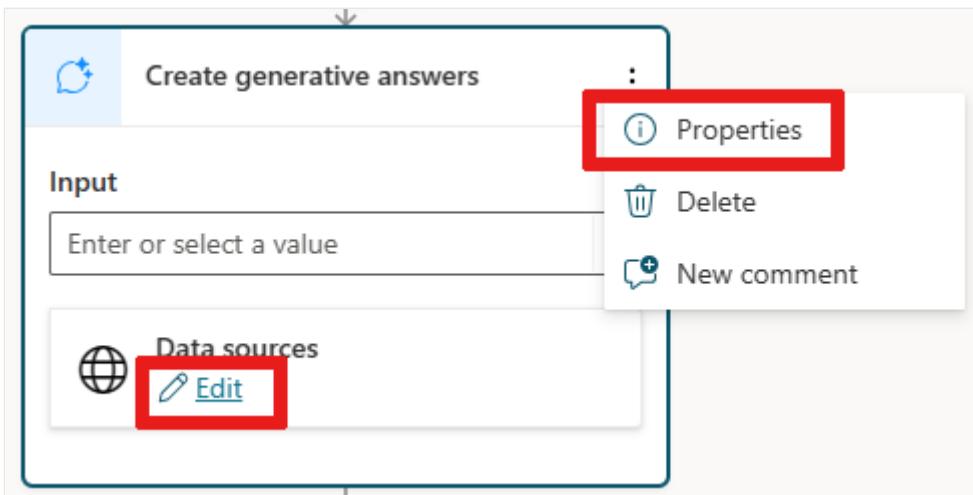
You can use the default Bing Search configuration or build a tailored search for specific content using [Bing Custom Search](#). Bing Search helps you boost your copilot by including many sources without configuring each source individually. To use Bing Custom Search as a source, you need a custom search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

3. Under **Search public data** you can choose 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, choose **Search public websites**.
- To use a Bing Custom Search, choose **Search with Bing Custom Search** and enter your Custom Configuration ID in the **Configuration ID** input box.

💡 Tip

To get the **Configuration ID**:

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

4. Save changes to your topic.

ⓘ Important

Information sources defined in the **Generative answers** node take priority at the copilot level. Copilot level sources function as a fallback.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#)

Connect your data to Azure OpenAI for generative answers

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.

When you connect data through Azure OpenAI Service in Microsoft Copilot Studio, your copilots 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](#).

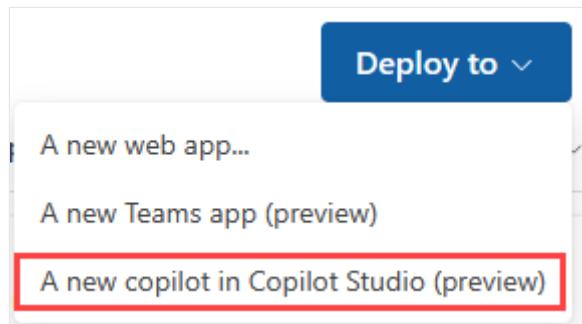
If you're building a production-ready copilot, see [Microsoft Copilot Studio Overview](#).

Prerequisites

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

The screenshot shows the Azure AI Studio interface with the 'Chat' tab selected in the sidebar. A modal window titled 'Assistant setup' is open, containing options for 'System message' and 'Add your data (preview)'. Below the modal, a note says 'Gain insights into your own data source. Your data is stored securely in your Azure subscription.' with a link to learn more about data protection. A checked checkbox says 'Limit responses to your data content'.

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



ⓘ Note

A **Conversational boosting** system topic is automatically generated when the copilot is created from Azure OpenAI Studio.

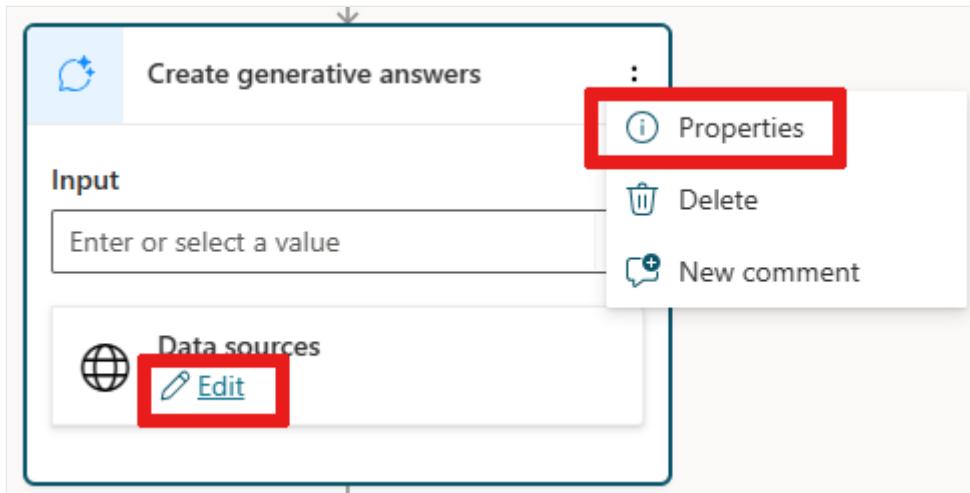
The screenshot shows the 'Topics' section of the Azure AI Studio interface. The sidebar has 'Topics' selected. The main area shows a table of topics under the 'System' tab. The table columns are 'Type', 'Name ↓', and 'Trigger ↑'. The rows show three topics: 'Conversation Start' (triggered on Conversation Start), 'Conversational boosting' (triggered on Unknown Intent), and 'End of Conversation' (triggered on Redirect). A note at the bottom states: 'System topics are the core of your bot. Without them, your bot will not function or handle errors properly. You can't delete these topics.'

boosted topic

Use Azure OpenAI in Copilot Studio

1. Open the **Data source** configuration pane from one of two places:

- On the **Create generative answers** node, select **Edit** under **Data sources**.



- 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 >

The 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' (with a 'Add connection' button and a 'Connection properties' box). There is also a 'Custom data' section with a text input field.

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

Information sources defined in the **Generative answers** node take priority at the copilot level. Copilot level sources function as a fallback.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use content on SharePoint or OneDrive for generative answers

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.

ⓘ Note

This article describes the classic method to add SharePoint or OneDrive sources. For the updated method, see [Add knowledge to a copilot](#).

This capability works by pairing your copilot with a URL, such as *contoso.sharepoint.com/sites/policies*. When a copilot user asks a question and the copilot doesn't have a topic to use for an answer, the copilot searches the URL and all sub-paths. 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.

✖ Caution

When you publish your copilot, the calls using generative answers are made on behalf of the user chatting with the copilot, using the authentication settings configured in the copilot, which must use manual authentication with the **Service Provider** type of **Microsoft Entra ID** (formerly Azure Active Directory).

- To authenticate your copilot, see [Authentication](#).
- You can find instructions for how to create the needed Microsoft Entra ID application registration in [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 EntralID App Registration.

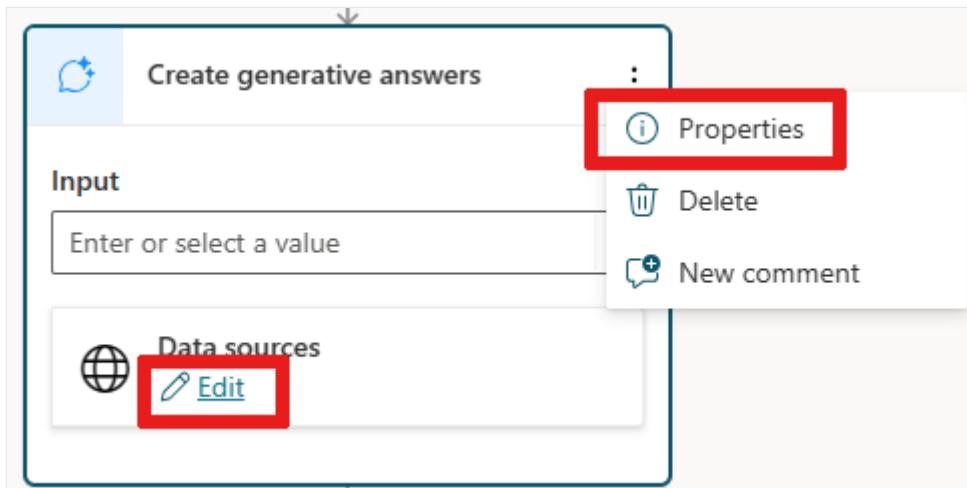
- Selecting **No authentication** and **Only for Teams** authentication options mean your copilot won't retrieve information from SharePoint or OneDrive for Business.
- When configuring authentication, scopes control user access. Specify **Sites.Read.All**, **Files.Read.All** in the scopes field near the existing **profile** and **openid** values in Copilot Studio. These scopes don't give users increased permissions when using the copilot, but allow their permissible content from the SharePoint site as configured for generative answers.
- This configuration only impacts Generative Answers. It does not impact Power Platform Connectors.

Due to memory limits, 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.

Use SharePoint or OneDrive for Business

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 >

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 for adding items. 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 or OneDrive for Business URLs you want to use. Separate multiple URLs with manual line breaks (with Shift + Enter).

The screenshot shows the Microsoft Copilot interface with two input sections. The top section is for 'Public websites' and the bottom section is for 'SharePoint'. Both sections have an 'Enter text' input field and a '+' button. A red box highlights the 'SharePoint' section.

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

A best practice is to omit *https://* from the URL. Recognized SharePoint URLs will be from the *sharepoint.com* domain.

Sharepoint site URLs cannot be more than two levels deep.

Only modern pages are supported. Content from classic **aspx** pages on SharePoint won't be used to generate answers.

4. Select **Save** at the top to save your topic changes.

5. Test your copilot with phrases that you expect to return content.

If the user account you used to sign in to *powerva.microsoft.com* doesn't have access to the SharePoint site, you won't get content or you might see a system error.

ⓘ Important

Information sources defined in the **Generative answers** node take priority at the copilot level. Copilot level sources function as a fallback.

Feedback

Was this page helpful?

👍 Yes

👎 No

Provide product feedback ↗

Use a custom data source for generative answers

Article • 06/27/2024

ⓘ 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.

In some cases, your data might not exist in a supported data source. However, you can provide your own data through [Power Automate cloud flows](#) or through [HTTP requests](#). This would typically return a JSON object literal that you can then parse into a `Table` format to generate answers.

The custom data field takes a `Table` as an input, with the below properties:

[] Expand table

Name	Required	Description
<code>Content</code>	Required	Source content that is used to be summarized and answer the input question
<code>ContentLocation</code>	Optional	URL for the citation of the source content
<code>Title</code>	Optional	Title for the citation of the source content

Here's a sample table in Power Fx format, after a JSON object literal has been parsed:

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"
}
]
```

Copilot answers are generated from `Content` and include the link to the data source in `ContentLocation`. If a `Title`, is it used for the citation.

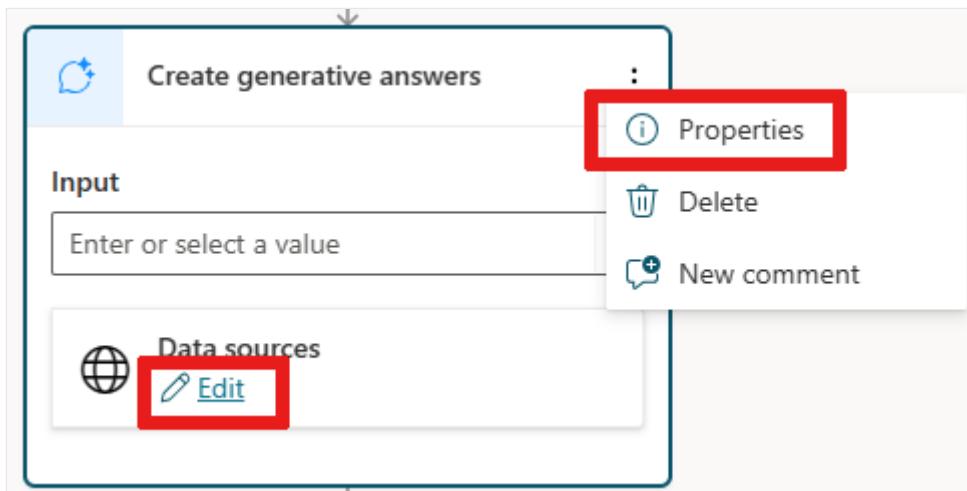
ⓘ Important

Only the first 3 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, 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 >

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.

ⓘ **Important**

Information sources defined in the **Generative answers** node take priority at the copilot level. Copilot level sources function as a fallback.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Generative answers with Search and summarize content

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.

When you create your copilot, you [enter a URL your copilot uses to generate responses](#). This URL is used in all features of the copilot. However, you can boost your copilot's conversations by using multiple internal and external information sources within individual topics.

Within a copilot's topic, you can add a **Create generative answers** node. This node allows you to specify more sources that the node searches based on your inputs. Information sources defined in the **Generative answers** node override sources you specified at the copilot level, which functions as a fallback.

These sources include:

- External sources:
 - Bing Search
 - Bing Custom Search
- Internal resources:
 - Azure OpenAI on your data
 - Documents uploaded to Dataverse
 - SharePoint (only [modern pages](#))
 - OneDrive for Business
 - Custom data (internal or external): Supply your own content from any source.
For example, a Power Automate Flow, a Skill, or other source.

Information sources

Different information 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

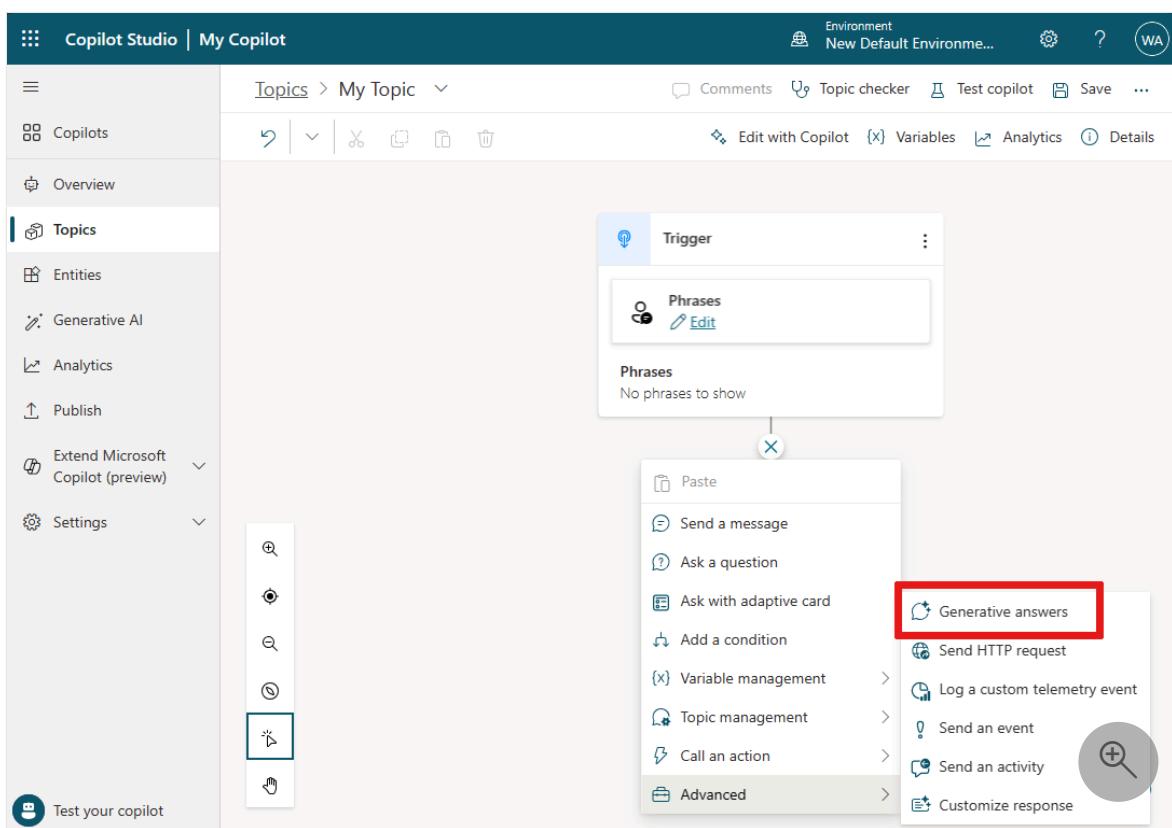
Name	Source	Description	Number of inputs supported in general answers	Authentication
Public Website	External	Searches the query input on Bing, only returns results from provided websites	4 public URLs (for example, <i>microsoft.com</i>)	None
Documents	Internal	Searches documents uploaded to Dataverse, returns results from the document contents	Limited by Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	4 URLs	Copilot user's Microsoft Entra ID authentication
OneDrive	Internal	Connects to a OneDrive URL, uses GraphSearch to return results	4 URLs	Copilot user's Microsoft Entra ID authentication
Dataverse	Internal	Connects to the connected Dataverse environment and uses retrieval-augmented generative technique in Dataverse to return results	Two Dataverse knowledge sources (and up to 15 tables per knowledge source)	Copilot user's Microsoft Entra ID authentication
Enterprise data via graph connections	Internal	Connects to the connected Dataverse environment and uses retrieval-augmented generative technique in Dataverse to return results	Two per custom copilot	Copilot user's Microsoft Entra ID authentication

(!) Note

Copilot user authentication for information sources means that when a specific user asks a question of the copilot, the copilot only surfaces content for that specific user to access.

Add a generative answers node

1. Select the **Topic** 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 and opens the **Properties** pane automatically. Or, access the **Properties** pane by selecting the node's **...** then **Properties**.

You can specify and configure your new data sources:

- Use a [Bing Custom Search](#) to search many websites
- Use a connection to [Azure OpenAI](#) on your data
- Upload documents to use as a data source (preview)
- Use content on [SharePoint](#) or [OneDrive for Business](#)
- Use a custom data source

Authentication

Some sources require authentication, since the copilot 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 copilot 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 Microsoft 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 won't be used to generate answers.

Feedback

Was this page helpful?



Yes



No

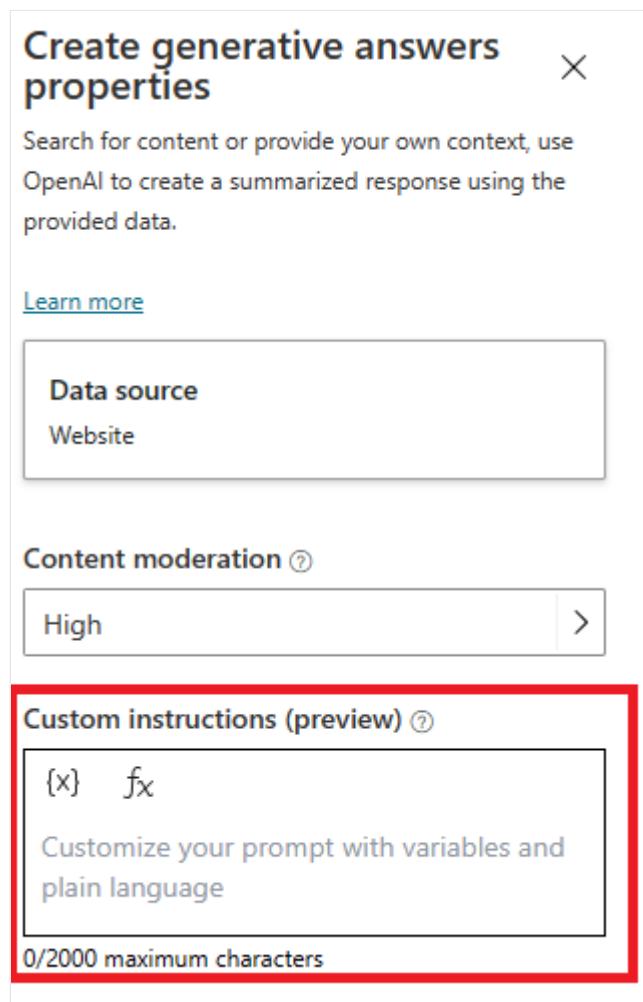
[Provide product feedback](#)

Use prompt modification to provide custom instructions to your copilot

Article • 03/26/2024

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 copilot to get the desired response.

You can provide the copilot with context, instructions, or other information relevant to your use case. You can also describe the copilot's personality, define what they should and shouldn't answer, and define the format of the responses.



Samples of prompt modifications and the resulting response:

Expand table

Custom Instruction	User question	Copilot response
You're an AI assistant that helps people find information and	What can you tell about me,	Dear John, I'm sorry to say, But I don't have info on you today. I'm just an AI

Custom Instruction	User question	Copilot response
responds in rhyme. If the user asks you a question you don't know the answer to, say so.	John Doe?	with knowledge in my brain, But without your input, I can't explain. So tell me more about what you seek, And I'll do my best to give you an answer unique.
Format the response as a series of steps, starting each step with 'Step 1:', 'Step 2:', etc. Don't use numbered lists. Each step must be bolded.	How do I add a new email account?	Step 1: Open the email application on your device. Step 2: Select the option to add a new account. Step 3: Enter your email address and password. Step 4: Follow the on-screen instructions to complete the setup process.
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 copilot 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 copilot understand your expectations. Examples help the copilot 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 copilot can process them effectively.
- **Give the copilot an “out”:** Give the copilot 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 copilot 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 copilot's responses.

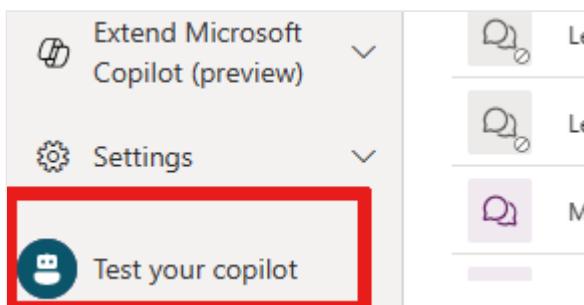
Capabilities

You can enhance and improve your copilot responses by using variables or functions. These capabilities are useful as your copilot increases scale.

- **Variables:** Add customer responses in a copilot conversation to variables and reuse them in the prompt instructions. For more information, see [variables](#).
- **Functions:** Add logic to your prompt instructions, using Power Fx. For more information, see [creating expressions](#).
- **Character limit:** Prompt instructions are limited to 2,000 characters.

Test your new prompt

1. Select **Test your copilot** at the bottom of the navigation pane.



2. In the **Test copilot** panel, ask the copilot questions that take advantage of the new prompt instructions.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

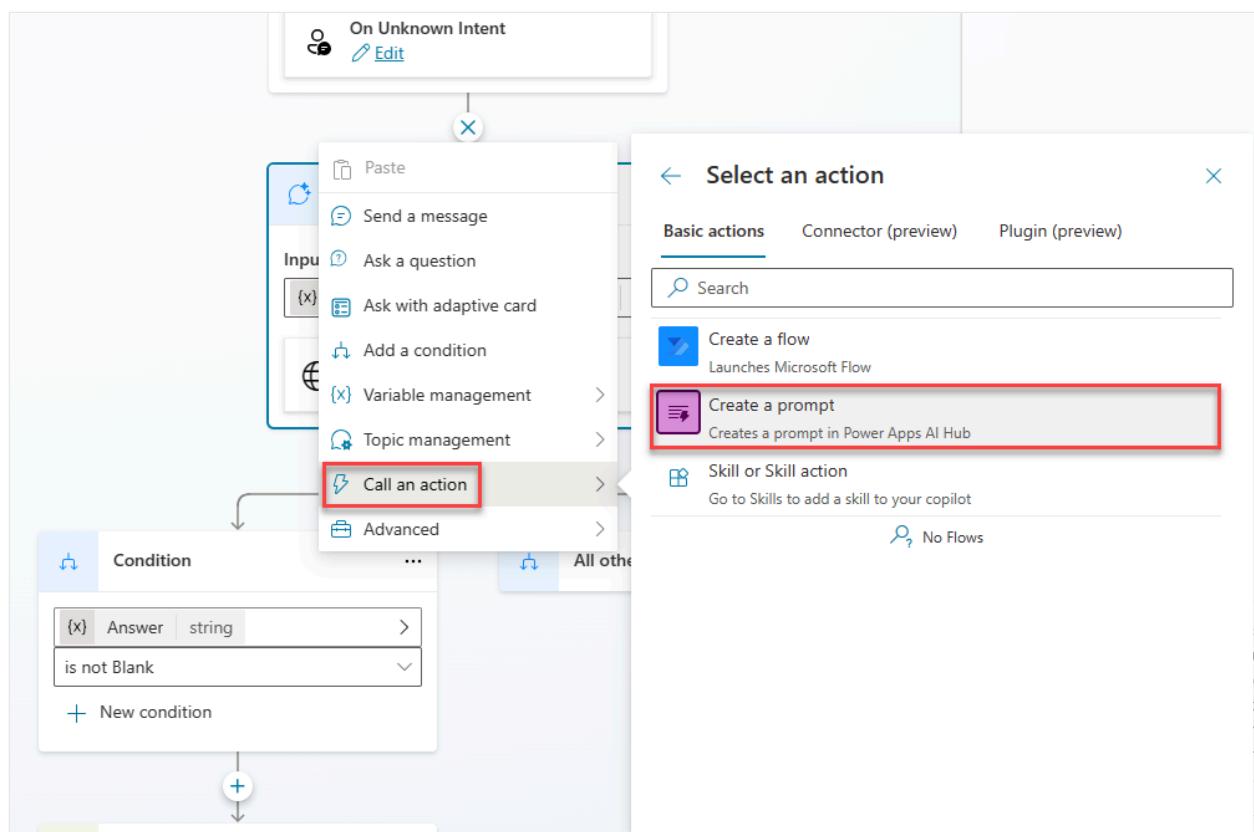
Use prompts so your copilot performs a specific task

Article • 05/21/2024

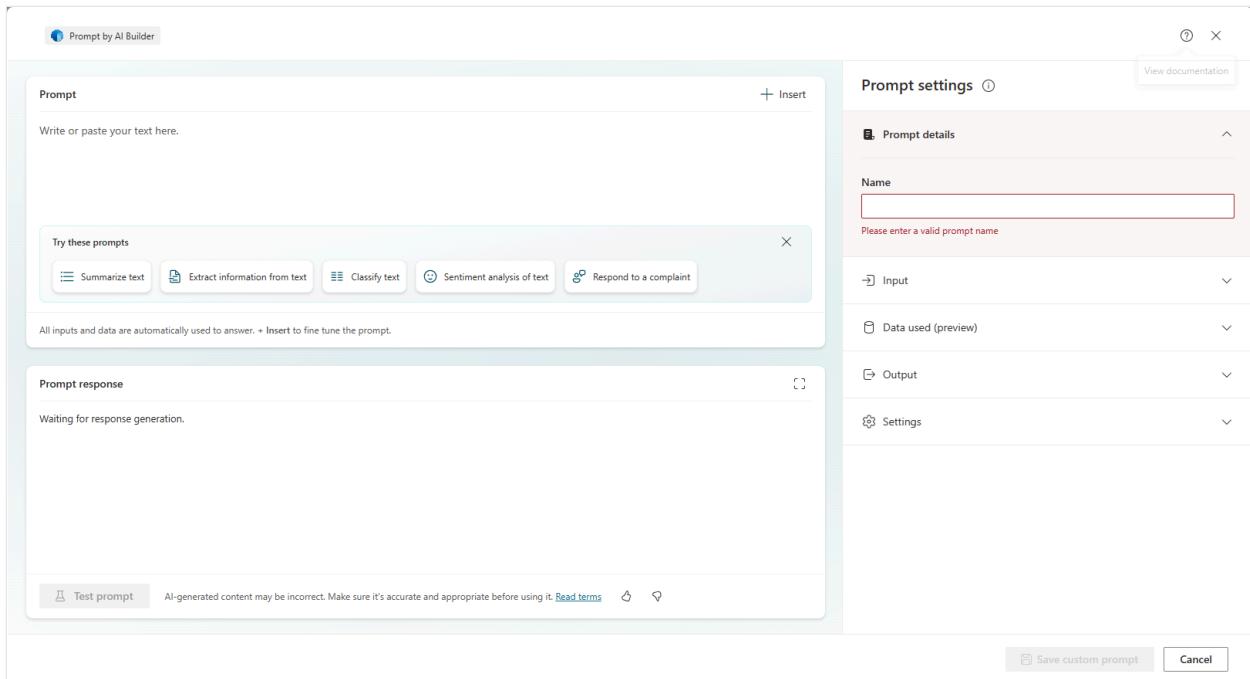
Using the prompt node, you can integrate with AI Builder's [custom prompt authoring](#) to give instructions to the GPT model for your custom copilot.

Add a prompt node in a topic

1. Open a topic and add a node.
2. From the menu, select **Call an action > Create a prompt**.



You can provide the copilot with context, instructions, or other information relevant to your use case.



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 copilot 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 copilot understand your expectations. Examples help the copilot 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 copilot can process them effectively.
- **Give the copilot an "out":** Give the copilot 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 copilot 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 copilot's responses.

[!IMPORTANT]

- Prompts are running on GPT-3.5 Turbo model powered by Azure OpenAI Service.
- This capability is limited to some regions.
- This capability might be subject to usage limits or capacity throttling.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create and edit topics with 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.

Normally, you can manually [create and edit topics](#), but Copilot Studio lets you create and iterate on topics with a description. By describing what you want and having AI build it for you, time is saved in creating and updating copilot topics.

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.

Prerequisites

- An account for Copilot Studio.

ⓘ Note

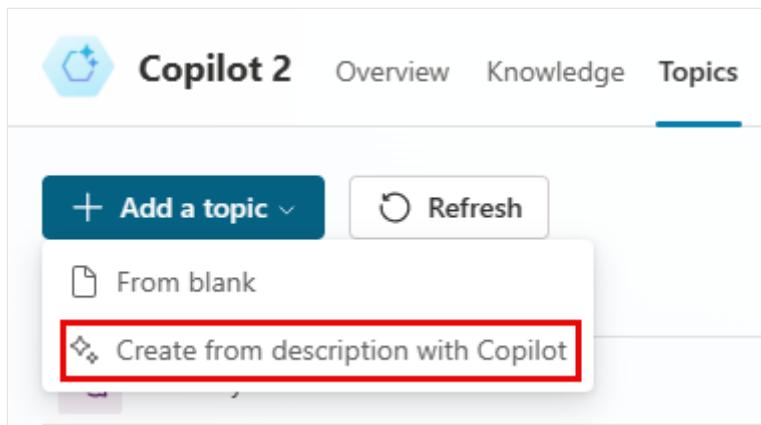
To create an account or get more familiar with Copilot Studio, see the [Quickstart guide for building copilots with generative AI](#).

- Review the [AI response generation training, model, and usage FAQ](#) and [Learn more about Azure OpenAI](#).
- This capability might be subject to usage limits or capacity throttling.
- To see the language support, go to [Supported languages](#).

Use Copilot Studio to create a topic

You can create a new topic from a description on the **Topics** page in your copilot.

1. Select **+ Add > Topic > Create from description**.



2. In the **Create from description with Copilot** window, add a name in the **Name your topic** field.
3. In the **Create a topic to...** field, 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 behavior you want the copilot to take.

A screenshot of a modal dialog titled 'Create from description with Copilot'. It contains instructions: 'Write a description of what you'd like your copilot to cover, and Copilot will create your topic.' followed by a link 'Learn more'. There are two main input fields: 'Name your topic *' which has 'Find a job' typed into it, and 'Create a topic to... *' which has 'Let the user ask about job opportunities for positions such as sales clerk, manager, packer, cashier, greeter.' typed into it. Both fields have a small asterisk indicating they are required.

You can select any of the examples to automatically insert them into the **Create a topic to...** field. Select **View more examples** to generate new suggestions.

4. Select **Create**.

The authoring canvas generates and opens a new topic. The topic includes trigger phrases, and one or more nodes, based on your description.

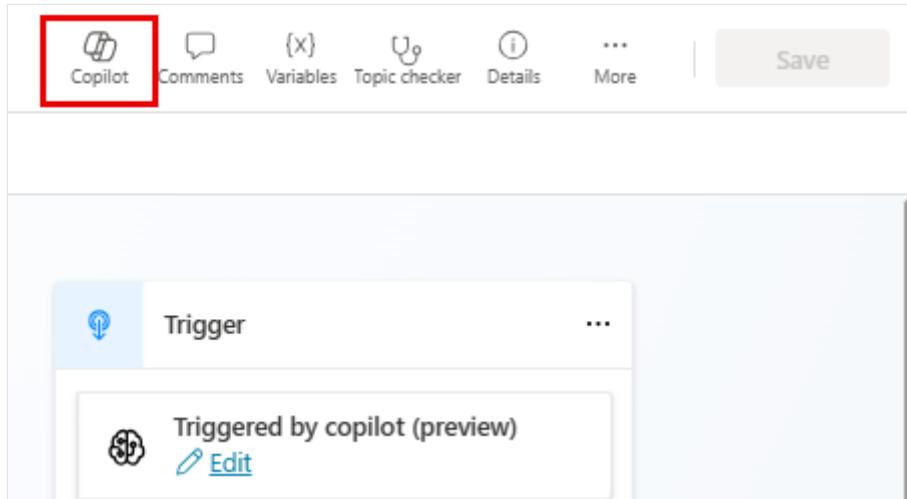
💡 Tip

For more information on editing topics manually, see [Create and edit topics in your Copilot Studio copilot](#).

Use Copilot to edit a topic

You can make changes to any topic using the power of natural language understanding (NLU).

1. From the **Topics** page in your copilot, select the topic you want to modify, and then select **Copilot** at the top of the page.



💡 Tip

When describing a task for your copilot, there's no need to mention canvas elements. For example, adding *ask the user their full name* to a question node, you don't have to direct the AI with *ask the user their full name in the question node*.

2. In the **Edit with Copilot** panel, in the **What do you want to do?** field, describe what you'd like to change.

Move nodes in the canvas. To make additions and changes to nodes, tell Copilot what you want to do. [Learn more](#)

What do you want to do?

Explain what you want to change or add from the topic. Select a node to be more specific.

Update No nodes selected

Examples that work

Add a question to ask the user for their name.

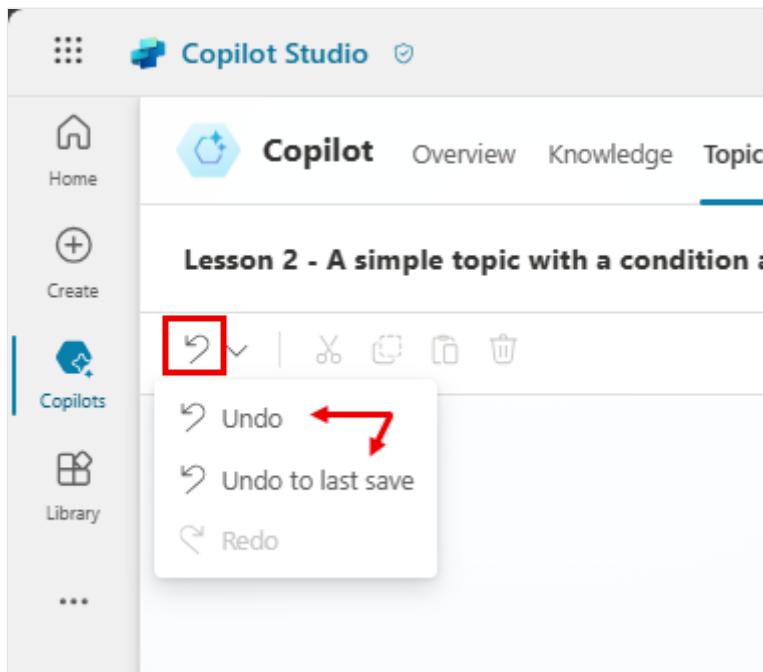
Use simple, plain English to direct the AI, like in these examples:

- *add a question to ask the user for their date of birth*
- *add 2 message variations to all questions in the topic*
- *summarize the information collected from the user in an Adaptive Card*

3. Select Update.

You can continue to [edit topics](#) or describe more changes.

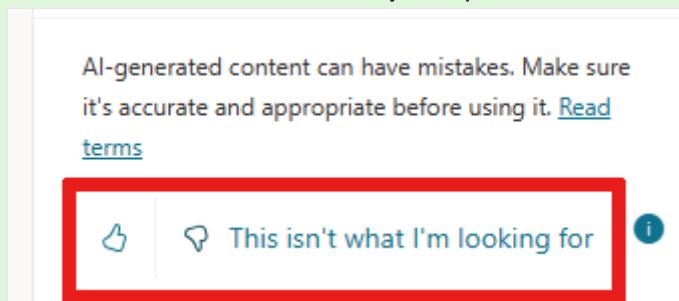
If you don't like the changes, select the Undo icon (undo icon) and select **Undo** to go back one action or select **Undo to last save** to go back to the last save point.



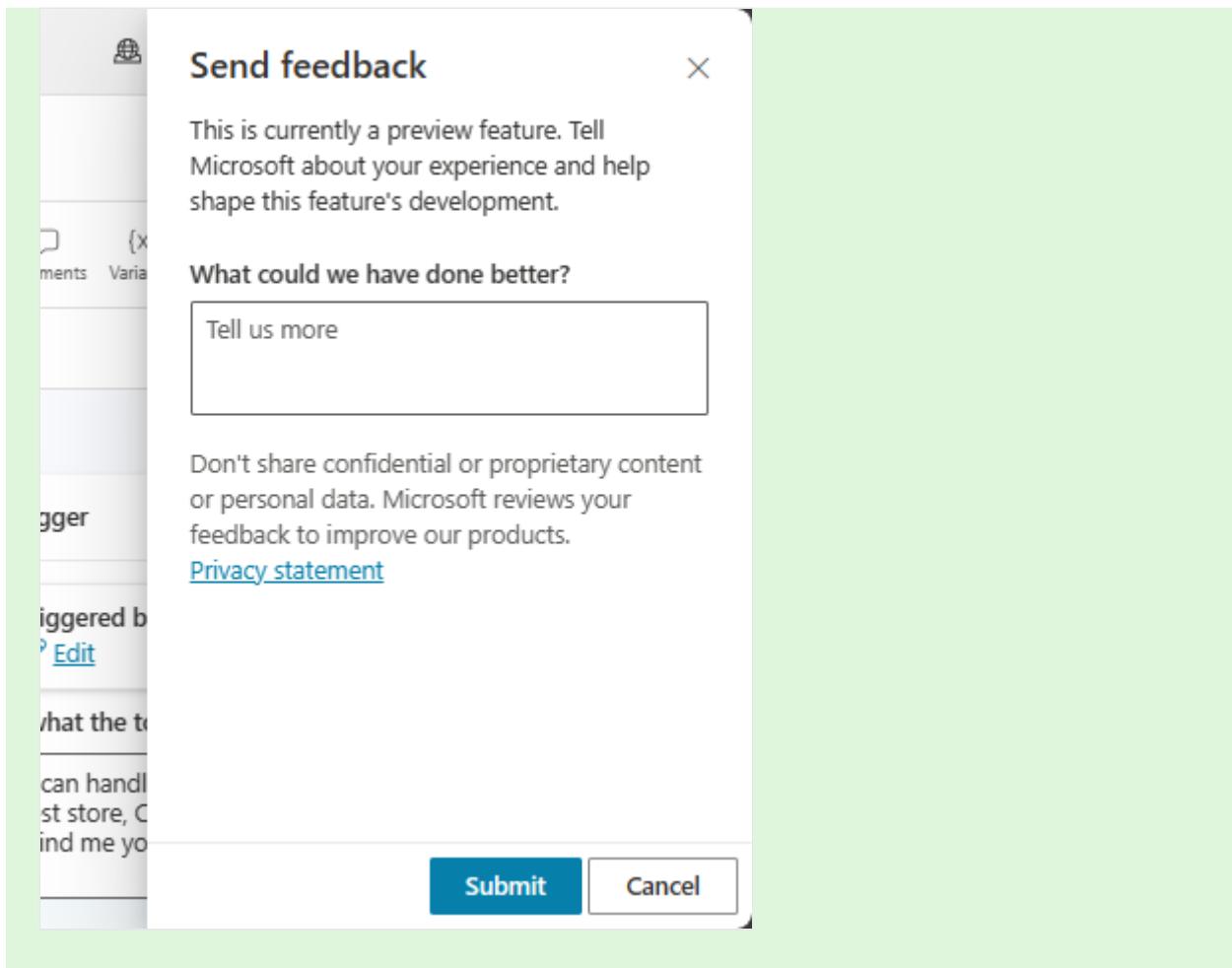
You can update your description under the **What you asked for** label on the **Edit with Copilot** pane.

💡 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 to see the **Send feedback** pane and add detailed feedback. We'll use this feedback to improve the quality of the AI.



What's supported

Topic node support

Copilot can create **Message**, **Question**, and **Condition** nodes when building and iterating on topics. Other node types aren't currently supported.

There are some limitations on how Copilot Studio works with nodes:

- 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 on a Question node.
 - You can still manually edit these properties.
- **Condition** nodes created by Copilot Studio that branch a topic from a multiple choice question show errors in the generated conditions.
 - You can [troubleshoot these errors](#) manually.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

AI features for Teams and Classic copilots

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.

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

Copilots 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 copilots and support some scenarios. In most cases, you should [create or convert your copilots 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 copilot [using the unified authoring canvas](#) in the Copilot Studio web app.

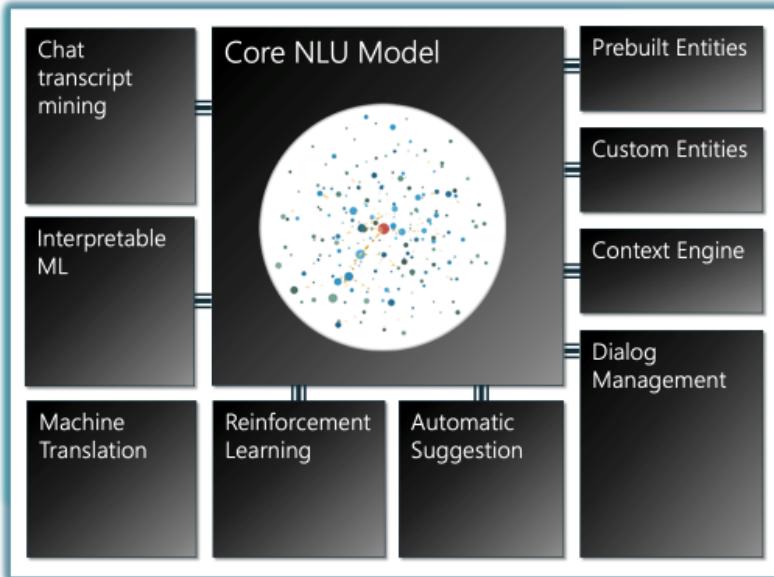
Prerequisites

You can use the AI features if:

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

AI models in Microsoft 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 copilot 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 Microsoft 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 modal window titled "Topic overlap details" with a sub-section "Languages supported in Microsoft 365". It displays two overlapping trigger phrases: "For which languages is Microsoft 365 available?" and "Can I use Microsoft 365 in languages other than the one I originally purchased?". Both phrases are associated with the topic "Languages supported in Microsoft 365". The "Save" and "Cancel" buttons are at the bottom right.

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

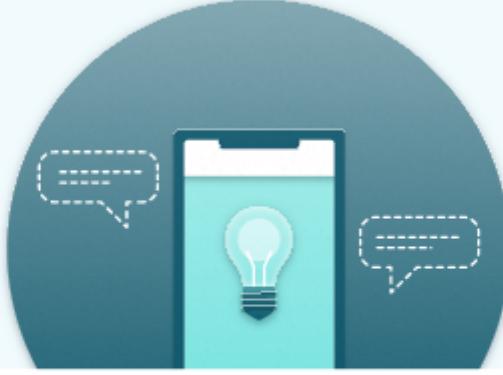
Trigger phrases that are too similar blur the triggering, review overlapping topics and tri

Updated 6 minutes ago

Topic suggestion from chat transcripts (preview)

This feature analyzes sessions between your copilot 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 copilot. 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

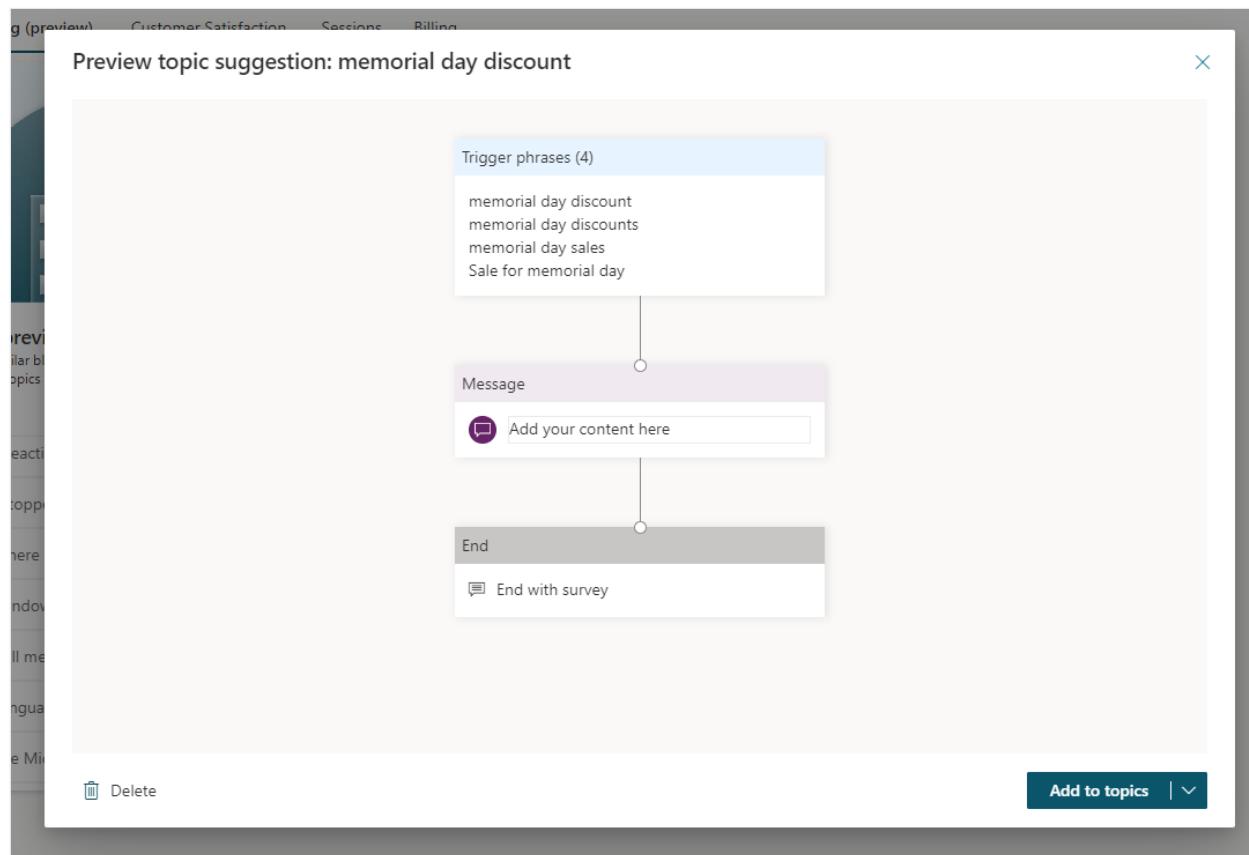
Topic suggestions (4)	Sessions ↓
memorial day discount	4
holiday deals	3
Surface laptop 4	3
Hololens	2

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 copilot 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 copilot 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 copilot, 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 copilots *not* marked **Classic**. This includes any copilots created (or converted) in the web app after May 23, 2023, or any copilots that aren't created with the unified authoring canvas.

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

To enable these older AI capabilities in Microsoft Copilot Studio:

1. Open your copilot:

- In the web app, the copilot 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.

The screenshot shows the 'Generative AI' settings page. On the left, there's a sidebar with options like Overview, Generative AI (which is selected and highlighted with a red box), Topics, Entities, Analytics, Publish, Settings (with Details, AI integration tools, and Channels), and Channels. At the top right, a message says: 'Your bot may not work as intended right now due to your organization's data loss prevention policies. Address affected admin.' with a 'Learn more' link. The main content area is titled 'Generative AI' and says 'Select options to enable intelligent features. Review supplemental terms and conditions to learn more about features in preview.' It lists two features: 'Topic overlap detection' and 'Topic suggestions from chat transcripts (preview)'. Both have descriptions and 'On' checkboxes. A magnifying glass icon is in the bottom right corner.

Your bot may not work as intended right now due to your organization's data loss prevention policies. Address affected admin. [Learn more](#)

Generative AI

Select options to enable intelligent features. Review [supplemental terms and conditions](#) to learn more about features in preview.

Topic overlap detection
Helps you improve triggering accuracy by finding overlaps between topics. Reduces the need for the bot to ask clarifying questions before triggering a topic. [Learn more](#)
[See overlapping topics](#)

On

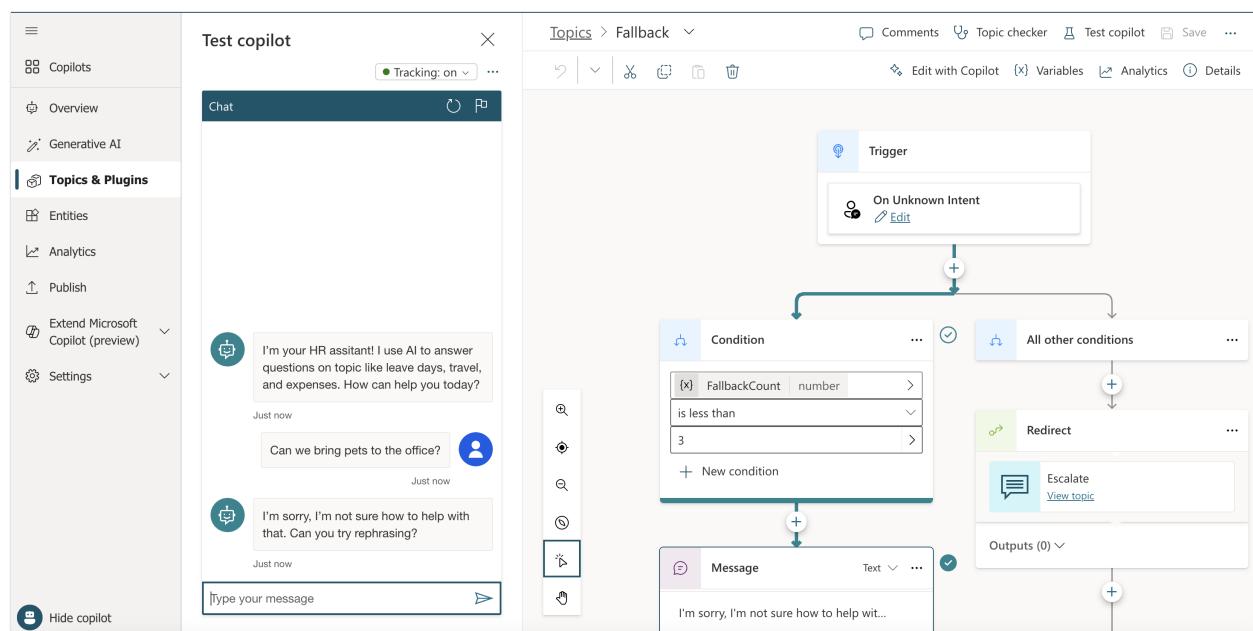
Topic suggestions from chat transcripts (preview)
Analyzes chat transcripts for end user queries that don't trigger a topic. The system provides suggestions for new topics that will help you address those gaps. [Learn more](#)
[See suggestions from transcripts](#)

On

Generative answers pointing to SharePoint or OneDrive sources don't return results

Article • 03/26/2024

Generative answers allow makers to create copilots that respond to questions grounded in data sources, like public websites or SharePoint, by pointing the copilot at those data sources. However, sometimes the copilot 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).



Why doesn't the 'Create generative answers' node respond?

When a SharePoint or OneDrive data source is configured, there could be several different factors preventing generative answers from returning a response, such as the following potential factors:

1. Search results are missing
2. The user accessing the copilot doesn't have sufficient permissions on the data source
3. Files are larger than the 3 MB size limitation
4. The app registration or the copilot are misconfigured

5. Content blocked by content moderation

Note

Before continuing, please make sure you have followed the instructions on how to [set up generative answers over SharePoint or OneDrive](#).

Search results are missing

Generative answers for a SharePoint or OneDrive data source rely on making calls to the Graph API search endpoint. Only the top three results coming back from Graph API are used to summarize and generate a response. If no results come back from Graph API, the generative answers node doesn't provide a response.

To diagnose whether Copilot Studio isn't returning results from the Graph API, you can make direct calls to the Graph API search endpoint. This call simulates the way Copilot Studio works behind the scenes. Calls to the Graph API 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/OneDrive 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:\\"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 Graph API, resulting in a query similar to the following example:

The screenshot shows the Microsoft Graph Explorer interface. The top navigation bar includes 'Tenant Contoso', 'Run query', and 'AL'. The left sidebar has sections for 'Sample queries', 'Resources', and 'History', with a search bar. Under 'Getting Started (8)', there are several GET requests listed. The main area shows a POST request to 'https://graph.microsoft.com/v1.0/search/query'. The 'Request body' tab displays the following JSON:

```
{  
  "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  
      }  
    }  
  ]  
}
```

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 after running the search query. The top navigation bar includes 'Tenant Contoso', 'Run query', and 'AL'. The left sidebar has sections for 'Sample queries', 'Resources', and 'History', with a search bar. Under 'Getting Started (8)', there are several GET requests listed. The main area shows a POST request to 'https://graph.microsoft.com/v1.0/search/query' with a green status bar indicating 'OK - 200 - 420 ms'. The 'Response preview' tab displays the following JSON response:

```
{  
  "value": [  
    {  
      "SearchTerms": [  
        "policy",  
        "perks",  
        "benefits"  
      ],  
      "hitsContainers": [  
        {  
          "total": 0,  
          "moreResultsAvailable": false  
        }  
      ]  
    }  
  ],  
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#Collection(microsoft.graph.searchResponse)"  
}
```

How to fix

1. Ensure that your Create generative answers node points to a SharePoint or OneDrive 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 or OneDrive, 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 and OneDrive rely on [delegated permissions](#) when making calls to Graph API. At a minimum, a user must have read permissions on the relevant sites and files, or the call to Graph API doesn't return any results.

If the user is missing permissions, no results are returned from Graph API, 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 copilot are misconfigured

When admins configure generative answers over SharePoint and OneDrive, 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 copilot authentication settings, or if consent wasn't granted to the required scopes, no results are returned, nor any errors or exceptions. For an end user, it appears as if no documents were found.

How to fix

Add the necessary scopes to the App Registration and/or the copilot's authentication settings, and grant consent.

The following example is a reference to a well configured app registration:

Configured permissions					
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent					
API / Permissions name	Type	Description	Admin consent requ...	Status	
Microsoft Graph (5)					***
Files.Read.All	Delegated	Read all files that user can access	No	✓ Granted for Contoso	***
openid	Delegated	Sign users in	No	✓ Granted for Contoso	***
profile	Delegated	View users' basic profile	No	✓ Granted for Contoso	***
Sites.Read.All	Delegated	Read items in all site collections	No	✓ Granted for Contoso	***
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for Contoso	***

The following example shows the required authentication settings in Copilot Studio:

Authentication

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 limitation

Currently, generative answers can only process files up to 3 MB in size. Larger files can be stored in SharePoint and are returned by a Graph API search, but aren't processed by generative answers.

ⓘ Note

This limitation doesn't apply to customers eligible for [M365 Semantic Indexing](#).

How to fix

If files relevant for your conversational AI experience exceed the 3 MB limitation, you might want to explore alternative architectures, such as using [Microsoft 365 Semantic Indexing](#) or [connect your data to Azure Open AI for Generative Answers](#).

Content blocked by content moderation

When generating responses, Copilot Studio moderates 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 Copilot Studio is configured to [send telemetry data to Azure Applications Insights](#).

After connecting your copilot to Azure App Insights, you can use the following Kusto Query Language (KQL) query to find out if content was filtered:

```
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 Microsoft Application Insights Log Analytics workspace. On the left, there's a navigation sidebar with sections like 'Application map', 'Smart detection', 'Live metrics', 'Transaction search', 'Availability', 'Failures', 'Performance', 'Troubleshooting guides (preview)', 'Monitoring' (with 'Logs' selected), 'Usage', 'Users', 'Sessions', and 'Events'. The main area has a search bar at the top, followed by a 'New Query 1*' tab, a 'Run' button, and a time range selector ('Last 7 days'). Below this is a table with columns: 'nid', 'topic', 'message', and 'result'. A single row is visible: 'nid' is blacked out, 'topic' is 'Conversational boosting', 'message' is blacked out, and 'result' is 'Filtered by High Content Mod...'. The bottom of the table shows '0s 874ms | Display time (UTC+00:00)' and 'Query details | 1 - 1 of 1'. The right side of the interface includes various buttons for feedback, queries, and export.

How to fix

1. Try to adjust [content moderation](#), but keep in mind that lower content moderation settings might result in answers that are less accurate or relevant.
2. 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 • 06/04/2024

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.

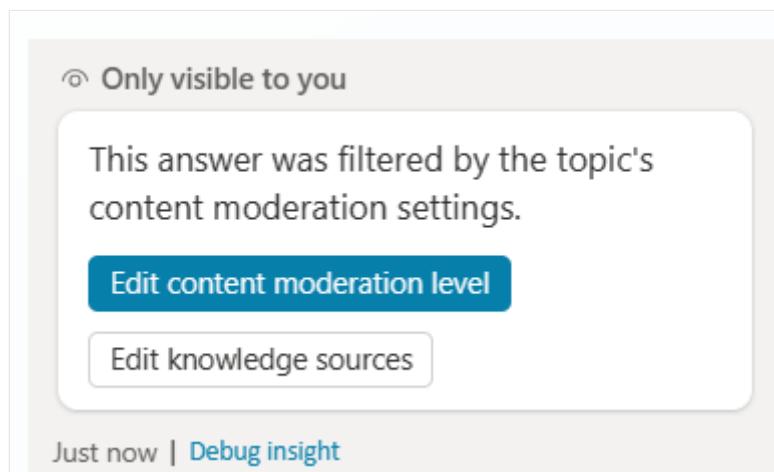
Some articles and screenshots may refer to Power Virtual Agents while we update documentation and training content.

Generative answers allow makers to create copilots that respond to questions grounded in knowledge sources, like public websites or SharePoint. However, sometimes the copilot doesn't provide a response. When a copilot 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

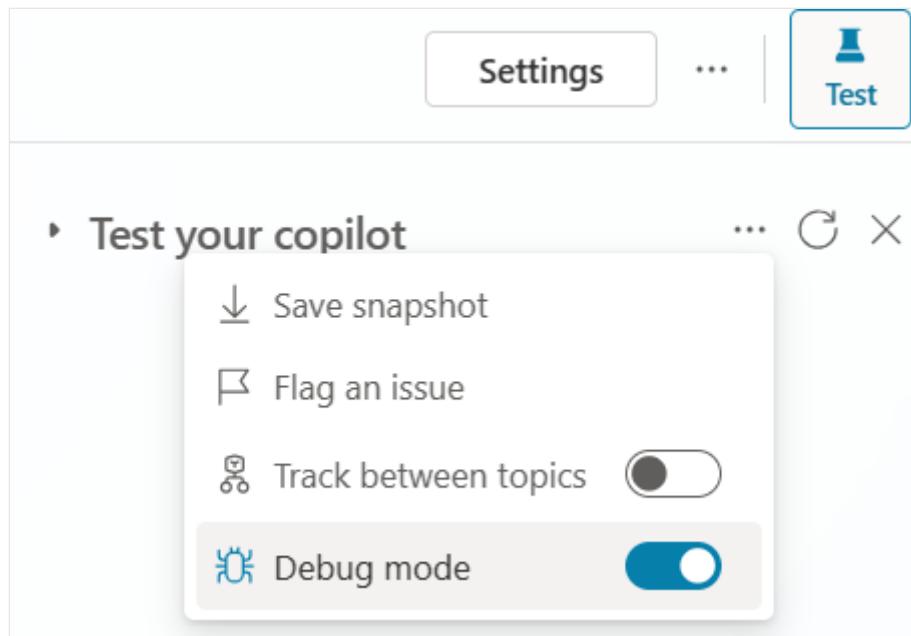
When debug mode is enabled, insights are displayed in the test web chat when generative answers query is made and no results are returned. Debug insights only show in the test chat and aren't shown to users in a published copilot.

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 or OneDrive sources don't return results

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Assign licenses and manage access to Microsoft Copilot Studio

Article • 04/15/2024

ⓘ 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 is written for tenant administrators. If you want to sign up for Copilot Studio as an individual, see the [Sign up for a Microsoft Copilot Studio trial](#) topic.

To create and manage copilots 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 copilots.
- 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 copilots don't need a special license. After you publish your copilot, anyone who can access the published copilot can interact with it.

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.

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

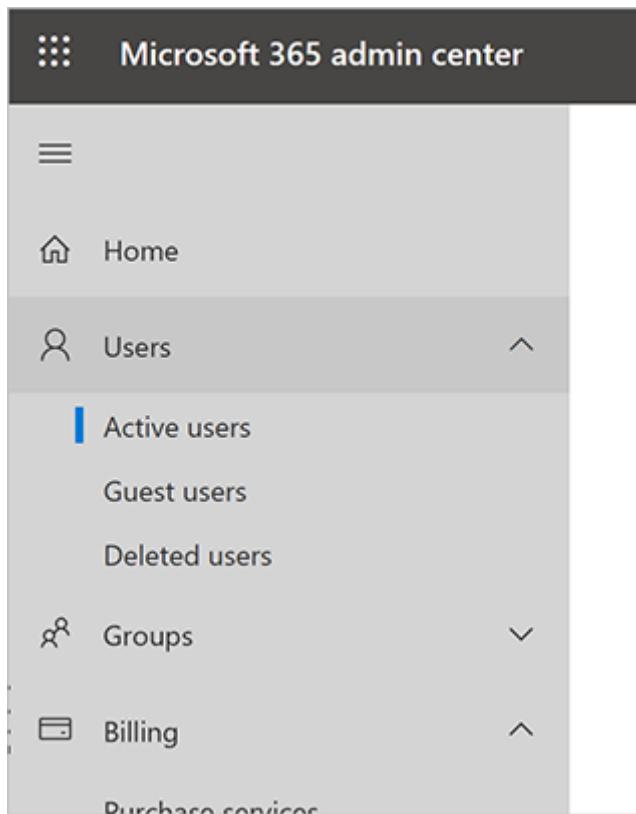
1. After you've purchased a Copilot Studio license from the Microsoft 365 admin center, you need to purchase user licenses to give users access to the product. In the Microsoft 365 admin center, go to **Billing**, and then select **Purchase services**.
2. Scroll down to the **Add-ons** section.
3. Look for **Copilot Studio User License**. Select the number of licenses you need, and complete the checkout process.

i Important

If you purchased a Microsoft 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**.

The screenshot shows the "Active users" page. At the top, there are buttons for "Add a user", "Refresh", "Reset password", "Assign to group", "Manage product licenses" (which is highlighted in grey), and "Manage roles". Below this is a table with columns: "Display name ↑", "Username", and "Licenses". The table contains two rows: one for "gds" (Username: gds, Licenses: Unlicensed) and one for "Alex Wu" (Username: Alex Wu, Licenses: Unlicensed). A checkmark is visible next to "Alex Wu".

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 [Microsoft 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 copilot](#).

See [Quotas and rate limit capacity considerations for Microsoft Copilot Studio](#) for more information.

Using Power Automate with a Copilot Studio license

- See [Quotas, limits, and configuration values](#)
- See [how to use Power Automate with Microsoft Copilot Studio](#)

Microsoft Copilot Studio for US Government customers

Article • 04/15/2024

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 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 [Microsoft 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.

Microsoft 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. This is in contrast to GCC, which 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 FBI's Criminal Justice Information Services (CJIS) Policy, or
- 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 Screening and Background Checks ¹	Description
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
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.

- ¹. 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 multi-tenant, 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.

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 end users' access points:

GCC customers

- *.azure.net*
- *.azure.us*
- *.azure-aphub.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 end 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.

 Expand table

Feature or capability	Available in GCC	Available in GCC High
Copilot Studio Analytics ¹	✗ 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](#)

Sign up for a Microsoft Copilot Studio trial

Article • 04/15/2024

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

Follow the instructions to sign up for a Copilot Studio trial. After you complete the process, you can use Copilot Studio to create and publish copilots.

1. Go to the [sign-up page](#).
2. Enter your email address and then select **Sign up**.



You're then taken to the Copilot Studio sign-in page where you can start using your account.

Trial expiration

The screenshot shows the Copilot Studio interface. At the top, there's a banner stating "Your free trial expires in 7 days. Contact your admin to discover the plan that's right for you. [See pricing](#)". Below the banner, the navigation bar includes "Start over", "Local promotions", "Undo", "Redo", "Topic checker", and "Save". A green button at the bottom says "Trigger phrases (8)".

After you finish the sign-up process, your time-limited free trial for Copilot Studio starts. You'll see notification and receive emails to inform you about the trial expiry. When the trial expires, you'll be able to extend the trial by 30 days.

The screenshot shows the Power Virtual Agents interface. It features a central message: "Your Power Virtual Agents trial has expired" and "To continue, extend your trial for 30 days or see pricing options." Below this are two buttons: "Extend trial" and "See pricing". The main area shows a chat window titled "Contoso bot" and a flow editor titled "SmartPrinter X troubleshooter". The flow editor contains several nodes: "Condition", "Question", "Identify", "Ask a question", "Options for user", "Save response as", "Condition", "Condition", and "Transfer to agent". The "Ask a question" node contains a question: "I can contact your SmartPrinterX serial number 418cd1e-994b-49b5-acb -d6e108c00d3 directly to get more information on your issue. Would you like me to do so?". The "Options for user" node provides two choices: "Yes, please look up the issue for me" and "No, I'll describe the issue myself". The "Save response as" node is set to "PrinterIssueText". The "Transfer to agent" node includes a note: "Send our records down something else. Let me connect you to one of our agents that can assist you further. One moment please." At the bottom, it says "Done using Power Virtual Agents? Permanently delete your bots".

⚠ Note

Your copilot 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 need to use your work or school account to finish the sign-up.

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](#).



Working with environments in Microsoft 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.

With Microsoft Copilot Studio, you can create copilots in different environments and easily switch between them.

An environment is a space to store, manage, and share your organization's business data. The copilots you create are stored in an environment (apps and flows are also stored in environments). Environments may also have different roles, security requirements and target audiences, and each environment is created in a separate location. See the following topics for more information:

- [Supported data locations](#)
- [Power Platform environments overview](#)

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 copilots for each audience.
- Create separate environments for different global branches of your company.
- Create separate environments to satisfy data residency requirements.

You can also build all your copilots 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 copilots that you want to deploy to production.

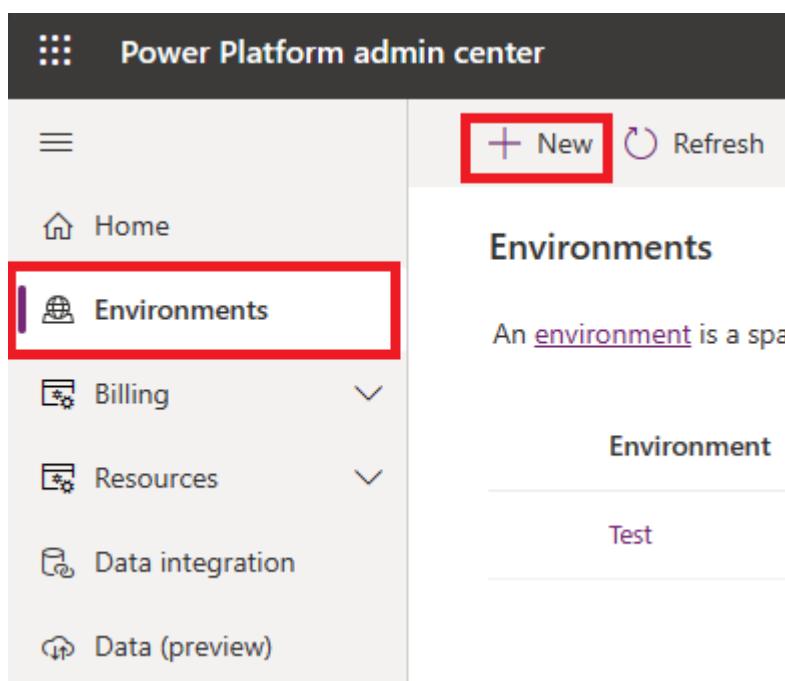
[Learn more about types of environments.](#)

Create a new environment for your copilots

When you first sign in and create a new copilot, 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 Microsoft 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 Microsoft 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 *

Choose a name

ibility and contr

Region *

United States - Default



A local region can provide quicker data access

Type ⓘ *

Production



Purpose

Describe the environment's 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 a copilot. For more information, see: [Supported data locations](#).

3. Specify the details of the Dataverse data store, such as the language, currency type, and security group and then select **Save**.
4. You are returned to the **Environments** tab. Select **Refresh** to see the environment. It may take a few minutes to fully create the environment.
5. After the environment is created, you can return to the [Microsoft Copilot Studio portal](#) and use the environment to create a new copilot.

Create a copilot in an existing environment

1. Select the environment in the environment switcher on the top menu bar.

Create a copilot in an existing environment where you don't have access

To create a copilot 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 a copilot in the environment (this step installs the necessary Microsoft Copilot Studio solutions).
2. [Assign the security role of "copilot author" to yourself](#) in the environment.
3. Return to the [Microsoft Copilot Studio portal](#) and create a copilot in the environment.

Trial environments

When you trial Microsoft Copilot Studio, you can create trial environments that expire after 30 days. When the environment expires, all the copilots in the environment are deleted. This causes all the data associated with the copilot, 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 Microsoft Copilot

Studio portal, you also see a notification if you have copilots 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 copilots for longer than 30 days, you must [convert the trial environment to a production environment](#).

Supported operations

Microsoft 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 a copilot

When you're creating your copilot, 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 copilot.

The environment doesn't show up in the drop-down menu of Microsoft 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 to [create a database in your environment](#).
- The environment is created in an unsupported region. For more information, see: [Supported data locations](#).

Related links

- [Power Platform - Environments overview](#)
- [Power Platform - Configure environment security](#)

Configure data movement across geographic locations for generative AI

Article • 05/28/2024

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.

With the [generative AI features](#) of Copilot Studio, you can instantly build useful copilots 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 admins 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 copilots 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 copilot.

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 admins 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 copilots and generative AI features](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Managing extensions for Copilot for Sales (preview)

Article • 06/26/2024

ⓘ 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 is prerelease documentation and is subject to change.]

Microsoft Copilot Studio can be used to extend 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, refer: [Copilot connectors and actions overview](#).

As a Copilot for Sales administrator, you can manage plugins for Copilot for Sales within your CRM instance by enabling or disabling specific plugins from Microsoft Copilot Studio.

ⓘ Note

- By enabling a connector action for Copilot for Sales, you may be allowing Copilot for Sales users on Microsoft 365 and Teams applications to send and receive data from external sources using Copilot for Sales even if the same connector action has been disallowed for use directly with Microsoft 365. It is recommended to ensure that this action complies with your organization policies before enabling it.
- The data and insights that connector actions bring to 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 such connector actions for compliance with your organization policies.

Who can manage plugins for Copilot for Sales?

To manage plugins for Copilot for Sales in Microsoft Copilot Studio, you must be:

- Assigned as a "Copilot for Sales Administrator". Refer: [Administrator settings for Copilot for Sales](#)
- Given the role of "Microsoft Copilot Administrator" in Dataverse. For Salesforce administrators, this role can be given for the msdyn_viva Power platform environment. Refer: [Assign a security role to a user](#)
- Given the role of "Environment Maker" or "System Customizer" in Dataverse. For Salesforce administrators, this role can be given for the msdyn_viva Power platform environment. Refer: [Assign a security role to a user](#)

Enable a plugin for Copilot for Sales

To enable a plugin for Copilot for Sales:

1. Log in to Microsoft Copilot Studio
2. Select the right Power platform environment for your CRM instance
 - To find this information for Dynamics 365 CRM, refer: [Find your environment and organization ID and name](#)
 - For Salesforce CRM, you can use *msdyn_viva*
3. Navigate to Copilot for Sales > Manage Here you can see all the plugins that are enabled in this environment.
4. Click on *Enable Plugins*
5. Search for the plugin you want to enable
6. Select the plugin you want to enable. You can review the information about the plugin and choose to enable it.
7. Toggle *Enable plugin* to Yes
8. Select the users you want to enable the plugin for. You can use one of the three options:
 - Only Copilot Administrators (default)
 - Entire organization
 - Specific users/groups

 **Note**

If a plugin is enabled, it is automatically enabled for all Copilot Administrators.

9. Click **Save** This enables the plugin for the users you selected. You also need to:

- [Create and assign security role to user](#)
- [Share the plugin with the user](#)

The plugin is now enabled for the users.

Assign security role to user

Once you enable a plugin for a user, you need to add appropriate security permissions for the users to be able to use the specific plugins. Assign the role "Microsoft Copilot User" to the users. For details, refer: [Configure user security in an environment](#)

Share plugin with user

The plugins need to be shared with the users as well. To do that:

1. Log in to the Power Apps Maker portal (<https://powerapps.com>)
2. Select your environment.
 - To find this information for Dynamics 365 CRM, refer: [Find your environment and organization ID and name](#)
 - For Salesforce CRM, you can use *msdyn_viva*
3. Navigate to *Solutions*. Select the solution that contains your plugin.
4. Click the plugin to open the details of the plugin.
5. Click *Share > Manage access*
6. Select the user or team that you want to share the plugin record
7. Select *Read* for the permission
8. Click *Share*

Disable a plugin for Copilot for Sales

To disable a plugin for Copilot for Sales:

1. Log in to Microsoft Copilot Studio
2. Select the right Power platform environment for your CRM instance

- To find this information for Dynamics 365 CRM, refer: [Find your environment and organization ID and name](#)
 - For Salesforce CRM, you can use *msdyn_viva*
3. Navigate to *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. Toggle *Enable plugin* to *No*.
 6. Click *Save*. This disables the plugin in Copilot for Sales. NOTE: It may take a few minutes for the settings to take effect.

Change the users for a plugin

To change the users for a plugin:

1. Log in to Microsoft Copilot Studio
2. Select the right Power platform environment for your CRM instance
3. Navigate to *Copilot for Sales > Manage* Here you can see all the plugins that are enabled in this environment.
4. Select the plugin you want to change.
5. Select the users you want to enable the plugin for. You can use one of the three options:
 - Only Copilot Administrators (default)
 - Entire organization
 - Specific users/groups

ⓘ Note

If a plugin is enabled, it is automatically enabled for all Copilot Administrators.

6. Click *Save*. This enables the plugin for the users you selected. You also need to:
 - [Create and assign security role to user](#)
 - [Share the plugin with the user](#)

See Also

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Key concepts - Authoring Copilot Studio copilots

Article • 04/26/2024

ⓘ 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.

When you create copilots with Copilot Studio, you author and edit topics. Topics are specific conversation paths in a copilot that allow the user and copilot conversation to feel natural and flow appropriately.

Creating a copilot with Copilot Studio is easy to do with the [no-code authoring canvas](#). There are many ways you can [manage how topics interact](#) and how you want the conversation to flow. You determine what a conversation should feel like.

You can [test the copilot](#) without having to fully deploy it whenever you make a small change.

Use the [lesson topics](#) to guide you through topic authoring—from simple to complex scenarios, including default [system topics](#).

You can also [choose a language for your copilot](#) to best fit your needs.

Accessibility

The copilot authoring canvas is built for accessibility in accordance with [Microsoft Accessibility Guidelines](#) and supports standard navigational patterns. For complex scenarios, users who author content with keyboard navigation can use routing anchors.

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.

In this section

[] Expand table

Topic	Description
Create your first copilot	Create your first copilot in less than 30 minutes.
Use lesson topics	Use a series of prebuilt topics that guide you through authoring topics—from simple to advanced.
Create and edit topics	Create and edit topics using the no-code authoring canvas.
Use a system fallback topic	Use a system fallback topic when your copilot can't find an answer.
Manage topics	Copy, edit, and manage which topics are available in your copilot.
Test your copilot	Test your copilot directly within the Copilot Studio portal without needing to deploy or publish it.

Topic	Description
Supported languages	Choose a language when you create your copilot. Languages affect the user interface, system, and template topic localization.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create and delete copilots

Article • 04/26/2024

ⓘ 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 create a copilot, using built-in content building blocks containing topics, trigger phrases, and preauthored copilot conversations.

ⓘ Note

Copilots created in the web app and in Microsoft Teams are preconfigured to use **Only for Teams and Power Apps** authentication.

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

To create your first copilot, see [Quickstart: Create and deploy a Copilot Studio copilot](#).

Web app

If you create a copilot with the Copilot Studio app in Microsoft Teams, you won't see the copilot in the [web app of Copilot Studio](#).

New copilots created in the web app and in Teams are configured with **Only for Teams and Power Apps** authentication.

Create a copilot

1. Go to the [Copilot Studio home page](#).
2. Select **+ New copilot** from the **Copilots** page, or select **Home** then choose **+ Create a copilot**.
3. For **Copilot name**, enter a name for your copilot.

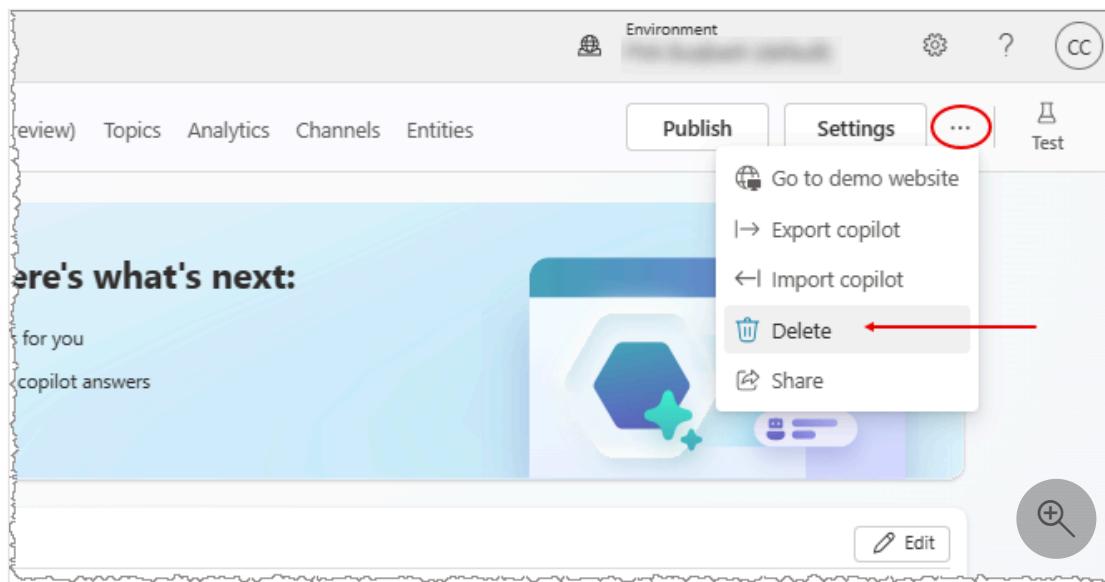
4. For **What language do you want your copilot to speak?**, select the language you want your copilot to speak. For more information, see [Supported languages](#).

5. Select **Create**.

Delete a copilot

You can delete copilots to remove them from your environment.

1. Go to the [Copilot Studio home page](#).
2. In the navigation menu, select **Copilots**.
3. Select the copilot you want to delete, then select the **...** at the top.



4. Confirm the deletion of the copilot by typing the copilot's name. Once you confirm, all copilot content will be deleted after a few minutes.

If your license is expired, you can delete your copilot(s) by selecting **Permanently delete your copilots**.

Known issues with creating a copilot

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

Insufficient permissions for the selected environment

You might 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 copilot again.

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

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

IT Helpdesk

Article • 07/03/2024

Publisher: Microsoft

IT Helpdesk 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 copilot 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 copilot to view details and statuses, enabling you to keep track of your inquiries.

ⓘ Note

Templates are currently available in English only. Additionally, templates should be limited to internal use within your organization.

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

[https://learn-video.azurefd.net/vod/player?id=42d8a595-ca0c-4a03-b6e8-6c79ea6ce9ea&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-it-helpdesk ↗](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 copilots, see [Quickstart: Create and deploy a Copilot Studio copilot](#)
- Copilot Studio message capacity
- ServiceNow connection and access to an instance with the knowledge base plugin enabled

Integrate with ServiceNow during installation

Details

Name * 11/42

 [Change icon](#) Used to represent the copilot. Icon should be in PNG format and less than 30 KB in size.

Description 81/1000

Instructions 504/8000

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

 ServiceNow [Set up connection](#)  

 [Edit](#)  [Delete](#)

Language
The primary language your copilot will use
 

Once created, the primary language cannot be changed. [Learn more](#)

Knowledge
Add knowledge sources 

In the settings of your copilot, 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 copilot.

Connection setup

ServiceNow

* 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

The screenshot shows a 'Connection setup' dialog box for ServiceNow. At the top left is the ServiceNow logo. At the top right is a close button ('X'). Below the logo is a red header bar containing the word 'ServiceNow' and an information icon ('i'). The main area contains four input fields with validation stars: 'Authentication Type' (set to 'Use Oauth2'), 'Instance Name' (description: 'The instance name used to identify the ServiceNow Site URL'), 'Client Id' (description: 'The ID used to identify this application with the service provider'), and 'Client secret' (description: 'The shared secret used to authenticate this application with the service provider'). A 'Sign in' button is located at the bottom of the form. At the bottom right are 'Confirm' and 'Close' buttons.

Capabilities

Employees can use the *IT Helpdesk* copilot to troubleshoot technical issues they encounter with their devices or software. The copilot provides instructions to resolve common problems based on your organization's existing knowledge articles hosted in ServiceNow, or ones that you add.

If the copilot 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 copilot to check the status of their tickets.

The *IT Helpdesk* copilot 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 copilot to ask their question using natural language. The copilot, with its built-in

knowledge base and conversational capabilities, can understand the context of the question and provide a relevant response.

If the copilot can't surface relevant information, it can assist an employee with creating a ServiceNow ticket to escalate their issue to the correct support team.

An employee can use *IT Helpdesk* to search for ServiceNow tickets submitted through the interface by supplying the ticket ID. Alternatively, an employee can use the copilot to see all ServiceNow tickets, submitted through the interface, conveniently listed with relevant information.

Extension Opportunities

In cases where the *IT Helpdesk* copilot is unable to provide a solution, it can escalate the conversation to a human support agent by [configuring an engagement hub](#). The copilot can provide the agent with a summary of the issue and any relevant information gathered during the conversation.

With the help of copilot extensions, the copilot 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.

An *IT Helpdesk* copilot 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](#)

Case Management

Article • 07/03/2024

Publisher: Microsoft

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

ⓘ Note

Templates are currently available in English only. Additionally, templates should be limited to internal use within your organization.

Watch this video to get an overview of how to use the *Case Management* copilot:

[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 copilots, see [Quickstart: Create and deploy a Copilot Studio copilot](#)
- Copilot Studio message capacity
- ServiceNow connection and access

Integrate with ServiceNow during installation

Details

Name * 11/42

 [Change icon](#)
Used to represent the copilot. Icon should be in PNG format and less than 30 KB in size.

Description [\(i\)](#) 81/1000

Instructions [\(i\)](#) 504/8000

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

 ServiceNow [Set up connection](#)  

Language
The primary language your copilot will use
 
Once created, the primary language cannot be changed. [Learn more](#)

 [Edit](#)  [Delete](#)

Knowledge
Add knowledge sources [\(+\) Add knowledge](#)

In the settings of your copilot, 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 copilot.

Connection setup

ServiceNow

* 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

The screenshot shows a 'Connection setup' dialog box for ServiceNow. At the top left is a red power icon, followed by the text 'ServiceNow'. At the top right is a close button (X). Below the title, there are four input fields with validation stars: 'Authentication Type' (set to 'Use Oauth2'), 'Instance Name' (description: 'The instance name used to identify the ServiceNow Site URL'), 'Client Id' (description: 'The ID used to identify this application with the service provider'), and 'Client secret' (description: 'The shared secret used to authenticate this application with the service provider'). A 'Sign in' button is located at the bottom of the form. At the bottom right are 'Confirm' and 'Close' buttons.

Capabilities

A *Case Management* copilot 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 copilot a question and get helpful information with citations returned from the provided documentation from the company. If unsatisfied with the provided information, the copilot helps in the creation of a new case.

Use Cases

A customer purchases a new washer but has difficulty with its operation. They open a *Case Management* copilot to ask for assistance. The customer describes the problem. The copilot 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. The *Case Management* copilot assists the customer in creating a

new support case by gathering details about the washer issue, such as error codes or specific symptoms. The copilot 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](#)

Team Navigator

Article • 06/12/2024

Publisher: Microsoft

Team Navigator is a helpful tool that assists employees in navigating their company's organizational structure within Teams. Team Navigator 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 streamlines the process of finding the right people within the organization.

ⓘ Note

Templates are currently available in English only. Additionally, templates should be limited to internal use within your organization.

Watch this video to get an overview of how to use the *Team Navigator* copilot:

[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 copilot for a coworker's reporting line using their name or email.
- Employees can ask the copilot for details on a colleague using their name or email. The copilot 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 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 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 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 copilot 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 copilot interface.
- Get the copilot to understand your questions and context on its own by turning on the Generative (preview) AI feature. Enabling this feature allows your copilot 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

Article • 06/12/2024

Publisher: Microsoft

Safe Travels is a Business-to-Employee (B2E) copilot designed to provide employees of a company with travel assistance. This copilot helps ensure employees are well-prepared and informed for their next work trip. This copilot 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 copilot currently only covers US travel destinations. You can replace the default website with your own knowledge source.

ⓘ Note

Templates are currently available in English only. Additionally, templates should be limited to internal use within your organization.

Watch this video to get an overview of how to use the *Safe Travels* copilot:

[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 template.

Installation

When using this template, you're equipped with the default travel sites attached to the template. If you want to add more data, you can do so from the **Knowledge** page of the copilot.

Open your copilot in Copilot Studio and select **Knowledge** at the top of the page.

The screenshot shows the Copilot Studio interface. At the top, there's a navigation bar with icons for Home, Create, Copilots, Library, and three dots. The title 'Safe Travels Copilot - 2' is displayed next to a profile icon. The 'Knowledge' tab is highlighted with a red box and a red arrow points to the 'Add knowledge' button. Other tabs include Configuration, Actions, Topics, Analytics, and Channels. On the right, there are buttons for Publish, Settings, Test, and a search bar labeled 'Search knowledge'. Below the tabs, there's a status message 'Last refreshed 1 minute ago'. A table lists a single item: 'US Travel Website' (Type: Public website), last modified 34 minutes ago, and a green 'Ready' status with a checkmark.

Name	Type	Last modified	Status
US Travel Website	Public website	34 minutes ago	Ready

Use cases

- Employees planning an international trip can use the copilot to review the specific travel documentation required, such as passports or visas, ensuring they have everything they need before departure.
- Employees can use the copilot 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 copilot would provide easy access to reservations directly through the copilot interface.
- Extending the copilot with information on local emergency services and company resources would allow employees to quickly connect with emergency assistance while traveling.
- Integrating the copilot 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 copilot interface.
- Employees can use the copilot 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

Article • 06/14/2024

Publisher: Microsoft

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

ⓘ Note

Templates are currently available in English only. Additionally, templates should be limited to internal use within your organization.

Watch this video to get an overview of how to use the *Store Operations* copilot:

<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 template is easy to configure and tailor for your organization's needs. For demonstration purposes, the template comes preconfigured with two Microsoft knowledge sources—the *Microsoft Store & billing* help page and the *Microsoft Store* website—and with instructions to the LLM (large language model) 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 with the title 'Store Operations - cc'. At the top, there are tabs for 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'. Below this is a 'Details' section with an 'Edit' button. It includes a icon, a 'Name' field with 'Store Operations - cc', a 'Description' field with 'Improve the efficiency of retail frontline workers by enabling easy access to store procedures and policies', and an 'Instructions' field containing a template. The 'Instructions' field text is: 'Your name is Store Operations. You are a custom Copilot that assists Retail employees in answering customer questions about Microsoft store policies and helping performing store operations. Maintain a formal tone when helping with customer inquiries. Avoid discussing policies from other companies or other stores. If the user asks about information from companies other than Microsoft don't include that information in the response.' The word 'Microsoft' is highlighted with a red box.

2. Replace the template'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 template:

- 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 template 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 template 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](#)



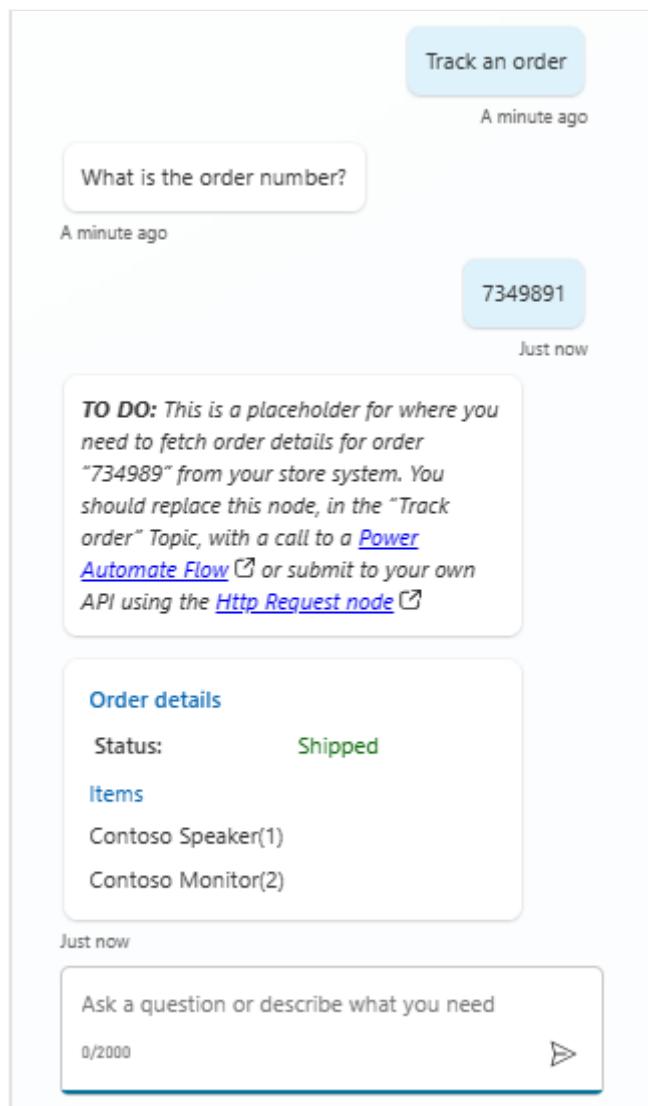
Just now

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 block of text provides step-by-step instructions for starting a return. At the bottom left, there's a note about 1 reference, a link to "Microsoft Store Refund & Return", and a timestamp indicating the response was generated by Azure OpenAI. On the right, there's a text input field for asking more questions.

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 • 06/12/2024

Publisher: Microsoft

The *Sustainability Insights* Copilot template enables users to easily get insights and data about their company's sustainability goals and progress. It also enables users to compare their company's sustainability efforts year over year and to that of other organizations, while also providing general knowledge around sustainability.

ⓘ Note

Templates are currently available in English only. Additionally, templates should be limited to internal use within your organization.

Watch this video to get an overview of how to use the *Sustainability Insights* copilot:

[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: Pointers to other organizations' public sources on sustainability for comparison

Capabilities

You can easily configure and tailor this template to suit the needs of your organization. For demonstration purposes, the template comes preconfigured to three Microsoft knowledge sources—the *Microsoft's Sustainability* website, *Corporate Responsibility* website, and *Sustainability Reports* website. You can also add your template to other data sources, such as from a peer organization in your industry, to benchmark your organization's sustainability efforts.

1. On your copilot's **Overview** page, replace knowledge sources to pointers of 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.
2. (Optional): Add more sources of sustainability information from industry peers, or any other organization with which you wish to benchmark performance. You're free to choose whatever names you like for these knowledge sources.

 **Note**

For best results, this knowledge source will have overlapping data points and similar information as to your organization's source in order to perform effective comparisons.

Use cases

The specific use cases and questions that can be addressed are dependent on the knowledge source utilized. However, the template was designed to:

- Answer questions about sustainability progress and efforts.

Sample prompts include queries like "**What is our total greenhouse gas emissions?**", "**What is 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**

A copilot built from this template might ask if you want to compare reports across two companies or compare reports year-over-year for a single company. The answer to this question helps the copilot distinguish between

use case #2 and use case #3. There are built-in quick replies to help the user with responses.

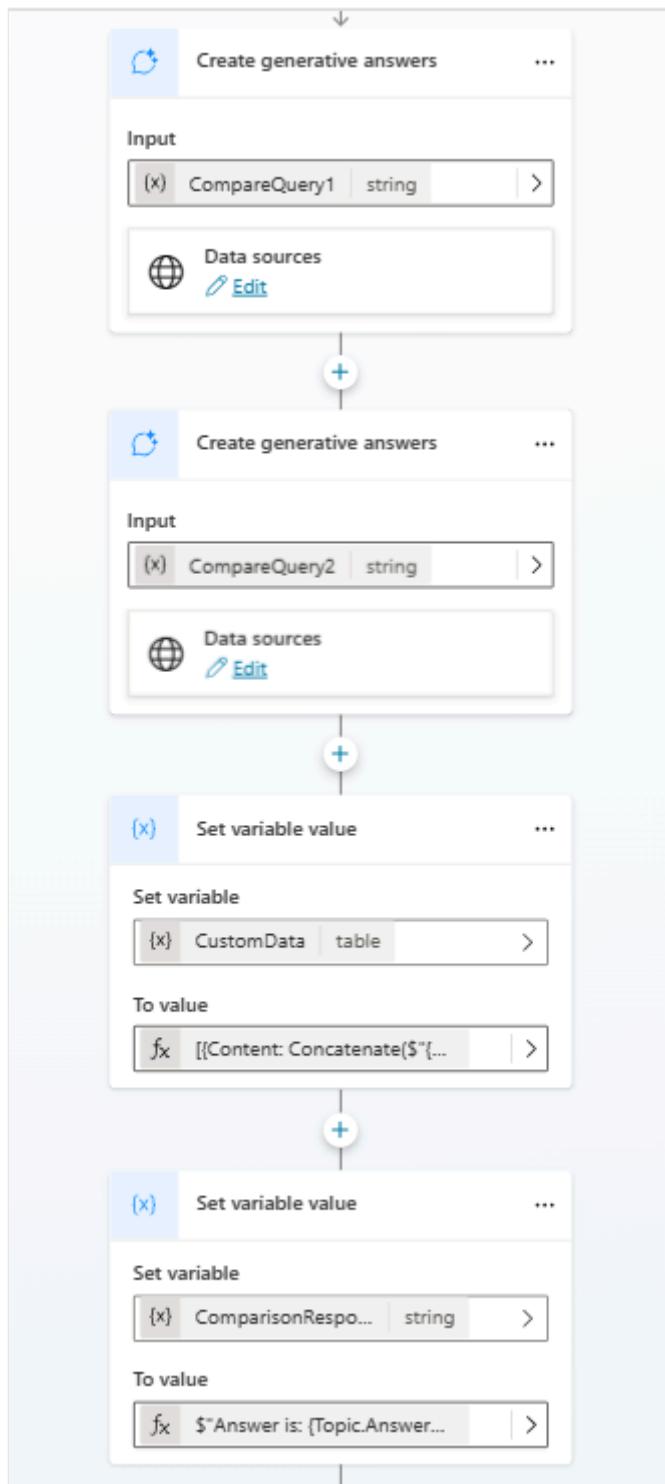
Configure the Compare topic to benchmark with other organizations

This template has a predefined pattern for comparing data across different companies from their respective data sources.

- Organizational variables can be set in the *Conversation Start* topic with the company's name (for example, your company and the company you want to compare/benchmark with – whether it be an industry peer, supplier, or anything else).
 - Variables: `OrganizationName` (Maker's company), `OrganizationToCompare` (Company for comparison purposes)
 - These variables are used both in prompts and as part of LLM (large language model) queries to ensure the data being retrieved across data sources is for that particular company.
- As previously mentioned, knowledge sources need to be configured for both `OrganizationName` and `OrganizationToCompare`.
- There's a three-step pattern that 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 | Surfaced 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 | Surfaced with Azure OpenAI |

Ask a question or describe what you need

0/2000

Extension opportunities

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

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

Article • 06/12/2024

Publisher: Microsoft

The *Weather* copilot 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 copilot with local sites to make the insights as personalized and accurate as possible.

ⓘ Note

Templates are currently available in English only. Additionally, templates should be limited to internal use within your organization.

Watch this video to get an overview of how to use the *Weather* copilot:

[https://learn-video.azurefd.net/vod/player?id=9245998b-afe8-4e75-a54c-247b0577dcd7&locale=en-us&embedUrl=%2Fmicrosoft-copilot-studio%2Ftemplate-weather ↗](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](#)

Use lesson topics in Copilot Studio

Article • 04/29/2024

ⓘ 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 includes sample topics with every new copilot. 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 copilots.

Prerequisites

- An existing copilot, or if you need to create one, see [Quickstart: Create and deploy a Copilot Studio copilot](#).
- 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

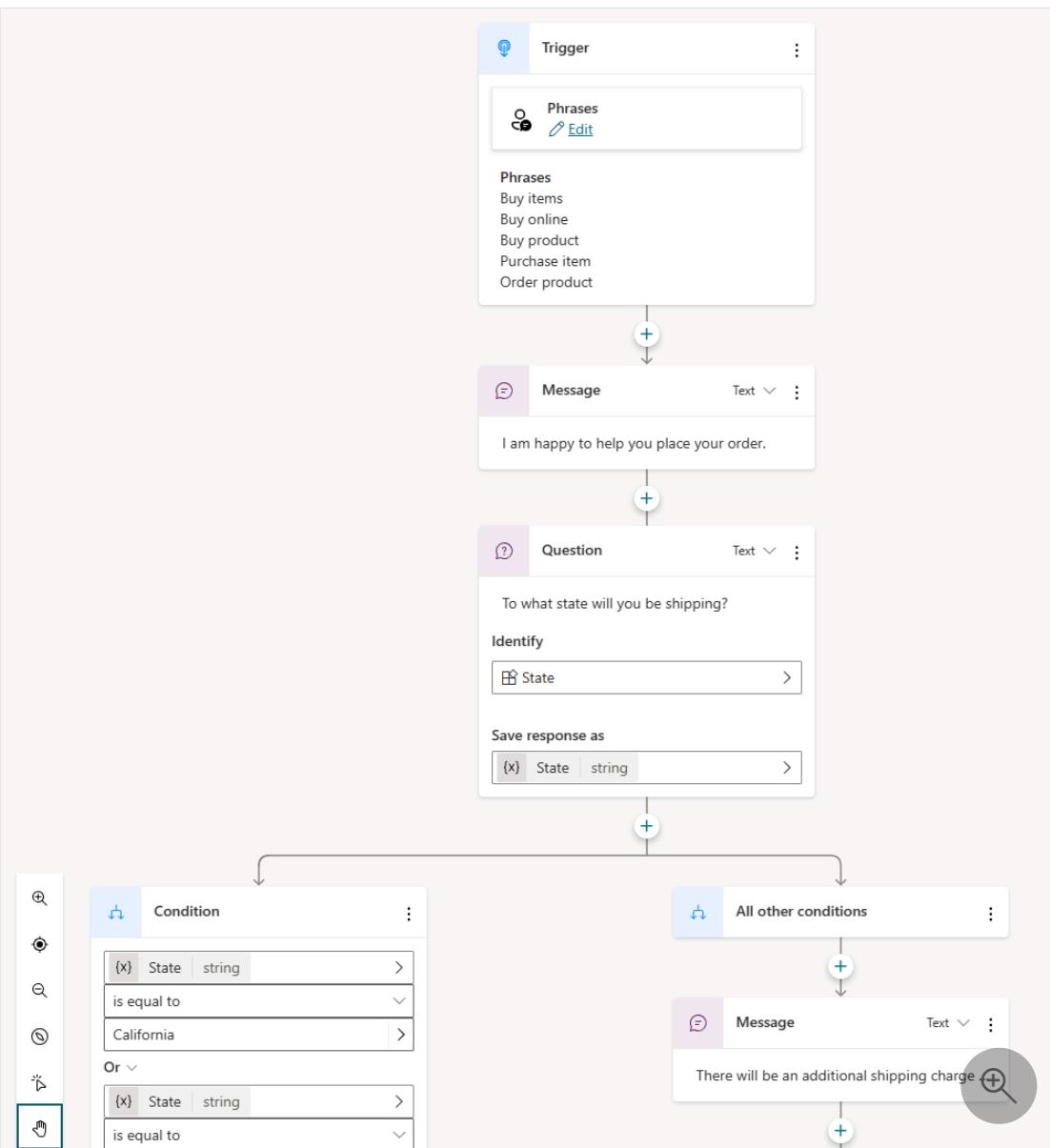
1. Open your copilot, or [create one](#), and go to the **Configure** page.
2. Review the copilot's details for name and description. The copilot derives topic phrases from the description (if any).
3. Select **Topics** at the top of the page to view the copilot's conversation flow.

The screenshot shows the Copilot Studio interface. On the left, there's a sidebar with icons for Home, Create, and Library. The main area has a header 'Copilot Studio' with a gear icon. Below it, there's a 'Copilots' section with a 'Custom copilots' dropdown containing 'Copilot'. Under 'Microsoft', there are entries for 'Copilot For Microsoft ...' and 'Copilot For Sales'. The right side is the 'Topics' tab, which is highlighted with a red box. It contains a 'Create' button, a 'Refresh' button, and a search bar with 'All' selected. A table lists seven topics:

	Name	Trigger	Description
1	Goodbye	Phrase	This topic triggers when the user says 'goodbye'
2	Greeting	Phrase	This topic is triggered when the user says 'hello'
3	Lesson 1 - A simple topic	Phrase	
4	Lesson 2 - A simple topic with a condition...	Phrase	
5	Lesson 3 - A topic with a condition, vari...	Phrase	
6	Start Over	Phrase	
7	Thank you	Phrase	This topic triggers when the user says 'thank you'

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 copilot'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 copilot 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 copilot 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 in 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.

System topics are common topics used in conversations with a copilot.

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 a copilot automatically when you create it. System topics help your copilot 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 copilot 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 copilot and its capabilities.

- Triggers when a copilot first engages with a user in conversation.

Conversational boosting

- Creates generative answers from external data sources.
- Triggers when the copilot 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 copilot 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 copilot 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 copilot pane, a detailed error message is included to help the copilot 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 copilot to use the latest published copilot content.
- Triggers with a redirection.

Sign in

- Prompts customers to sign in when user authentication is enabled. Learn how to [add user authentication](#).
- 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?



[Provide product feedback ↗](#)

Create and edit topics

Article • 06/12/2024

ⓘ 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.

In Copilot Studio, a topic defines how a copilot conversation progresses.

To author topics, you can [customize provided templates](#), create topics from scratch, or [get suggestions from existing help sites](#).

In Copilot Studio, a *topic* represents some portion of a conversational thread between a user and a copilot. 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 copilots 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 a copilot, see [Quickstart: Create and deploy a Copilot Studio copilot](#).

Natural language understanding

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

For copilot and topic limits, see [Quotas, limits, and configuration values](#).

Topics in Copilot Studio

The screenshot shows the Copilot Studio interface with the 'Web app' tab selected. The main content area is divided into two sections: 'Topic types' and 'Node types'. The 'Topic types' section contains text about system and custom topics, a list of system topic properties, and links to more information. The 'Node types' section lists available node types.

Topic types

A copilot can include two types of topics, *system*, and *custom* topics. Every new copilot 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 copilot'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 copilot 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 copilot entirely.
 - All topics that you create are custom topics.

Node types

The following types of node are available:

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.
Add a condition	Branch your conversation based on a condition.
Variable management	Set a value, parse a value, or clear all variables.
Topic management	Redirect, transfer, or end the topic or conversation.
Call an action	Call a flow like Power Automate or Excel Online, or use a connector or a plugin.
Advanced	Generative answers , HTTP requests , events , and more.

Create a topic

1. Open your copilot from the list on the **Copilots** page. For better visibility, close the **Test your copilot** 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 copilot 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 copilot topic details:

- Enter a name to identify the topic, such as "Store hours." The **Topics** page lists all the topics defined in your copilot, by this name.
- If desired, enter a display name that your copilot will show to the customer when it 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 copilot makers on your team. Customers never see topic descriptions.

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

Important

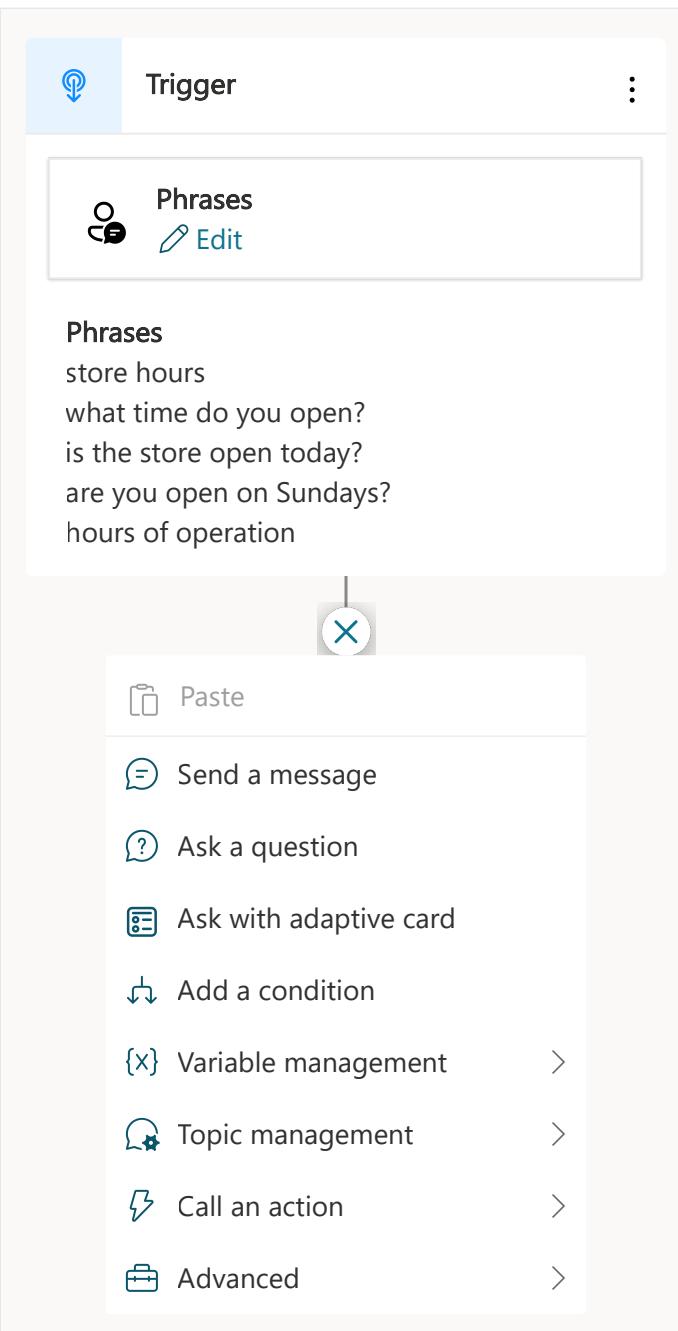
Avoid using periods (.) in your topic names. It isn't possible to export a solution that contains a copilot with periods in the name of any of its topics.

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.

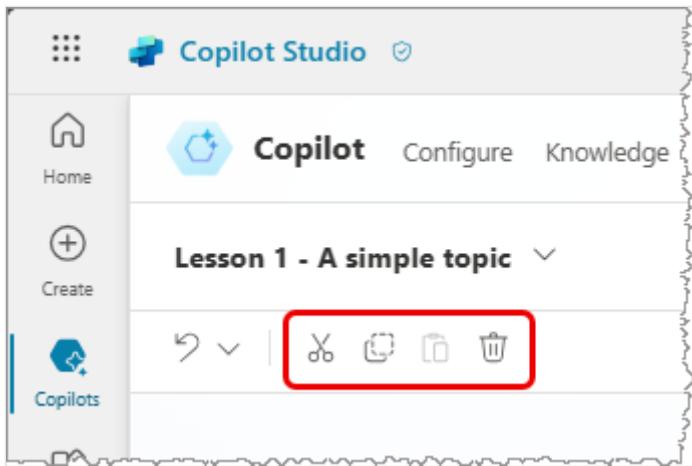


Delete a node

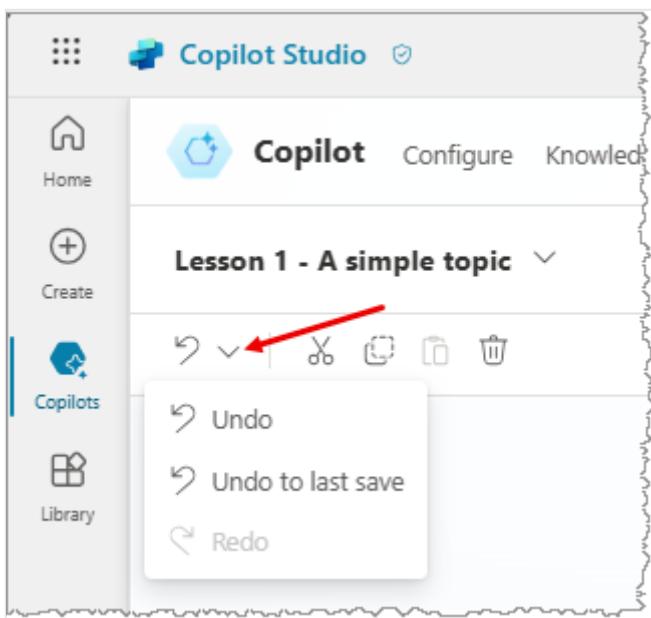
Select the menu icon (:) of the node you want to delete, and then select **Delete**.

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 copilot 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

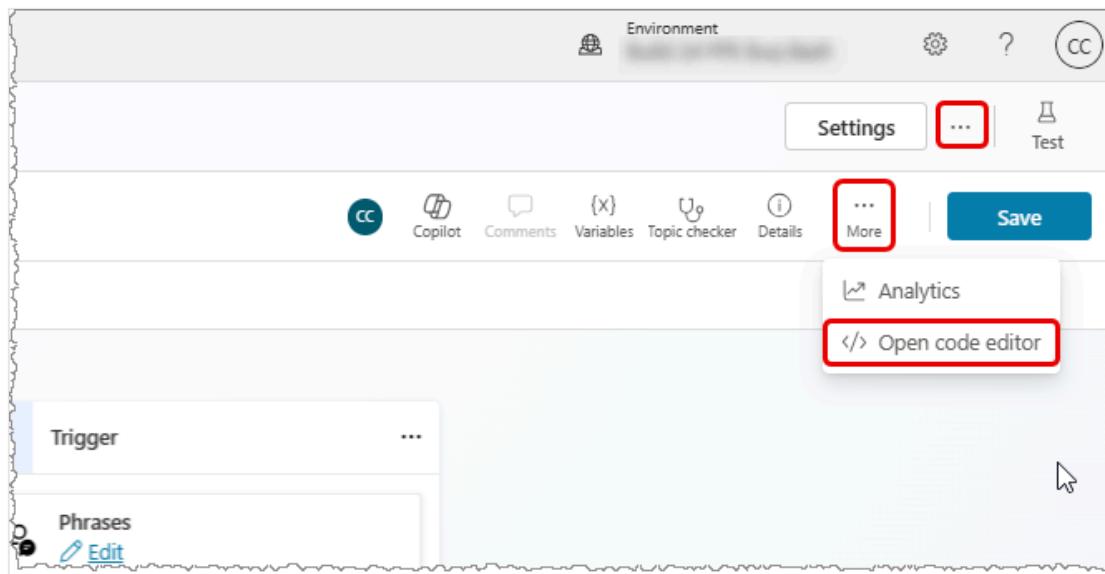
The code editor shows the topic in [YAML](#), a markup language that's easy to read and understand. Use the code editor to copy and paste topics from other bots—even ones created by other authors.

ⓘ 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 the **...** to see *More options*, 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: pbR5L0
          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.

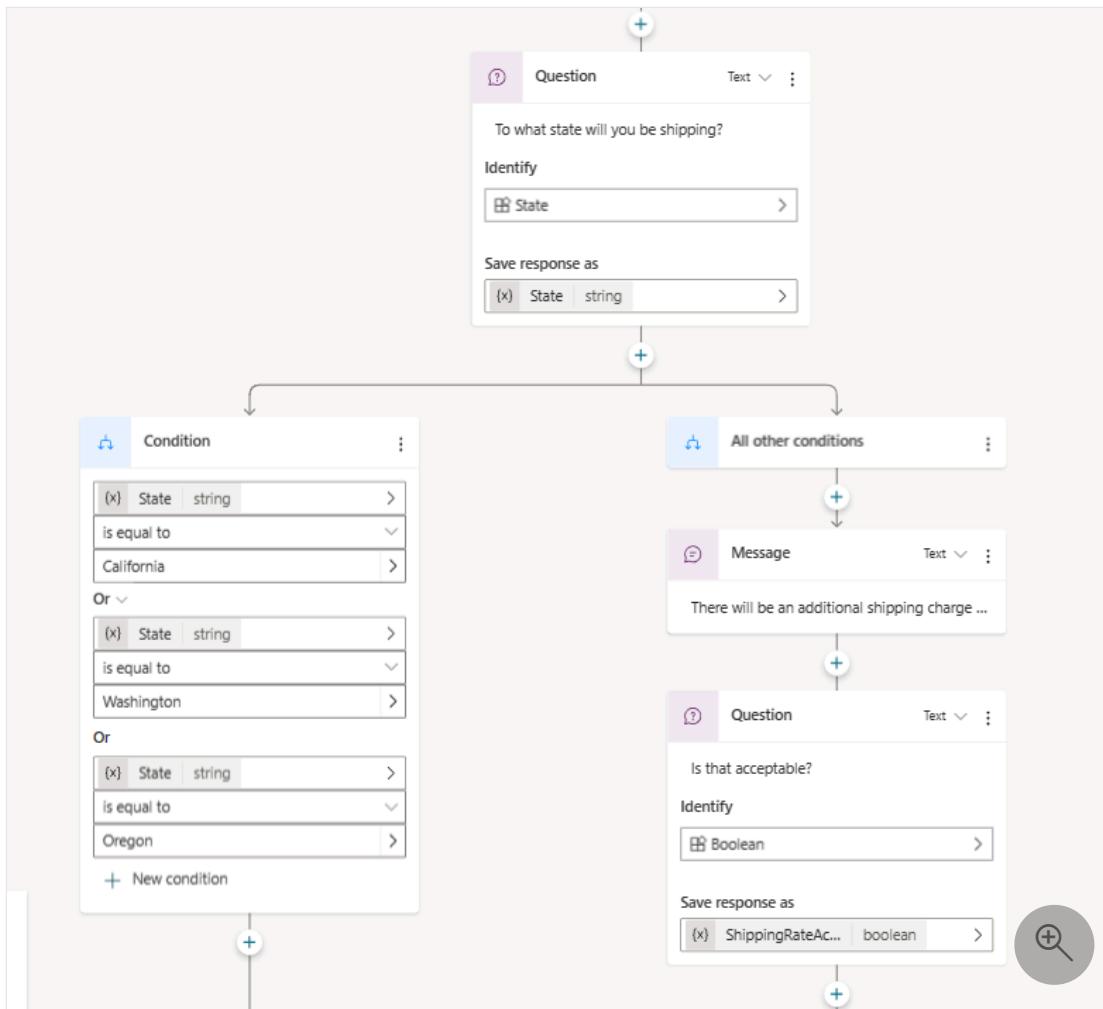
        - kind: Question
          id: 6lyBi8
          alwaysPrompt: false
          variable: init:Topic.ShippingRateAccepted
          prompt: Is that acceptable?
          entity: BooleanPrebuiltEntity

      - 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 copilot

Test your copilot when you make changes to your topics, to make sure everything works as expected.

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

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Using topic triggers

Article • 04/26/2024

ⓘ 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.

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. [Create and edit topics in your Microsoft Copilot Studio copilot](#).

ⓘ 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 copilot, 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 a copilot, the payload is a **message** type of **Activity**. Copilots can receive activities of other types with new changes in the chat.

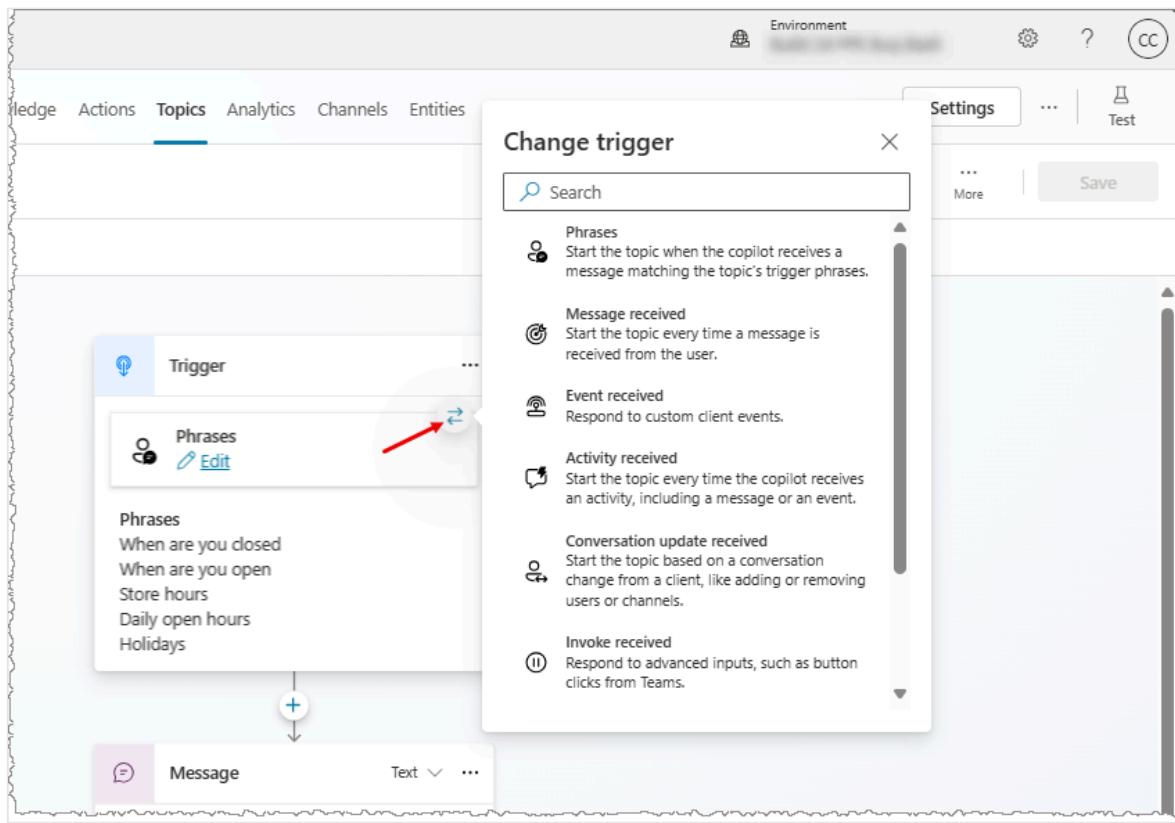
For example, Microsoft Teams sends hidden activities of type **invoke**, which signifies a user action, such as interacting with a Message Extension.

[+] 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 copilot.
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 copilot over time. The time period can be configured.

Changing the trigger for a topic

1. Open your copilot 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.

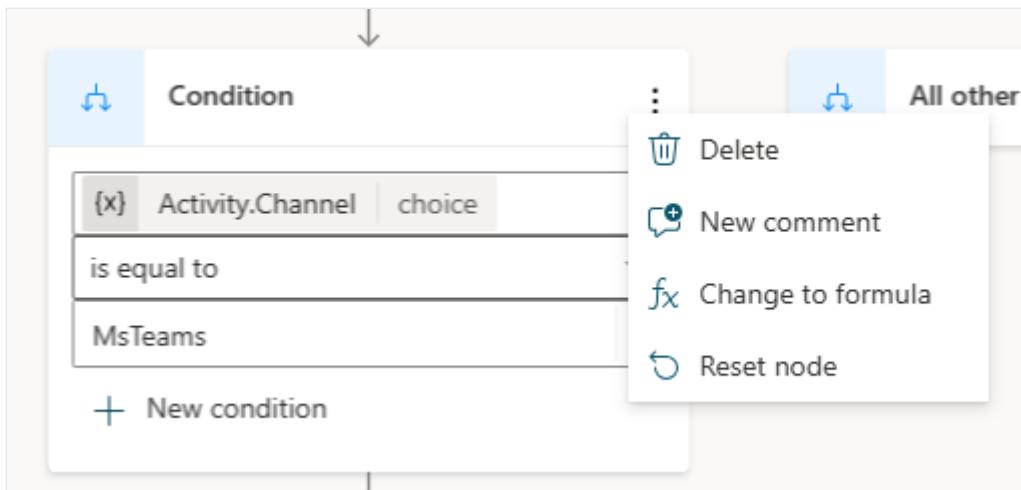
Discard

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 [Author using conditions \(preview\)](#).

Trigger priority

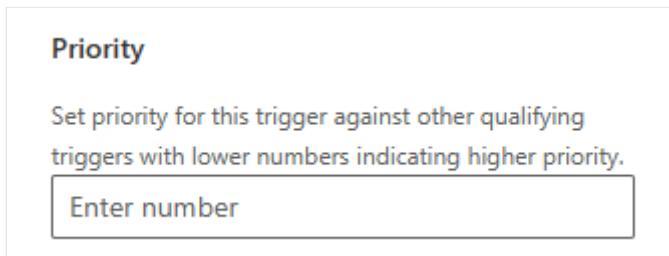
More than one trigger can fire for a single incoming Activity, such as a message. The trigger type determines the order the triggers fire.

Order of execution:

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

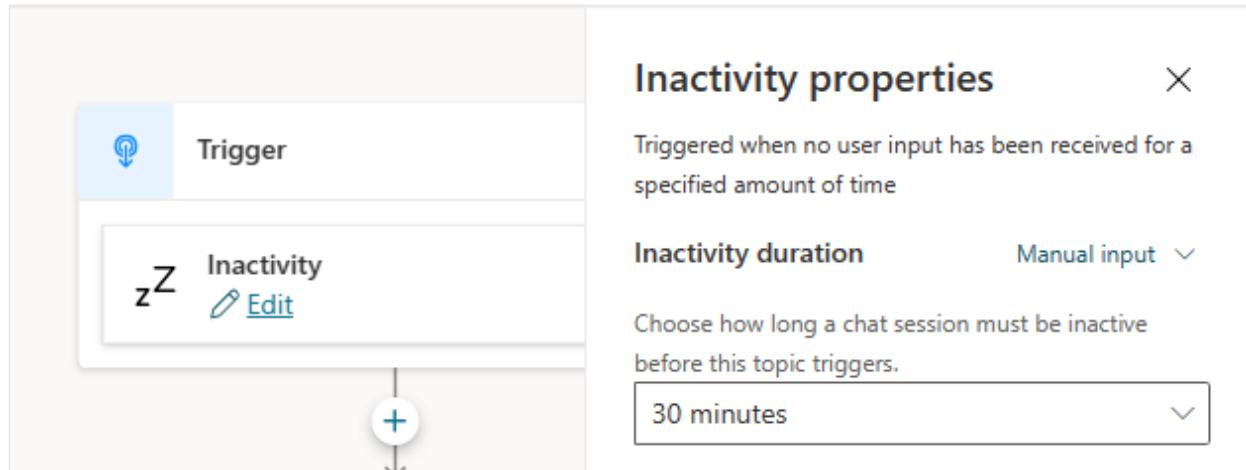
If there's more than one qualifying trigger of the same type, the triggers get executed in the order of creation (oldest first).

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 ↗](#)

Use lesson topics in Copilot Studio

Article • 04/29/2024

ⓘ 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 includes sample topics with every new copilot. 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 copilots.

Prerequisites

- An existing copilot, or if you need to create one, see [Quickstart: Create and deploy a Copilot Studio copilot](#).
- 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

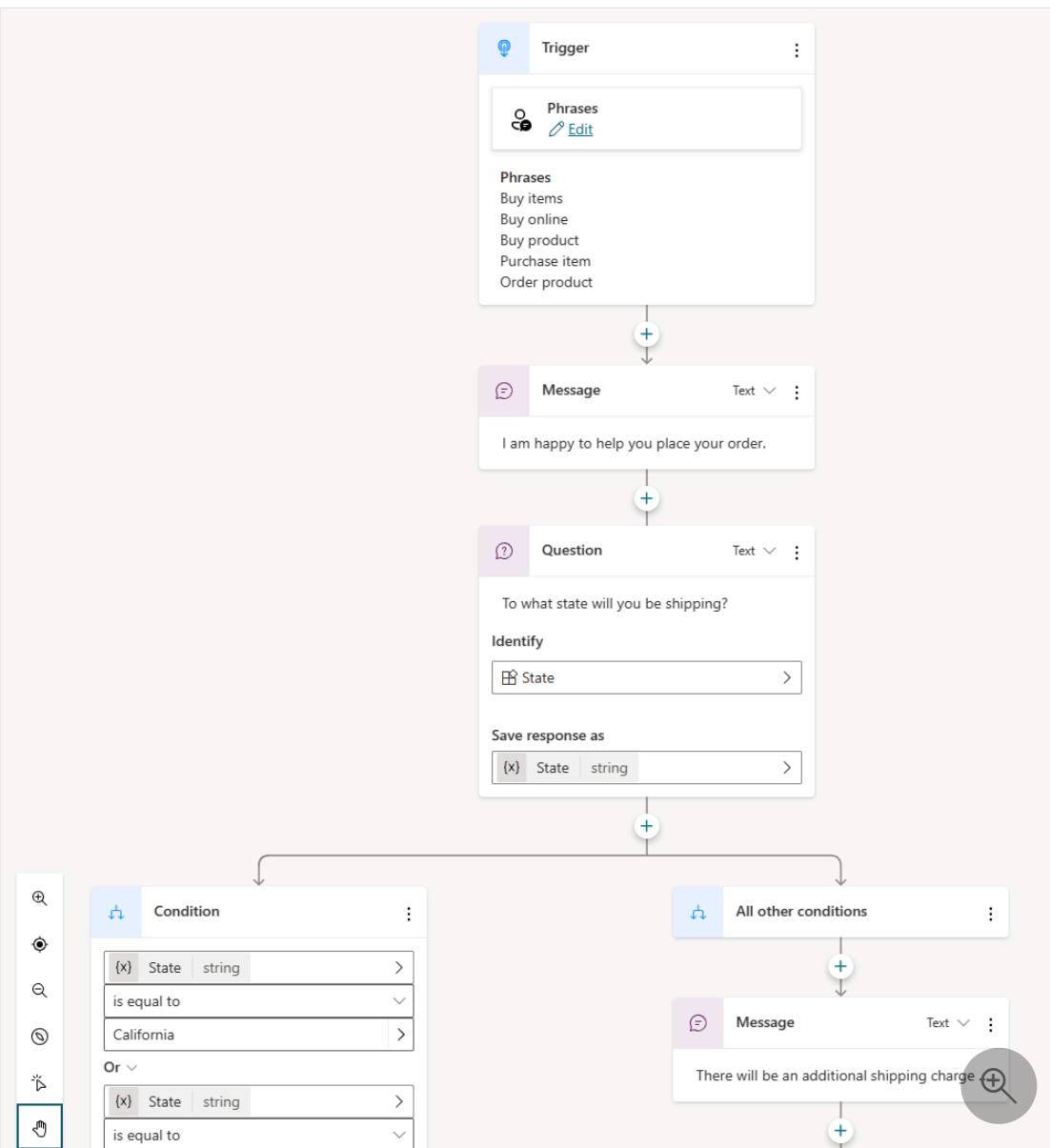
1. Open your copilot, or [create one](#), and go to the **Configure** page.
2. Review the copilot's details for name and description. The copilot derives topic phrases from the description (if any).
3. Select **Topics** at the top of the page to view the copilot's conversation flow.

The screenshot shows the Copilot Studio interface. On the left, there's a sidebar with icons for Home, Create, and Library. The main area has a header 'Copilot Studio' with a gear icon. Below it, there's a 'Copilots' section with a 'Custom copilots' dropdown containing 'Copilot'. Under 'Microsoft', there are entries for 'Copilot For Microsoft ...' and 'Copilot For Sales'. The right side is the 'Topics' tab, which is highlighted with a red box. It contains a 'Create' button, a 'Refresh' button, and a search bar with 'All' and a magnifying glass icon. A table lists seven topics:

Name	Trigger	Description
Goodbye	Phrase	This topic triggers when the user says 'goodbye'
Greeting	Phrase	This topic is triggered when the user says 'hello'
Lesson 1 - A simple topic	Phrase	
Lesson 2 - A simple topic with a condition, vari...	Phrase	
Lesson 3 - A topic with a condition, vari...	Phrase	
Start Over	Phrase	
Thank you	Phrase	This topic triggers when the user says 'thank you'

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 copilot'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 copilot 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 copilot 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 ↗](#)

Create a copilot from a classic chatbot

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.

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

Eligibility

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

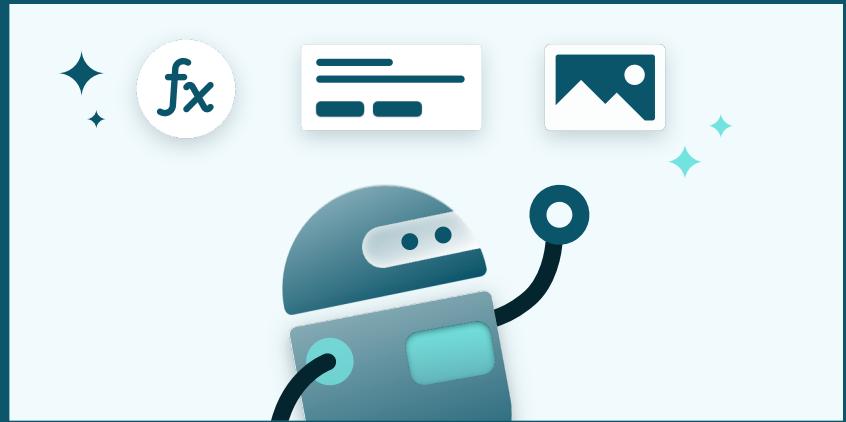
- Languages other than English
- Handoff to Omnichannel for Customer Service
- Knowledge Management extensions

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 copilot is created. However, the cloned topic has the same input and output variables as the original.

Clone a classic chatbot

1. Open your classic chatbot in Copilot Studio.
2. On the **Overview** page, select **Copy this chatbot**.



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 GPT.

[!\[\]\(b0874c0b5e5987fd16e1f0a71afb1e70_img.jpg\) Copy this chatbot](#) [!\[\]\(6047c10a2893f0ea3cfc6ce4870d482a_img.jpg\) Learn more](#)

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 copilot instead.

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

Related topics

[Create and edit topics](#).

Send a message

Article • 04/29/2024

ⓘ 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.

The **Message** node, within the flow of a topic, sends a message from the copilot 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.

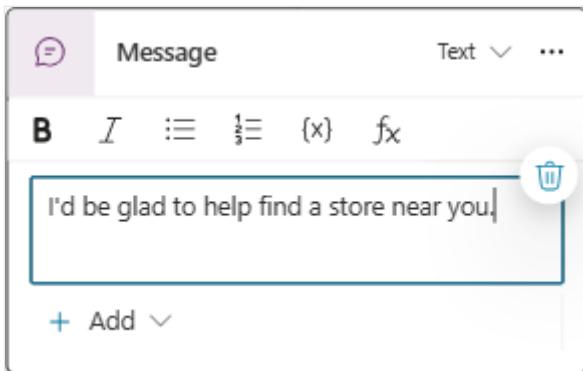
Prerequisites

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

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

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 copilot to send.
3. Optionally, use the tools in the node's menu bar to add basic formatting, [variables](#), and [message variations](#).

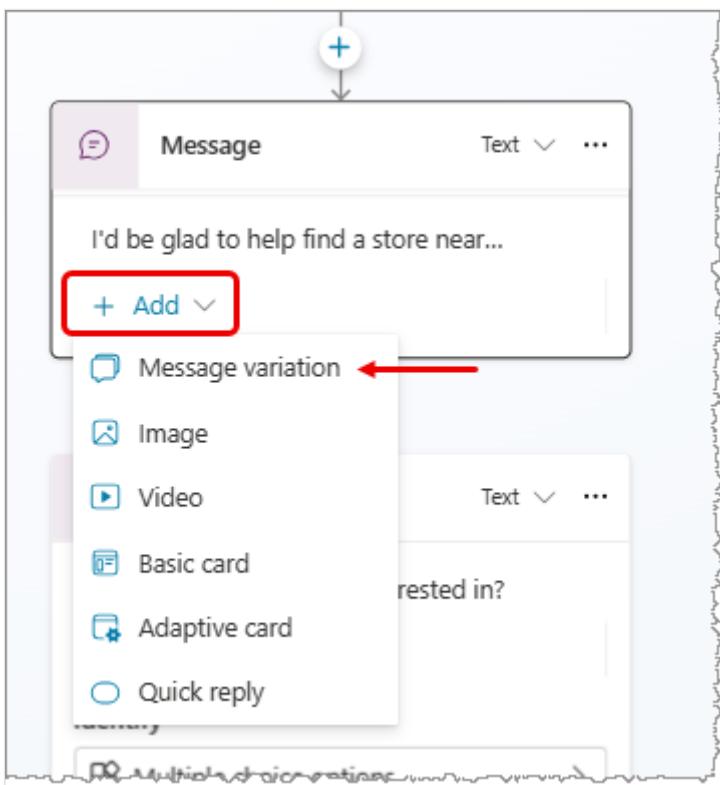


Use message variations

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

Add a message variation

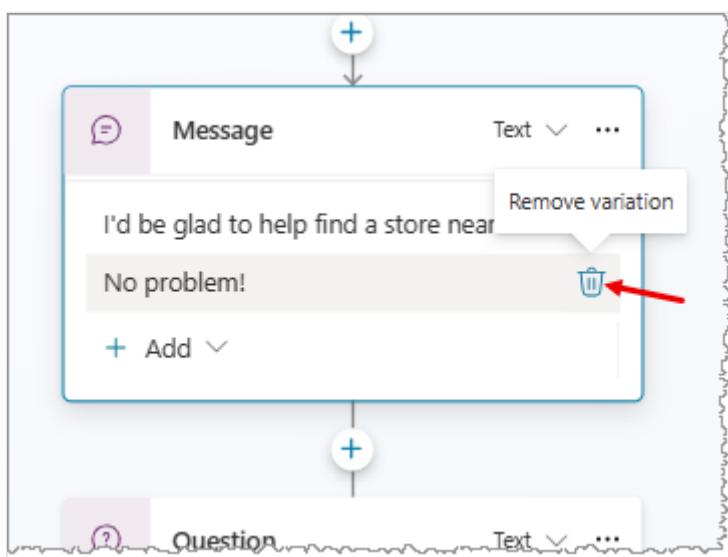
1. In the node's menu bar, select **Add**, and then select **Message variation**.



2. Enter what you want the copilot 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 trashcan icon (**Remove variation**).



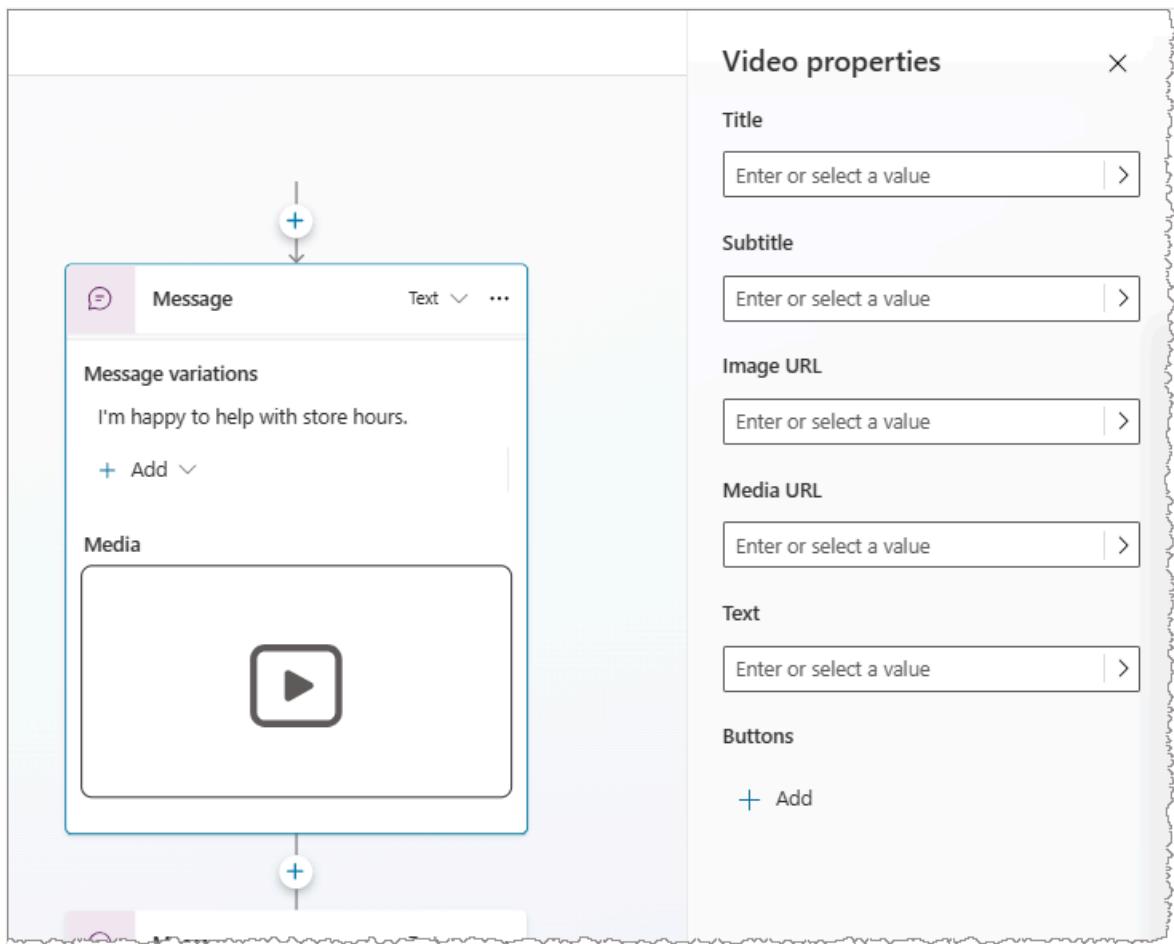
Add an image

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 a copilot 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

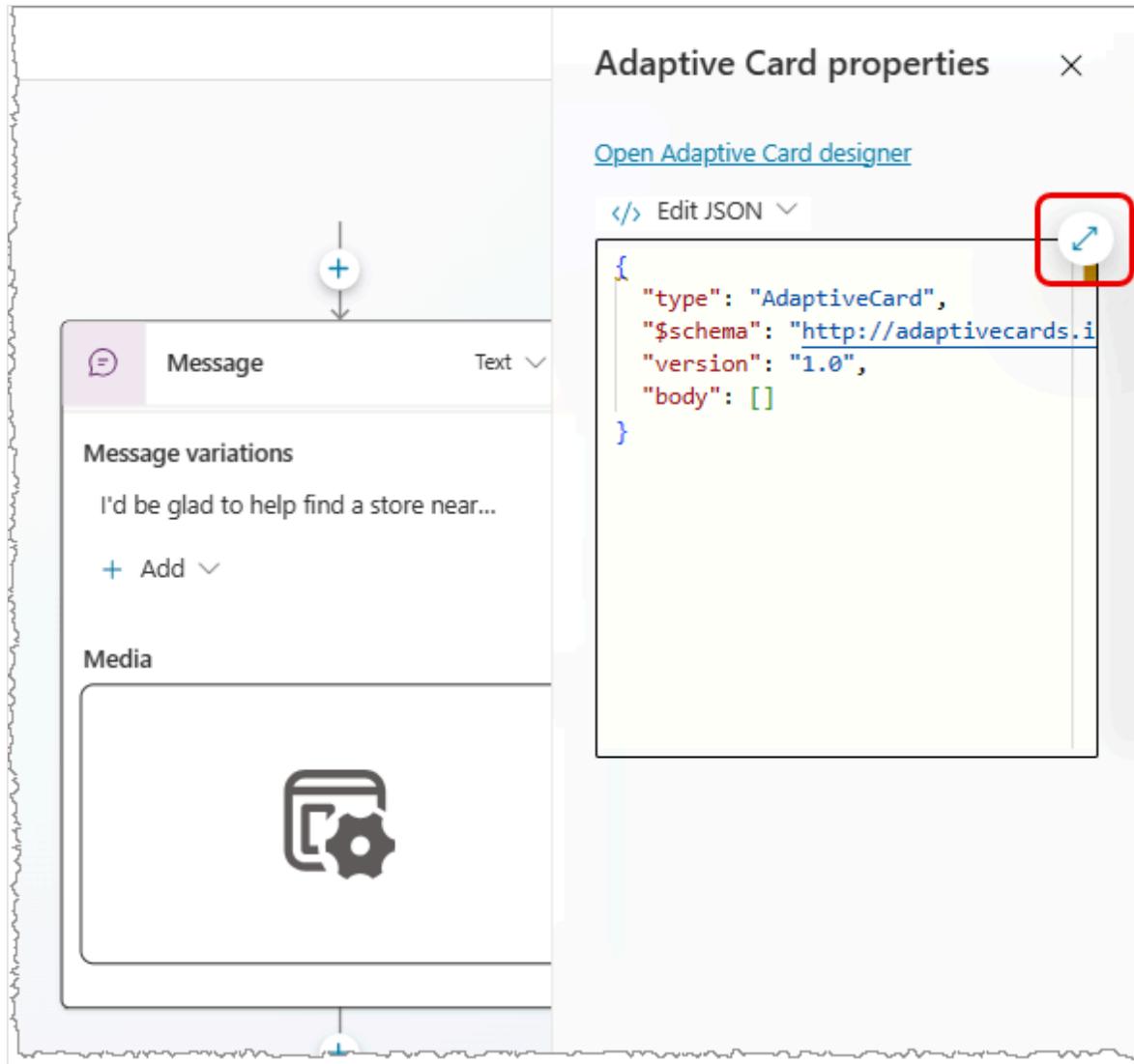
[Adaptive Cards](#) are platform-agnostic cards that you can customize. You can design Adaptive Cards using the [Adaptive Cards Designer](#) or author the JSON directly.

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](#)

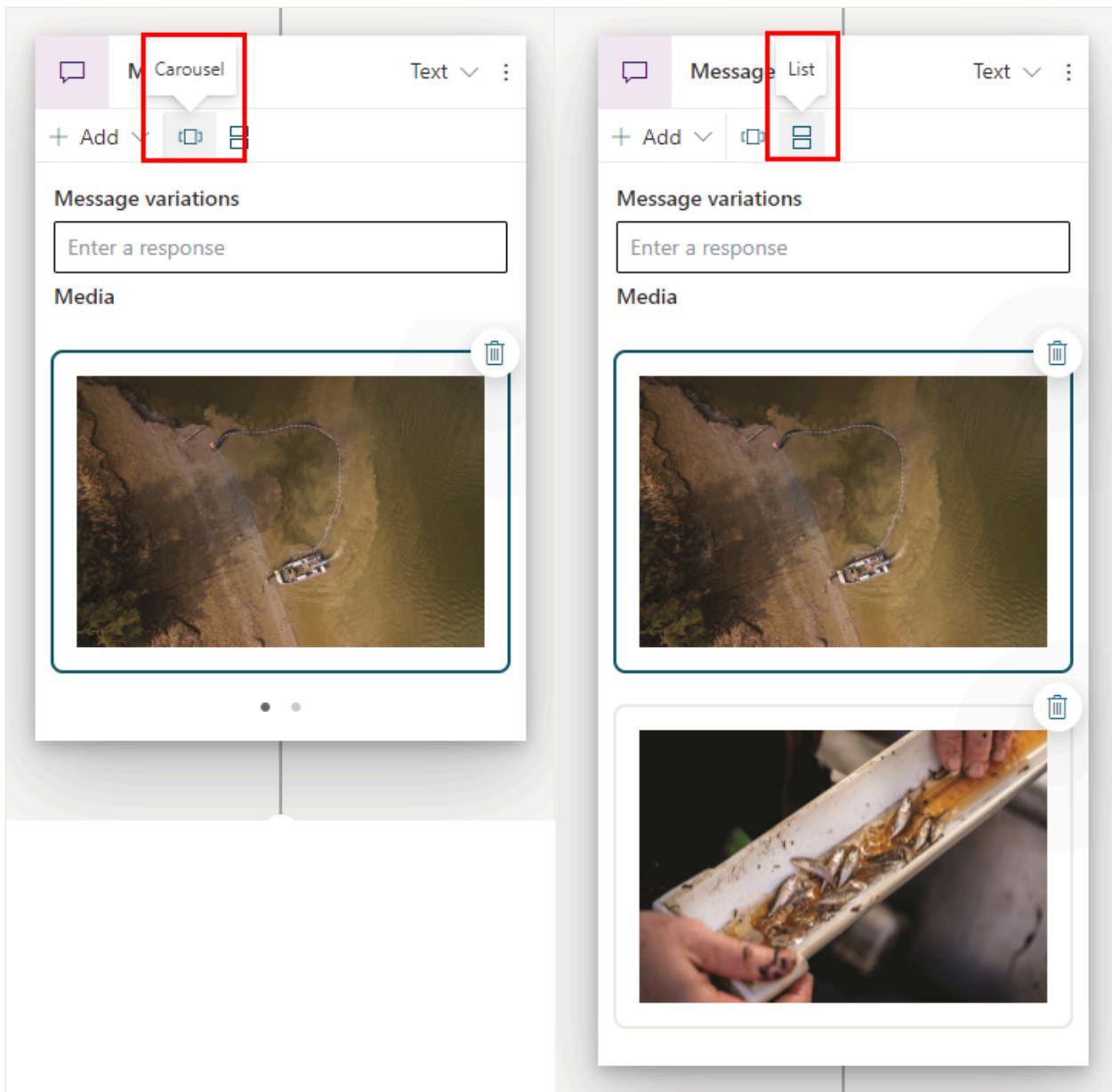
[Edit JSON](#)

```
{ "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" } ] }
```

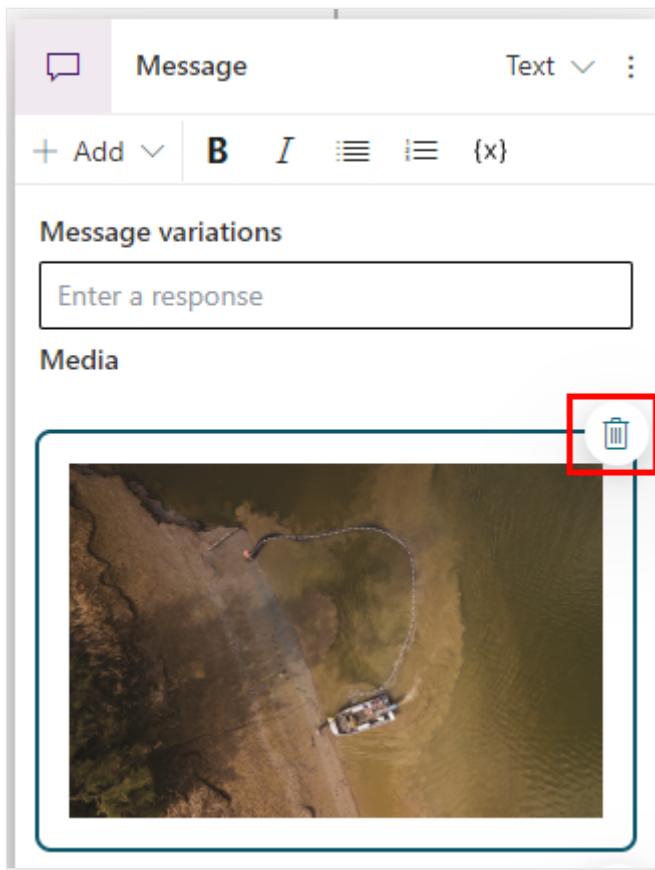
View multiple cards in the same node and remove a card

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 trash can icon (**Remove**) at the upper-right corner of the item.



Use quick replies

i 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 copilot.

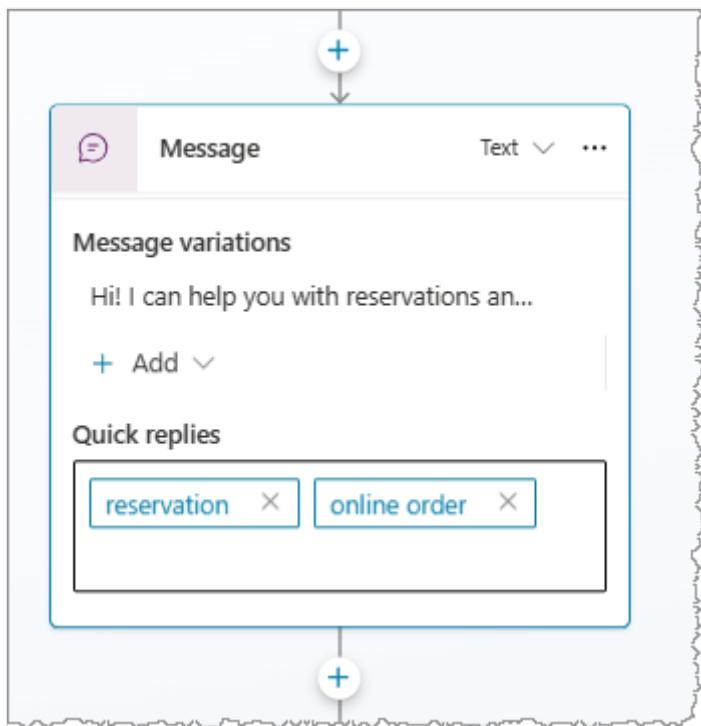
If the user chooses a quick reply text option, it gets shown in the chat history. The quick reply buttons are removed from the chat history when the copilot or user sends another activity.

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 Copilot interface. On the left, there's a preview window showing a message variation: "Hi! I can help you with reservations an..." with a "reservation" quick reply button. Below it, another message variation shows "online order" as a quick reply. On the right, two "Quick replies properties" windows are open, one for each variation. Both windows show "Send a message" selected as the type, "reservation" and "online order" as the text, and "Enter or select a value" as the title. There are also "Advanced options" sections and a "+ Add" button.

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

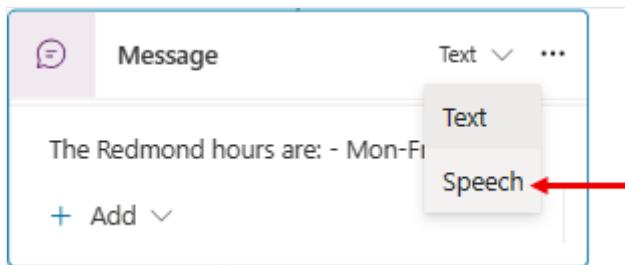
- **Send a message** (default): Send a message to the copilot and show it in the chat history.
- **Open URL**: Open a URL. The URL must start with `https://`.
- **Make a call**: Call a phone number. Your entry must be in the format `tel:1234567890`.
- **Send a hidden message to the copilot**: Send a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

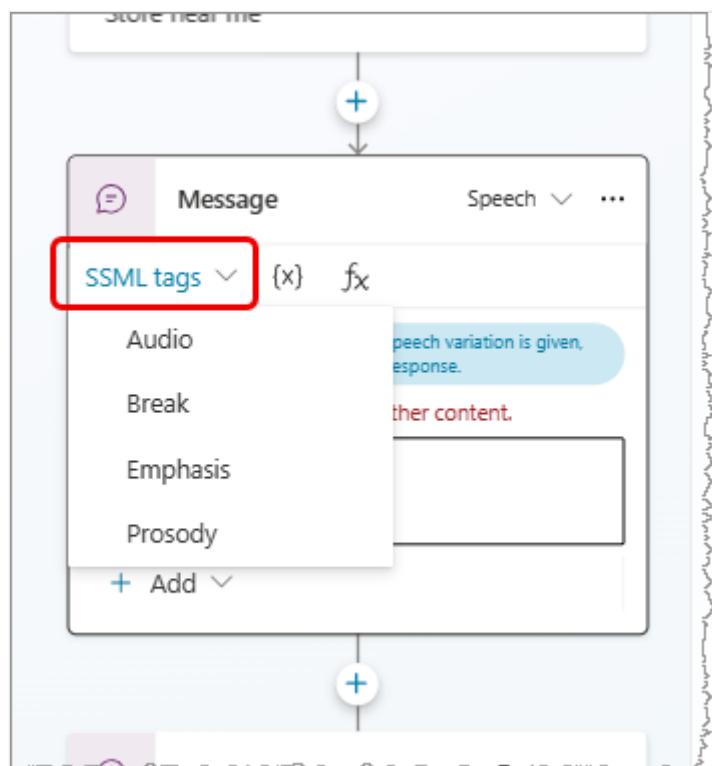


2. Add a message variation to the **Message** card. You see the menu bar changes.
3. Enter what you want the copilot to say. Add as many variations for the speech response as needed.
4. Optionally, [add SSML to your message variations](#) to control how the copilot speaks the message.

Use SSML to customize speech responses

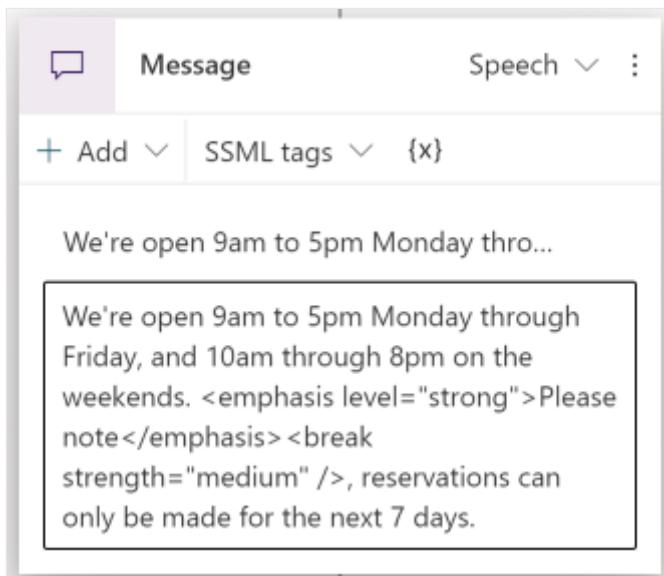
Speech Synthesis Markup Language (SSML) is an XML-based markup language that specifies how text is converted into synthesized speech. SSML is one of the many ways Copilot Studio makes your copilot's responses sound like natural speech.

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 ↗](#)

Configure the system fallback topic

Article • 04/29/2024

ⓘ 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.

During a conversation, a Microsoft Copilot Studio copilot [triggers the appropriate topic](#) based on the user's input. If the copilot can't determine the user's intent, it prompts the user again. After two prompts, the copilot escalates to a live agent through a system topic called **Escalate**.

You can customize how the copilot 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.

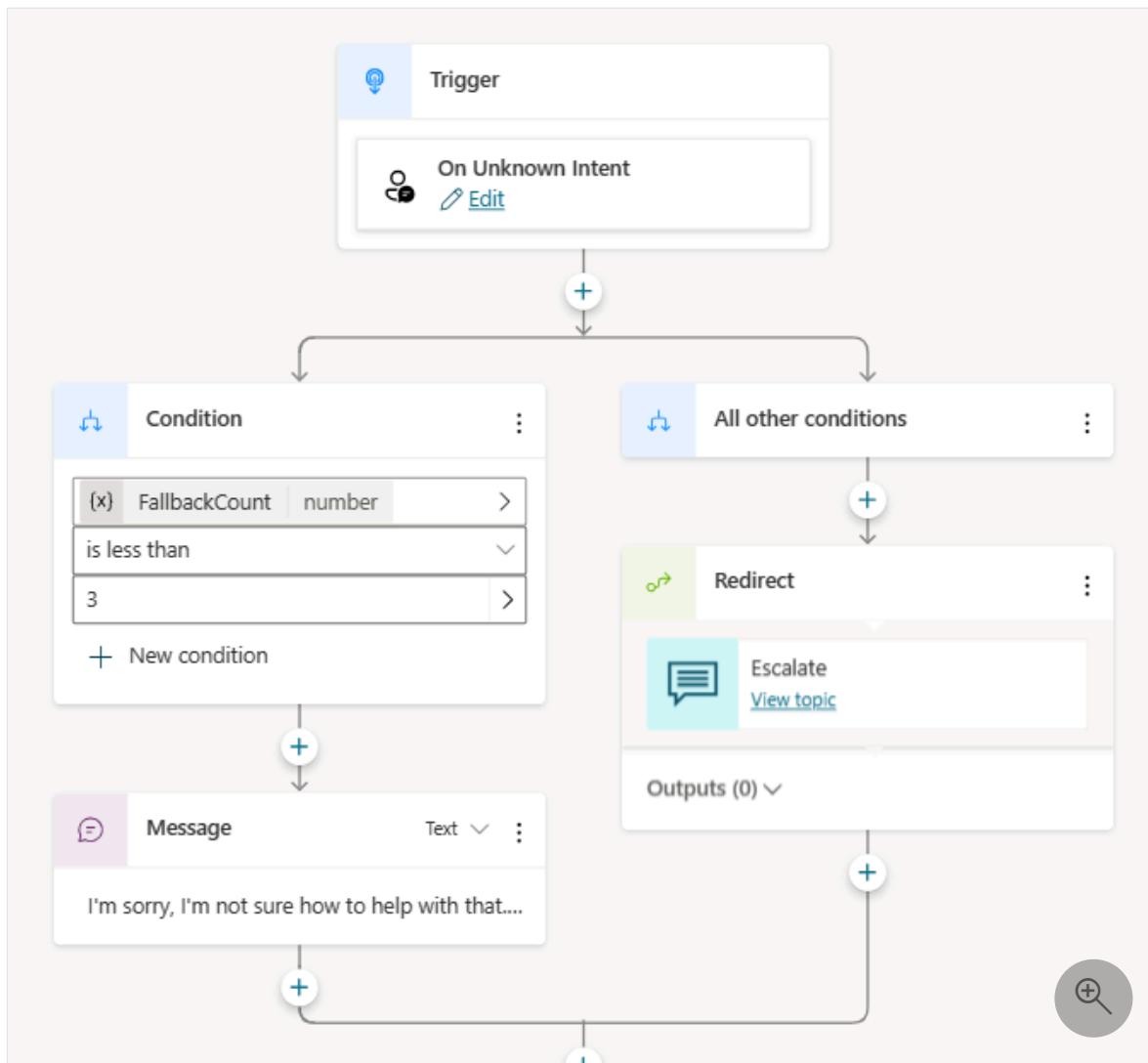
View the default system fallback topic

1. Open your copilot in Copilot Studio.
2. At the top of the page, select **Topics > System**, then open the **Fallback** topic.

The screenshot shows the Copilot interface with the 'Topics' tab selected. A red box highlights the 'Topics' tab in the navigation bar. A red arrow points to the 'System (9)' button in the top right corner of the toolbar. The main content area displays a table of system topics:

	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 copilot 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 copilot 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](#).

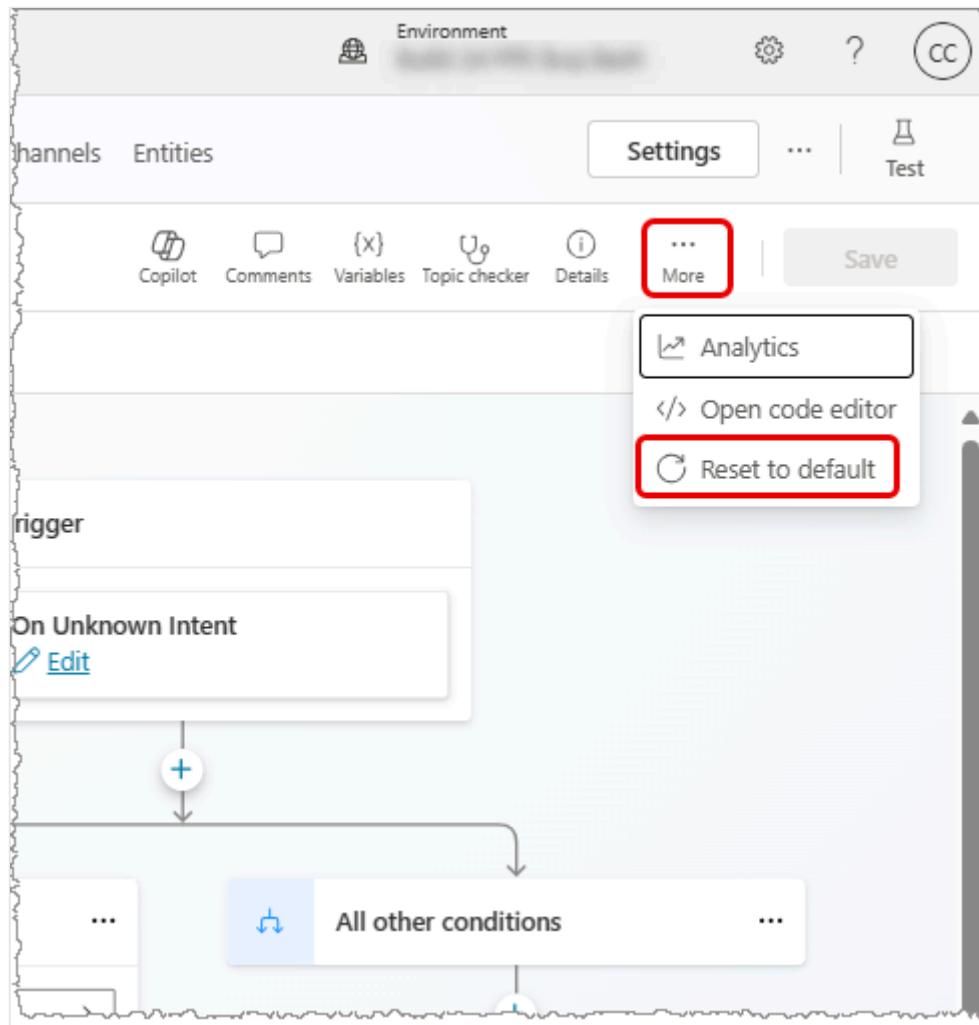
⚠ Note

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 ↗](#)

Manage topics in Copilot Studio

Article • 04/30/2024

ⓘ 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.

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 copilot. 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 copilot from working. You must fix them before you can publish your copilot.
- **Warnings** don't stop your copilot 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 copilot 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 a copilot 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 a copilot, 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.

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 copilot** pane to confirm that your copilot 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 copilot 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 ↗](#)

Ask a question

Article • 06/12/2024

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.

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.

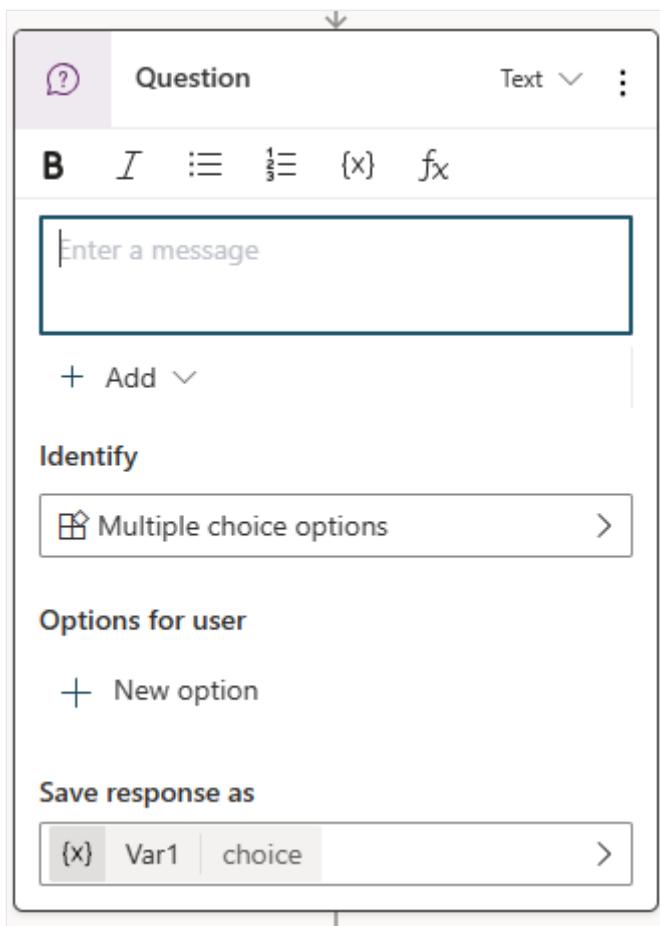
Like **Message** nodes, **Question** nodes can include images, videos, cards, quick replies, and message variations. For more information, see [Send a message](#).

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](#).
4. Depending which **Identify** option you selected, there might be more properties to set.

For example, for **Multiple choice options**, you need to 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.

5. Select the name of the default variable 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.

Configure question behavior

Question behavior properties allow you to control how the copilot responds to an invalid response or how it validates user input.

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

The **Question behavior** panel is where you can adjust behaviors like prompting, validation, and interruptions.

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 **Question behavior** pane.

- **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.

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.

Interruptions

Interruptions determines whether the user can switch to a different topic during the question.

- **Allow switching to another topic:** The user can abandon the question for a new topic.
- **Only selected topics:** Specify for which topics the user can abandon the question.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Add comments to copilots

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.

Comments are notes that are associated with items in your copilot. Use comments to help your team collaborate on the copilot design and provide feedback, or provide additional information on implementation details of your copilot.

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.

Prerequisites

- [Learn more about what you can do with Microsoft Copilot Studio.](#)

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 a copilot with a solution](#).
- You can't @ mention other copilot makers in a comment.
- New comments are displayed for [other copilot 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.

Test your copilot in Copilot Studio

Article • 04/29/2024

ⓘ 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.

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

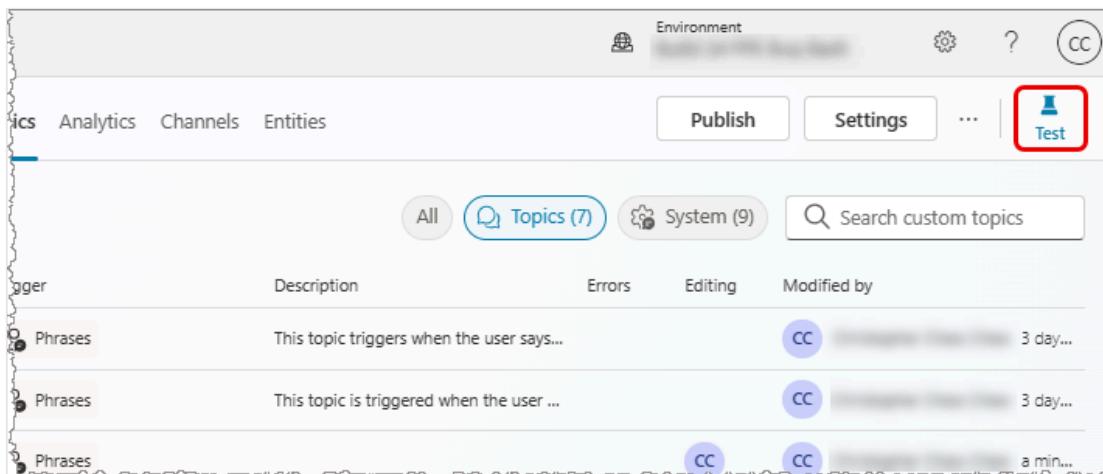
To help you find and fix unexpected behavior, you can enable tracking between topics to take you through the copilot conversation step by step, and go to the corresponding node in the authoring canvas.

Web app

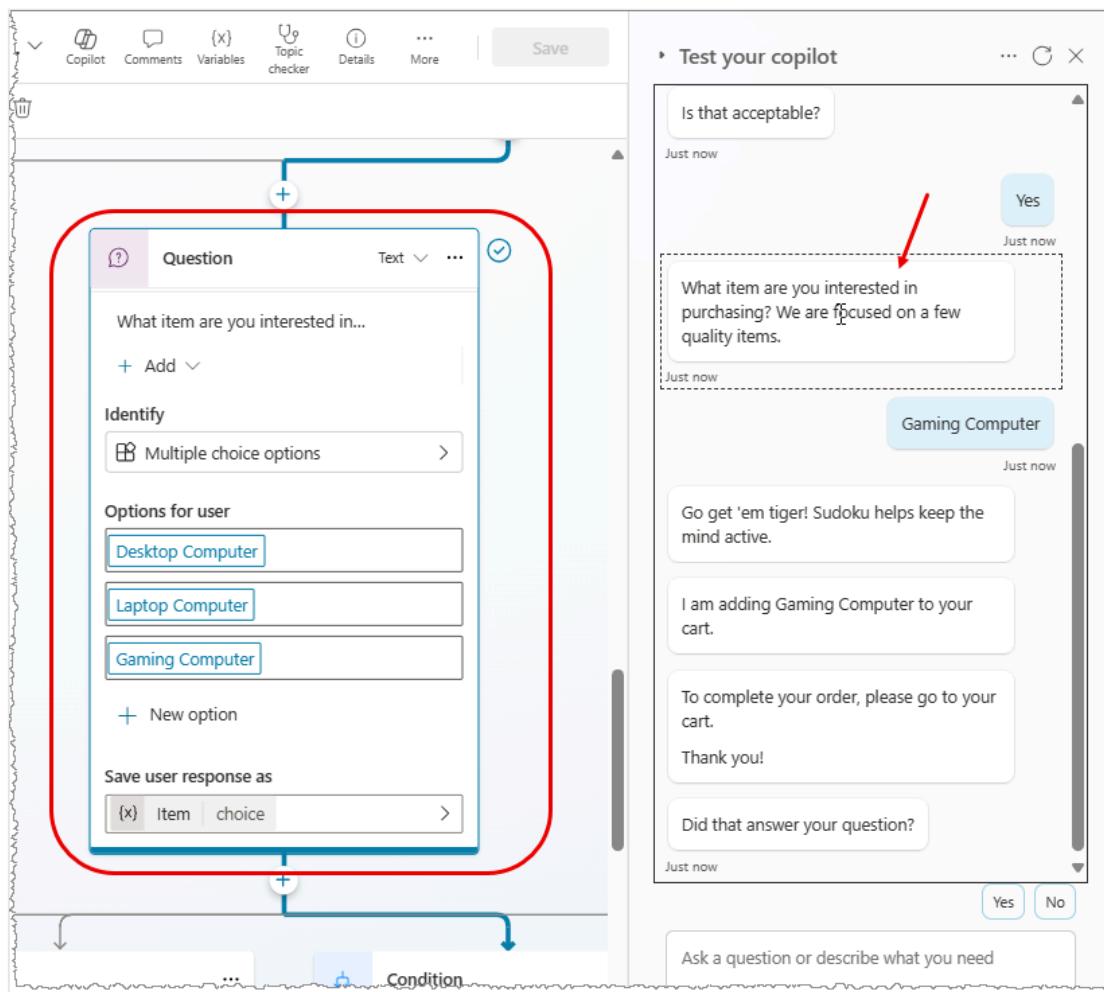
Use the **Test your copilot** pane to walk through your copilot conversations as a user. It's a good way to make sure your topics are working and that conversations flow as you expect.

Preview a conversation

1. If the **Test your copilot** pane is hidden, open it by selecting **Test** at the top of any page.

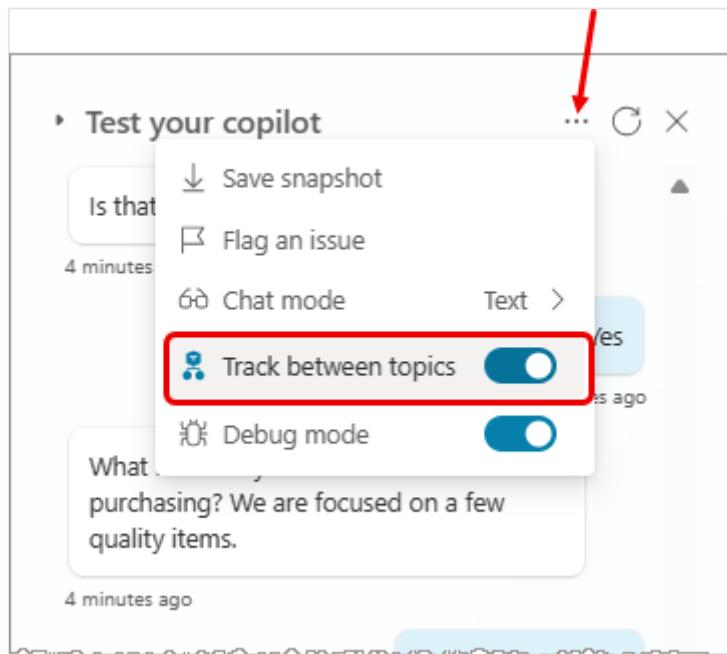


2. In the field at the bottom of the **Test your copilot** pane, enter some text. If the text is similar to a trigger phrase for a topic, that topic begins.
3. Select the copilot response in the **Test your copilot** pane. This takes you to the topic and the node that sent the response. Nodes that have fired have a colored checkmark and a colored bottom border.



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.

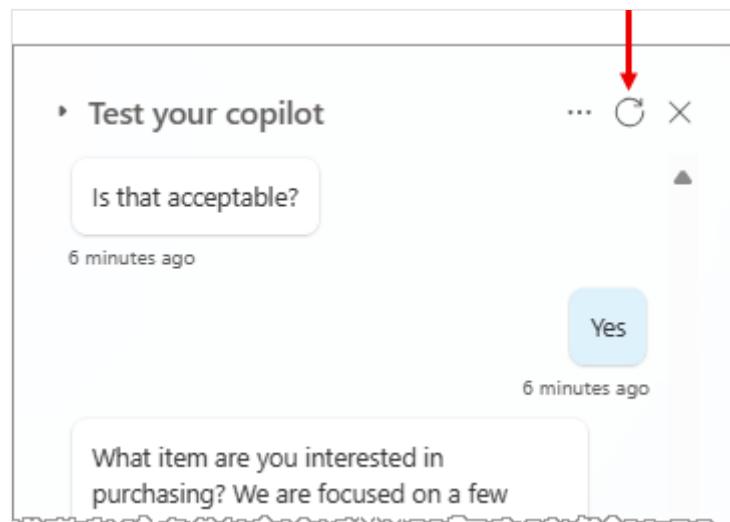
4. If you would like to follow the whole conversation automatically, as it moves from topic to topic, turn on **Tracking** at the top of the **Test your copilot** pane.



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 copilot. Simply save your topic to apply changes and continue the conversation with your copilot.

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



Test variable values

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

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.

A screenshot of the 'Variables' panel in the Microsoft Bot Framework Emulator. The title 'Variables' is at the top, followed by tabs for 'Browse' and 'Test' (which is underlined). A search icon is also present. Below the tabs, there are three sections: 'Topic (2)', 'Global (0)', and 'Environment (0)'. Under 'Topic (2)', there are two entries: 'CoffeeSize' (choice type, value 'Undefined') and 'CoffeeType' (choice type, value 'Espresso').

Category	Variable	Type	Value
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](#).

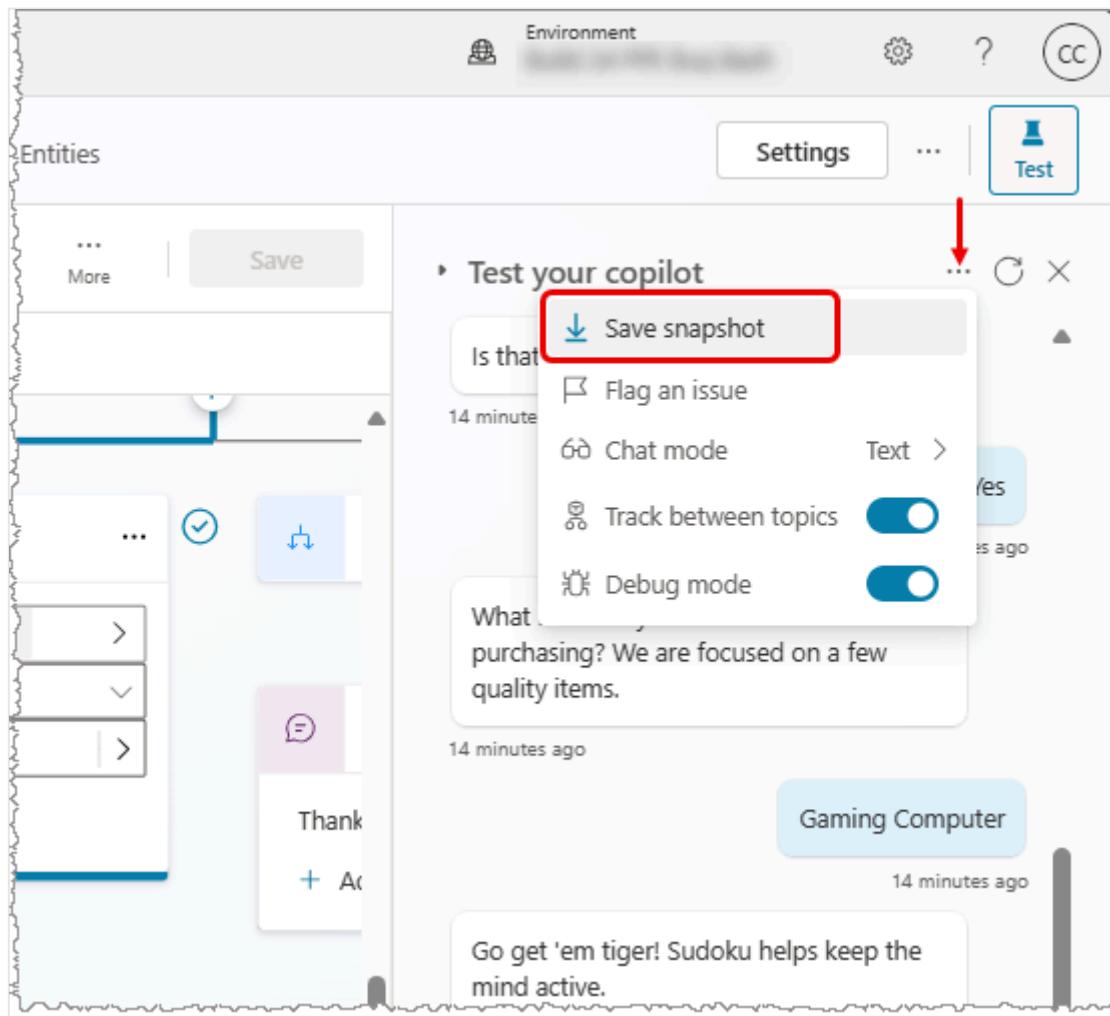
Save conversation snapshots

While you're testing your copilot, 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 copilot not responding in the way you expect.

⚠ Warning

The snapshot file contains all your copilot content, which may include sensitive information.

1. At the top of the **Test your copilot** 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 copilot 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 copilot's topics and other content, including entities and variables.

Manage connections

If your copilot requires [end 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**.

① 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 ↗](#)

Supported languages

Article • 06/04/2024

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.

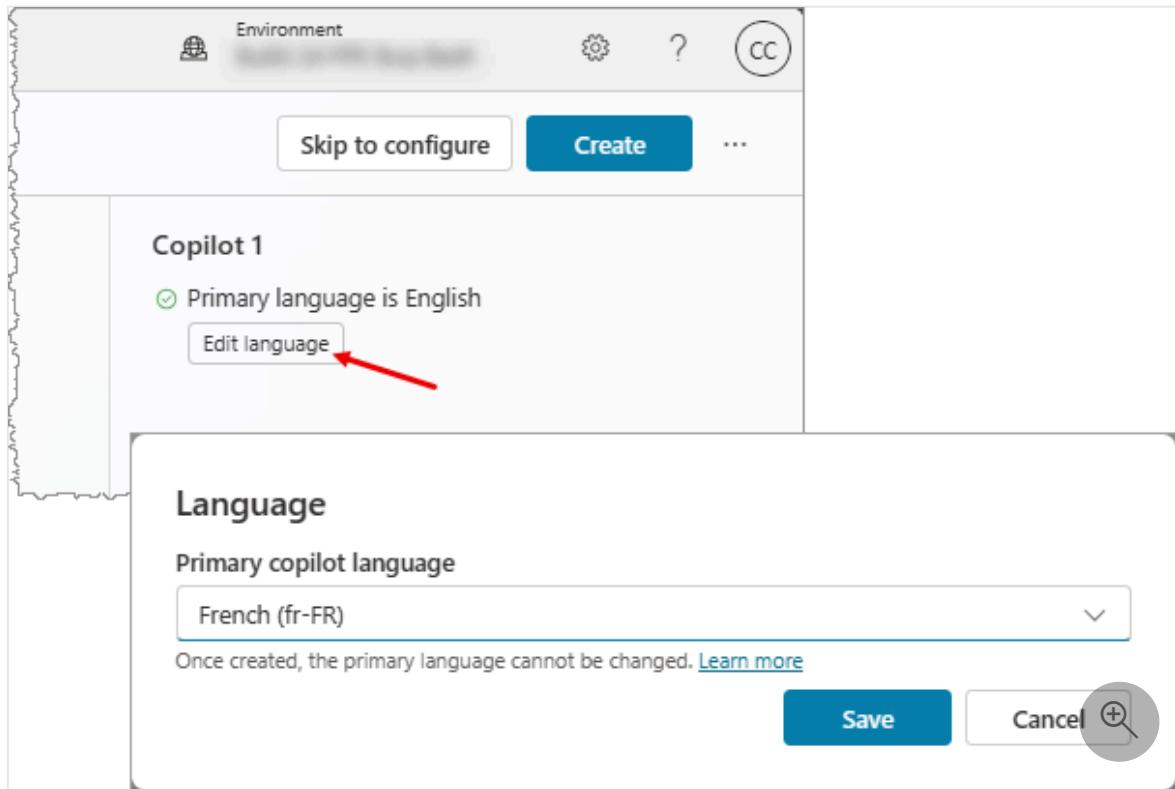
Some articles and screenshots may refer to Power Virtual Agents while we update documentation and training content.

You can create chat-based copilots in Microsoft Copilot Studio in many languages. Your copilots can reach a broader audience and engage with more markets around the world.

Specify a language for your copilot

You specify the target language when you create a copilot in your [Copilot Studio canvas](#).

1. From the **Home** page, select **Create > New custom copilot**.
2. Select **Language**, then select a language from **Primary copilot language**, and then **Save**.



3. Select Create at the top of the page.

Default content in the target language

After your new copilot 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 copilot, author copilot trigger phrases, and create copilot messages in your target language.

You can test the copilot in the chat pane. In chat, you see how well the copilot 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 copilot 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: Copilot creator sees the Copilot Studio UI in the target language.
- End-user language: The language a user can type in chat to ask questions. Exceptions include copilots with generative AI features that aren't yet available in some languages.
- Generative answers: A Copilot Studio feature that autogenerates chat responses.
- Generative actions: A Copilot Studio feature that uses plugins in a conversation.

ⓘ Note

Generative Actions is currently only in English.

[+] Expand table

Language	Authoring canvas	End-user language	Generative answers	Generative actions
Chinese (Simplified)	✓	✓	✓	Preview
Chinese (Traditional)	✓	✓		Preview
Czech	✓	✓	✓	Preview
Danish	✓	✓	✓	Preview

Language	Authoring canvas	End-user language	Generative answers	Generative actions
Dutch	✓	✓	✓	Preview
English	✓	✓	✓	Preview
Finnish	✓	✓	✓	Preview
French	✓	✓	✓	Preview
German	✓	✓	✓	Preview
Greek	✓	✓	✓	Preview
Hindi	✓	✓	✓	Preview
Indonesian	✓	✓	✓	Preview
Italian	✓	✓	✓	Preview
Japanese	✓	✓	✓	Preview
Korean	✓	✓	✓	Preview
Norwegian	✓	✓	✓	Preview
Polish	✓	✓	✓	Preview
Portuguese (Brazil)	✓	✓	✓	Preview
Russian	✓	✓	✓	Preview
Spanish	✓	✓	✓	Preview
Swedish	✓	✓	✓	Preview
Thai	✓	✓	✓	Preview
Turkish	✓	✓	✓	Preview

Related content

- [Regional settings and data locations for organizations using Copilot Studio](#)
- [Configure and create multilingual copilots](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Configure and create multilingual copilots

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.

Multilingual copilots are copilots that can communicate with customers in different languages while keeping all the content in a single copilot. In many cases, they can automatically detect the desired language based on the copilot 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 our [Supported Languages article](#).

ⓘ Note

Multilingual copilots are not supported for the following scenarios:

- Microsoft Copilot Studio classic bots
- Voice bots with Dynamics Omnichannel

Configuring a multilingual copilot

When a copilot is initially created, it's created with a primary language. (See the [Supported Languages article](#) for more details on how to select the copilot's primary language).

To enable a multilingual copilot:

1. With a copilot open, select **Settings** at the top of the page, and then select **Languages**.

2. Select Add Languages on the Settings page.

The screenshot shows the Copilot Studio Settings page. On the left sidebar, the 'Languages' option is highlighted with a red box. In the main content area, under the 'Languages' heading, it says 'Add languages and customize your copilot's voice per language.' Below this is a 'Primary language' section showing 'English (United States) (en-US)' with a green checkmark indicating 'Voice features supported'. A red arrow points to a button labeled '+ Add Languages' at the bottom of the 'Add languages' section. The message 'No additional languages' is displayed below the button.

3. Choose the languages you want to add to the copilot in the **Add Languages** pane and select **Add Languages** at the bottom.

4. Confirm the list of languages is correct on the **Languages** page.

The screenshot shows the Copilot Studio Languages page. The 'Languages' section header is visible. A green success message at the top states 'Languages were added successfully.' Below it is the 'Primary language' section with 'English (United States) (en-US)' selected. The 'Add languages' section contains a note about voice support being temporarily unavailable for secondary languages. At the bottom, there is a table listing added languages:

Secondary language ↑	Voice support	Localization	Last modified
English (United Kingdom) (en-GB)	Supported	Upload	-
Italian (Italy) (it-IT)	Supported	Upload	-

A red box highlights the table containing the two added languages.

ⓘ Note

You can add as many secondary languages as desired to the copilot, as long as they are in the list of supported languages.

Managing localization in a multilingual copilot

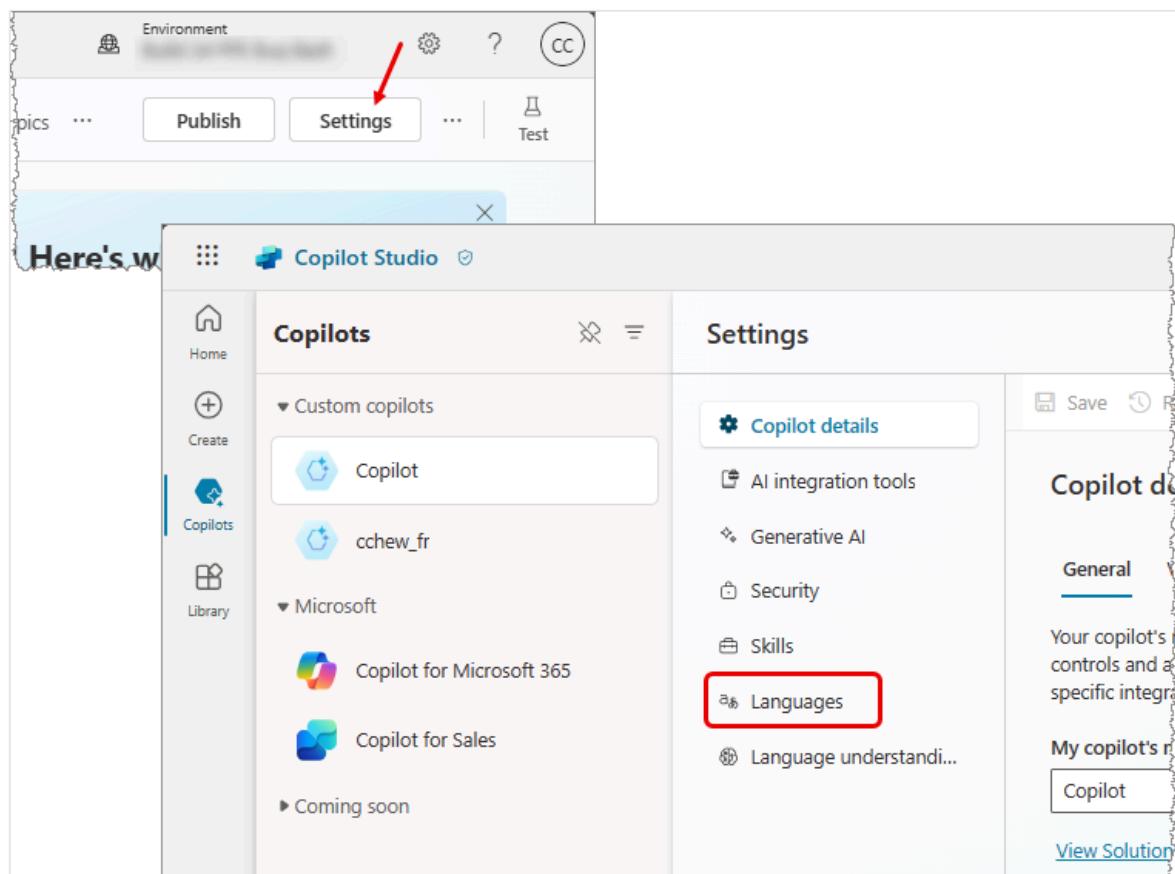
ⓘ Note

All copilot topic and content editing must be done in the copilot's primary language. To edit the copilot's secondary language to localize the strings, you must use the steps in this section.

Once the copilot strings are localized and uploaded to the copilot as described here, you can see the localized content in the authoring canvas by [switching the language in the test copilot](#).

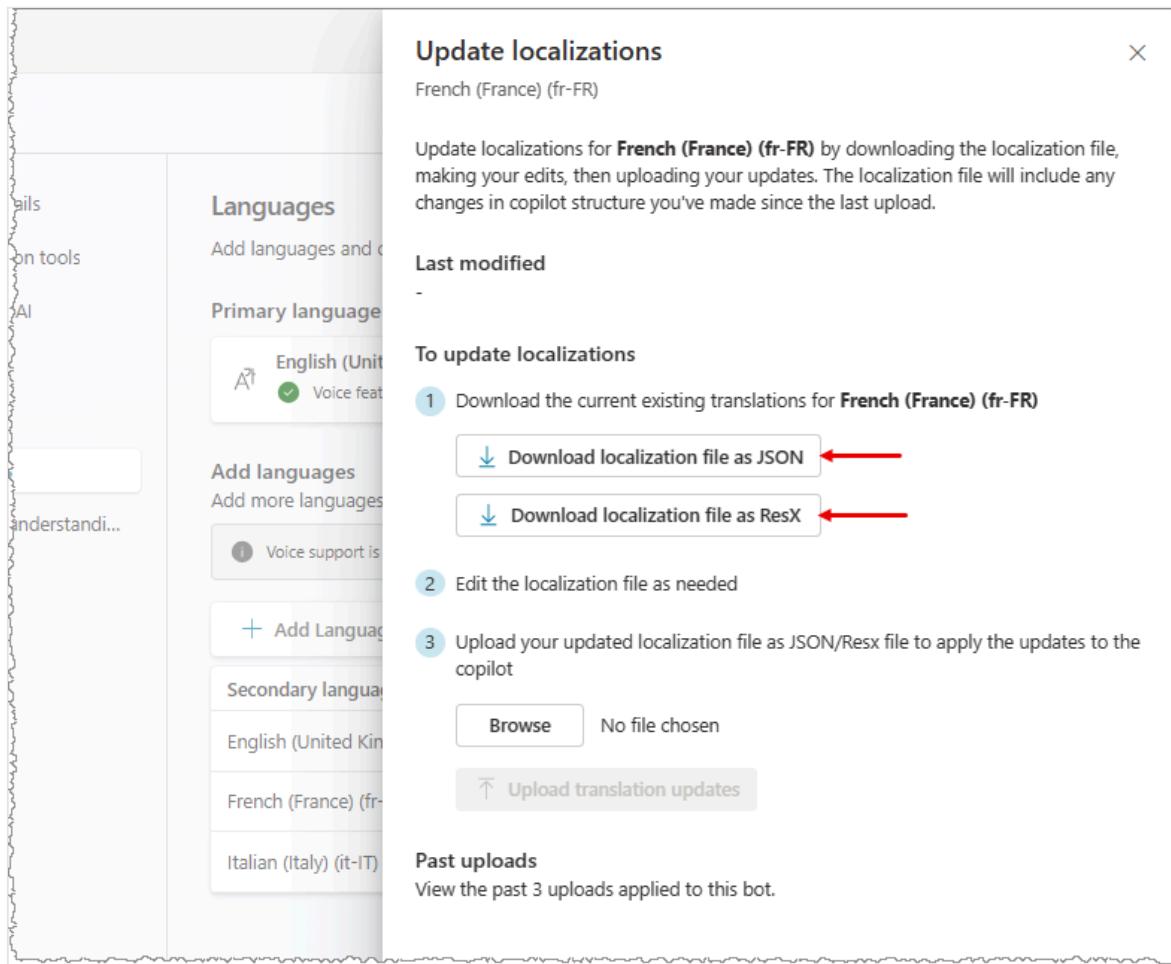
To add the localization strings to your multilingual copilot:

1. With a copilot open, select **Settings** at the top of the page and then on the **Settings** page, 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.



➊ Note

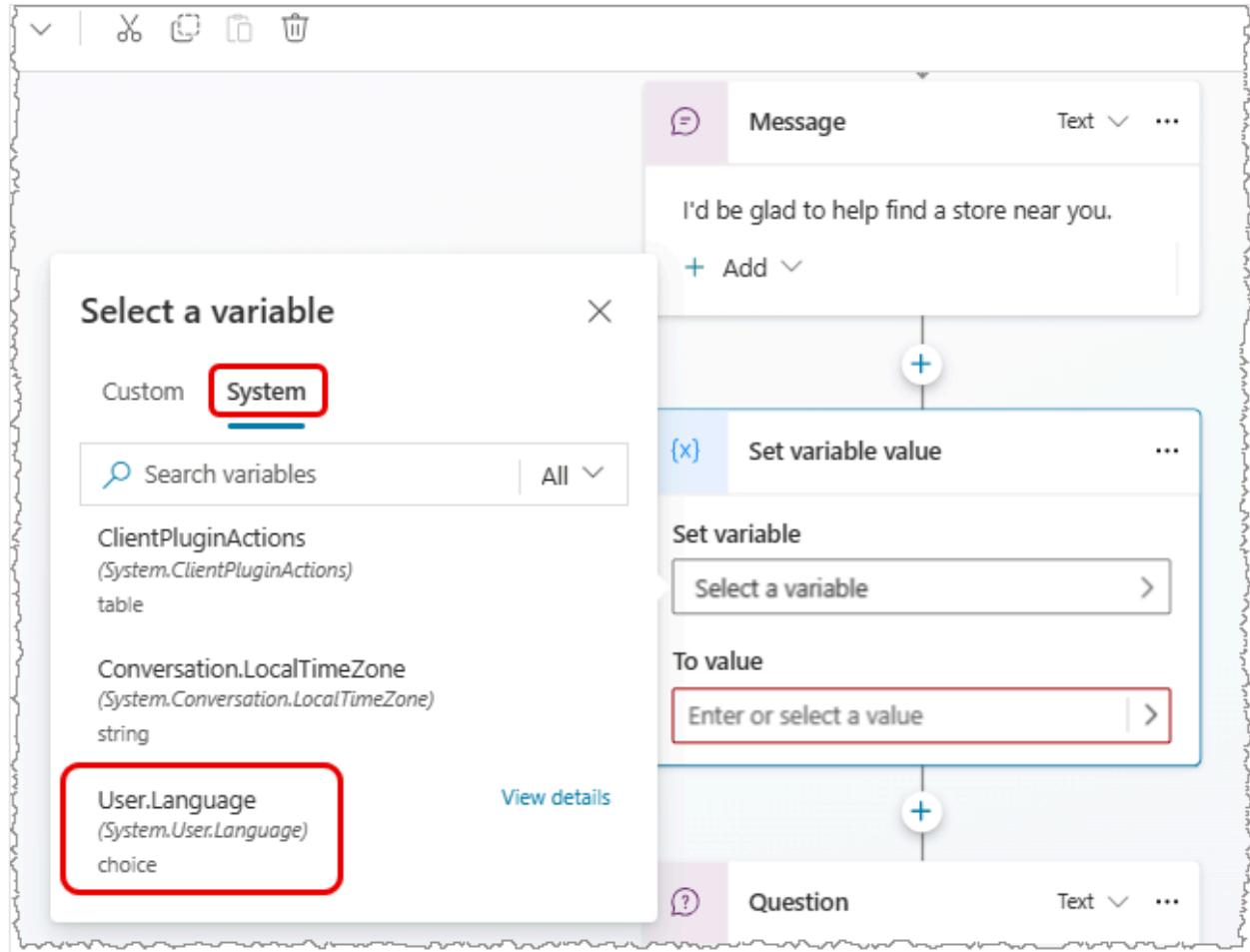
The downloaded file will contain the latest localization content for the copilot. If you need to download previous versions of the localization file, [open the copilot's solution in Power Apps to download previous versions](#).

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.

Authoring considerations for multilingual copilots

When authoring, you can configure the copilot to change the current spoken language in the middle of a conversation. That logic can reside in any topic in the copilot.

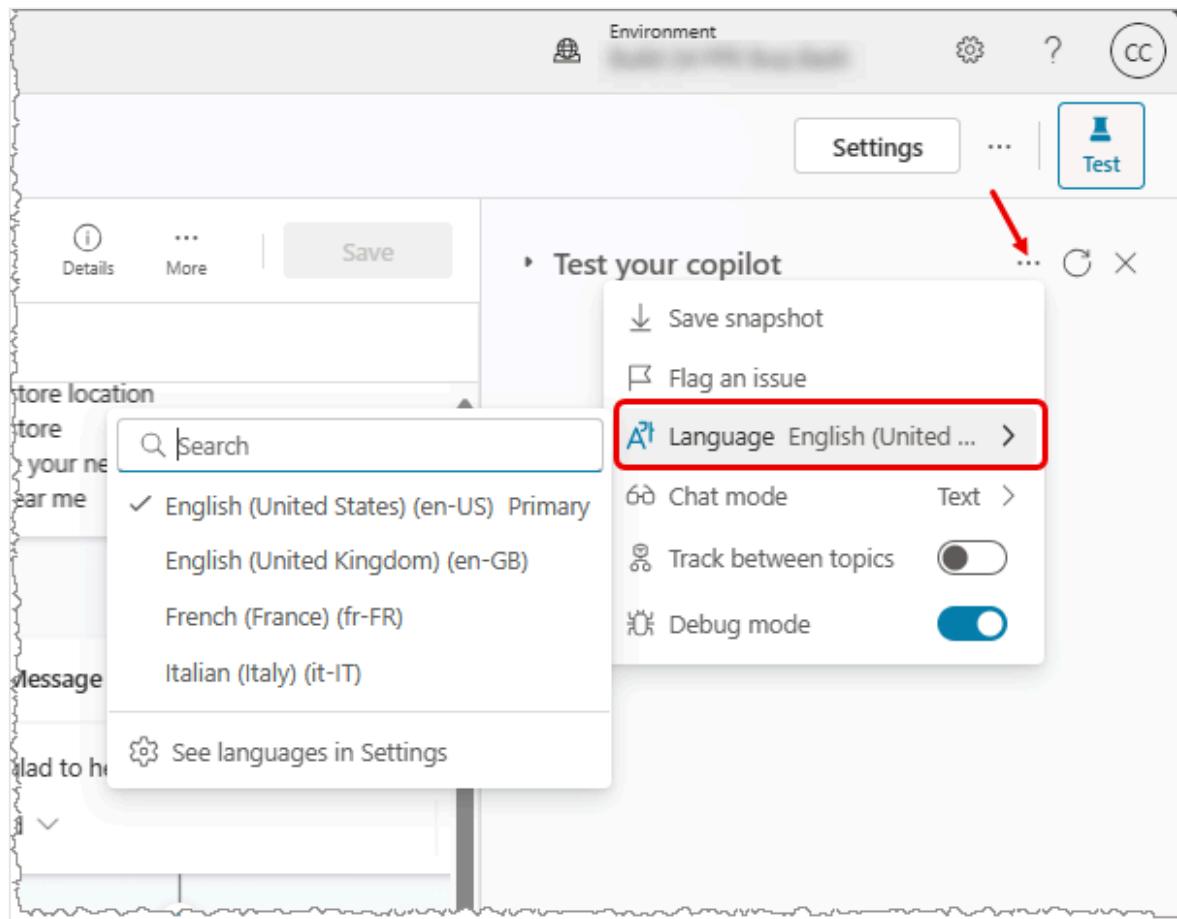
To change the copilot's current language, you can set the `User.Language` variable value to one of your copilot's secondary languages. This selection changes the language spoken by your copilot immediately.



Testing a multilingual copilot

To test a multilingual copilot:

1. Open the **Test your copilot** pane.
2. Open the menu at the top of the test copilot, and select one of the secondary languages.



3. The test copilot will reload itself, this time using the selected secondary language.
The authoring canvas shows the secondary language.
4. Enter a message in the secondary language to test the copilot.

You can also set your browser language to the secondary language supported by your copilot, and open the [Demo copilot website](#). The demo website opens in the secondary language, and the copilot chats using the secondary language strings.

Multilingual copilot behavior for languages that are not configured

If a copilot user has their browser configured for a language that is not configured in the copilot through the steps above, the copilot will always fallback to its primary language (the language the copilot was created in originally). The copilot's primary language cannot be changed after copilot creation.

Multilingual copilot behavior for missing translations

If the copilot author makes a change to the copilot in the primary language and doesn't [upload the translations](#), the copilot will show the untranslated changes in the copilot's primary language. This leads to a confusing dual-language experience. Always make sure the translations are up-to-date after making changes to the copilot in the primary language.

Related topics

- [Supported Languages](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Understand error codes

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.

When a copilot 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 copilot should give this error code to their administrator.

As a copilot maker, if a problem occurs when you are using the test pane to [test your copilot](#), 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 copilot](#).

Error list

Web app

ⓘ Note

The term *dialog* used in some error messages refers to a *topic*.

[+] [Expand table](#)

Error code	Description
ContentError	There's an error in the topic content.
DataLossPreventionViolation	There was a data loss prevention violation.
FlowActionException	An error occurred while executing a cloud flow .
FlowActionBadRequest	A request made to a cloud flow was malformed.

Error code	Description
FlowActionTimedOut	A cloud flow took more than 100 seconds to run and timed out.
InvalidContent	Invalid content was added to the code editor.
InfiniteLoopInBotContent	A node was executed too many times.
LatestPublishedVersionNotFound	Unable to retrieve the published version of the copilot.
RedirectToDisabledDialog	A topic is redirecting to a disabled topic.
RedirectToNonExistentDialog	A topic is redirecting to another topic that no longer exists.
SystemError	A system error occurred in Microsoft 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 copilot'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 copilot. 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 [Configure user authentication](#).
- One or more connectors that are used in the copilot aren't in the same data group. See [Microsoft Copilot Studio connectors](#).
- One or more connectors that are used in the copilot 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 copilot 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 copilot. 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 copilot, place these actions after the 'Return value(s) to Microsoft Copilot Studio' step in your cloud flow.

InvalidContent

Error message: "A total of {TotalComponents} component(s) exist in the copilot, 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 copilot."

Resolution: [Publish the copilot](#).

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](#).

Key concepts - Enhanced authoring in Copilot Studio

Article • 05/02/2024

ⓘ 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.

You can go beyond basic authoring in Copilot Studio by taking advantage of advanced features.

Web app

These features enhance and improve upon the [authoring experience](#) by letting you:

- Expand the types of information the copilot understands and can parse back.
- Use variables to retrieve information from one part of a topic and use it in another, to ensure a consistent experience for your copilot users.
- Utilize existing support content from website pages, such as FAQ pages, to quickly create a robust set of topics that can be used in a copilot.
- Allow users to sign in directly within a copilot, and then use that authentication to retrieve information from backend systems.
- Use advanced Power Automate flows to retrieve and send information to other databases.
- Add special nodes that take the context of the entire copilot conversation and share it with a human help support agent.

In this section

[+] [Expand table](#)

Topic	Description
Use entities and slot filling	Use prebuilt entities, and define custom ones, to expand the

Topic	Description
	types of information the copilot can parse.
Use variables	Retrieve information for consistent usage across topics.
Create topics from existing support content	Utilize your existing web content to quickly create copilots and topics.
Add end-user authentication	Allow users to sign in from within the copilot.
Add actions using Power Automate	Connect to existing and create new flows that let you automate and send and receive data.
Trigger handoff to a live agent	Send entire transcripts of a copilot conversation to a live agent when the copilot isn't able to resolve the issue.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use entities and slot filling in Copilot Studio copilots

Article • 06/12/2024

ⓘ 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.

A significant part of copilot 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 copilot 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||Msg=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, a copilot can smartly recognize the relevant information from a user input and save it for later use.

Prebuilt entities

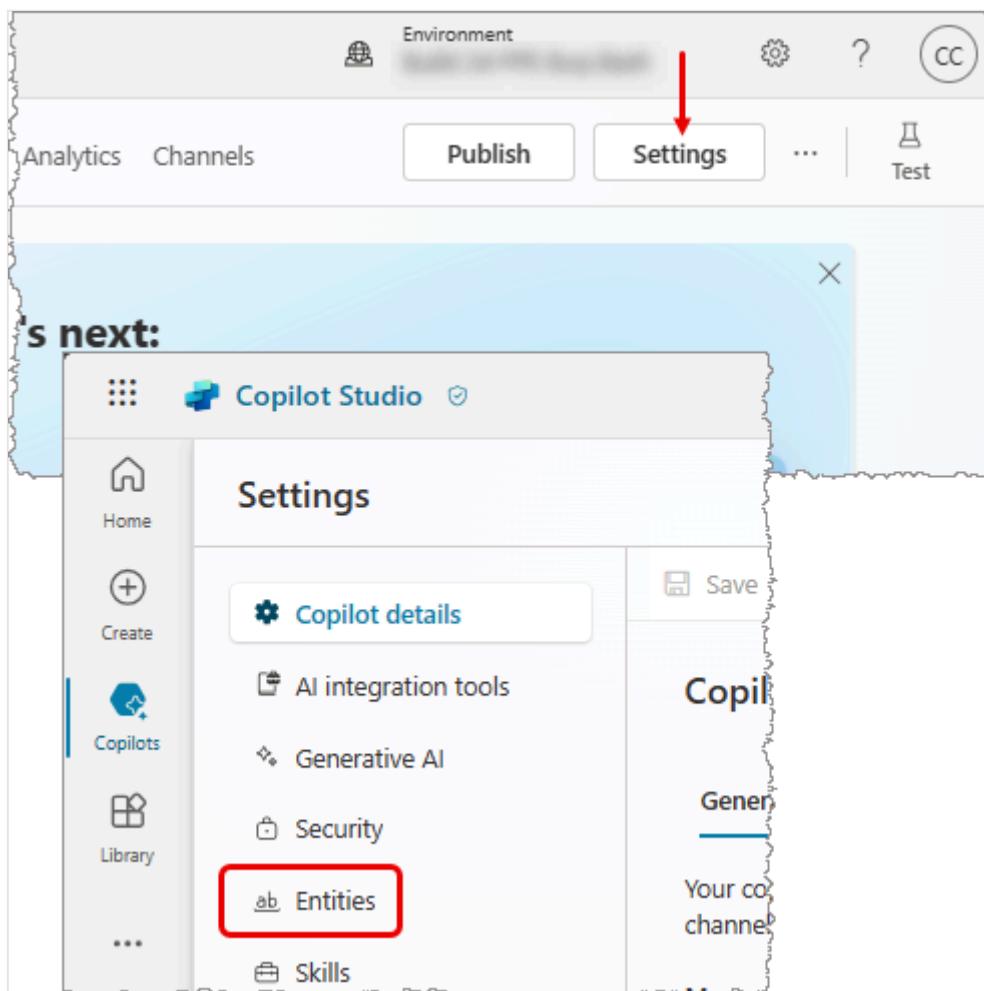
Entities in copilots let you store information in similar groups.

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, a copilot 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 copilot 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.

+ Add an entity		Search entities		
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 copilot knows that "1000 dollars" represents the *money* type of information. When the copilot 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 a copilot that serves a specific purpose, you might need to teach the copilot's language understanding model domain-specific knowledge.

For instance, let's say you want to build a copilot for an outdoor store. In this case, you need to teach the copilot 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 copilot the knowledge of all outdoor product categories.

1. Open your copilot 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**.

The screenshot shows the Copilot Studio interface with the 'Settings' tab selected. On the left sidebar, under the 'Copilots' section, the 'Entities' option is highlighted. A modal window titled '+ Add an entity' is open, with a red arrow pointing to the 'New entity' button. The main pane displays a table with two rows: 'Age' (Boolean) and 'City' (City names). To the right of the table, there is descriptive text: 'Age of a person' and 'City names'.

3. A new pane opens where you can choose the 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.

The screenshot shows the 'Create an entity' dialog. On the left, there is a decorative icon of a pen writing on a speech bubble. Below it, text says 'Make your copilot smarter with entities.' and 'An entity is a piece of information (person, place, thing, or concept) that the copilot might want to pick out from what your customer says.' A link 'Learn how to use entities' is also present. On the right, a 'Choose a method' section is shown with two options: 'Closed list' (selected) and 'Regular expression (Regex)'. The 'Closed list' section contains a sub-section for 'Example: List of sizes' with items: 'Extra small', 'Small', 'Medium', and 'Large'. A red box highlights this section. The 'Regular expression (Regex)' section contains a sub-section for 'Example: Article ID pattern that starts with "kb" and ends with 6 digits, such as 'kb123456''. Below the examples, there is sample regex code: '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 *

List items

Add

Description

These are activities that users can ask about.

Method

List

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

Modified by

14 minutes ago

Smart matching

on

The Smart matching option enables the bot's understanding of natural language. This can help match misspellings, grammar variations, and words with similar meanings.

If the bot isn't matching enough related words, enhance the bot'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.

Edit synonyms

X

Think about other words and expressions your customer might use to mean the following:

"hiking"

mountaineering

Add

trekking



Done

Cancel

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

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

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

Smart matching



on

The Smart matching option enables the copilot's understanding of natural language. This can help match misspellings, grammar variations, and words with similar meanings.

If the copilot isn't matching enough related words, enhance the copilot's understanding further by adding synonyms to your list items.

[Learn more about entities](#)

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 copilot.

Create an entity



Make your copilot smarter with entities.

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

[Learn how to use 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

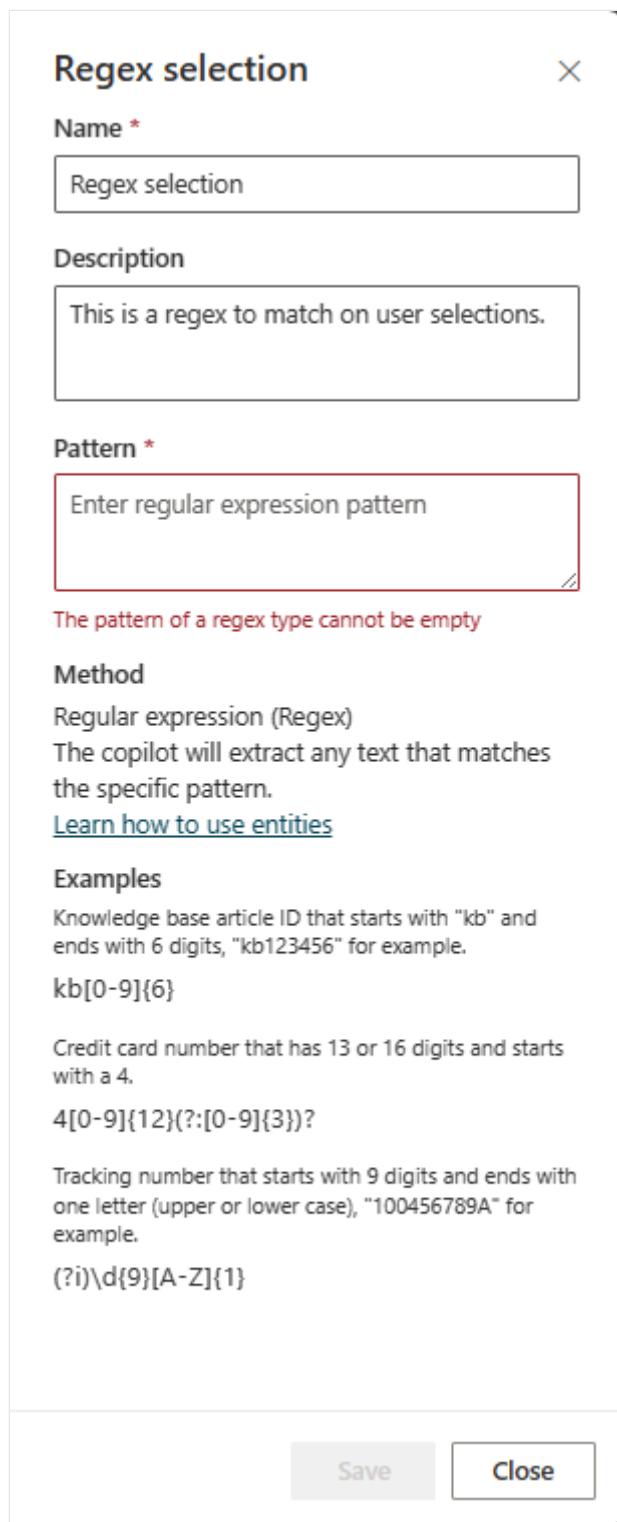
Regular expression (Regex)

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.



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.

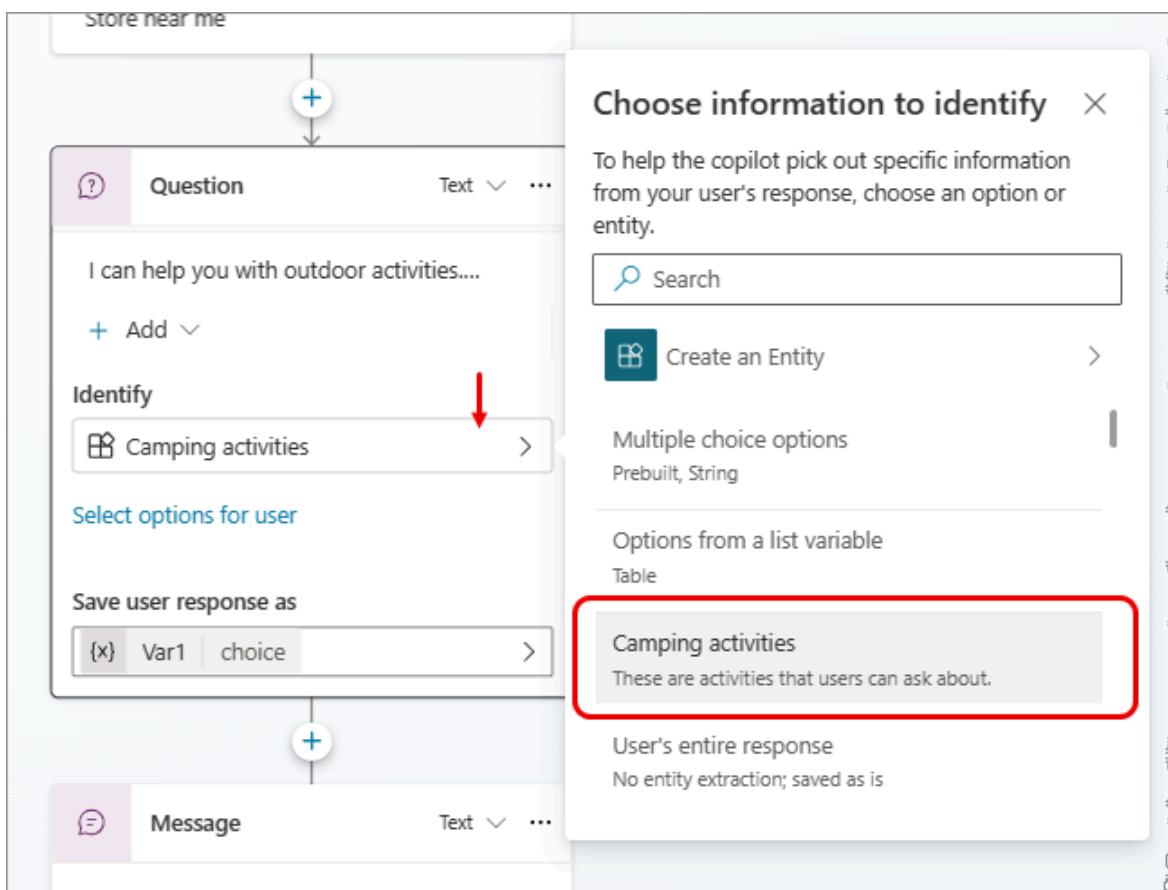
If you need to make the regular expression accommodate multiple patterns you can use the alternation operator `|` to concatenate regular expressions. For example, you could combine two regex patterns to look for both the old and new formats of a product ID code.

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

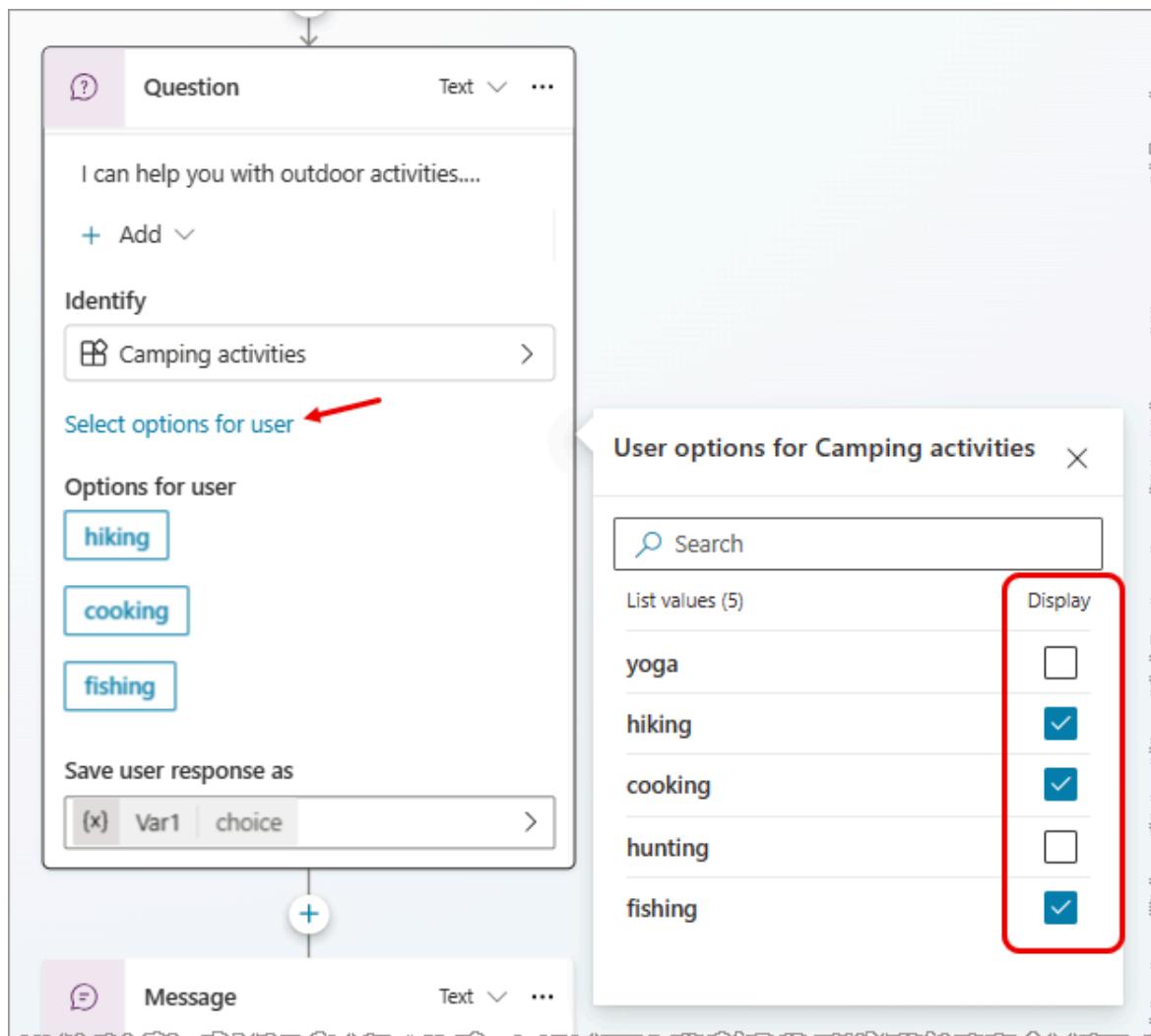
Now that you've done the work of giving the copilot the knowledge about outdoor gear by creating that product category entity and a few other custom entities, you can start to use them when constructing a copilot conversation.

1. Go to the [Topics](#) page for the copilot 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 Copilot Studio interface. On the left, there's a flowchart node labeled "Store near me" with a "Question" node below it. The "Question" node has the text "I can help you with outdoor activities....". Below it is an "Identify" section with a dropdown menu containing "Camping activities". A red arrow points from this dropdown to a callout box on the right. The callout box is titled "Choose information to identify" and contains the following text: "To help the copilot pick out specific information from your user's response, choose an option or entity." It includes a search bar, a "Create an Entity" button, and a "Multiple choice options" section. At the bottom of the callout box, a section is highlighted with a red border, containing the text "Camping activities" and "These are activities that users can ask about." Below the callout box, another section says "User's entire response" and "No entity extraction; saved as is".

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



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

Slot filling

Slot filling is a natural language understanding concept that means saving an extracted entity to an object. 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 copilot 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 copilot is running to the question node you just edited.

The screenshot shows a virtual assistant interface with two main sections: a configuration panel on the left and a test session on the right.

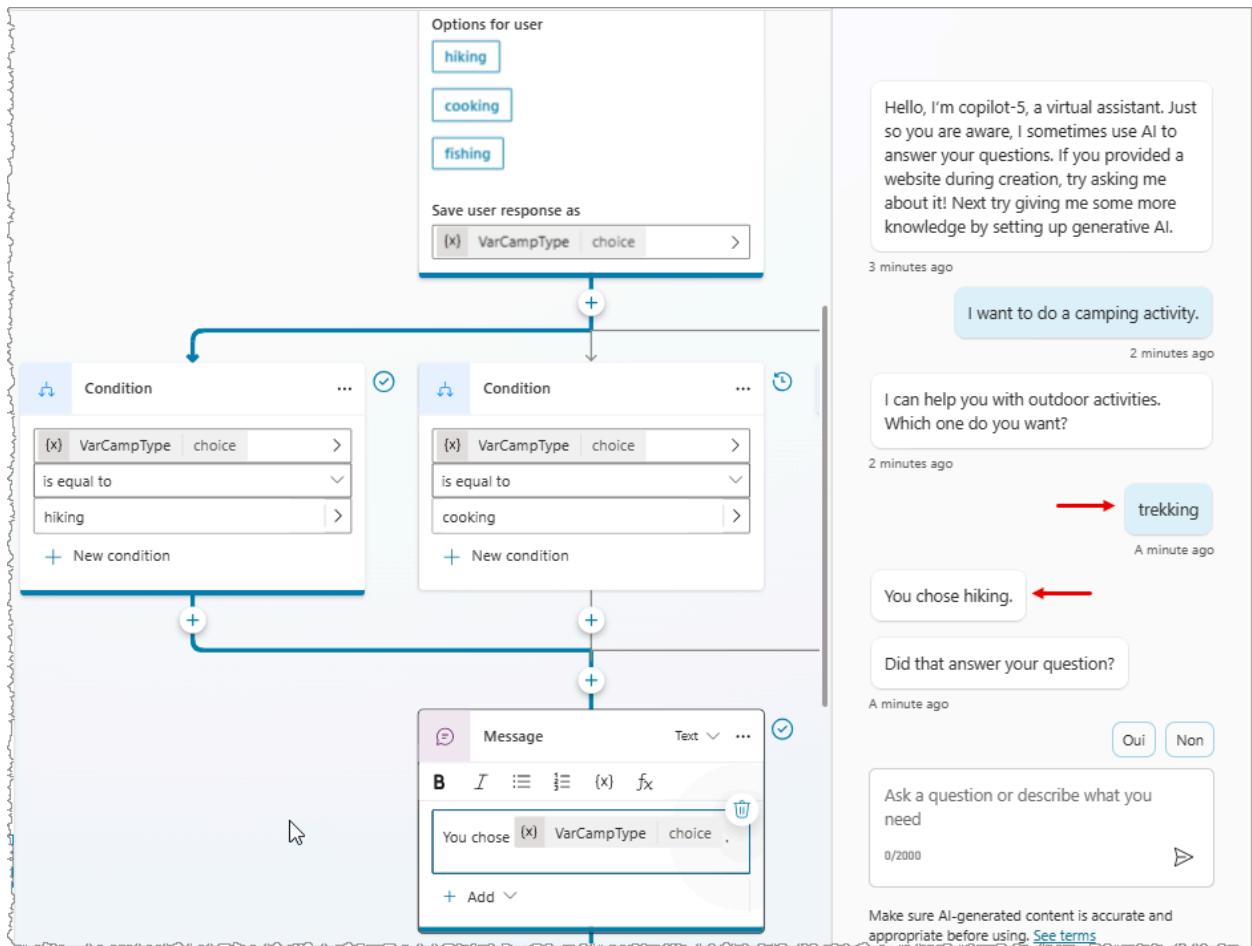
Configuration Panel (Left):

- Top Bar:** Copilot, Comments, Variables, Topic checker, Details, More, Save.
- Phrases Section:** Contains a list of predefined phrases:
 - I want to do an outdoor activity.
 - I'm bored.
 - I want something to do.
 - What activities can I do outdoors?
 - What can I do when I'm camping?
- Question Section:**
 - Question:** I can help you with outdoor activities. Which...
 - Add:** + Add ▾
 - Identify:** Camping activities >
 - Select options for user:** Options for user
 - Buttons:** hiking, cooking, fishing
 - Save user response as:** (x) Var1 choice >

Test Session (Right):

- Title:** Test your copilot
- Message from AI:** Hello, I'm copilot-5, 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.
- Timestamp:** 2 minutes ago
- User Message:** I want something to do.
- AI Response:** Just now
- Message from AI:** I can help you with outdoor activities. Which one do you want?
- Timestamp:** Just now
- Buttons:** hiking, cooking, fishing
- Text Input:** Ask a question or describe what you need
0/2000 >
- Text at Bottom:** 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".

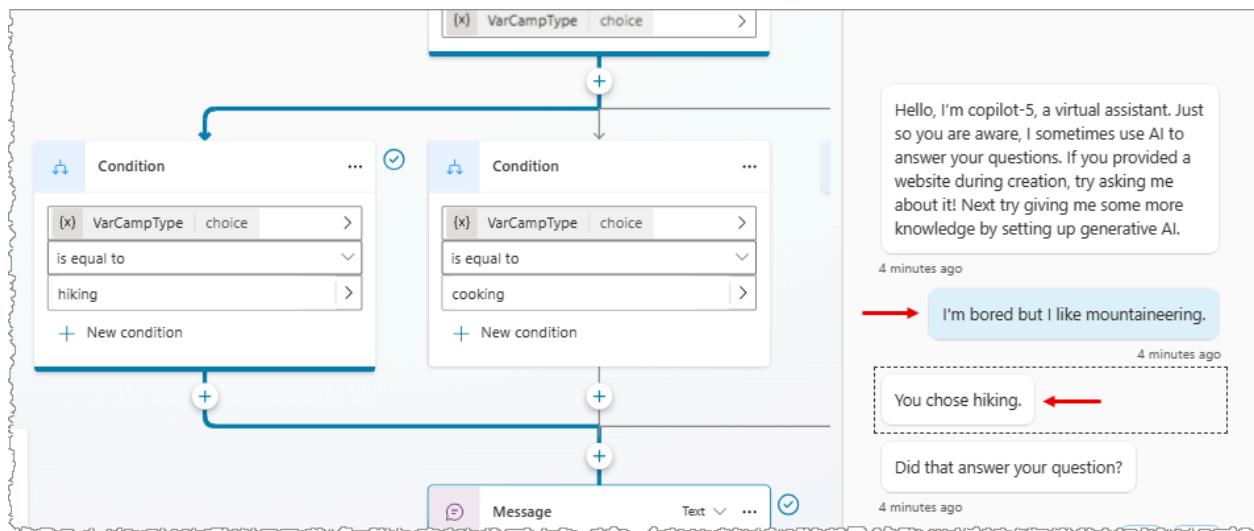
Essentially, slot filling has happened by inserting the extracted entity "Hiking" into the variable `VarCampType`.

The screenshot shows the 'Variables' window in the Microsoft Bot Framework. It displays a list of variables under 'Topic (1)': 'VarCampType choice' is set to 'hiking'. Other sections like 'Global (0)' and 'Environment (0)' are also listed. A red arrow points to the 'hiking' value in the 'VarCampType choice' row.

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 copilot 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 copilot fills in both the entity for choosing an outdoor activity, and for the type of activity.



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

The copilot 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 copilot only the product category, the user can say "I want to buy a pair of hiking boots under \$100". In this example, the copilot 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.

The screenshot shows the Microsoft Bot Framework designer interface. On the left, a logic flow diagram illustrates proactive slot filling. It starts with a 'Question' node asking 'How much time do you have for this activity?'. A condition node follows, checking if 'VarCampType' is equal to 'hiking'. If true, it leads to another condition node checking if 'VarCampType' is equal to 'cooking'. If both conditions are met, it triggers a proactive message to the user. On the right, a 'Test your copilot' window shows a conversation where the bot asks for time and price range, and then sends a proactive message about the user's chosen activity and budget.

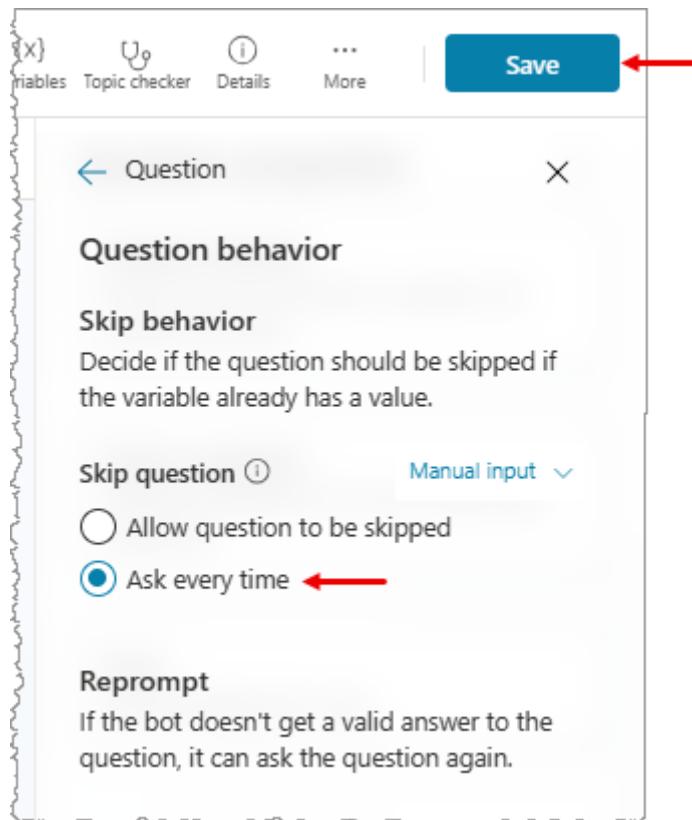
Proactive slot filling can be manually controlled at the node level. If you'd like to always prompt for the question within a specific node, regardless of whether the slot has been filled from previous user responses, you can disable the **Skip question** option for that question node. Perform the following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

The screenshot shows the 'Question properties' panel for a specific question node. The 'Properties' option is highlighted with a red arrow. Other options shown include 'Delete' and 'New comment'.

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 ↗

Use variables

Article • 05/07/2024

ⓘ 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.

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 copilot 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 three 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 copilot. They provide more contextual information about the conversation or the user. They're available in all topics.

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 that have 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"` will result in an error.

When you're testing a copilot, a variable may 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. That is, 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. That is, 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 that's associated with prebuilt entities.

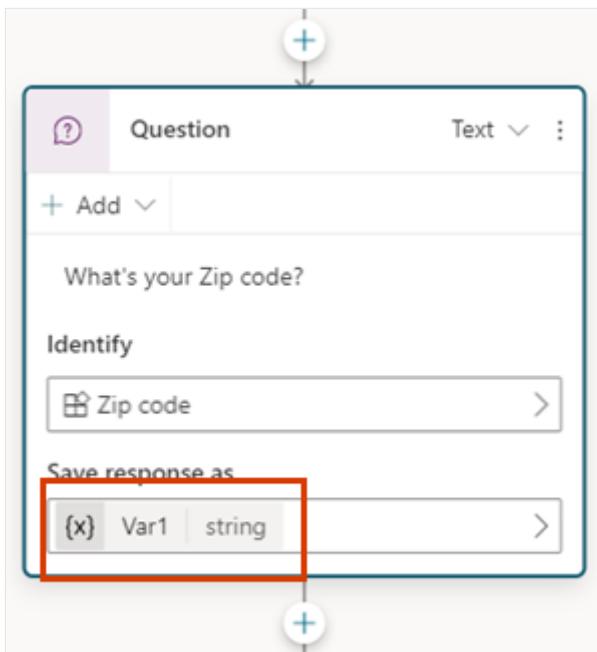
 Expand table

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.



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 copilot should listen for.

The screenshot shows the Microsoft Copilot interface with the 'Variables' tab selected. A red arrow points to the 'Identify' section in the left sidebar. The 'Multiple choice options' node is currently selected. A modal window titled 'Choose information to identify' is open, listing several entity types:

- Create an Entity: Names of organizations, associations, and corporations, extracted as a string
- Percentage: Percentages in numerical or text form, extracted as a number
- Person name: A person's partial or full name, extracted as a string
- Phone number: Phone numbers in the standard US format, extracted

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 copilot.

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'll use a **Question** node to save user input to a variable. There may 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 'Set variable value' step in a flow on the right. In the 'Select a variable' dialog, the 'Custom' tab is selected, and the search bar contains 'Var1'. A red box highlights the 'Create a new variable' button. Below it, 'Var1' is listed with type 'string'. The flow on the right shows a 'Save user response as' step where '{x} Var1 string' is selected. This leads to a 'Set variable value' step, which has a 'Set variable' section showing '{x} Var1' and a 'To value' section with a red box around 'Enter or select a value' and the error message 'Variable selection is missing'.

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

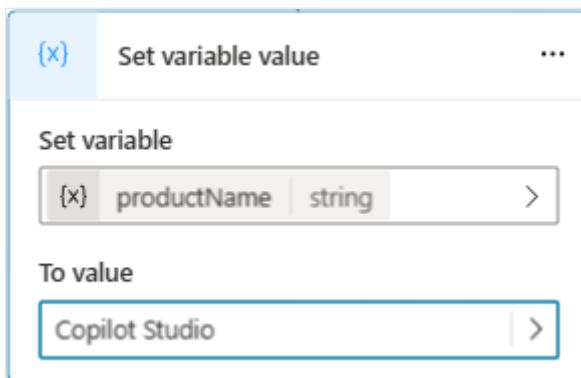
The screenshot shows the 'Set variable value' step in a flow. It has a 'Set variable' section with '{x} Var3' selected and a 'To value' section with a red box around 'Enter or select a value' and the error message 'Variable selection is missing'.

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.

Use literal values

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



The screenshot shows the configuration of a 'Redirect' node. It has an input 'user Name' of type boolean set to 'John Doe'. Below the node, there is a 'Greeting' section with a message icon and a link to 'View topic'.

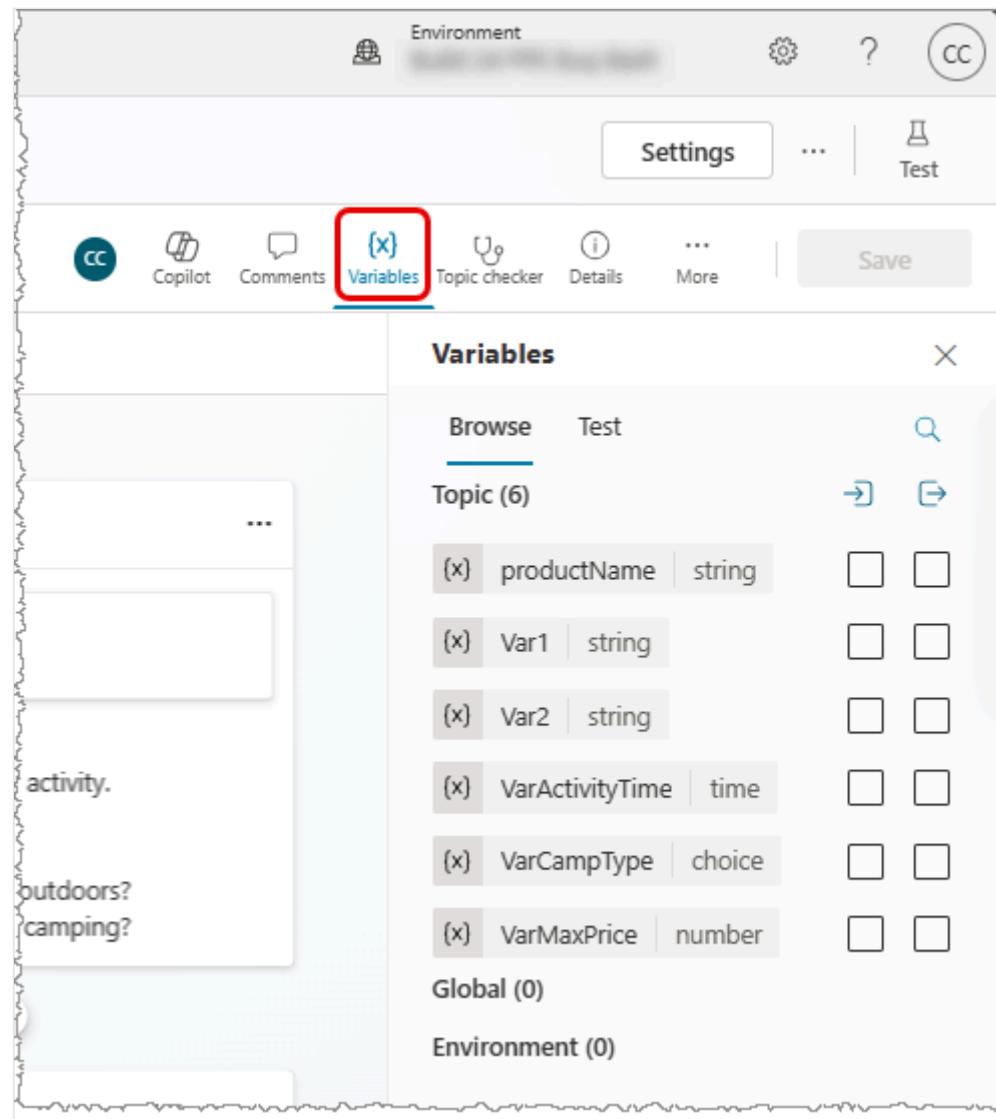
The node attempts to interpret literal values as a string, a number, or a boolean type. For example, `123` is interpreted as a number. If you want it to be interpreted as a string value instead, you can wrap the value in double quotes, like this: `"123"`.

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 Microsoft Copilot interface. On the left, a question card is displayed with the title "Question" and the text "Text". Below the card, there is a section titled "Identify" containing a button labeled "Camping activities". Under "Select options for user", there are three buttons: "hiking", "cooking", and "fishing". Below these, a section titled "Save user response as" contains a button labeled "(x) VarCampType choice". A red arrow points from this button to the variable "VarCampType" in the "Variables" panel on the right.

Variables

Browse Test

Topic (6)

(x) productName | string

(x) Var1 | string

(x) Var2 | string

(x) VarActivityTime | time

(x) VarCampType | choice (x) VarCampType | choice

(x) VarMaxPrice | number

Global (0)

Environment (0)

System variables

Every copilot comes with built-in system variables that provide additional information about a conversation.

Select a variable X

Custom **System** Formula

Search variables All ▾

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*)

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, you'd need to refer to it as `System.User.DisplayName`.

□ [Expand table](#)

Name	Type	Hidden	Definition
Activity.Channel	choice	visible	The channel ID of the current conversation
Activity.ChannelData	any	hidden	An object that contains channel-specific content
Activity.ChannelId	string	hidden	The channel ID of the current conversation, as a string
Activity.From.Id	string	hidden	The channel-specific unique ID of the sender
Activity.From.Name	string	hidden	The channel-specific user-friendly name of the sender

Name	Type	Hidden	Definition
Activity.Name	string	visible	The name of the event
Activity.Text	string	visible	The most recent message sent by the user
Activity.Type	choice	visible	Type of activity
Activity.TypeId	string	hidden	Type of activity , as a string
Activity.Value	any	hidden	Open-ended value
Bot.Name	string	visible	The name of your copilot
Channel.DisplayName	string	hidden	The display name of the channel
Conversation.Id	string	visible	The unique ID of the current conversation
LastActivity.Id	string	visible	The ID of the previously sent activity
LastMessage.Id	string	visible	The ID of the previous message sent by the user
LastMessage.Text	string	visible	The previous message sent by the user
Recognizer.TriggerMessage.Id	string	visible	The ID of the user message that triggered the current topic
Recognizer.TriggerMessage.Text	string	visible	The user message that triggered the current topic
User.DisplayName	string	visible	The display name of the signed-in user

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.

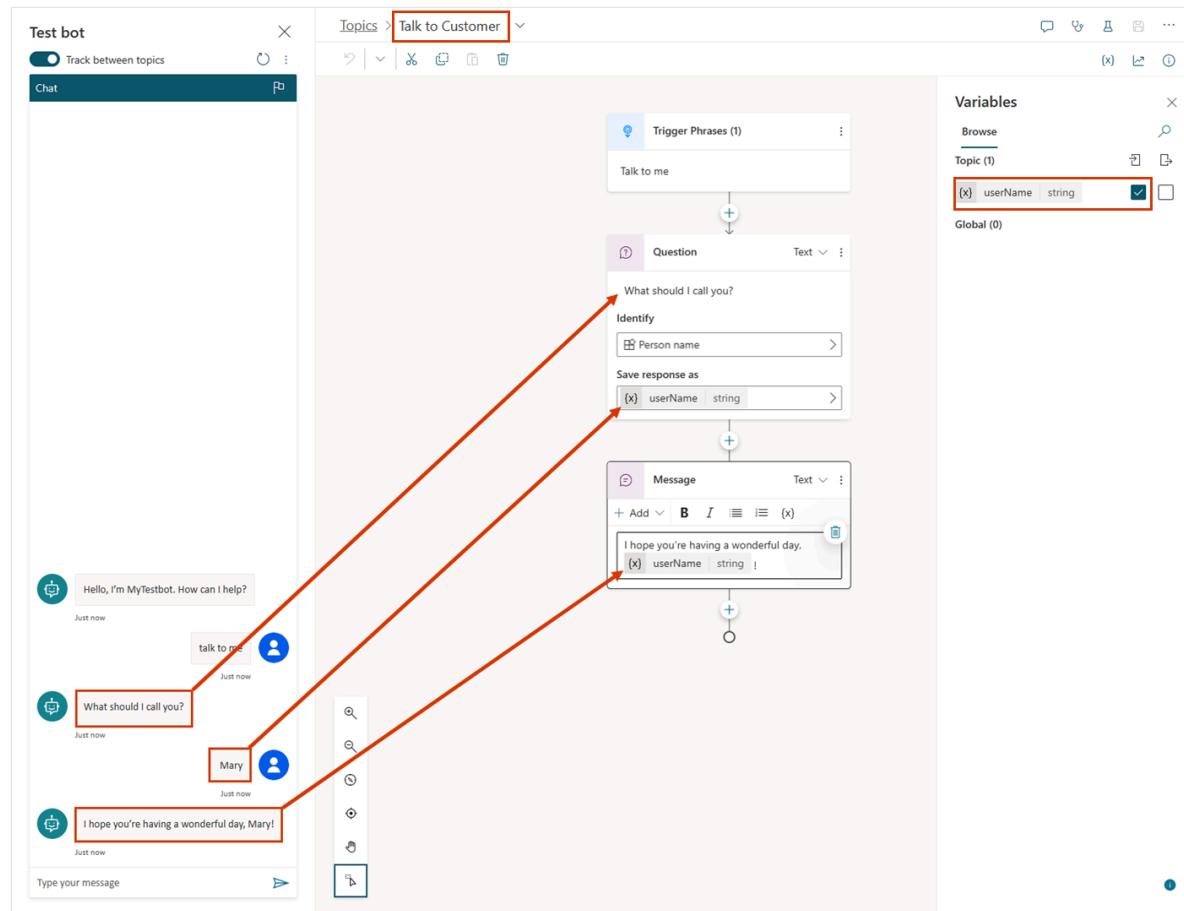
Receive values from other topics

When a topic defines a variable (for example, in a Question node), the copilot asks the user the question to fill in the variable's value. If the copilot 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 copilot is seamless.

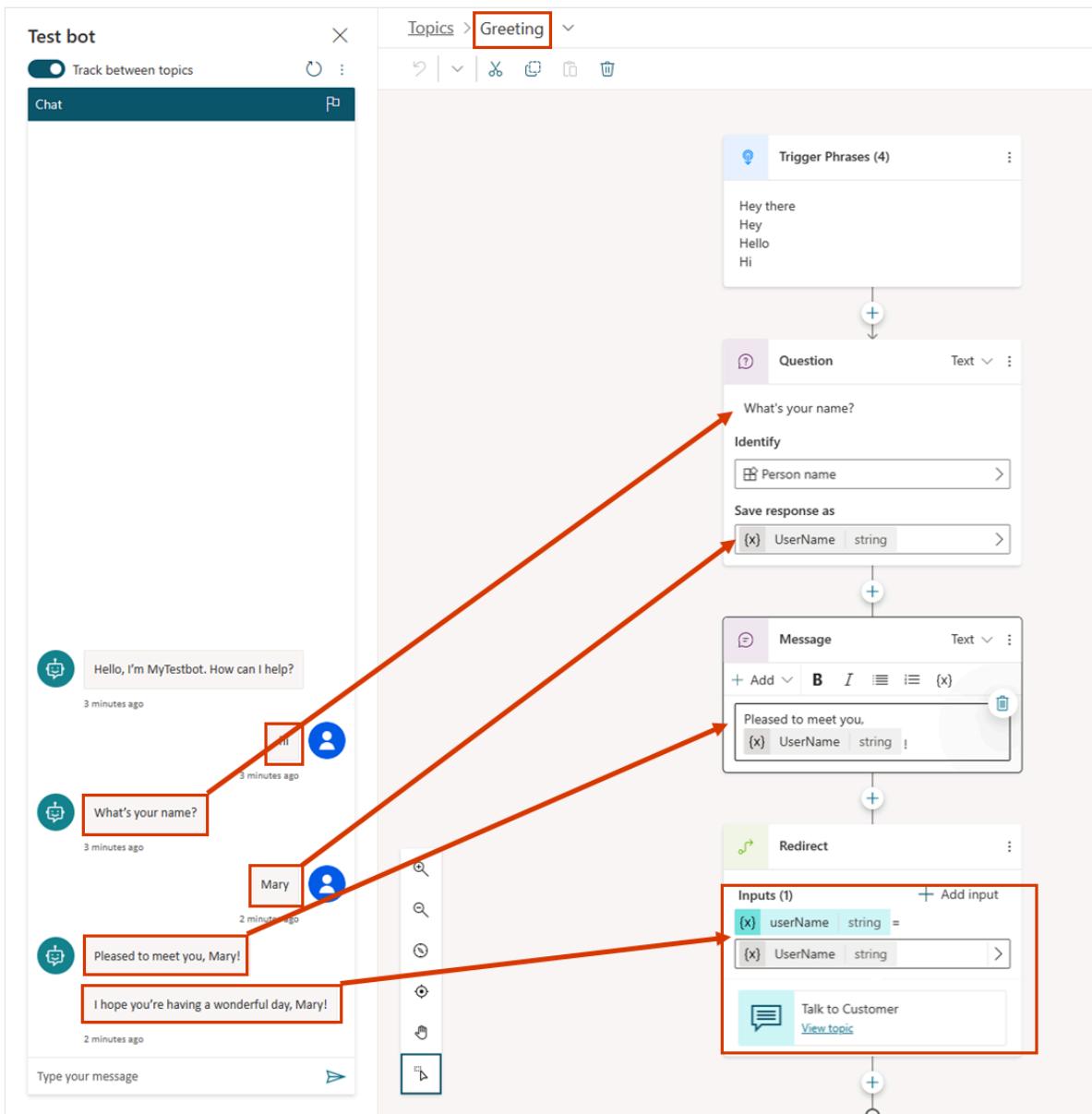
In this example, we'll use two topics, Greeting and Talk to Customer. Both topics ask for the customer's name. However, if the Greeting topic runs first, the Talk to Customer topic skips its question. Instead, it uses the value of the variable that's passed from the Greeting topic.

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



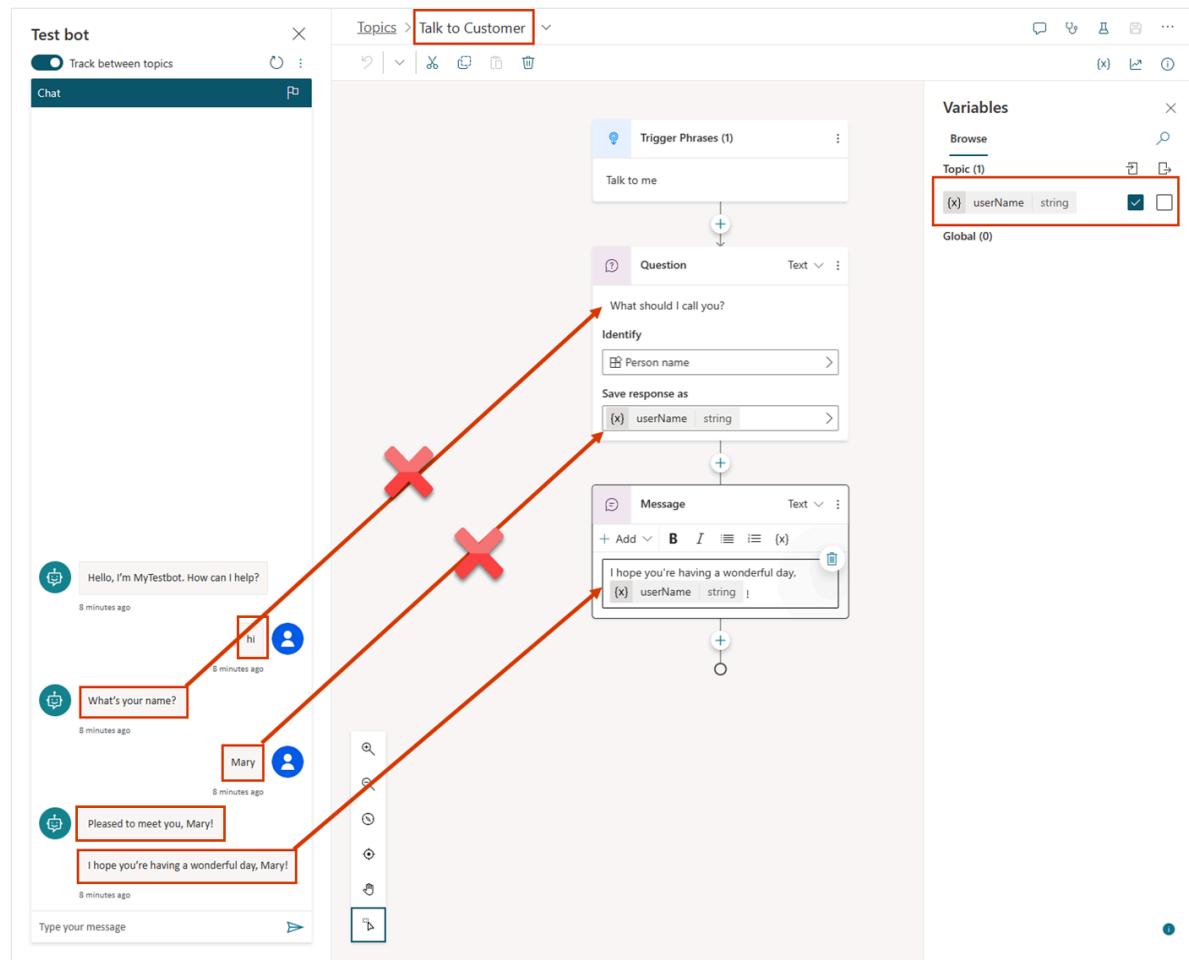
As shown in the Test copilot pane, if this topic is triggered first, it asks the user, "What should I call you?" It stores the value in a string variable called `userName`. The `userName` variable is also set to get its value from other topics. The topic concludes with the message, "I hope you're having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

Here's the flow of the Greeting topic:



As shown in the Test copilot 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, {userNmae}!" 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**.

The screenshot shows the Mindsay AI builder interface. On the left, a flowchart node is connected to a Question node. The Question node has a 'Trigger Phrases (1)' input with 'Talk to me' and an output pointing to a 'Save response as' field. The 'Save response as' field contains '{x} **userName** string'. On the right, the 'Variable properties' panel is open for the variable 'userName'. It shows the variable name, type (string), reference to the Question node, and usage settings. The 'Receive values from other topics' checkbox is checked.

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.

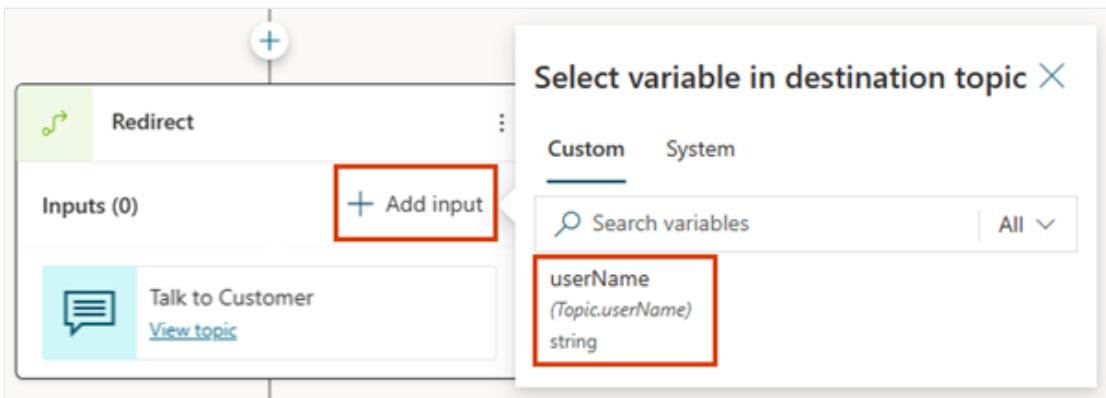
Set up the source topic

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.

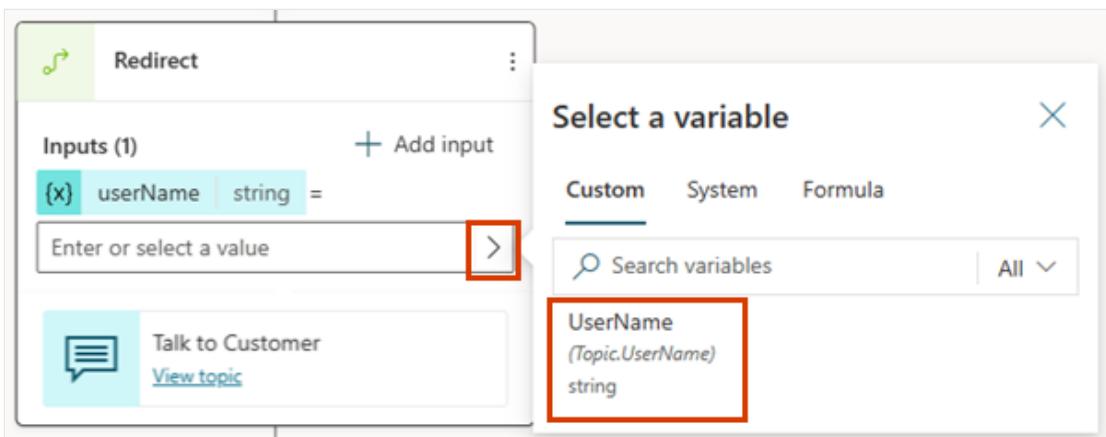
1. Go to the source topic.

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

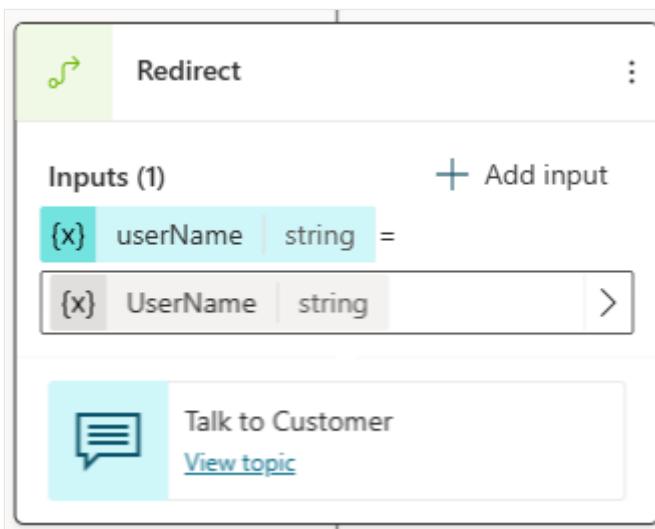
3. Select **+ Add input**, and then select the variable from the destination topic that you want to pass a value to.



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.

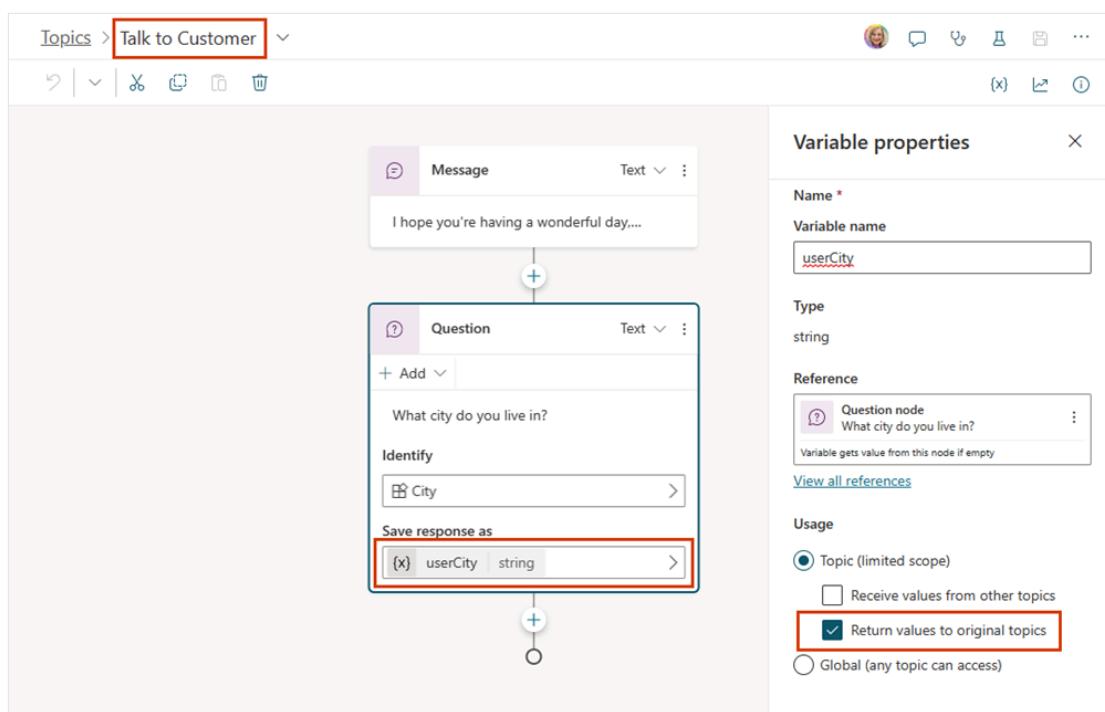
Information the copilot obtains is thus available across topics, reducing the need for [global variables](#).

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.

The screenshot shows the Microsoft Bot Framework Composer interface. On the left, a chat window for 'Test bot' shows a conversation where the bot asks for the user's name and greets them. On the right, the 'Topics > Greeting' editor shows the logic for this topic. A 'Redirect' node is highlighted with a red box. Red arrows point from the 'userCity' variable in the message to the 'userCity' input in the Redirect node, and from the 'userCity' output in the Redirect node to the 'userCity' variable in the subsequent message node. The 'userCity' variable is also highlighted with a red box in the message node.

```

graph TD
    Start((Start)) --> Question[Question]
    Question --> Identify[Identify]
    Identify --> Save[Save response as]
    Save --> Plus1((+))
    Plus1 --> Redirect[Redirect]
    Redirect --> Talk[Talk to Customer]
    Talk --> Plus2((+))
    Plus2 --> Message1[Message]
    Message1 --> Plus3((+))
    Plus3 --> Message2[Message]
    
```

3. Save the topic.

Related topics

- Reuse variables across topics
- Customize the look and feel of a copilot

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Reuse variables across topics

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.

Variables store your customers' responses to questions from your copilot. For example, you can save a customer's name in a variable called `UserName`. The copilot 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 copilot 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 copilot.

Web app

Copilot variables apply during a single user session. You specify which variables should be treated as copilot 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`.

The screenshot shows the Microsoft Copilot Studio interface. On the left, there's a 'Trigger Phrases (0)' section with a 'Trigger phrases...' button and a plus sign icon. Below it is a 'Question' node configuration. The 'Text' field contains the question 'What's your email address?'. Under 'Identify', there's a dropdown set to 'Email'. Under 'Save response as', a dropdown is set to '{x} Global. userEmail string', which is highlighted with a red box. A plus sign icon is below this. On the right, the 'Variable properties' panel is open. It shows the 'Name *' field as 'Global. userEmail' and the 'Type' as 'string'. In the 'Reference' section, a 'Question node' is listed with the question 'What's your email address?'. The 'Usage' section has two options: 'Topic (limited scope)' and 'Global (any topic can access)', with 'Global' selected and highlighted with a red box. There are also two unchecked checkboxes: 'Allow to carry between sessions' and '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

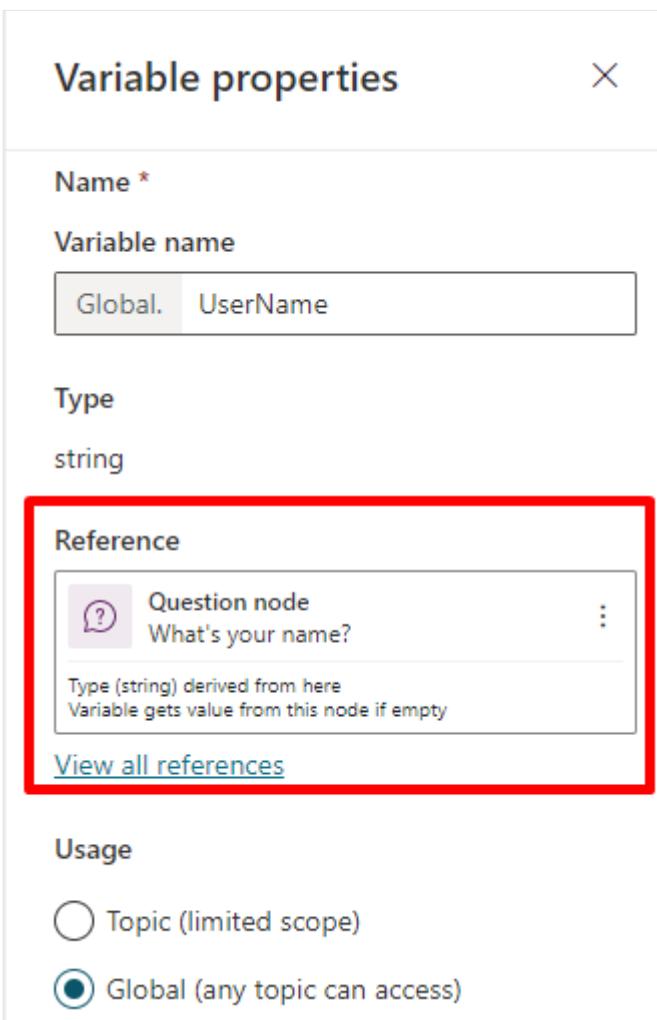
When you're composing a copilot message in a Message node or a Question node, select the {x} icon to view the variables that are available to the topic. Global variables appear in the **Custom** tab along with any topic variables. Variables are listed in alphabetical order.

The screenshot shows the Microsoft Copilot Studio interface. On the left, there's a 'Question' node configuration. The 'Text' field contains the question 'Is this the best email address to reach you?'. Below it is an 'Identify' section with a dropdown set to 'Multiple choice options'. On the right, a 'Select a variable' dialog is open. It has tabs for 'Custom' (which is selected and highlighted with a red box) and 'System'. Below the tabs is a search bar with 'Search variables' and a dropdown set to 'All'. A list of variables is shown, with 'UserEmail (Global.UserEmail) string' highlighted with a red box.

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 copilot, 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 copilot 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 copilot 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 copilot

If you're [embedding your copilot in a simple web page](#), you can append variables and their definitions to the copilot'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 copilot'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 copilot's URL is <https://web.powerva.microsoft.com/webchat/bots/12345>.
- To pass in the user's name when starting a copilot 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 copilot, 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 copilot, 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 copilot'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 topics

- [Use variables](#)
- [Customize the look and feel of the bot](#)

Manage topic inputs and outputs

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.

[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 copilot uses [generative AI to decide how to respond to users](#), it can automatically fill inputs, before running the topic, by using conversation context or generate questions to collect values from the user's response. This behavior is similar to how generative slot filling for actions works.

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 copilot uses generative AI to decide how to respond to users, more properties are available for input variables. Specifically **How will the copilot fill this input?** determines how the copilot fills this variable with a value before running the topic. By default, each input is set to **Dynamically fill with the best option**. The copilot 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 copilot 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 copilot captures the correct type of value for the input, you must also specify an entity type under **Identify as**.

The screenshot shows the 'Topic details' page in the Microsoft Copilot interface. The 'Input' tab is selected. Key visible elements include:

- Title:** A field containing 'Title' with a trash icon.
- Variable name:** A field containing 'Title' with a help icon.
- How will the copilot fill this input?**: A dropdown menu set to 'Dynamically fill with best option (default)'.
- Variable data type:** A dropdown menu set to 'String'.
- Display name:** A field containing 'Title'.
- Identify as:** A dropdown menu set to 'User's entire response'.
- Description:** A text area containing 'The title of the ticket to be created.'
- Additional settings:** A section with a link 'Additional settings' preceded by a right-pointing arrow.
- Create a new variable:** A button at the bottom left.

There are a number of **additional settings** that can be configured for inputs, that can control the copilot's behavior when asking for a value or validating a user's answer.

One of these properties, **Should prompt user**, is only available for topic inputs. It determines if the copilot 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 • 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.

When you configure [inputs for topics](#) or [inputs for actions](#), common settings allow you to determine how your copilot gathers input.

💡 Tip

The additional settings are always available for action inputs. However, for topic inputs, these settings are only available if your copilot is configured to use generative AI to determine how to respond to a user.

The following additional settings are available.

Copilot prompts user for input

By default, the copilot will use the name and description for the input to generate an appropriate question to prompt the user for a value. However, you can use the **Copilot prompts user for input** setting to override the question.

To override and provide your own question, tick **Customize**. You can then author a question, including adding references to variables or Power Fx formulas.

② Copilot prompts user for input

Configure how the copilot 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?

Copilot doesn't identify entity

Retry prompt 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](#).

- **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 question used to ask a user again for input, where the basic entity validation has failed, select **Customize**, and then enter the new question. e.g. If you were expecting a number, you might enter a question like
`What is the ID? This 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. e.g. If you received a number, but it was above 10, breaking the validation rule, you might enter a question like `What is the ID? This needs to be less than 10..`

No valid entity found

No valid entity found determines what happens when your copilot stops trying to get a valid response from the user, because the maximum retry count has been reached. You can escalate to a human agent 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. e.g. Your message might say `Looks like you're struggling. I am redirecting you to one of our team.`

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Make HTTP requests

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.

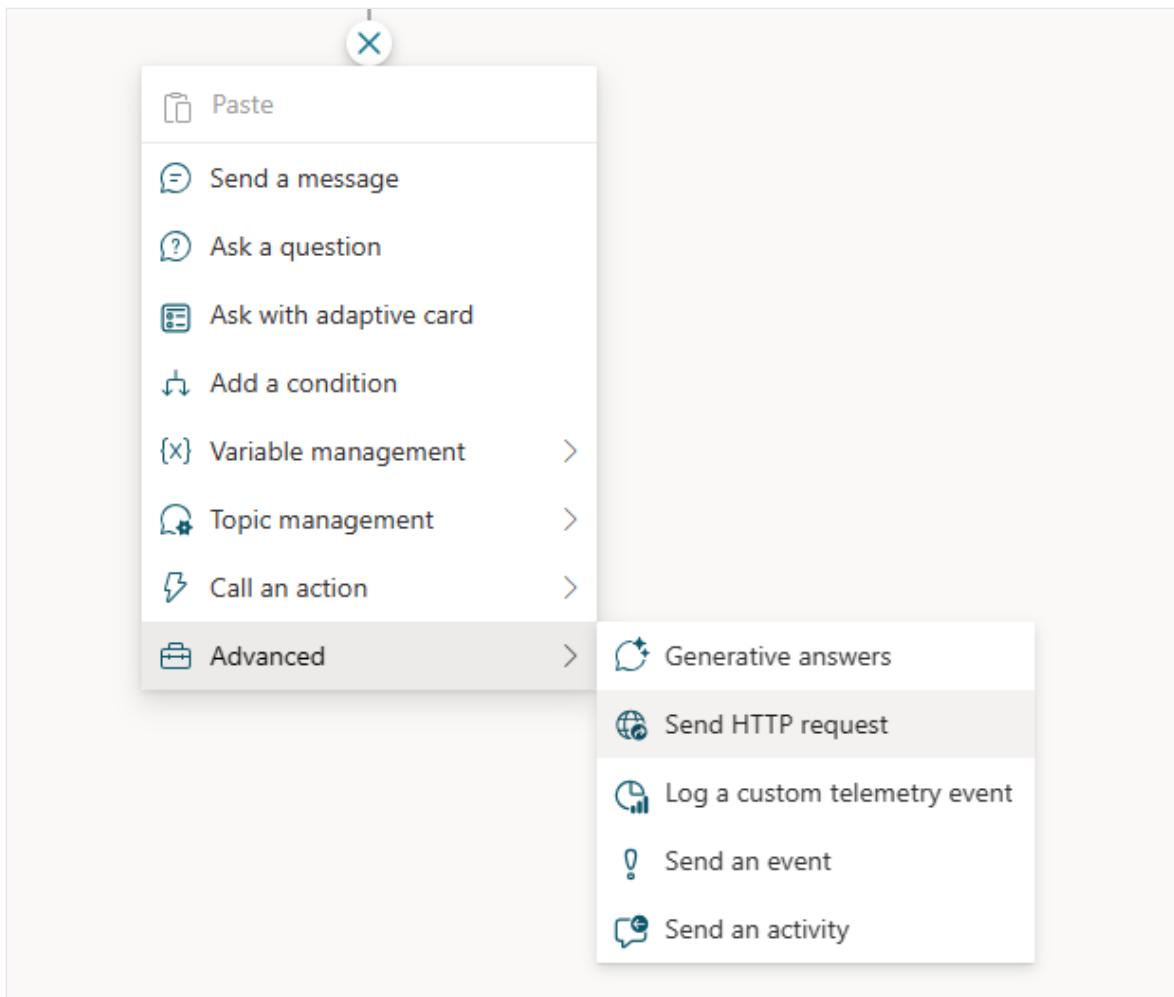
In Copilot Studio, you can extend a copilot's capabilities by calling external REST APIs using the **HTTP request** node. This can be helpful when you need to retrieve data from an external system or manipulate data on that system.

Prerequisites

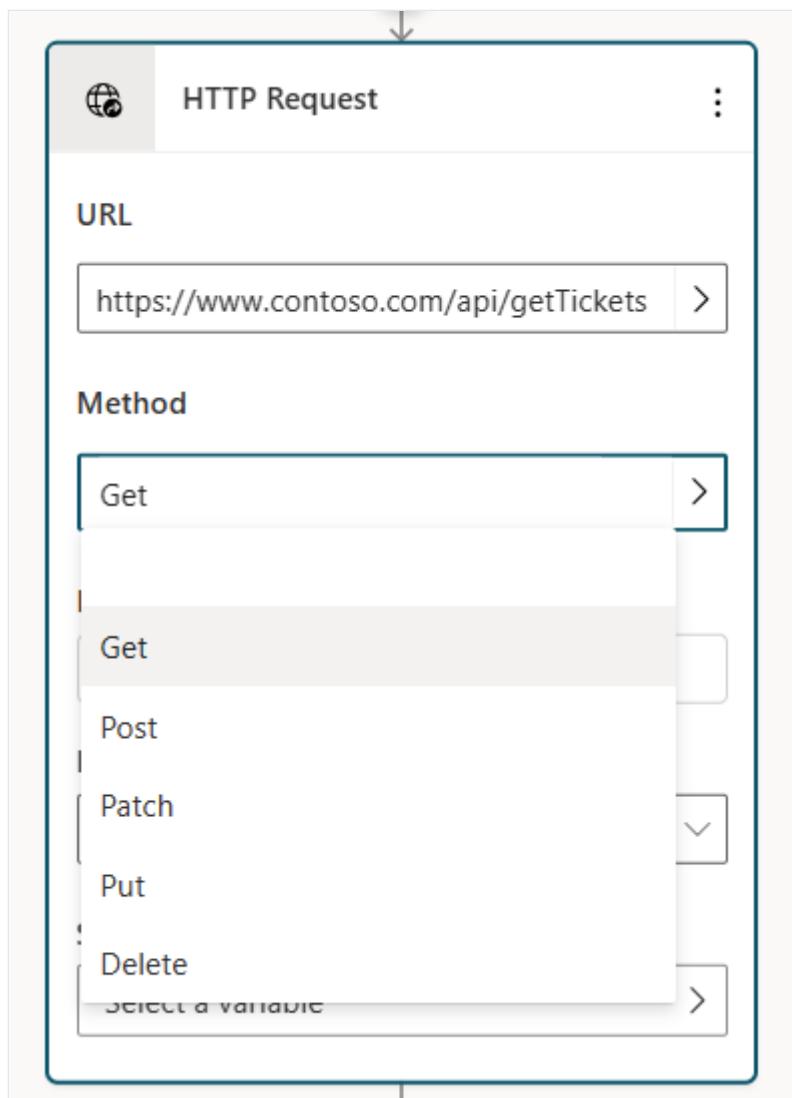
- [Create and edit topics](#)

Use the HTTP request node

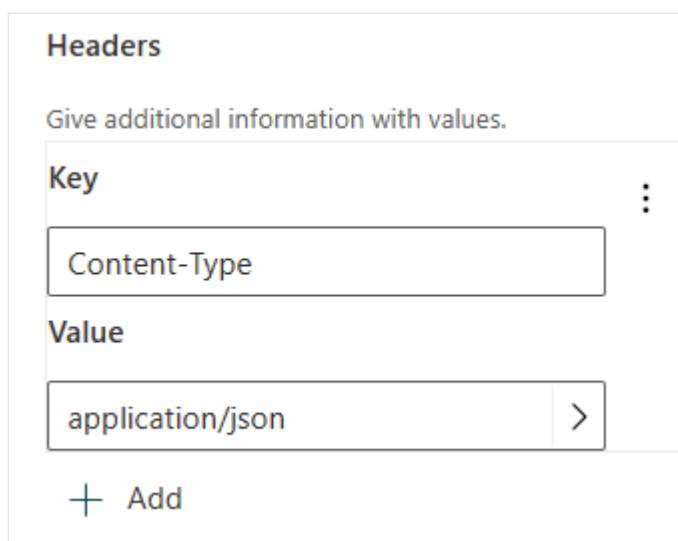
1. Select **Add node** (+), open the **Advanced** sub menu, and then select **Send HTTP request**.



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.



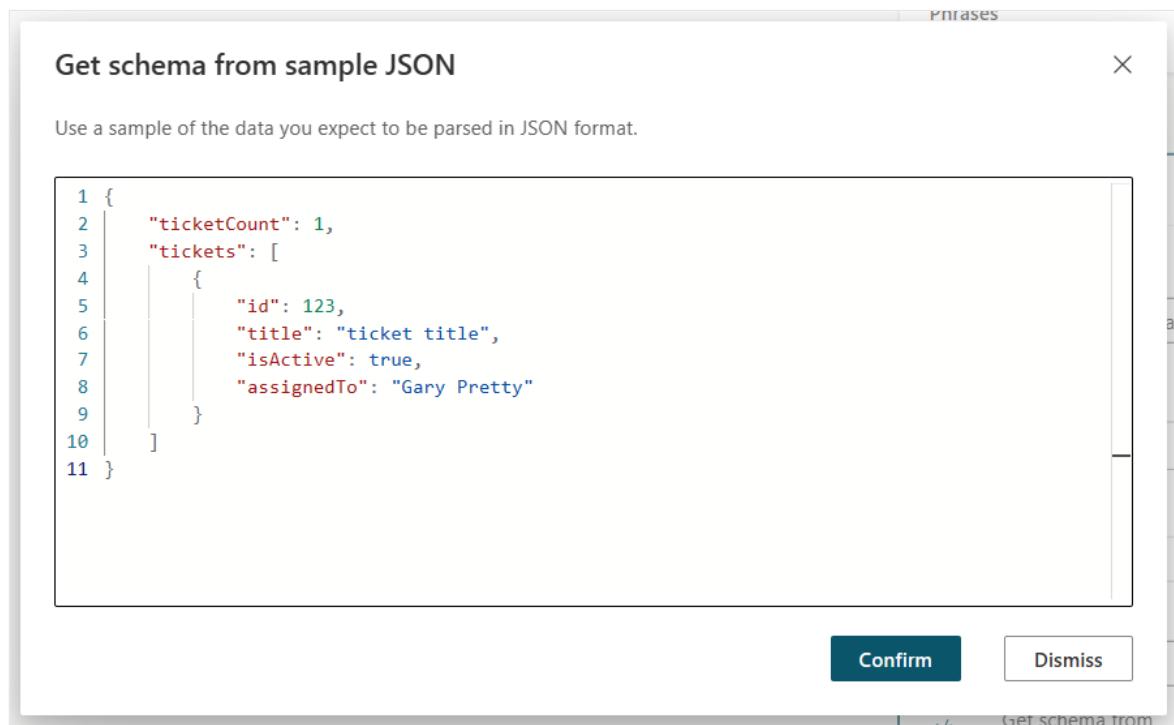
4. Under **Headers and body** on the node, click the **Edit** button. This will open the **HTTP request properties** pane.
5. You can optionally add one or more headers to the HTTP request. Click the **Add** button to add each header's key and value.



6. By default, no content is sent in the body of the HTTP request, but you [specify content to be sent on 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 are calling), which will generate a strongly typed Power FX variable, allowing you to use it elsewhere in your authoring canvas with intellisense support in the Power FX editor. To do this, choose **From Sample Data** and then click **Get schema from sample JSON**, pasting your sample response into the editor that is shown and choosing **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, click the **Edit** button. This will open the **HTTP request properties** pane.
2. Under **Body**, select the appropriate content type. By default this 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 is often used when making POST or PUT requests. When this option is selected, you are presented with an editor where you can enter your JSON content.

You can also enter specify your JSON content using Power FX, allowing you to include dynamic values and reference variables. To do this, select the **Edit JSON** button, choose **Formula** and then enter a Power FX object (any existing JSON that has already been entered will be converted into a Power FX object for you), which will be converted into JSON when the HTTP request is made.

Body

Decide what to send or send nothing

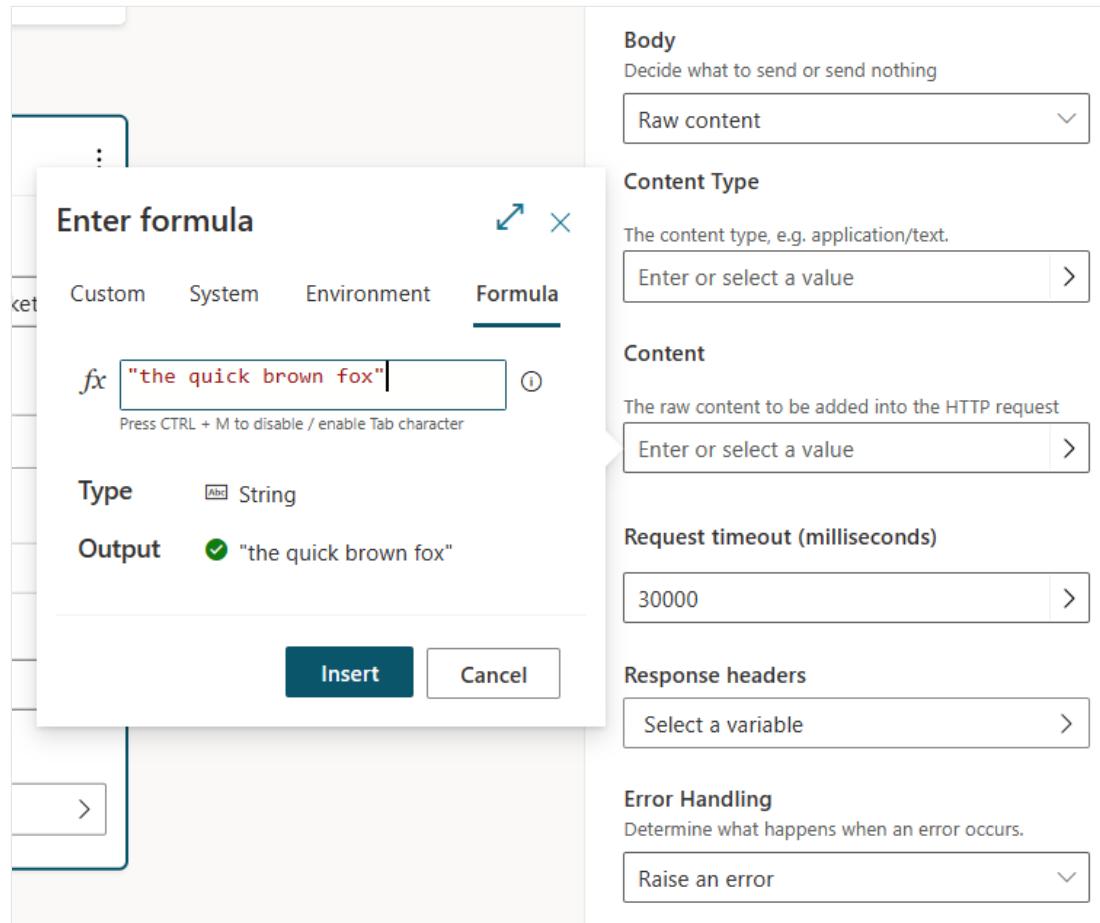
JSON content

</> 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 example below.



Error handling and timeout

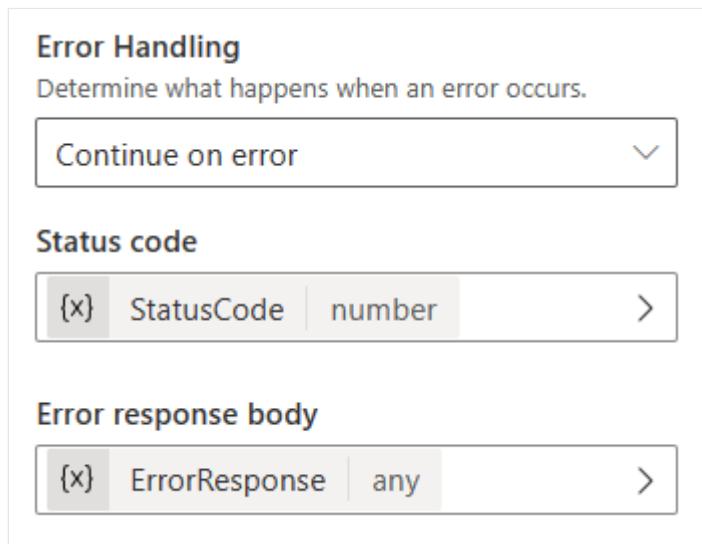
The **HTTP request** node has multiple options for handling situations where the **HTTP request** may fail or return an error.

By default, the feature will "Raise an error". This means that when an **HTTP request** returns an error or fails to execute, the copilot will stop its operation and trigger the **Error system topic**, displaying an error message.

However, you can configure the error handling to not trigger the **Error system topic** and, instead, populate 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 is useful in scenarios where you want the copilot to continue operation even if a particular **HTTP request** fails.

To configure the error handling behavior.

1. Under Headers and body on the node, click the **Edit** button. This will open 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 copilot will store the HTTP status code in the Topic.StatusCode variable and the response body in the Topic.ErrorResponse variable. The copilot will then continue to its next steps in the topic.

The **Error response** variable type will be **Any**. You can use the [Parse Value node](#) to convert this 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.

Sending event activities

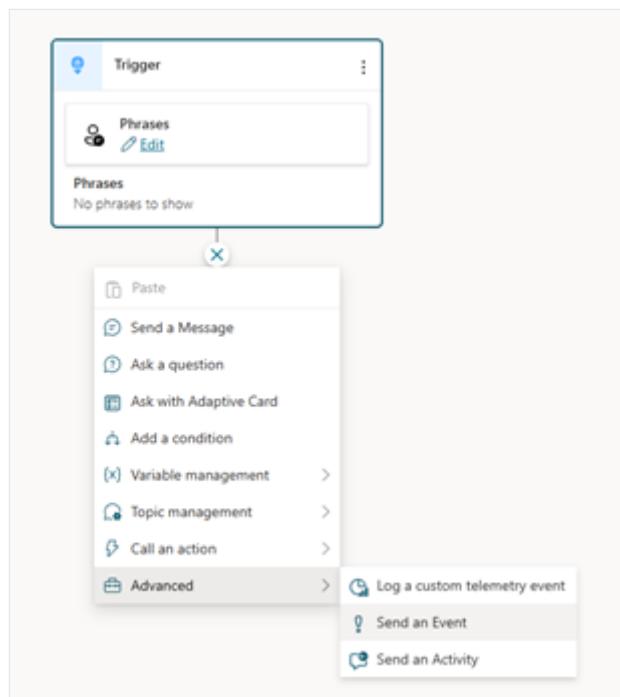
Article • 06/04/2024

ⓘ 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.

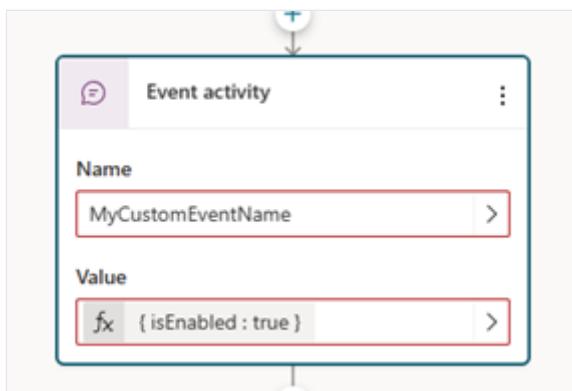
Some articles and screenshots may refer to Power Virtual Agents while we update documentation and training content.

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 copilot 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:

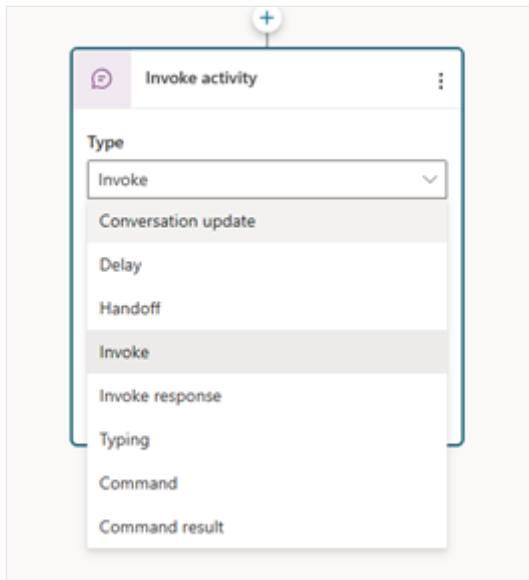
- You can configure a custom web chat control to handle events sent from the copilot. For example, you could look for an event coming back from the copilot and take an action on the page. The `04.api/c.incoming-activity-event` sample in the [Microsoft Bot Framework Web Chat samples repository on GitHub](#) 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 end user, 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

- [Reuse variables across topics](#)
 - [Customize the look and feel of a copilot](#)
-

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Add end-user authentication to a copilot

Article • 06/04/2024

ⓘ 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.

You can enable user authentication directly within a copilot conversation. User authentication means you can get a user's basic properties such as name and ID in variables, but also prompt a user to sign in using an authentication node, retrieve a user token for that user, and then use that token to retrieve the user's information from a back-end system.

ⓘ Note

Copilots created in the web app and in Microsoft Teams come with authentication pre-configured to use **Only for Teams and Power Apps** authentication.

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

Prerequisites

- [Use variables](#)
- [Configure end-user authentication](#)

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

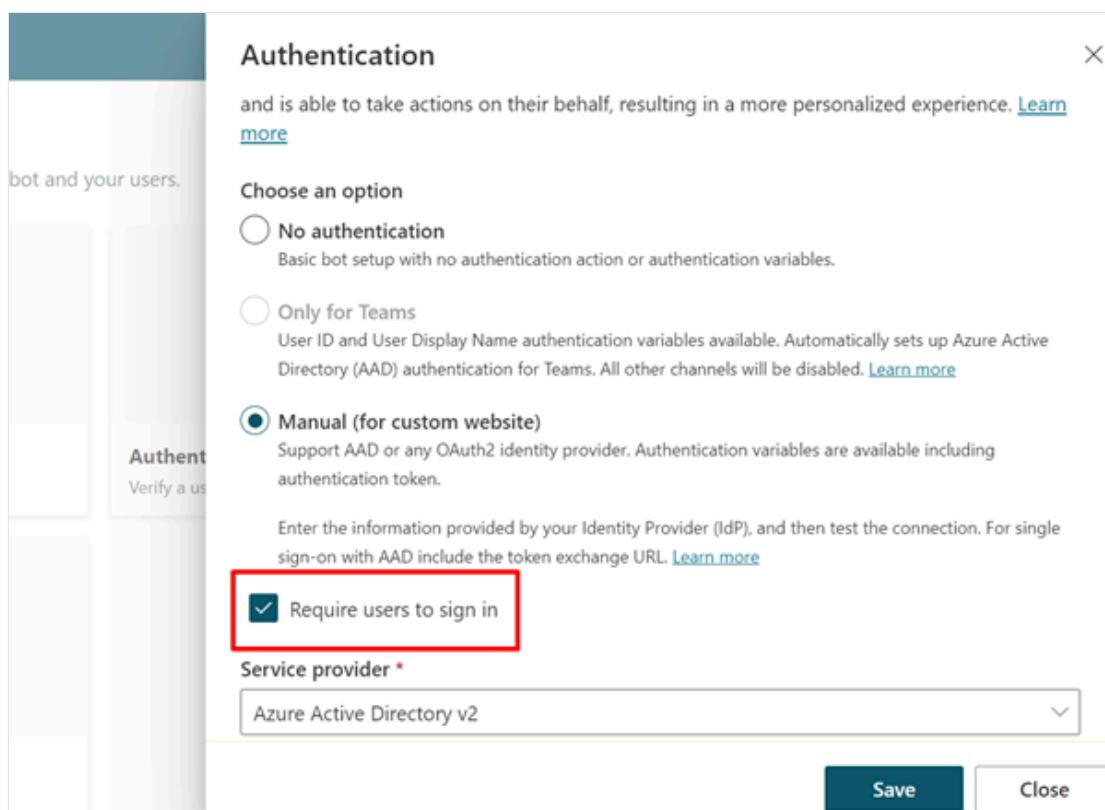
You need to configure user authentication with Microsoft Entra ID before you can use authentication in your topics.

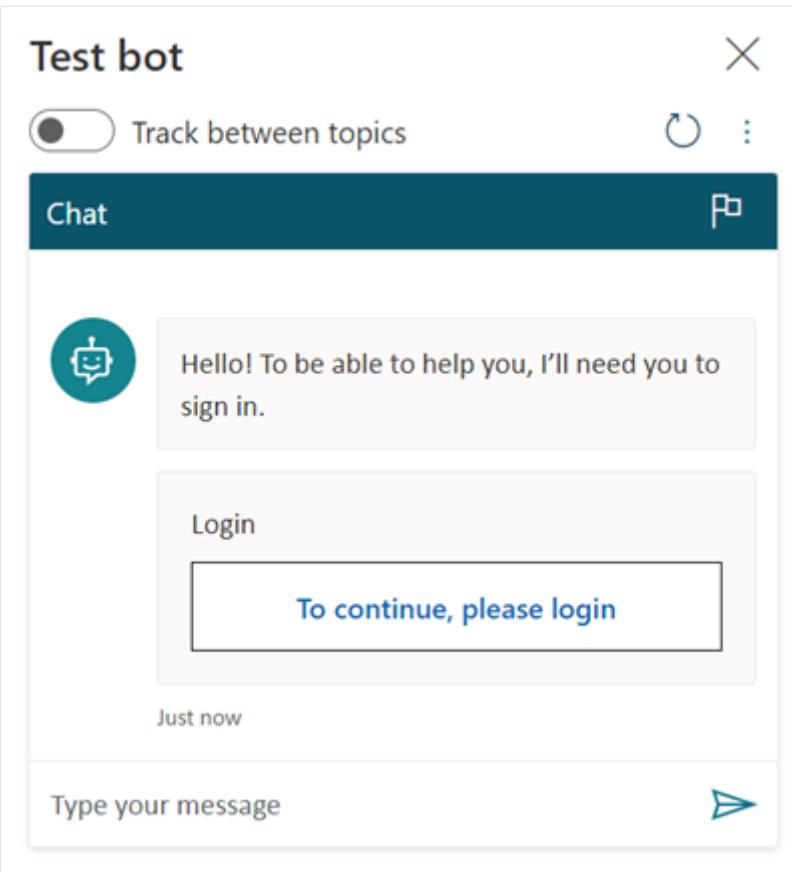
Follow the instructions in [Configure user authentication with Microsoft Entra ID](#).

Add user authentication with the Sign in system topic

When you create a copilot, Copilot Studio automatically adds a system topic called **Sign in**. To use it, you must set your copilot's authentication to manual and require users to sign in. When a customer starts a conversation with the copilot, the **Sign in** topic triggers and prompts the user to sign in. You can customize the **Sign in** topic as appropriate for your copilot.

1. In the navigation menu under **Settings**, select **Security**.
2. Select the **Authentication** tile.
3. Select **Manual (for custom website)**, and then select **Require users to sign in**.





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 username and password, they might be prompted to enter a validation code. After they've logged in, they won't be prompted again, even if they reach another **Authenticate** node.

1. In the navigation menu under **Settings**, select **Security**.
2. Select the **Authentication** tile.
3. Turn off **Require users to sign in**.
4. In the navigation menu, select **Topics**, and then open the topic you want to add authentication to.
5. Select **Add node (+)**, select **Call an action**, and then select **Authenticate**.
6. [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've configured user authentication for your copilot, you can use authentication variables in your topics. The following table compares the availability of these variables based on the authentication option you chose.

 Expand table

Authentication Variable	No Authentication	Only for Teams and Power Apps	Manual
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 that's 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 copilot, 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 userID that's 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 if the user hasn't signed in yet, and they reach a point in the conversation that uses authentication variables, they'll be 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 ↗](#)

Create expressions using Power Fx

Article • 04/29/2024

ⓘ 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.

Power Fx is a low-code language that uses Excel-like formulas. Use Power Fx to create complex logic that allows your copilots 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 Microsoft 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

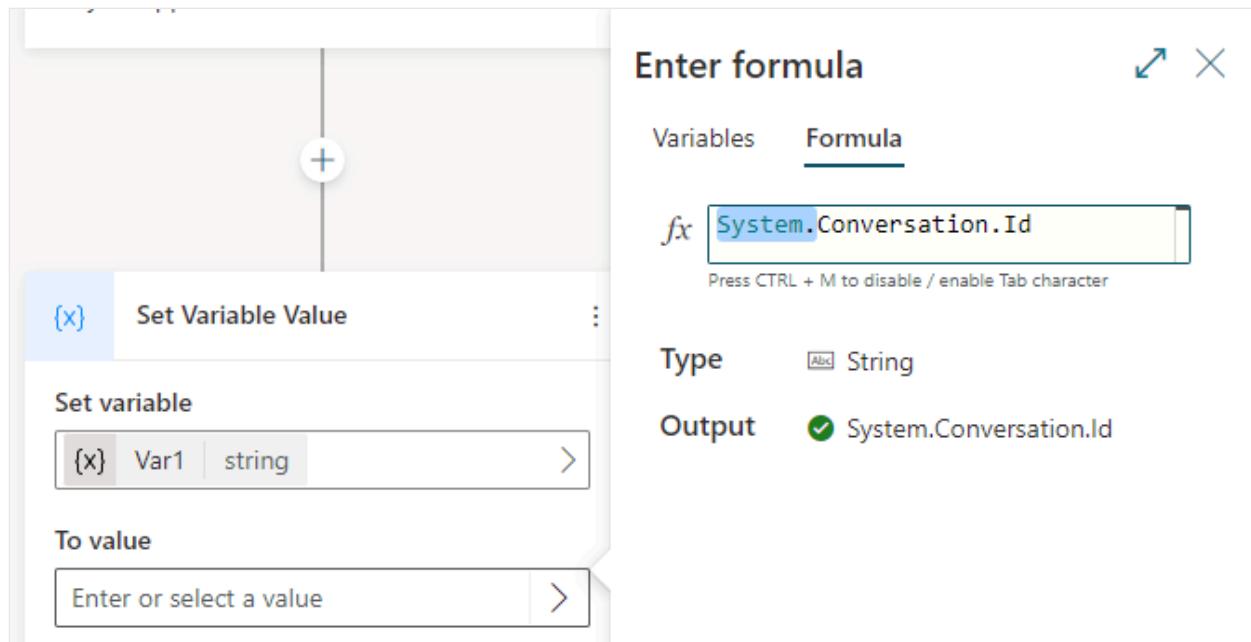
- [Create and edit topics](#)
- [Use variables](#)

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, you'd need to refer to it as `System.Conversation.Id`.



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.

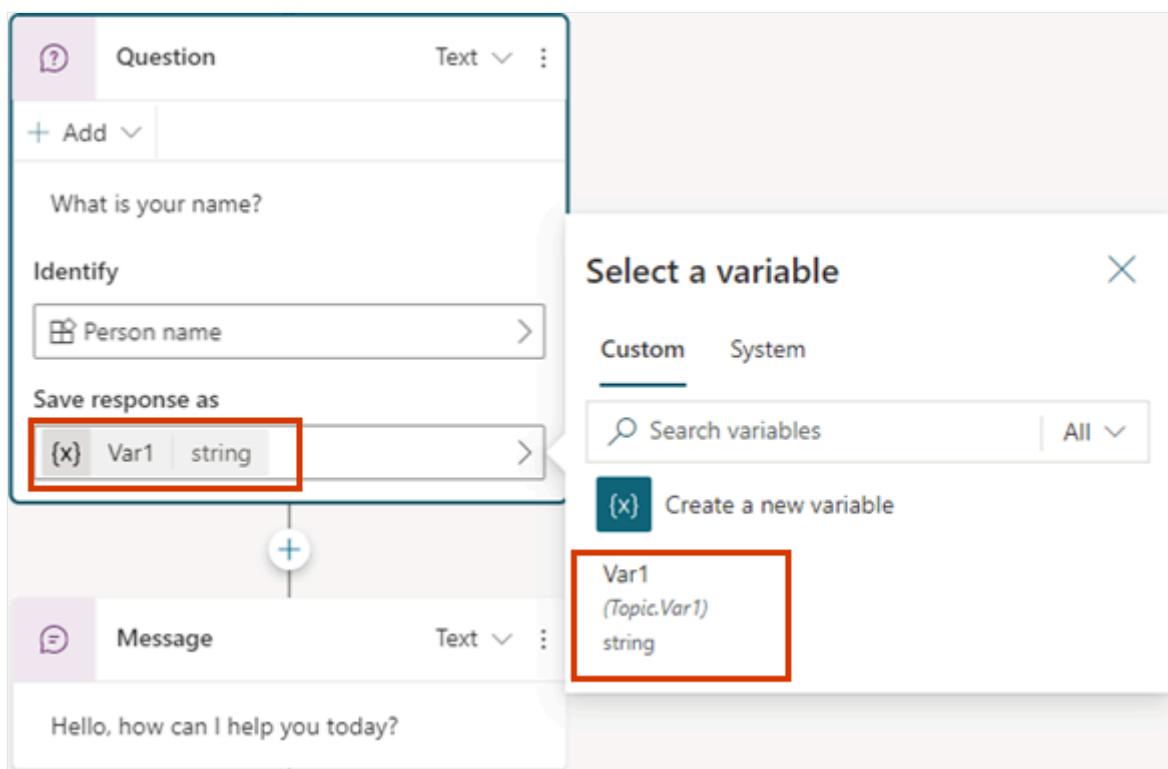
[+] Expand table

Type	Format examples
String	<code>"hi"</code> , <code>"hello world!"</code> , <code>"copilot"</code>
Boolean	Only <code>true</code> or <code>false</code>
Number	<code>1</code> , <code>532</code> , <code>5.258</code> , <code>-9201</code>
Table	<code>[1]</code> , <code>[45, 8, 2]</code> , <code>["cats", "dogs"]</code>
Record	<code>{ id: 1 }</code> , <code>{ message: "hello" }</code> , <code>{ name: "John", info: { age: 25, weight: 175 } }</code>
DateTime	<code>Time(5,0,23)</code> , <code>Date(2022,5,24)</code> , <code>DateTimeValue("May 10, 2022 5:00:00 PM")</code>
Choice	Not supported
Blank	Only <code>Blank()</code>

Use Power Fx to set a variable

In this example, we'll use a Power Fx expression to store the customer's name and output it 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**.

Question

Text

What is your name?

Identify

Person name

Save response as

{x} customerName string

Enter formula

Custom System Formula **Formula**

fx `Upper(Text(Topic.customerName))`

Type String

Output `Upper(Text(Topic.customerName))`

Set Variable Value

Set variable

{x} capsName blank

To value

Enter or select a value >

Insert Cancel

9. Under the **Question** node, select + and then select **Send a message**.

10. Enter **HELLO**, select {x}, and then select **capsName**.

Message

Text

+ Add < B I <= <= {x}

HELLO,

Select a variable

Custom System

Search variables All

capsName
(Topic.capsName)
string

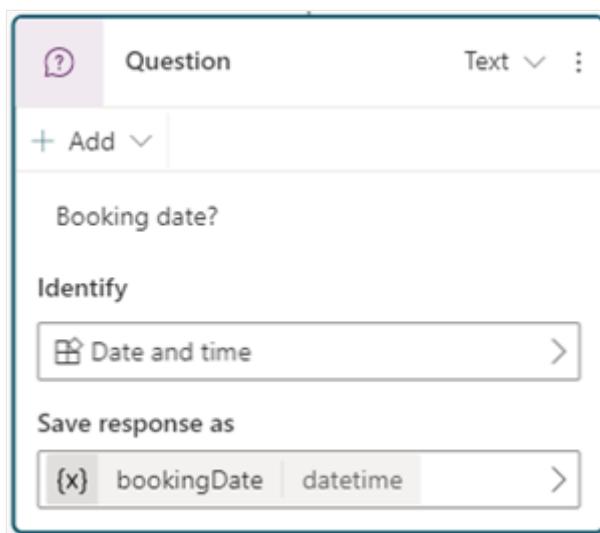
customerName
(Topic.customerName)
string

Use a Power Fx formula as a condition

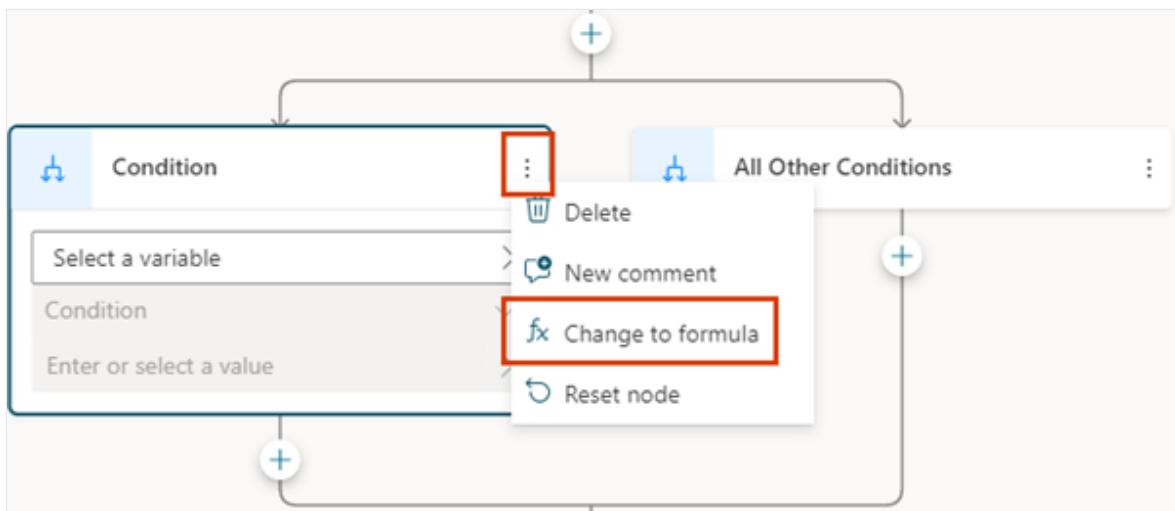
To evaluate more complex expressions, set up **Condition** nodes to use Power Fx formulas.

In this example, the copilot 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 + icon and then select **Add a condition**.
6. In the **Condition** node, select the **Node menu** (⋮), and then select **Change to formula**.

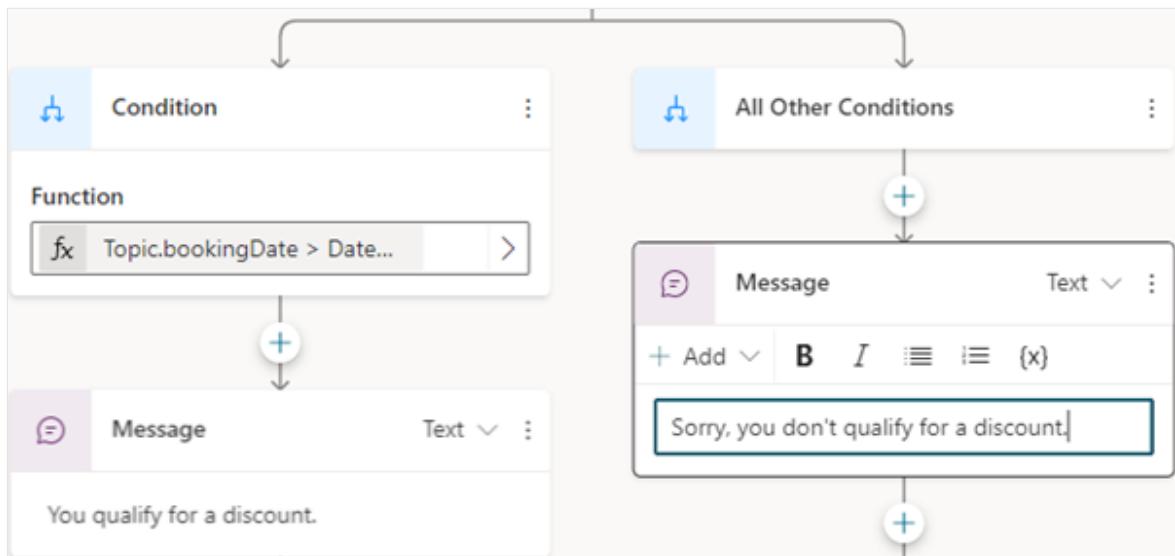


7. In the **Function** box, select the > arrow, and then select the **Formula** tab.

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 **Send a message** node and enter the message `You qualify for a discount.`

10. Under the **All Other Conditions** node, add a **Send 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

Ask with Adaptive Cards (preview)

Article • 06/07/2024

ⓘ 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.

ⓘ 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 copilot, see [Microsoft Copilot Studio Overview](#).

[Adaptive Cards](#) enable you to add snippets of content to Copilot Studio copilots that can also be openly exchanged with other cloud apps and services. To provide rich conversation capability for your copilot, 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.

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.



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 copilot. If it doesn't

and is only intended to show information to the copilot 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 =

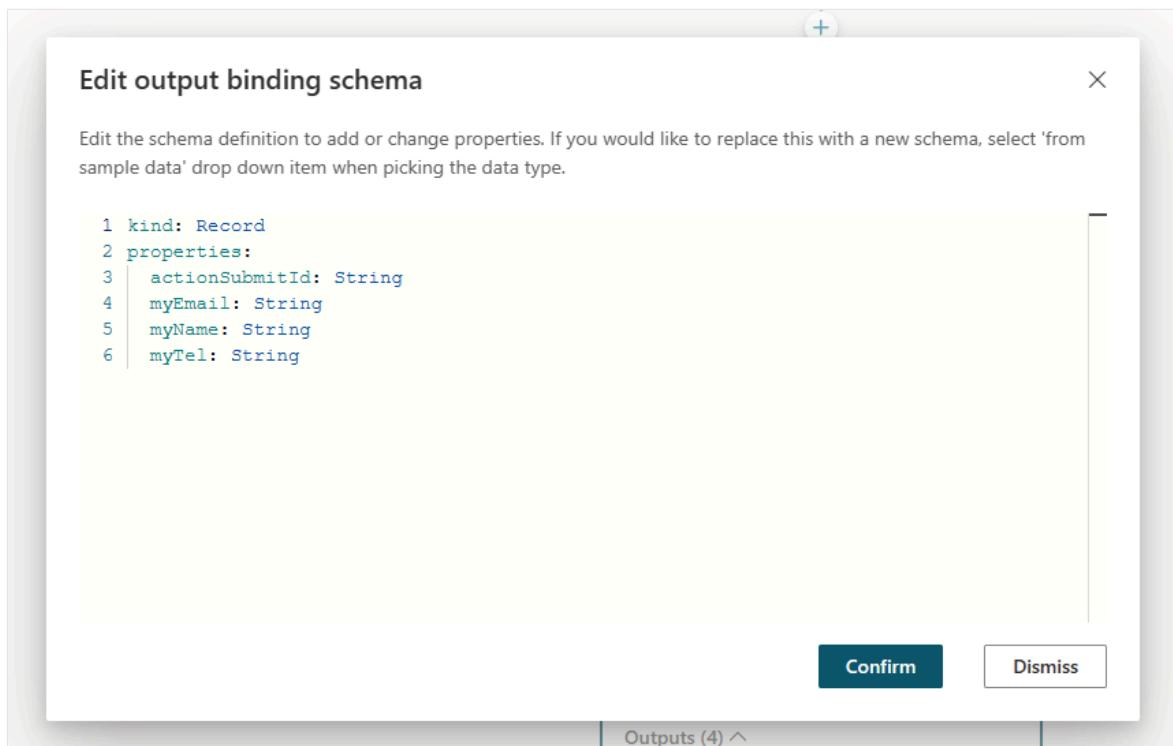
{x} myName | string >

{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 copilot 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 copilot responds to an invalid response
- If it can be interrupted

If the copilot 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 copilot 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 copilot 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.

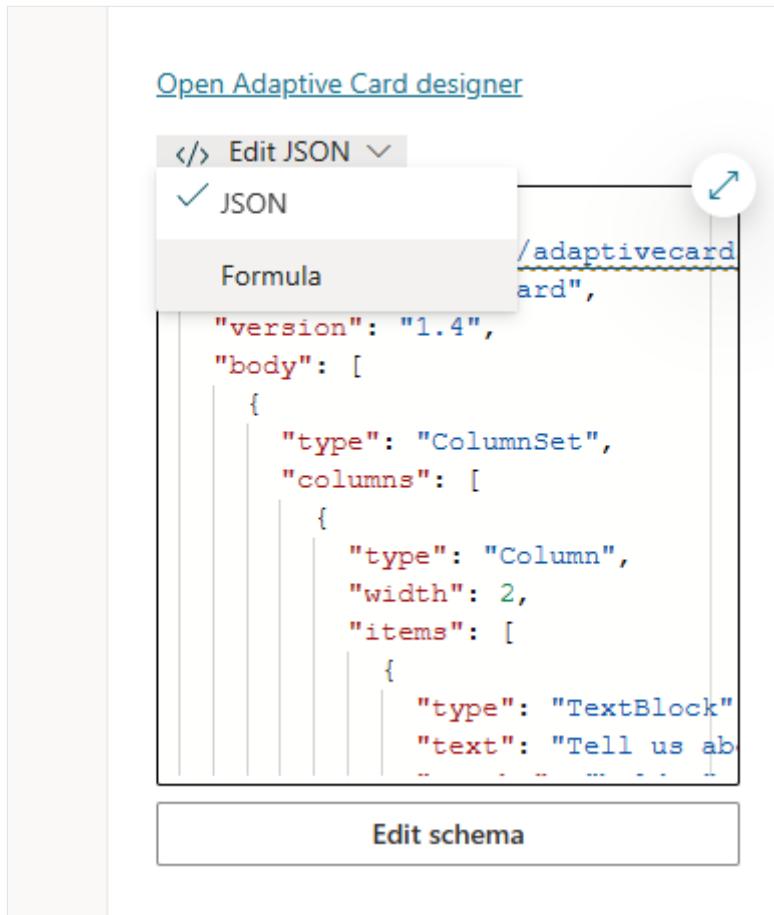
Use Power Fx to make your card dynamic

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

```
{
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "type": "AdaptiveCard",
  "version": "1.5",
  "body": [
    {
      "type": "ColumnSet",
      "columns": [
        {
          "type": "Column",
          "width": "2",
          "items": [
            {
              "type": "TextBlock",
              "text": "Topic.Title",
              "weight": "Bolder",
              "size": "Medium",
              "wrap": true,
              "style": "heading"
            },
            {
              "type": "TextBlock",
              "text": "Topic.Subtitle",
              "isSubtle": true,
              "wrap": true
            },
            {
              "type": "Input.Text",
              "id": "myName",
              "label": "Your name (Last, First)",
              "isRequired": true,
              "regex": "^[A-Z][a-z]+, [A-Z][a-z]+$",
              "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 ↗](#)

Send proactive messages Microsoft Teams

Article • 06/05/2024

ⓘ 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.

After you've [published your copilot](#) and [made the copilot available to end 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 copilot could send a reminder message to complete online training.

ⓘ Important

Proactive messages can be configured for copilots and classic chatbots created and customized in Copilot Studio.

A copilot can't deliver messages if the recipient:

- Doesn't have the copilot installed in Microsoft Teams.
- Has uninstalled the copilot.
- Has blocked the copilot.
- Doesn't have permission to chat with the copilot, in which case you need to [share the copilot with users](#).

Prerequisites

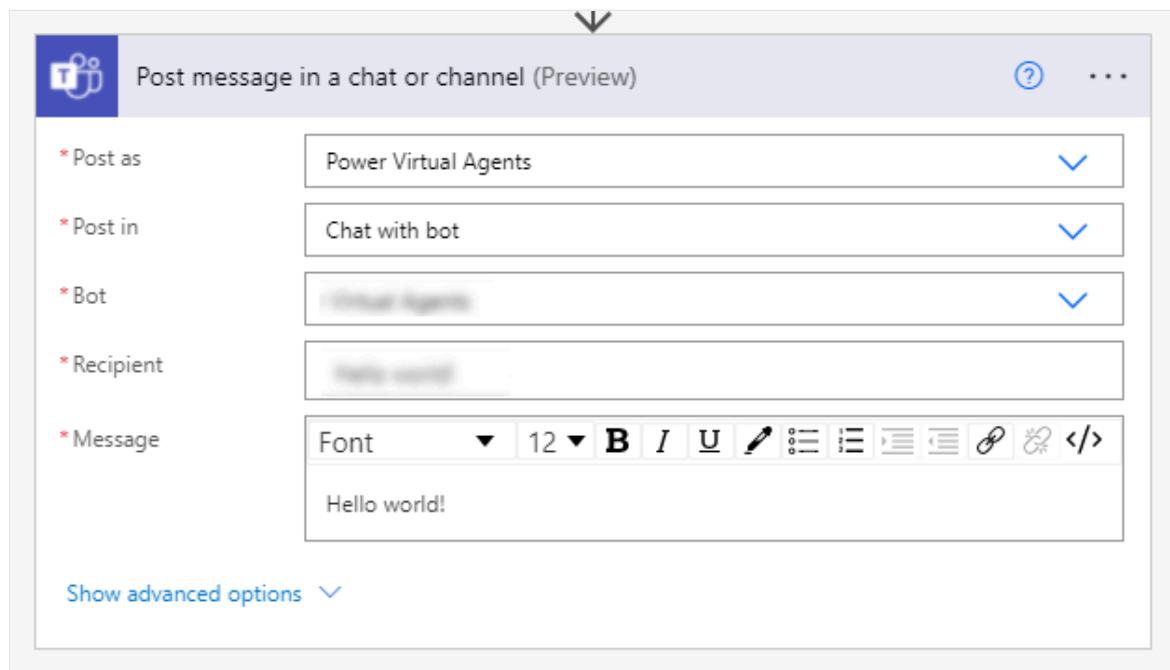
- Learn how to use and create flows
- Publish your copilot
- Add your copilot to Microsoft Teams
- Getting started with Power Automate

Known limitations

- If the copilot is disconnected and reconnected to Microsoft Teams, users won't receive proactive messages until after they reinstall the copilot.
- 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 copilot.

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 copilot 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.



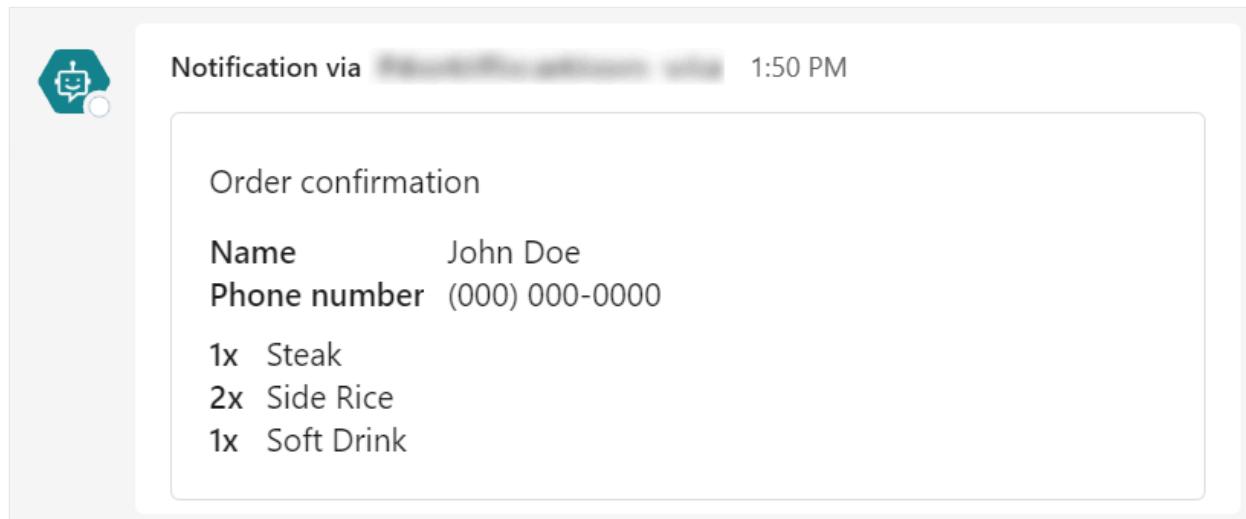
When the flow is run, the recipient will receive the proactive message from the copilot in Microsoft Teams.

Send a proactive Adaptive Card

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 a copilot. To learn how to send Adaptive Cards in an *active* conversation, see [Add Adaptive Cards with Composer](#).

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 copilot 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:

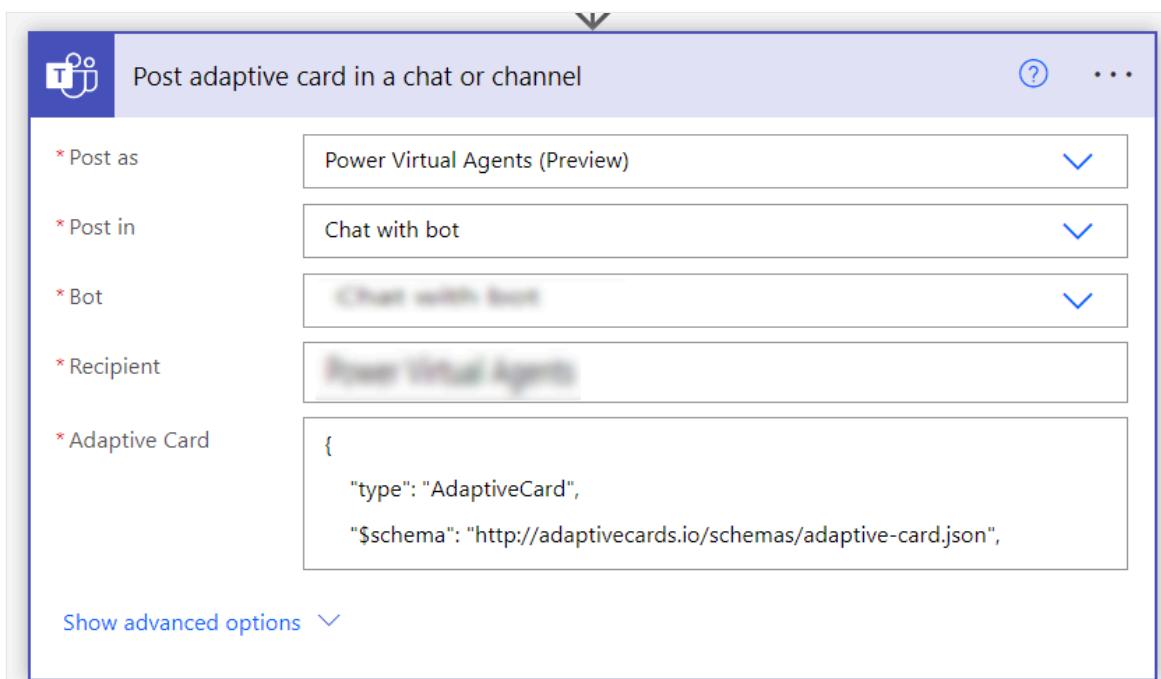
JSON

```
{  
  "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"
      }
    ]
  }
}

```



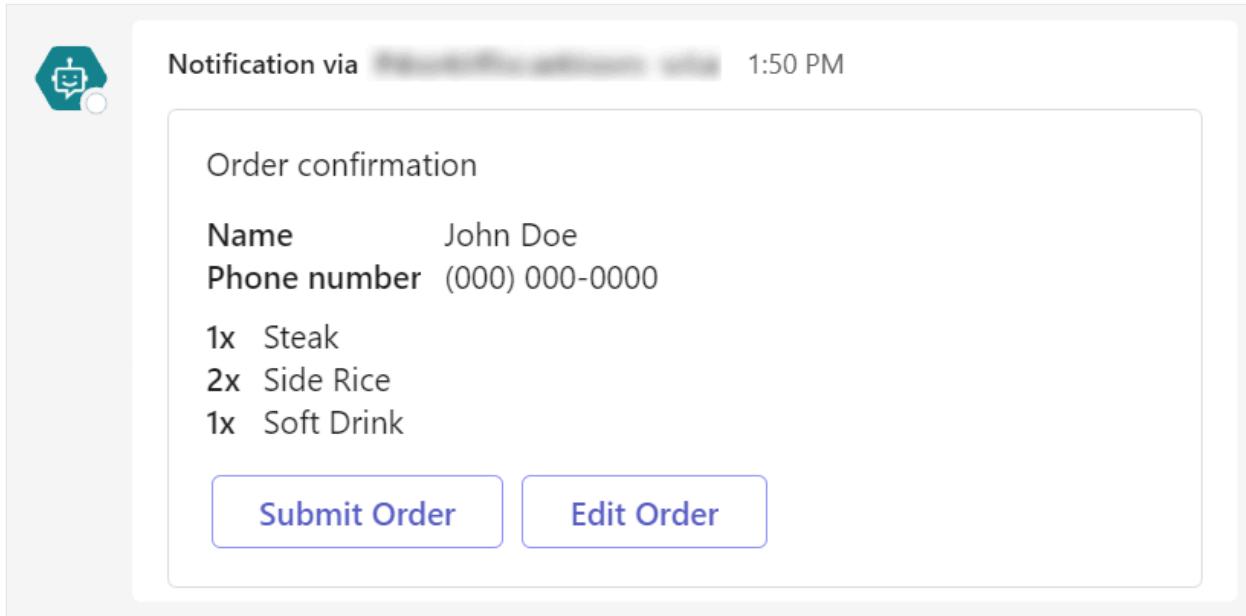
When the flow is run, the recipient will receive the Adaptive Card from the copilot in Microsoft Teams.

Wait for user response

ⓘ Important

This section details how to send proactive Adaptive Cards with Power Automate flows. Proactive Adaptive Cards enables users to receive information from an *inactive* conversation with a copilot. To learn how to send Adaptive Cards in an *active* conversation, see [Add Adaptive Cards with Composer](#).

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

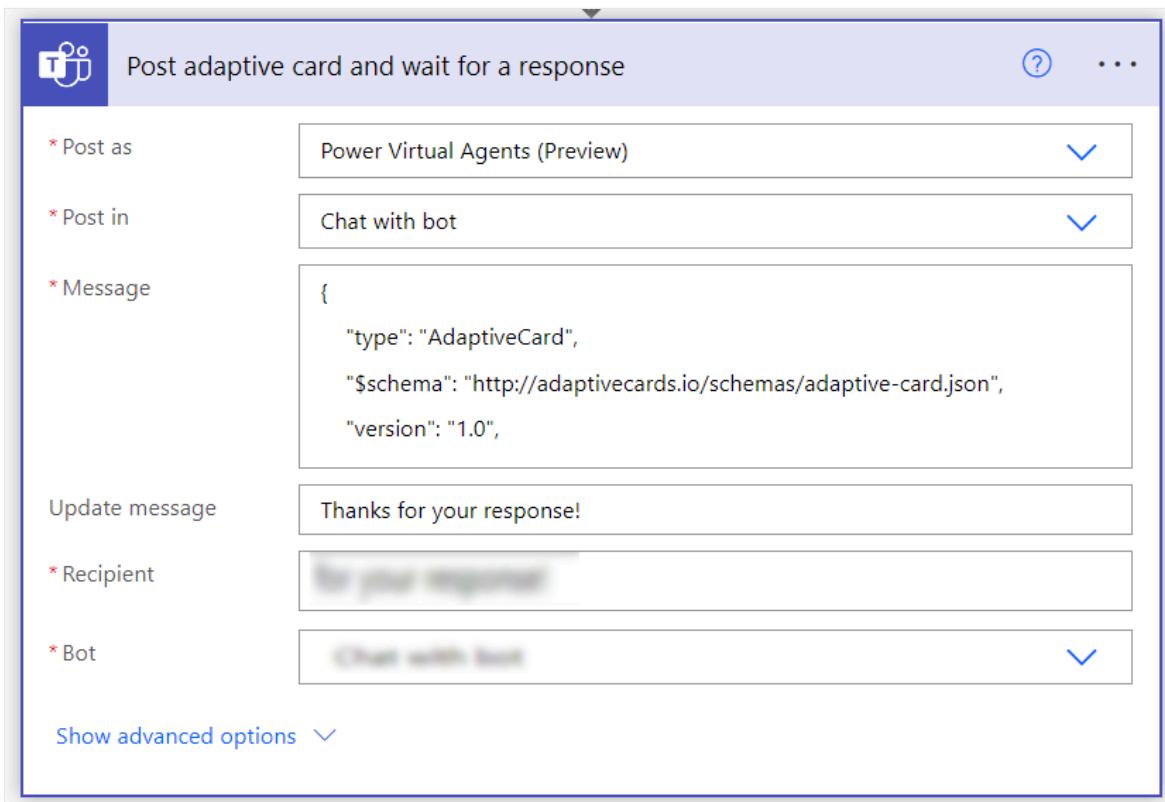
```
{  
    "type": "AdaptiveCard",  
    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",  
    "version": "1.0",  
    "body": [  
        {  
            "type": "TextBlock",  
            "text": "Order confirmation",  
            "wrap": true,  
            "style": "heading"  
        },  
        {  
            "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": "Place Order"  
            }  
        ]}  
    ]  
}
```

```

        "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 copilot that you want the card to be sent from.



When the flow is run, the recipient will receive the adaptive card from the copilot 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

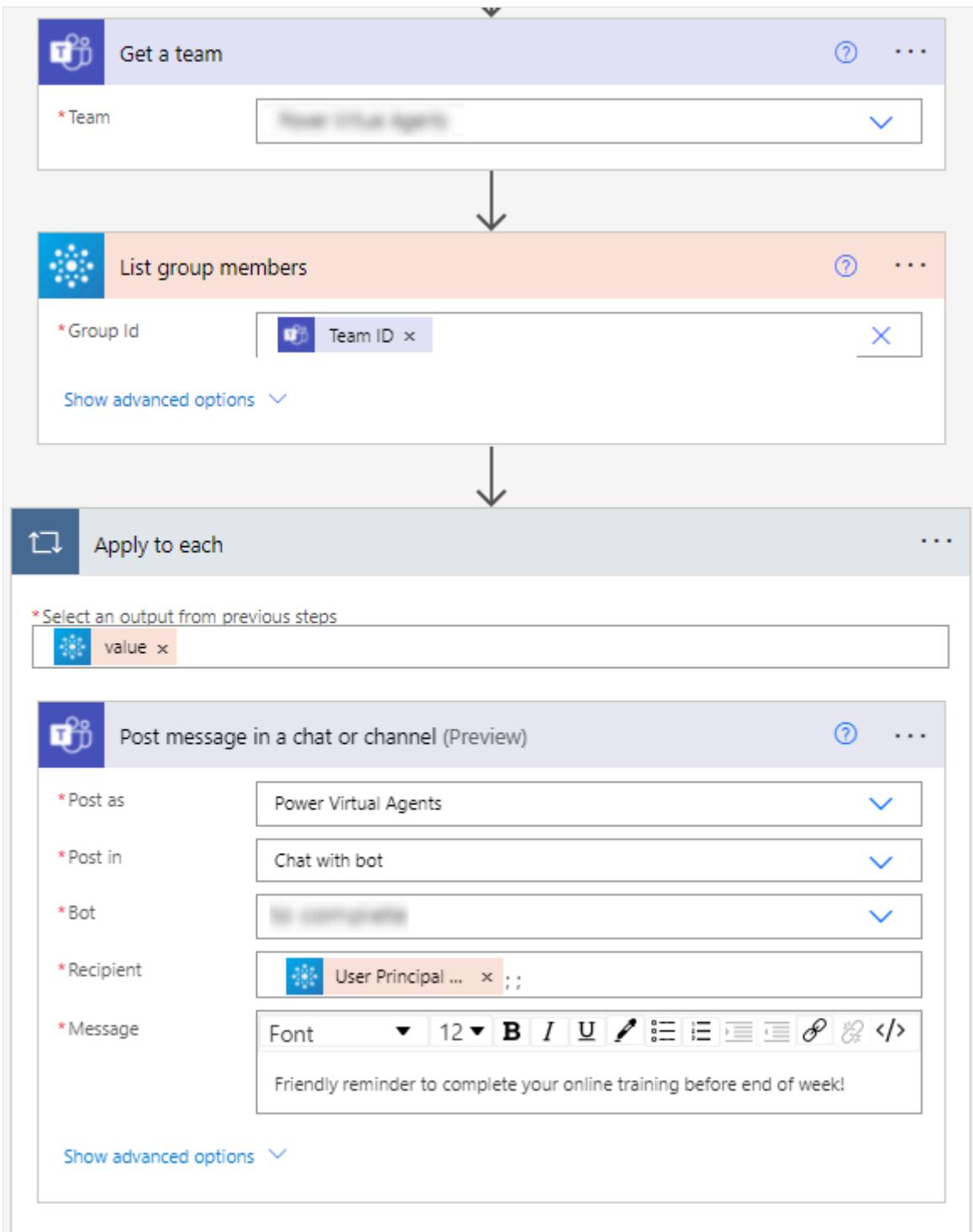
The prerequisites detailed in this topic apply to each recipient.

For example, you can send the same message to 10 recipients in one action. If three of the 10 recipients do not have the copilot installed, then those three recipients won't receive the message.

Send to teammates

In this example, the copilot 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 copilot 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.

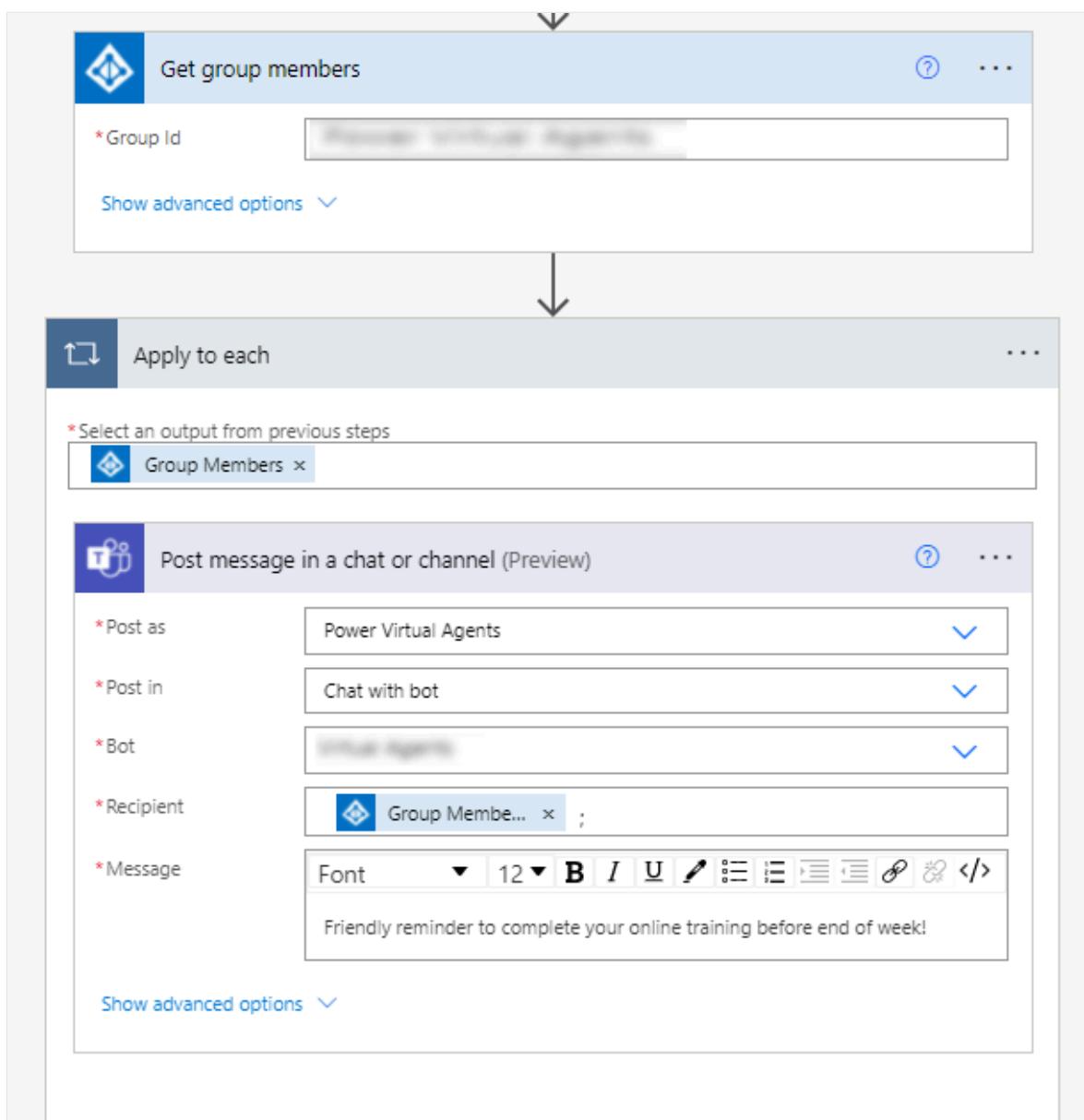
When the flow runs, each user in the team will receive the proactive message in a private chat with the copilot.

Send to a security group

In this example, the copilot 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 copilot 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 copilot.

Send proactive message to multiple recipients in parallel

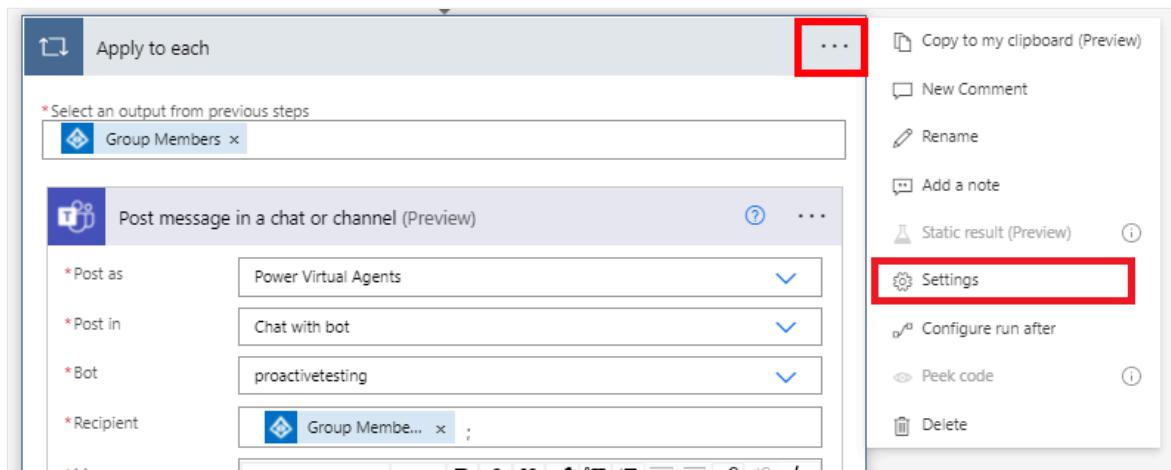
Normally when sending a proactive message to multiple recipients, your copilot will send one message after another. However, in some situations, it might be preferable to send the message to multiple recipients at the same time.

⚠️ Warning

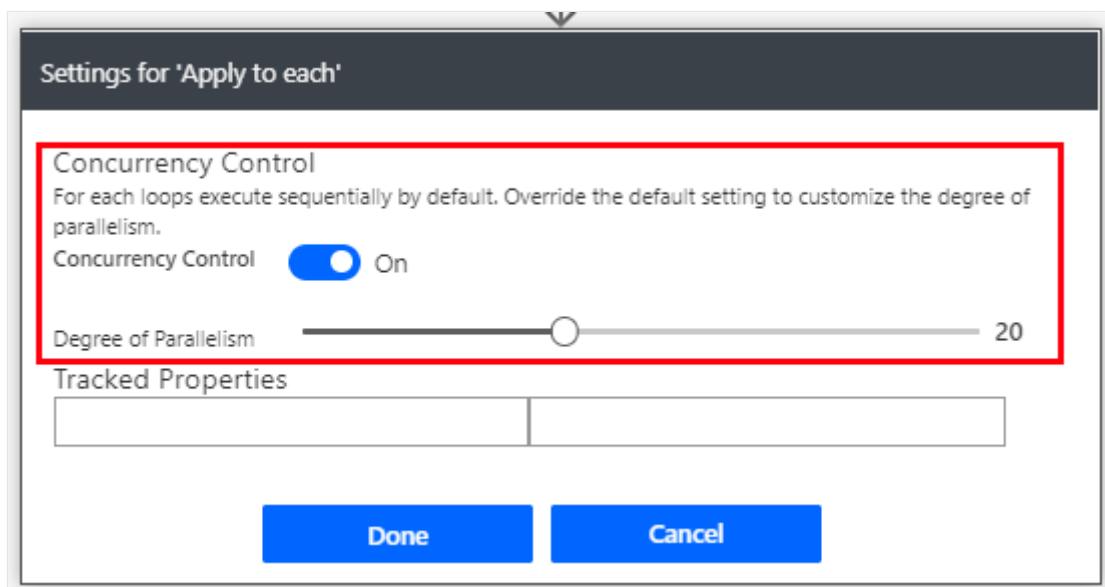
All proactive messages from Copilot Studio are subject to [limits on Power Automate and throttling limits of the Microsoft Teams connector](#).

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**.

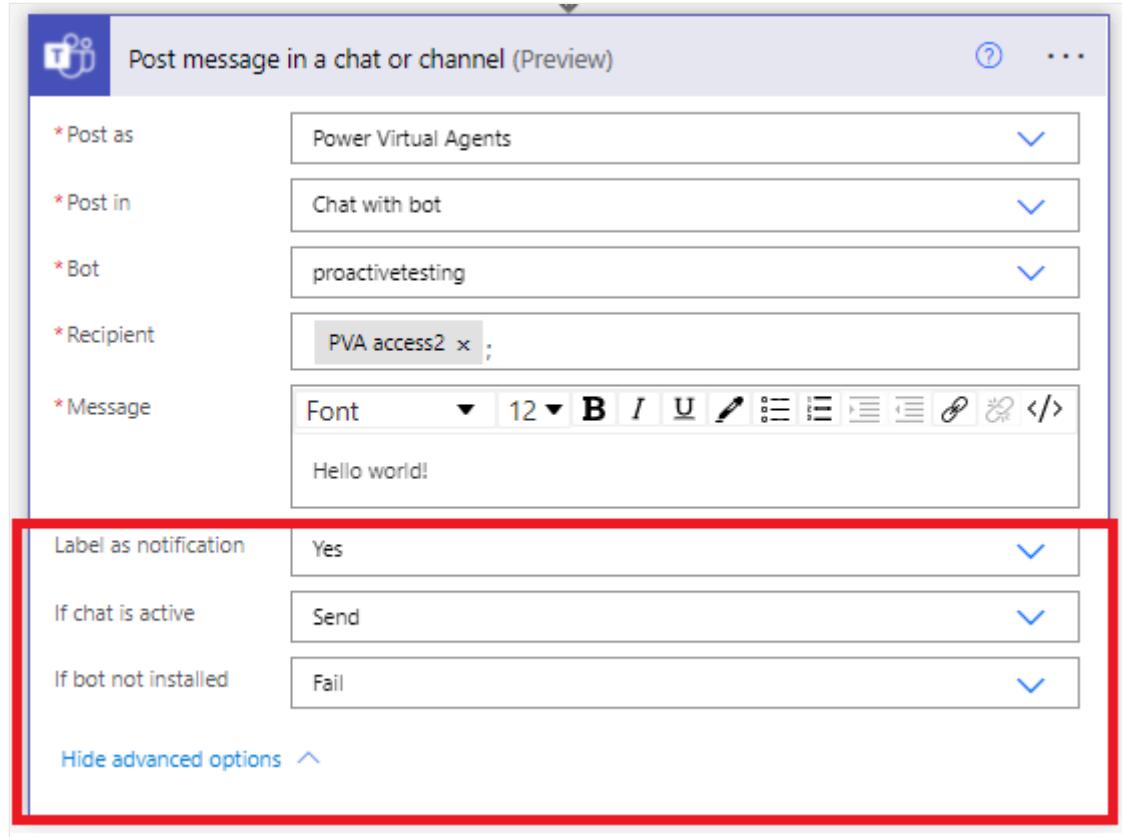


2. Turn on **Concurrency control** and set the degree of parallelism.



Configure advanced options for proactive messages

Copilot Studio allows you to control detailed behavior on your copilot 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 copilot's name. Labeling the copilot's response allows the recipient to identify the copilot's response to their inquiry.



When the recipient is currently in an active chat with the copilot

Sometimes the copilot might be sending a proactive message when the recipient is in an active conversation with the copilot.

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 copilot will send the proactive message as normal.
- **Don't send and succeed:** the copilot 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 copilot 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 copilot

The copilot can only deliver messages to recipients who have installed the copilot in Microsoft Teams. Recipients might not want to install the copilot or have uninstalled the copilot.

For lower importance messages, you can set the flow run to be marked as succeeded even when the recipient doesn't have the copilot 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 copilot 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 copilot. 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 copilot installed.
300	False	Message couldn't be delivered because the recipient is in an active conversation with the copilot.

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Accommodate time zones

Article • 03/26/2024

ⓘ 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.

The **Date and time** entity captures a date and time in Coordinated Universal Time (UTC). However, you may want to display the date and time based on the user's location instead.

Prerequisites

- [Use variables](#)
- [Use conditions](#)
- [Create expressions using Power Fx](#)

How Microsoft Copilot Studio determines the user's time zone

Microsoft 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

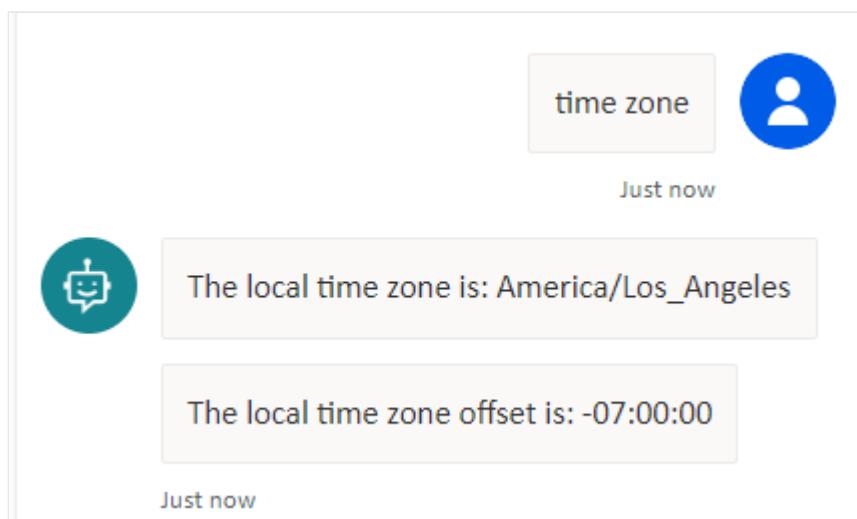
Use these system-level variables to get information about the user's time zone:

- `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

Microsoft Copilot Studio [automatically determines the user's time zone](#) when your copilot 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 type, 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 type, enter `The local time zone offset is:`, then select *Insert variable* (`{x}`), and insert the system variable `Conversation.LocalTimeZoneOffset`.
7. Save and test the copilot.



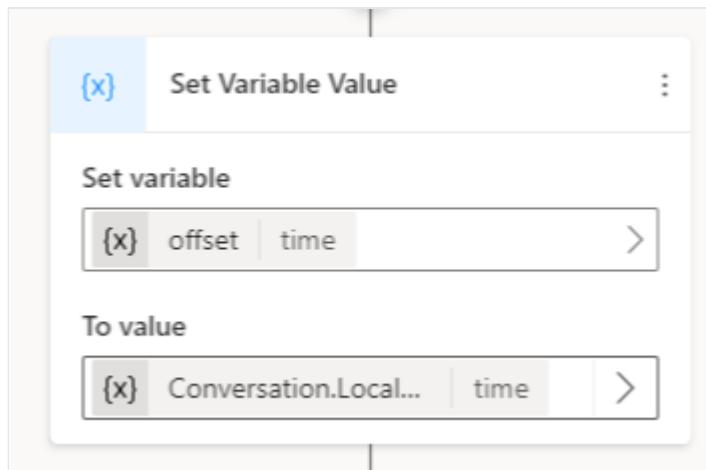
Display the date and time in the local time zone

Microsoft 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.

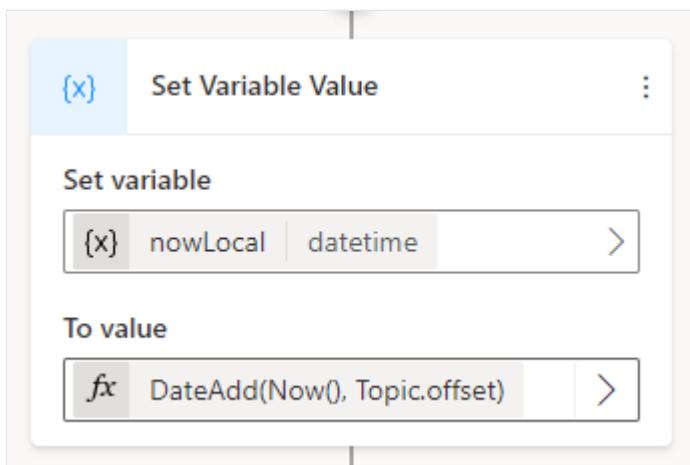
In this example, we'll get the current date and time using the Power Fx `Now()` function, and then add the time zone offset. It isn't possible to use the `Conversation.LocalTimeZoneOffset` system variable directly in a Power Fx formula.

Instead, we'll use a **Set Variable Value** node to create a variable and then assign it the value of `Conversation.LocalTimeZoneOffset`.

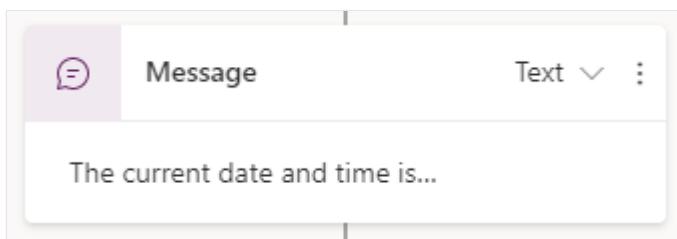
1. Create a topic and add the trigger phrase `what time is it.`
2. In the authoring canvas, select **Add node (+)**. Select **Set a variable value**.
 - For **Set variable**, create a variable named `offset`.
 - For **To value**, select the system variable `Conversation.LocalTimeZoneOffset`.



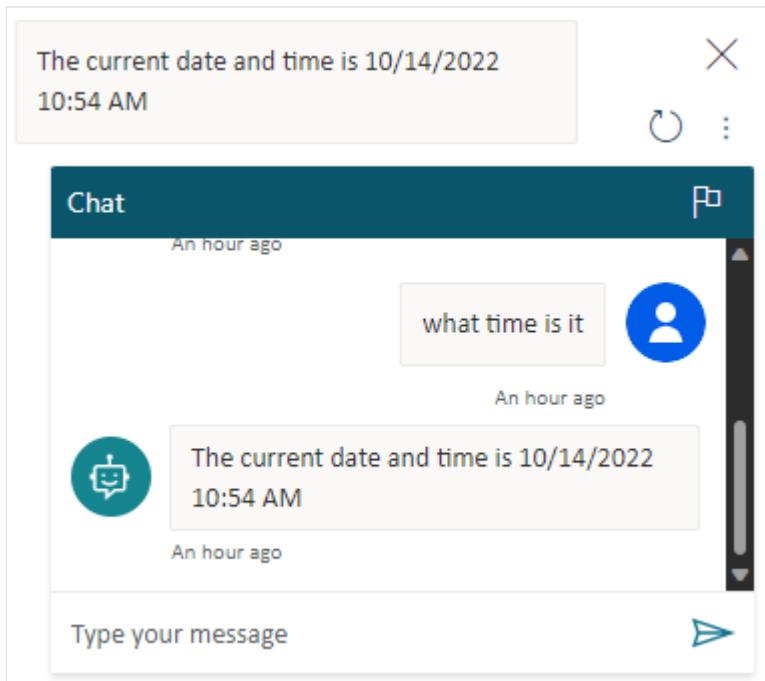
3. Add a second **Set Variable Value** node.
 - For **Set variable**, create a variable named `nowLocal`.
 - For **To value**, enter the Power Fx formula `DateAdd(Now(), Topic.offset)`.



4. Add a **Message** node. Enter the message `The current date and time is {Topic.nowLocal}`.



5. Test your copilot in the **Test copilot** pane to get the date and time in your time zone.



Related topics

[Power Fx date and time reference](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Hand off to a live agent

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.

With Microsoft Copilot Studio, you can configure your copilot to hand off conversations to live agents seamlessly and contextually.

When your copilot 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 the [Configure handoff to Omnichannel for Customer Service](#) article.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4n4G1?postJs||Msg=true> ↗

ⓘ Note

You can choose to escalate a copilot 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 copilot 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 a copilot from Dynamics 365 Customer Service, the **Escalate** system topic already includes a **Transfer conversation** node. However, copilots 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 copilot 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 copilot 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 copilot may be unable to determine the intent of a customer's conversation. For example, the customer may be asking a specific question for which there's no **topic**, or no matching option within a topic.

In other instances, the customer may ask to be handed off to a live agent immediately. For example, a customer may type "talk to agent" mid-way into a conversation.

When the copilot 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 copilot, 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 will be 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 copilot 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 will start 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 may 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
<code>va_Scope</code>	Route escalations to a live agent.	<code>"copilot"</code>
<code>va_LastTopic</code>	Route escalations to a live agent and help them ramp-up. Includes the last topic that was triggered by an utterance from the user.	<code>"Return items"</code>
<code>va_Topics</code>	Ramp-up a live agent. Only includes topics triggered by end user using a trigger phrase. Doesn't include topics that were redirected to.	<code>["Greetings", "Store Hours", "Return Item"]</code>
<code>va_LastPhrases</code>	Route escalation to a live agent and help them ramp-up.	<code>"Can I return my item"</code>
<code>va_Phrases</code>	Ramp-up a live agent.	<code>["Hi", "When does store open", "Can I return my item"]</code>
<code>va_ConversationId</code>	Uniquely identify a copilot conversation.	<code>6dba796e-2233-4ea8-881b-4b3ac2b8bbe9</code>
<code>va_AgentMessage</code>	Ramp-up a live agent.	<code>"Got a gift from: HandoffTest"</code>
<code>va_BotId</code>	Identify the copilot that is handing off a conversation.	<code>6dba796e-2233-4ea8-881b-4b3ac2b8bbe9</code>
<code>va_Language</code>	Route escalation to a live agent.	<code>"en-us"</code>
All user-defined topic variables	Ramp-up a live agent.	<code>@StoreLocation = "Bellevue"</code>

A customer may go through several topics prior to escalating. Your copilot 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 copilot promotes the most recently defined topic variable.

Key concepts - Publish your copilot

Article • 06/04/2024

ⓘ 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.

With Copilot Studio, you can publish copilots to engage with your customers on multiple platforms or channels. These include live websites, mobile apps, and messaging platforms like Microsoft Teams and Facebook.

After you've published at least once, you can connect your copilot to more channels.

Each time you update your copilot, you can publish it again from within Copilot Studio. Publishing your copilot applies to all the channels where you've inserted or connected your copilot.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4mGa9?postJslIMsg=true> ↗

Web app

You need to publish your copilot before your customers can engage with it. You can publish your copilot on multiple platforms, or "channels".

After you've published your copilot to at least one channel, you can connect it to more channels. Remember to publish your copilot again after you make any changes to it.

Publishing your copilot updates it across all the channels it's connected to. If you don't, your customers won't be engaging with the latest content.

The copilot comes with the **Only for Teams and Power Apps** authentication option turned on. The copilot automatically uses Microsoft Entra ID authentication for Teams and Power Apps without requiring any manual setup and only lets you chat with your copilot on Teams.

If you want to allow anyone to chat with your copilot, select **No authentication**.

⊗ Caution

Selecting the **No authentication** option will allow anyone who has the link to chat and interact with your bot or copilot.

We recommend you apply authentication, especially if you are using your bot or copilot 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 copilot, select **Manual**.

Authentication



Verify a user's identity during a conversation. The bot 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

Basic bot setup with no authentication action or authentication variables.

Only for Teams and Power Apps

User ID and User Display Name authentication variables available. Automatically sets up Azure Active Directory (AAD) authentication for Teams and Power Apps. All other channels will be disabled. [Learn more](#)

Manual (for custom website)

Support AAD or any OAuth2 identity provider. Authentication variables are available including authentication token.

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](#)

ⓘ Important

If you select **No authentication**, it won't be possible for your copilot to use [Copilot actions](#) with [end-user credentials](#).

Publish the latest content

1. With your copilot open for editing, in the navigation menu, select **Publish**.
2. Select **Publish**, and then confirm. Publishing can take a few minutes.

Test your copilot

Test your copilot after it's published. You can [make the copilot available to users in Microsoft Teams](#) with the installation link or from various places in the Microsoft Teams app store.

You can share your copilot later by selecting **Make the copilot available to others** from the **Publish** page, in Teams.

You can also install the copilot for your own use in Microsoft Teams by selecting **Open the copilot in Teams**.

If you selected the **No authentication** or **Manual** authentication option, select the **Demo website** link to open a prebuilt website in a new browser tab, where you and your teammates can interact with the copilot.

The demo website is also useful to gather feedback from stakeholders before you roll your copilot out to customers. Learn how to [configure the demo website and add the copilot to your live website](#).

💡 Tip

What's the difference between the test chat and the demo website?

Use the test chat (the **Test copilot** pane) while you're building your copilot 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 copilot. The demo website isn't intended for production use. You shouldn't share the URL with customers.

Publish

Excited to activate your bot? Publish it with a single click. Then, try it out on a website and configure ch

↑ **Publish**

Share your bot

After you publish, try out your bot on the [demo website](#) and invite team members to do the same.

Optimize your bot

Configure channels

After publishing your copilot at least once, you can add channels to make it reachable by your customers.

To configure channels for your copilot:

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. See the article for the desired channels, in the list below, for more information.

- [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 end-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 copilot content for specific channels.

 Expand table

Experience	Website	Microsoft Teams	Facebook	Dynamics Omnichannel for Customer Service
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 Chat . Not supported for other channels.
Did-You-Mean	Supported	Supported	Supported	Supported for Microsoft Teams , Chat , Facebook, and text-only channels (SMS via TeleSign and Twilio , WhatsApp , WeChat , and Twitter). Suggested actions will be presented as a text-only list; users will need to retype an option to respond

① Important

Users can't send attachments to Copilot Studio copilots. If they try to upload a file (including media, such as images), the copilot 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 end-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
Configure the demo website and add the copilot to your live website	Publish your copilot on your live website, or use a demo website to share internally.
Add a copilot to Microsoft Teams	Use Teams to distribute your copilot.
Add a copilot to Facebook	Add your copilot to Facebook Messenger.
Add a copilot to mobile and custom apps (web-based or native)	Add your copilot to mobile or custom native apps (developer coding required).
Add a copilot to Azure Bot Service channels	Add your copilot to Azure Bot Service channels (developer coding required).

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Configure the demo website and add the copilot to your live website

Article • 04/30/2024

ⓘ 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.

When publishing the copilot to the web, you can publish to a prebuilt demo website (which you can use to share the copilot with your teammates and stakeholders). You can also publish to your own live website.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4mGa9?postJsIMsg=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 copilot and share it with your teammates or other stakeholders who want to try out the copilot. It's not intended for production uses, for example, you shouldn't use it directly with customers.

You should publish and use the copilot on your live website for production scenarios, such as a help copilot 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 copilot, see [Microsoft Copilot Studio Overview](#).

A prebuilt demo website is automatically created for you when you publish your copilot. Your teammates and stakeholders can use the demo website to try out the copilot while you're building it. You might want to customize the demo site to guide your teammates' testing efforts. Publish the copilot 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 copilot, 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 copilot 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 copilot.

1. Under the copilot settings, select **Channels**.
2. Select the **Demo website** tile.
3. Under **Welcome message**, enter the message you want your teammates to see.
4. 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.

Demo Website

X

You created a bot. Great job! Let's set up a website to share with team members so they can try your bot.

Welcome message

Introduce your bot and its purpose to your team members.

Try out the chatbot we made!
[Edit this welcome message in the demo website channel configuration.]

*

Conversation starters

Provide some common trigger phrases to help your team members start a conversation with your bot.

"Hello"
"Start over"
"Talk to a person"

*

Share your website

To invite team members to see your bot in action, copy the link below.

Copy

Copy the demo website URL and share it with your teammates.

Add your copilot to your website

When your copilot 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 copilot to your Power Platform admin center.

1. Under the copilot settings, 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.

Custom website

X

This channel allows you to embed an HTML chatbot into your website. Learn about [embedding your bot in a web page](#).

Copy or share the code

Copy the following code snippet and paste it to your HTML website. If you do not have access to the website's code, share the code with the person responsible for your website.

Embed code

```
<!DOCTYPE html> <html> <body> <iframe src="https://powerva.microsoft.com/webchat/bots/XXXX" frameborder="0" style="width: 100%; height: 100%;"> </iframe> </body> </html>
```

Copy

Share to email

3. Provide the snippet to your web developer to add the copilot to your website.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Add a copilot to Power Pages

Article • 05/29/2024

ⓘ 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.

[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 a copilot 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 a Microsoft Copilot Studio bot.

Add a copilot 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 a copilot with design studio, see [Add a copilot to your Power Pages site](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

Add a copilot to Microsoft Teams

Article • 06/05/2024

ⓘ 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.

With your copilot published, you can make your copilot available to users in Microsoft Teams. This way you, your teammates, and your broader organization can interact with it. You need to [publish the copilot at least once](#) before users can interact with the copilot in Microsoft Teams.

You can:

- Customize your copilot's appearance in Microsoft Teams
- Install the copilot for yourself in Microsoft Teams
- Share the copilot's installation link with other users
- Show the copilot in Microsoft Teams app store.
 - Show the copilot to [shared users](#) by adding it to the Microsoft Teams app store **Built by your colleagues** section.
 - Share the copilot with the organization by submitting the copilot for admin approval to be featured in the Microsoft Teams app store **Built by your org** section.
- Add the copilot to a team channel.
- Download the pregenerated Teams app manifest to distribute it within your Microsoft Teams tenant.

When you add the copilot to Microsoft Teams, some of your data, such as copilot content and end-user chat content, is shared with Microsoft Teams. Your data might flow outside of your [organization's compliance and geographic or regional boundaries](#). For more information, see the [Microsoft Teams app privacy policy](#).

Prerequisites

- [Share with users](#) so they can install and chat with the copilot in Microsoft Teams.

- Your organization needs to allow Microsoft Power Platform apps to be added to Microsoft Teams before you or other users can directly install the copilot in Teams or find it in the **Built by your colleagues** 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.
- Copilots that are approved for the **Built by your organization** section (after submission to an admin) are managed separately by admins in the Microsoft Teams admin center's [Manage apps page](#).
- Before installing the copilot for yourself or sharing the copilot with others, you should [publish the copilot at least once](#).

Connect your copilot to the Microsoft Teams channel

After publishing your copilot at least once, you can connect your copilot to the Microsoft Teams channel to allow users to chat with the copilot in Microsoft Teams.

Note

We recommend you [turn on security settings](#) for copilots built for Microsoft Teams or internal employee use so you can ensure the copilot will not be accessible by individuals outside of your organization.

Connect to Microsoft Teams:

1. [Open the configuration panel for the Microsoft Teams channel](#).
2. Select **Turn on Teams**.

Microsoft Teams

X

Nothing can stop a team, and now your bot can help you achieve more together. To open the lines of communication, select **Turn on Teams**. After a quick installation, your users and bot can start chatting. [Learn more.](#)

Note that certain bot content may not appear the same on Microsoft Teams as it was authored in Power Virtual Agents. For details, refer to our article on [supported channel content](#).

Turn on Teams

Cancel

ⓘ Note

If you do not see this option, then the Microsoft Teams channel may already be enabled.

Customize the copilot's appearance in Microsoft Teams

Providing the right description and appearance for the copilot before making it available to other users is important as it educates them on what the copilot's purpose is and its branding, when applicable.

1. Open the configuration panel for the Microsoft Teams channel.
2. Select **Edit details** to change the copilot's icon, color, and descriptions. These options are shown in the Microsoft Teams app store and the app's About tab after the user installs the copilot. [Review the recommendations on Microsoft Teams icon formats.](#)

ⓘ Important

Updates to the copilot's detail will not apply to users who have already installed the copilot from a shared link or the [Built by your colleagues](#) section. They will need to reinstall the copilot to receive the updates.

Updates to the copilot's detail will not apply to copilots that have been approved by an admin. You will need to resubmit the copilot for approval to have the latest detail updated for users.

3. Select **More** to add additional information:

- Developer name
- Website
- Privacy statement
- Terms of use

You should provide this information to keep your user informed about the copilot. See [Privacy statement and terms of use](#) for more details and examples.

4. Select **Save** to return to the screen where you can make your copilot available in Teams. Your updates to the color, icon, and description are now visible.

 **Note**

Icon customization in this section only applies to the copilot avatar for copilots that are [approved by admin](#) and does not apply for users who install the copilot with a direct [installation link](#) or from the Microsoft Teams app store [Built by your colleagues](#) section. You can customize the copilot avatar at [Customize the look and feel of the copilot](#).

Install a copilot as an app in Microsoft Teams

With your copilot published, you can add the copilot to your own version of Microsoft Teams directly from the Copilot Studio app.

 **Important**

New content in a freshly published copilot 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 copilot.

You should add your copilot to your own version of Teams before sharing it with others.

1. [Open the configuration panel for the Microsoft Teams channel](#).

2. Select **Open copilot** to have the installation prompt launch in Microsoft Teams.

Select **Add** to add it as an app in Teams. This only installs the copilot for you.

Share a link so others can install the copilot

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 copilot in Microsoft Teams.

 **Important**

Only users who can access the copilot can use the installation link to install the copilot. You can [share the copilot to give users access to chat](#) with the copilot.

Show the copilot in the Microsoft Teams app store

Once you publish your copilot, you can show it in the Microsoft Teams app store so users can find and install it.

Show to shared users

You can share your copilot by adding it to the Microsoft Teams app store, **Built for your org** > **Built by your colleagues** section. Only [shared users](#) can find the copilot there.

 **Important**

Only shared users can find and install the copilot in the Microsoft Teams app store **Built by your colleagues** section. The copilot does not show for everyone in the organization even if it is configured to allow everyone to use the copilot. To [show the copilot to the organization](#), submit the copilot for an admin's approval to show it in the Microsoft Teams app store **Built by your org** section.

The copilot can only be shown in the Microsoft Teams app store **Built by your colleagues** section when the total number of shared users does not exceed the

tenant's [Teams app store discovery policy](#). To show the copilot beyond your organization's policy limit, partner with your admin to [show the copilot to the organization](#).

The screenshot shows the Microsoft Teams App Catalog interface. On the left, there's a sidebar with categories like Apps, Home, Featured, Categories, and App features. The main area displays a grid of apps. A red box highlights the 'Built for your org' section, which contains several Power Virtual Agents bots. Another red box highlights the 'Built by your colleagues' section, which also contains Power Virtual Agents bots. At the bottom right of the catalog, there's a note: 'Placeholder text'.

1. Open the configuration panel for the Microsoft Teams channel.
2. Select Availability options.
3. Select Show to my teammates and shared users and review who has access to the copilot.
4. Confirm the Visible in Build by your colleagues checkbox is selected and select Share.
5. This automatically adds the copilot to the Built by your colleagues section, and shows a green label Added to Teams in the availability options page.

ⓘ Note

If your app doesn't appear in the Built by your colleagues section, 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 web app of Teams).

Remove the copilot from the Built by your colleagues section

You can remove the copilot from showing in the **Built by your colleagues** section.

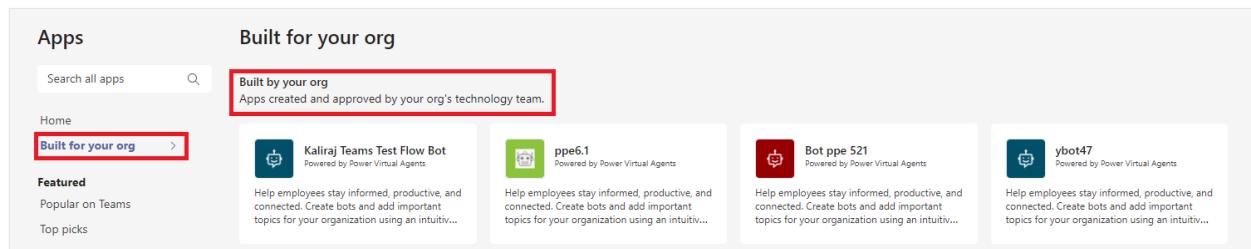
Removing the copilot here doesn't affect users with the copilot installed.

1. [Open the configuration panel for the Microsoft Teams channel.](#)
2. Select **Availability options**.
3. Select **Show to my teammates and shared users** and remove the **Visible in Built by your colleagues** checkbox selection.
4. Select **Share**. The copilot no longer appears in the Microsoft Teams app store **Built by your colleagues** section.

Show to the organization

You can share the copilot with your organization by submitting your copilot for admin approval to be featured in the **Built for your org > Built by your colleagues** section.

Featuring your app in **Built by your colleagues** is a great way to broadly distribute your copilot by partnering with your admin to roll out the copilot to everyone in the organization or a subset of users.



Admins can also [preinstall and pin the copilot](#) for users in the organization so they can access it directly from their app bar in Microsoft Teams.

Before submitting the copilot to your organization, make sure:

- Your copilot is ready for release and in compliance with company standards, rules, and policies.
- You coordinate with other copilot managers. Once you submit the copilot, a different user can't resubmit it by until an admin approves or rejects the previous submission.
- You remove the copilot, if applicable, from the **Built by your colleagues** section. Otherwise if you submit your copilot for admin approval your copilot could end up showing up in two places. See the instructions on [removing your copilot](#) for details.

Note

Once a copilot is approved by an admin, the copilot availability in the **Built by your org** section will be completely managed by your admin. Only an admin can remove the copilot from the **Built by your org** section.

Submit for the Built by your org section

1. [Open the configuration panel for the Microsoft Teams channel](#).
2. Select **Availability options**.
3. Confirm the copilot isn't shown to teammates or shared users. [Remove it](#) if it says **Added to Teams**.
4. Select **Show to everyone in my org**.
5. Review the requirements for submission, and select **Submit for admin approval**.

Note

Make sure you review the content and it's ready for an admin to review. You need to submit new requests to the admin to [update the icon and descriptions](#) for copilots that were previously approved.

Once approved, all new and existing users of the copilot in Teams receive the update.

This only applies to the copilot details. You can continue to make content updates to your copilot by publishing the latest content. You don't need to re-submit to an admin.

A final confirmation prompt appears.

6. Select **Yes** to continue. The copilot is sent to your admin [for approval](#), and you can see a note indicating the status of your submission.

Note

After submitting the copilot for admin approval, you should not change the copilot's access setting to be less than everyone in your organization.

Setting to less than everyone results in users in the organization unable to chat with the copilot after installing it from the Microsoft Teams app store **Built by your org** section.

Work with your admin on the [Microsoft Teams app policy](#) on who can find and install the copilot.

7. Wait for your [admin's approval](#). You can check the latest status by selecting **Refresh**, or opening the **Make the copilot available to others** window from the **Publish** page.
8. After your admin approves the copilot, the status is updated to **Approved**. You can directly open the copilot and share the approved version's link with other users.
9. Optionally, partner with your admin to prepin or install the app for users in the [Microsoft Teams app policy](#) so they can find the copilot directly in their Microsoft Teams left app rail.

Note

If your app doesn't appear in the **Built by 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).

Add copilot to a team

In Microsoft Teams, you can [add copilot to teams](#). When you add the copilots, team members can `@mention` the copilot in any team channels, and all teammates will see the response from the copilot. This feature is useful for scenarios where:

- Your copilot supports collaboration, like sharing the latest sales report insight with the rest of the Sales team.
- You want to allow every team member to engage with the copilot in a team channel or via private chat without needing to manually install the copilot themselves.

Allow users to add copilot to a team and Enable for Teams group and meeting chats

This two option controls if users can add the copilot to any team that they're part of, or to grout and meeting chats when adding the copilot in Microsoft Teams.

When you select these two options, the copilot gets the conversation history from the team channel, group chat, or meeting chat to be able to provide the best answer in context of the conversation.

After you enable at least one option, any user, who installs the copilot from the [Teams app store](#) or the [installation link](#), can see one of the following:

- the option to add the copilot to a team channel
- the option to add the copilot to group and meeting chats in Microsoft Teams

Team members can also [remove the copilot from a team](#).

Important

The copilot is configured to allow everyone in the organization to chat with it. This ensures all team members have permission to chat with the copilot in team channels. It's highly recommended to not change the security setting of the copilot after enabling this capability. Otherwise users will see permission errors when interacting with the copilot in team channels.

To allow users to add a copilot 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 copilot to a team**.
4. Select **Save**.

Microsoft Teams caches information, so it can take a while before users see the option to add the copilot to a team or group chat. Users can refresh the browser or sign in and out of the Microsoft Teams desktop client to reflect changes immediately.

If you previously [submitted the copilot for admin approval](#), you need to resubmit after enabling this option. The admin needs to approve these changes before users can add the copilot to their teams from the Microsoft Teams app store in the **Built by your org** section.

Disallow users from adding copilot to a team, or group and meeting chats

Disabling this option prevents users from adding a copilot to a new team, group, or meeting chat. If you added a copilot to a conversation already, disabling this option doesn't remove it. You need to manually [remove](#) the copilot.

To disallow users from adding a copilot 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 copilot to a team** checkbox.
4. Select **Save**.

Configure private greeting behavior

Private greetings from a copilot are useful for when you want team members to have one-on-one conversations with the copilot without needing to manually install the copilot themselves.

Note

Private greeting configuration isn't supported for copilots using the [upgraded unified authoring canvas](#).

When enabled, your copilot automatically starts a conversation with team members. To customize the copilot's greeting message, [modify the Greeting system topic](#).

Consider how relevant the copilot is to team members. Make sure the copilot'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 a copilot 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 copilot's Teams app manifest (advanced)

You can download the copilot'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 the app manifest into Microsoft 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**.
4. The manifest is generated as a .zip file, and is saved according to your browser's save settings.

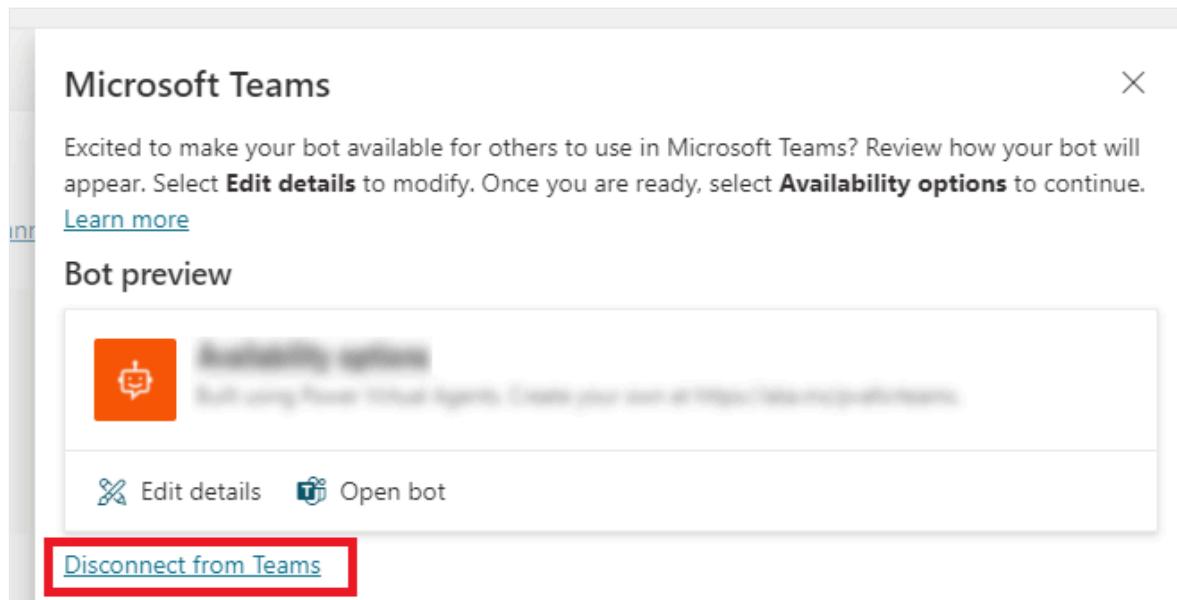
Disconnect your copilot from Teams

You can take your copilot offline so users can no longer chat with the copilot in Teams. When your copilot is offline, it doesn't respond to anyone who tries to chat with it, and disrupts anyone who might be using it. You should thoroughly consider the need to disconnect your copilot before you do so.

Copilots no longer show in the Microsoft Teams app store **Built by your colleague** section and can't be installed from the shared link. Copilots added by users in Microsoft Teams continue to exist until the user uninstalls the copilot.

For copilots that are approved by an admin, the copilots continue to appear in the Microsoft Teams app store **Built by your org** section. You need to contact your admin to remove the copilot.

1. [Open the configuration panel for the Microsoft Teams channel](#).
2. Select **Disconnect from Teams**.



3. Review the information about disconnecting your copilot, and select **Disconnect** to continue.

You can always reconnect your copilot after disconnecting by selecting **Turn on Teams** in the **Make the copilot available to others** window on the **Publish** page.

Open the configuration panel for the Microsoft Teams channel

1. Open your copilot 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 copilots using the [upgraded unified authoring canvas](#).
- Currently, only the **Personal** and **Team** scopes are supported for Microsoft Teams. **Group chat** isn't supported.
- [Setting the copilot avatar icon](#) on the details page doesn't show up for copilots approved by an admin. Copilots approved by an admin use the app icon as the copilot avatar.
- Teams has [rate limiting](#) on copilots. Depending on your business needs, you might want to keep your copilot content concise for copilots that are added to Microsoft Teams.

- Currently, if your copilot supports [end-user authentication](#), the user isn't able to explicitly sign out. This fails the Microsoft Teams AppSource certification if you're publishing your copilot in the Seller Dashboard. This doesn't apply to personal or tenant usage of the copilot. Learn more at [Publish your Microsoft Teams app](#) and [AppSource Validation Policy](#).
 - Disconnecting your copilot from Teams might take a few minutes before the copilot becomes unreachable by users on Microsoft Teams.
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Create a privacy statement and terms of use in Microsoft Teams

Article • 04/29/2024

ⓘ 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.

When you want to publish your bot so other Microsoft Teams users can access it, you need to provide the following information about your bot:

- A link to a compliant privacy statement
- A link to the terms of use (TOU) for the bot
- 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 Microsoft Copilot Studio chatbot, and suggestions on what you should enter instead:

[\[+\] Expand table](#)

Requirement	Default value	Suggestions
Privacy statement	Empty privacy statement and terms of use	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	Empty privacy statement and terms of use	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 Teams privacy policy guidelines .

Requirement	Default value	Suggestions
Developer website	Link to the Microsoft Copilot Studio overview page	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 bot.

Empty privacy statement and terms of use

Important

This app is not developed by Microsoft. If you are a user of a bot and see this page, it means the bot developer did not provide a terms of use and privacy statement for the bot.

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 chatbot users are redirected here.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Add a copilot to Facebook

Article • 06/05/2024

ⓘ 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.

You can add your copilot to Facebook Messenger to use the Facebook experience to engage with your end users.

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 copilot to a Facebook page, some of your data such as copilot content and end-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 copilot to Facebook Messenger

You need a Facebook app, and an associated developer account, to connect your Copilot Studio copilot to Facebook Messenger on your Facebook pages.

To connect your copilot 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

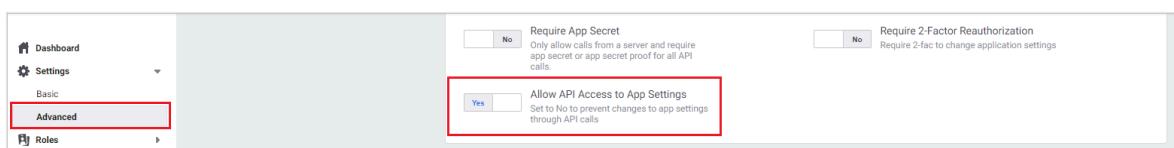
Retrieve Facebook app information

1. Sign in to the Facebook app that you want to add your copilot 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](#).



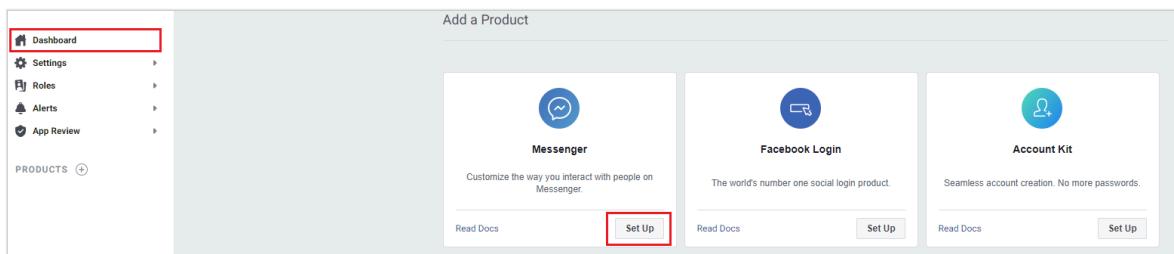
Enable API access for your Facebook app

1. Sign in to the Facebook app that you want to add your copilot 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.



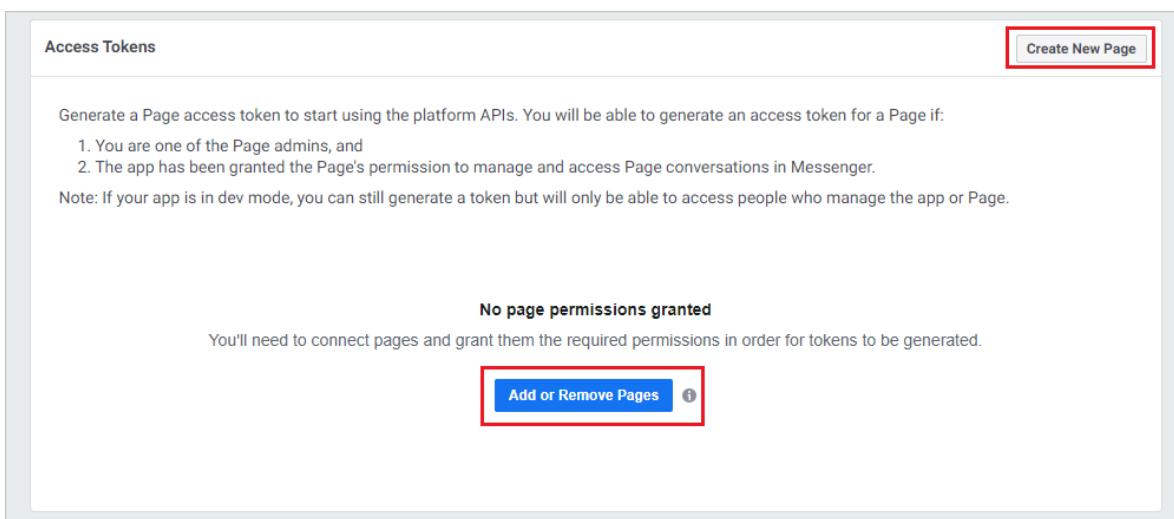
Add Facebook Messenger to your app

1. Sign in to the Facebook app that you want to add your copilot to at [Facebook for Developers](#).
2. Go to the **Dashboard**. Under the **Add a Product** section, select **Set Up** on the **Messenger** tile.

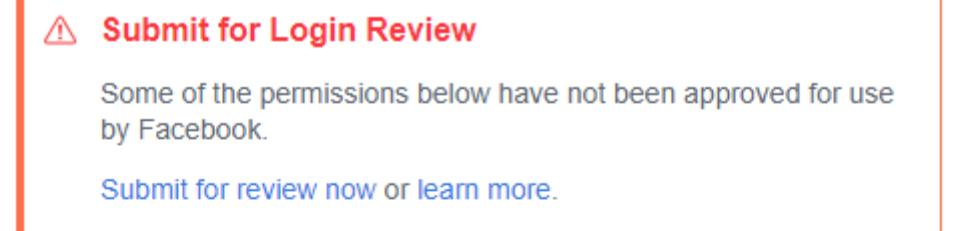


Configure Facebook pages

1. Sign in to the Facebook App that you want to add your copilot 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 copilot 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**.



What is [REDACTED] allowed to do?

(i) [REDACTED] may not work properly if you turn off these options.

Manage and access Page conversations in Messenger YES

New app

[Cancel](#) [Back](#) [Done](#)

[Help Center](#)

5. Copy the *Page ID* and *Token* for each of the pages that you want to add the copilot 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](#) [\(i\)](#)

Configure the Facebook channel in Copilot Studio

1. Open your copilot 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.

The screenshot shows the 'Facebook' configuration window. At the top, there's a note about setting up the channel for Facebook users to chat with your bot, with links to 'Learn how' and 'supported channel content'. Below this, a note about data sharing and compliance is present. The main section is titled 'Facebook credentials' with a sub-note about getting information from Facebook. A red box highlights the input fields for 'Facebook app ID *' (containing 'Your Facebook app ID'), 'Facebook app secret *' (containing 'Your Facebook app secret'), 'Page ID *' (containing 'Your Facebook page ID'), and 'Page access token *' (containing 'Your Facebook page access token'). There's also a 'Delete' icon next to the token field. Below these fields is a 'Add page' button. Further down, there's a 'Connection info for Facebook' section with 'Callback URL' and 'Verify token' fields, each with a '(Available after channel is added)' note and a 'Copy' button. At the bottom right are 'Add' and 'Cancel' buttons.

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

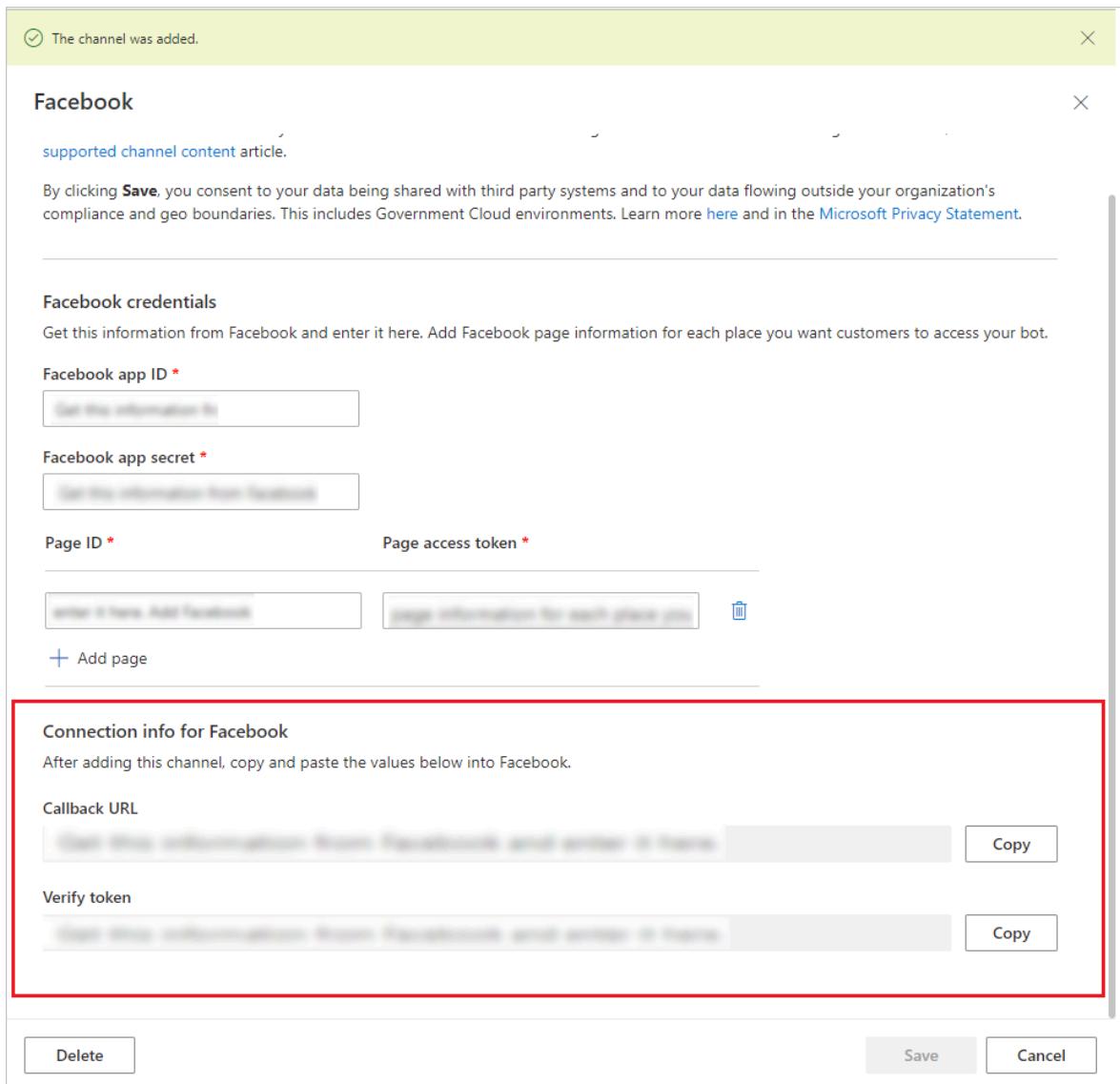
After adding this channel, copy and paste the values below into Facebook.

Callback URL

Add

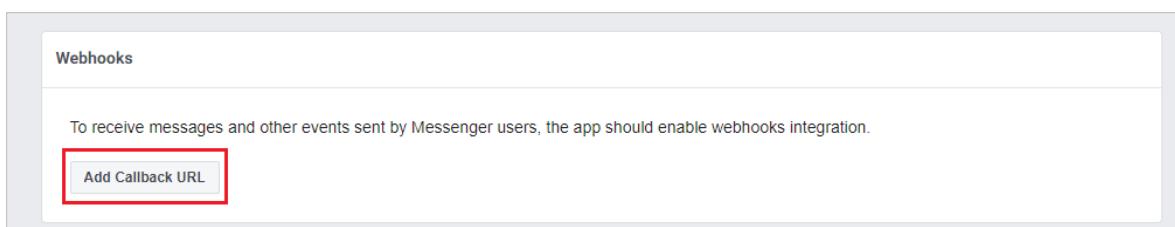
Cancel

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](#).

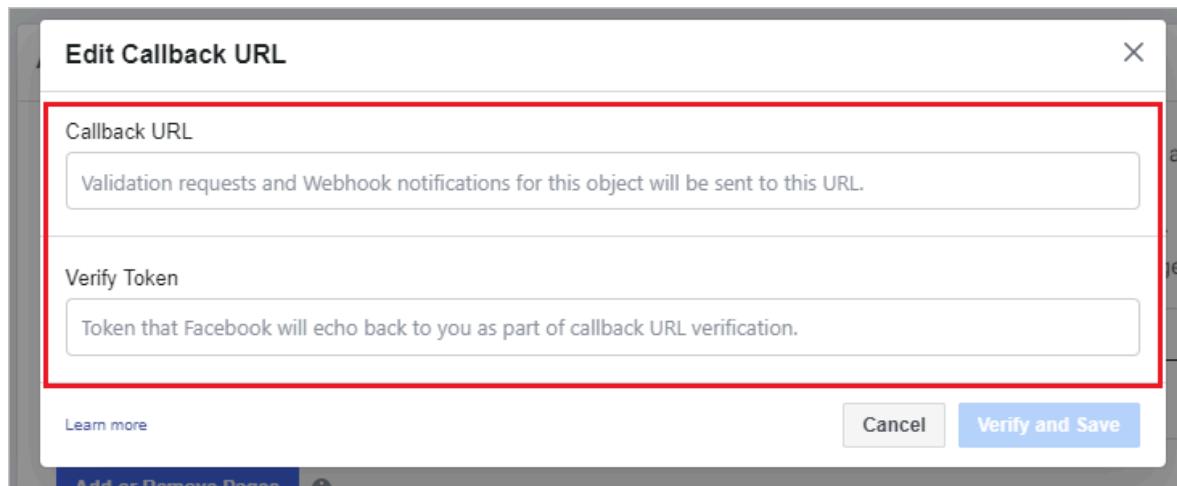


Connect your Facebook app to Copilot Studio

1. Sign in to the Facebook app that you want to add your copilot 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 copilot to.

The screenshot shows the "Webhooks" configuration page. It features a table where each row represents a page. The columns are "Pages ↑" and "Webhooks". Under "Webhooks", it says "0 Fields" and "—". To the right of each row is a blue "Add Subscriptions" button, which is highlighted with a red border for both rows.

Pages ↑	Webhooks	
D [REDACTED]	0 Fields —	Add Subscriptions
N [REDACTED]	0 Fields —	Add Subscriptions

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 copilot 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 copilot 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 copilot is published. The status appears in the **Pages** settings.

Remove your copilot from Facebook Messenger

When you do not want the copilot to be reachable in Facebook Messenger, you can remove the copilot from Facebook.

1. Open your copilot 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 copilot content for existing conversations on Facebook

To prevent disruptions during a chat between a user and the copilot, existing conversations are not updated to the latest copilot 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 copilot have the latest published copilot content.

Known limitations

- You need to create a new copilot if the Customer Satisfaction (CSAT) survey shows up as a non-interactive card image for a copilot created during public preview to access the latest CSAT content.
- It may take a few minutes before the copilot 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 copilot is removed fully and becomes unreachable on Facebook Messenger.
- After removing a Facebook page, it may take a few minutes before the copilot becomes unreachable by visitors to the removed page through Facebook Messenger.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Add copilot to mobile and custom apps

Article • 04/15/2024

ⓘ 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.

You can connect your copilot to a custom app so that the app's users can interact with the copilot 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 copilot 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 copilot into your app. Both procedures are described in this article.

Prerequisites

- [.NET Core SDK](#) version 2.1.
- Nuget package [Microsoft.Bot.Connector.DirectLine](#).
- A copilot created in Copilot Studio that you want to connect to your app.

Connect your copilot to a web-based app

1. In Copilot Studio, in the navigation menu under **Settings**, 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.

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

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.

[Copy](#)

Connect your copilot 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 [Connect your bot 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

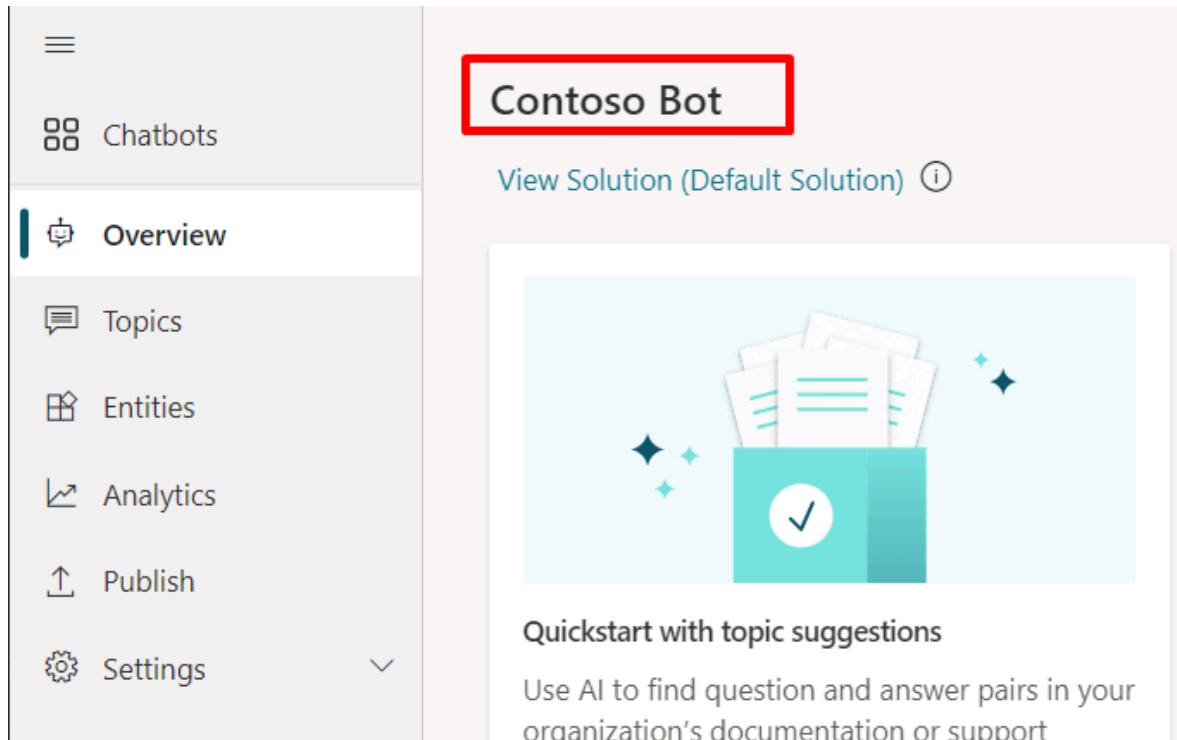
The instructions in this document reference the following source material:

- [Bot Framework Direct Line API](#)
- [Direct Line Authentication](#)
- [Context variables available upon handoff](#)
- [Microsoft Bot Framework Activity ↗](#)

Retrieve your Copilot Studio copilot parameters

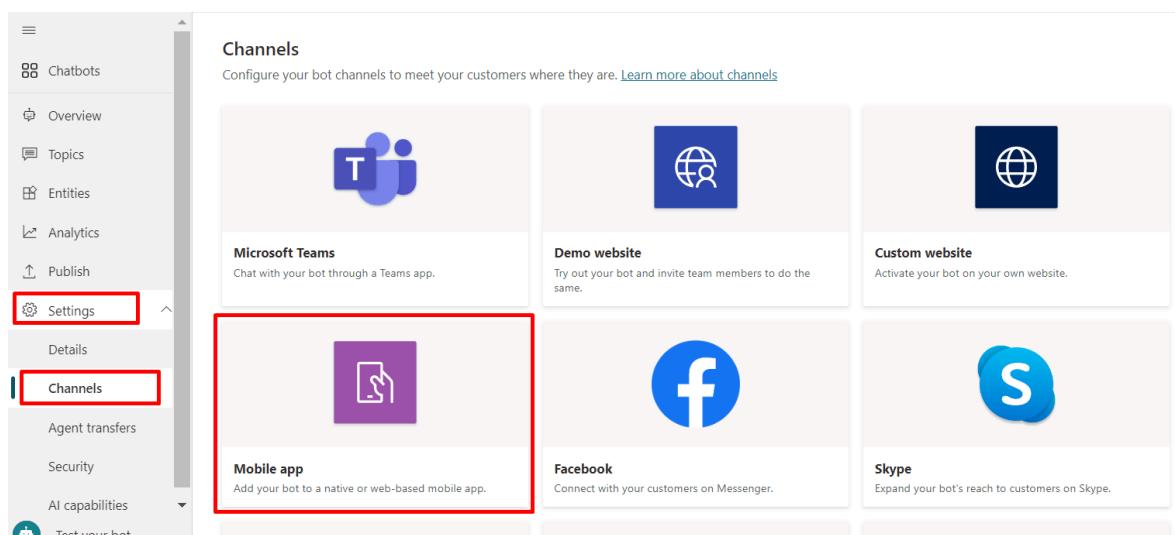
To connect to the copilot you built, you need to retrieve your copilot's name and token endpoint to identify it.

1. Copy your copilot's name in Copilot Studio.



2. In the navigation menu under **Settings**, select **Channels**.

3. Select **Mobile app**.



4. Next to **Token Endpoint**, select **Copy**. You need this endpoint for the [Get Direct Line token](#) step.

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

https://[REDACTED].com

Copy

Get Direct Line token

To start a conversation with your copilot, 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 copilot. Example:

```
Json
{
  "token": "RCurR_XV9ZA.cwA.BKA.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfiCpg4Fv0y8qqb0F5x8qb0F5xn",
  "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 copilot.

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.iaJrC8xpy8qb0F5xnR2vtCX7CZj0LdjAPGfiCpg4Fv0y8qb0F5xPGfiCpg4Fv0y8qqb0F5x8qb0F5xn",
  "expires_in": 3600,
  "conversationId": "abc123"
}
```

Use Direct Line to communicate with the copilot

After retrieving the Direct Line token, you are ready to have a conversation with your Copilot Studio copilot 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 copilot.

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 copilot's response using the same `token` and `conversationId`. The retrieved Direct Line response activities contains both the user's and copilot's messages. You can filter response activities by your copilot's name to get only the copilot'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 may need to add code to refresh the Direct Line token if your application has a lengthy conversation with the copilot. 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 copilot

After starting a conversation with the copilot, 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 copilot to send the Greeting system topic automatically when a user starts a conversation, you can send an activity with `Type=event` and `Name=startConversation`.

Add a Copilot Studio copilot to Azure Bot Service channels

Article • 04/15/2024

ⓘ 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.

You can connect your copilot to existing [Azure Bot Service channels](#). This can be helpful if you want to connect your copilot to end users on Azure Bot Service channels.

Adding your copilot 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 copilot 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 [Add copilot to mobile and 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](#).
- A copilot created in Copilot Studio that you want to connect to an Azure Bot Service channel.
- [Connect your copilot to mobile and custom apps](#).

Code samples

Code snippets used in this document are from [relay bot sample code](#).

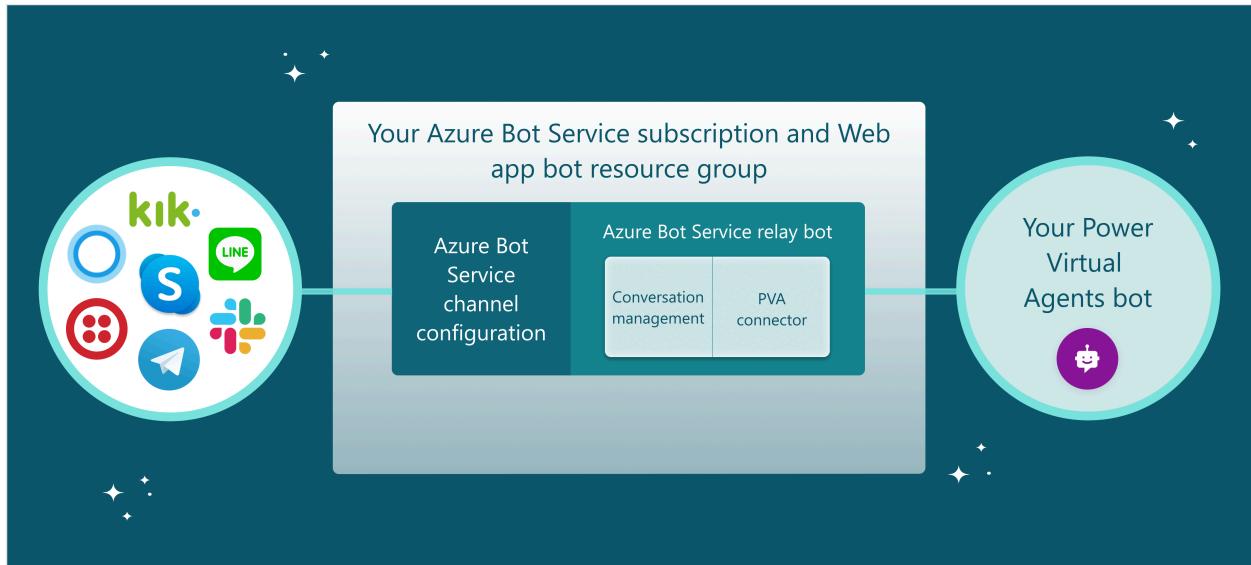
References

The instructions in this document reference the following:

- [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 copilot and Azure Bot Service channels.



The [relay bot sample code](#) is a good starting point if you do not have an existing Azure Bot Service bot. It is 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

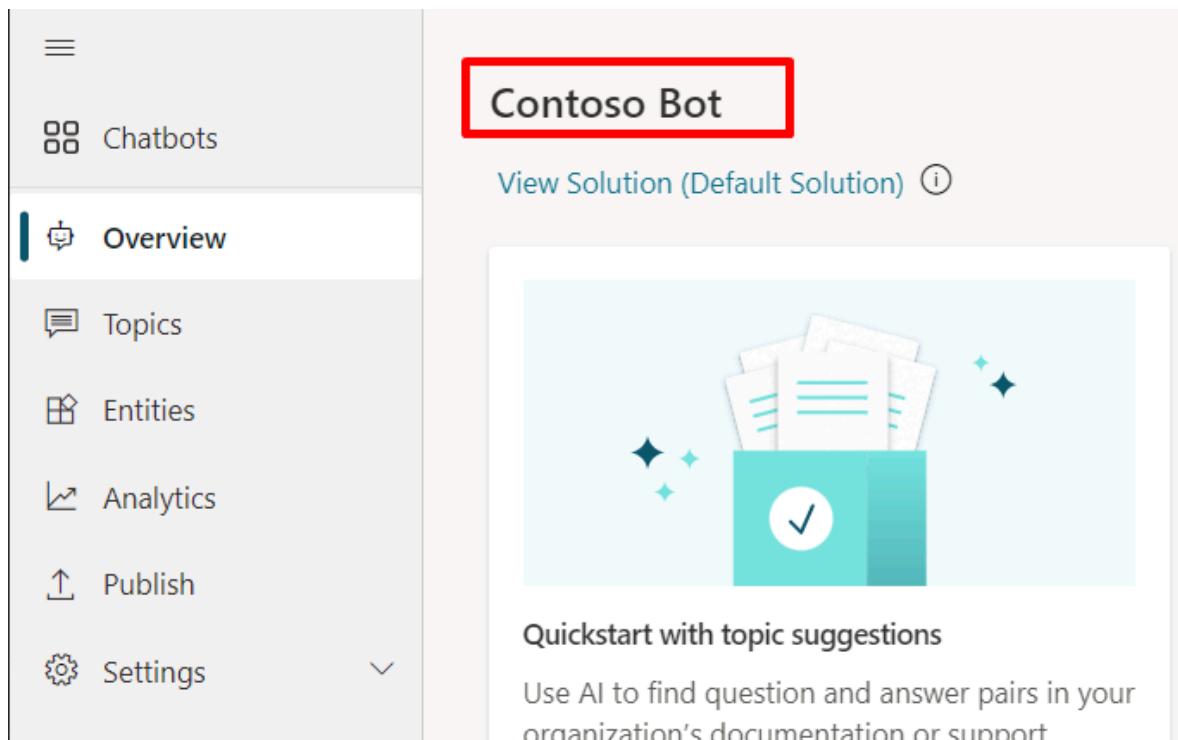
be used as a starting point and not intended to be used in production directly. You will need to add code and optimization to match your business needs.

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 copilot parameters

To connect to the copilot you have built with Copilot Studio, you'll need to retrieve your copilot's name and token endpoint.

1. Copy your copilot's name in Copilot Studio.



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 Azure Bot Service portal's left sidebar with 'Settings' and 'Channels' selected. The main area displays various channel integration cards: Mobile app, Facebook, Skype, Cortana, Slack, Twilio, Line, and Kik. The 'Slack' card is highlighted with a red box.

- To copy and save the **Token Endpoint** value, select **Copy**. You'll need your endpoint to connect your copilot to the Azure Bot Service channel.

The screenshot shows the Slack channel configuration page. It includes a description of Slack, a 'Get connected' section with instructions for developers, and a 'Token Endpoint' field containing a URL. A 'Copy' button is positioned next to the URL field, which is also highlighted with a red box.

Manage conversation sessions with your Copilot Studio copilot

There can be multiple conversations between the Azure Bot Service channels and the Direct Line connection with your Copilot Studio copilot.

Your Azure Bot Service bot will need to map and relay the conversation from the Azure Bot Service channel to the Direct Line conversation with the Copilot Studio copilot 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 copilot conversation. Refer to [Get Direct Line token](#) and [Use Direct Line to communicate with the copilot](#) 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 copilot conversations. A Copilot Studio copilot 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 copilot conversation to support those is defined as follows:

C#

```
/// <summary>  
/// Data model class for Copilot Studio copilot conversation  
/// </summary>  
public class PowerVirtualAgentsConversation  
{  
    public string ConversationId { get; set; } // The Copilot Studio  
    copilot 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 copilot  
}
```

3. When a new Copilot Studio copilot conversation starts, add a key value pair

```
(external_Azure_Bot_Service_channel_conversationID,  
PowerVirtualAgentsConversation)
```

C#

```
// After new Copilot Studio copilot 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 copilot, 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 copilot and get response
protected override async Task
OnMessageActivityAsync(ITurnContext<IMessageActivity> turnContext,
 CancellationToken cancellationToken)
{
    // Retrieve copilot conversation from mapping table
    // If not exists for the given external conversation ID, start a
    new Copilot Studio copilot 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 copilot](#) for how to get the Copilot Studio copilot's response. When the Copilot Studio copilot's response is

received, refer to [Parse conversation payload from the copilot](#) 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 have deployed to. View the specific instructions for each channel at [Azure Bot Service Channels](#).

Create copilots for customer engagement and handoff

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.

You can create copilots to interact with your customers and integrate with customer service and customer engagement hubs. Such copilots provide your customers with self-help based on generative AI. The copilot 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 copilot can transfer the customer to a live agent with integrated handoff to the customer engagement hub that you already use, including Dynamics 365 Customer Service, ServiceNow, Salesforce, LivePerson, and Genesys.

First, you need to create a copilot with generative AI, and tell it where it should look for information when interacting with your customer. The copilot will use that information to determine the best answer or response to the user's question and return the information in a conversational tone.

Once you've created and tested your copilot, you can configure integration between the copilot and your engagement hub, including handoff to a live agent.

Create a copilot and connect it to your knowledge source

If you haven't created copilots in Copilot Studio before, follow the steps in the [Quickstart guide for building copilots with generative AI](#).

If you already have a copilot that you've built with Copilot Studio, you can go straight to the instructions for [adding a generative answers node](#) to connect your copilot 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

[+] Expand table

Name	Source	Description	Number of Inputs	Authentication
Bing Search	External	Searches the query input on Bing; returning results only from provided websites	4 public URLs (for example, <i>microsoft.com</i>)	None
Bing Custom Search	External	Query input filtered based on a website configuration external to Microsoft Copilot Studio	Each search ID can use more than 4 URLs (Bing Custom Search also provides other functionality) but you can only connect to one search ID	None
Azure OpenAI on your data	Internal		Defined by your Azure OpenAI Service connection	Copilot user's Microsoft Entra ID authentication
Documents	Internal	Searches documents uploaded to Dataverse, returning results from the document contents	Limited by Dataverse file storage allocation	None
SharePoint	Internal	Connects to a SharePoint URL, uses GraphSearch to return results	4 URLs	Copilot user's Microsoft Entra ID authentication
OneDrive for Business	Internal	Connects to a OneDrive for Business URL, uses	4 URLs	Copilot user's Microsoft Entra ID authentication

Name	Source	Description	Number of Inputs	Authentication
		GraphSearch to return results		
Custom data	Internal	Uses a JSON code block to define the URLs and content to use	One variable, populated with the JSON results to be summarized	Dependent on source

Copilot user authentication for information sources means that when a specific user asks a question of the copilot, the copilot will only surface content that that specific user can access.

Configure handoff to customer engagement hubs

- [Dynamics 365 Customer Service](#)
- [ServiceNow](#)
- [Salesforce](#)
- [LivePerson](#)
- [Generic customer engagement hub](#)

Configuring Copilot behavior

The following fields are available under **Copilot details** on the side navigation pane and let you customize your copilot for tone and behavior.

[] Expand table

Field	Description
Greeting message	What the copilot says when greeted.
Start over message	What the copilot says when the user asks to start the conversation over.
Conversation start message	What the copilot says when opened.
Escalate link	A link for the user to talk to an agent.
No match message	What the copilot says when it doesn't have topics for the user's input.

Field	Description
Multiple topics matched	What the copilot says to ask the user to clarify their meaning.
Reset conversation message	What the copilot says after starting the conversation over.

Configure handoff to any generic engagement hub

Article • 04/15/2024

ⓘ 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 hand over copilot conversations seamlessly and contextually to a human agent through an engagement hub.

With some custom development, you can configure your copilot to hand off conversations to any engagement hub. This guide describes how you can do this.

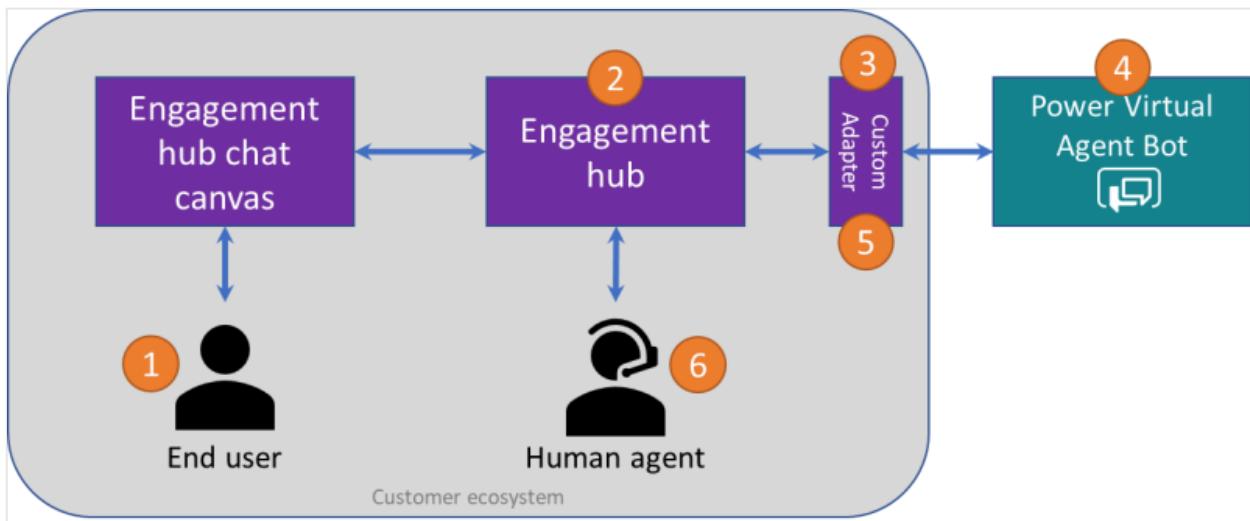
Prerequisites

- You need to have [built a copilot using Copilot Studio](#).
- You need an engagement hub that can interact programmatically using APIs or SDK.

ⓘ 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 who are looking to integrate third-party engagement hubs with Copilot Studio.

Overview



A full handoff to an engagement hub follows this pattern:

1. An end user interacts with the engagement hub's chat canvas.
2. The engagement hub routes the incoming chat through built-in chat routing capabilities to a copilot.
3. A custom adapter relays the incoming chat messages from the engagement hub to a Copilot Studio copilot.
4. Once the end user [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 end user'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 end users, copilots, and human agents.

Most popular agent engagement hubs provide SDKs or document their APIs publicly, enabling you to build such adapters.

While it is outside the scope of this document to cover what a custom adapter could contain, 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 copilot conversations to any generic engagement hub.

Sample handoff message payload

Handoff is currently only supported over Direct Line. Learn more about [interacting with a copilot over Direct Line](#). Upon handoff, an event activity called `handoff.initiate` is raised and sent to the adapter.

You can see a [full sample handoff message activity](#) on our GitHub site ↗.

Extract context from handoff message

To use [conversational context](#), you must parse the `handoff.initiate` event activity. The following snippet of code parses the `handoff.initiate` event activity and extracts the conversational context. See the [full code sample on GitHub](#) ↗.

C#

```
public void InitiateHandoff(string botresponseJson)
{
    BotResponse response = JsonConvert.DeserializeObject<BotResponse>(botresponseJson);

    // Look for Handoff Initiate Activity. This indicates that conversation
    // needs to be handed off to agent
    Activity handoffInitiateActivity =
        response.Activities.ToList().FirstOrDefault(
            item => string.Equals(item.Type, ActivityTypes.Event,
System.StringComparison.OrdinalIgnoreCase)
            && string.Equals(item.Name, HandoffInitiateActivityName,
System.StringComparison.OrdinalIgnoreCase));

    if (handoffInitiateActivity != null)
    {
        // Read transcript from attachment
        if (handoffInitiateActivity.Attachments?.Any() == true)
        {
            Attachment transcriptAttachment =
                handoffInitiateActivity.Attachments.FirstOrDefault(a =>
string.Equals(a.Name.ToLowerInvariant(), TranscriptAttachmentName,
System.StringComparison.OrdinalIgnoreCase));
            if (transcriptAttachment != null)
            {
                Transcript transcript =

```

```
JsonConvert.DeserializeObject<Transcript>
(transcriptAttachment.Content.ToString());
    }
}

// Read handoff context
HandoffContext context =
JsonConvert.DeserializeObject<HandoffContext>
(handoffInitiateActivity.Value.ToString());

// Connect to Agent Hub
// <YOUR CUSTOM ADAPTER CODE GOES HERE>
}
}
```

Configure handoff to Dynamics 365 Customer Service

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.

When your customers need to speak with a human agent, your copilot can seamlessly hand off the conversation. With the omnichannel capabilities of the [Chat Add-in for Dynamics 365 Customer Service](#), you can connect both voice (in Classic chatbots) and text (messaging) conversations.

ⓘ Note

Connecting to Dynamics 365 Customer Service Omnichannel for **voice** capabilities is available to Microsoft Copilot Studio classic chatbots only at this moment. This feature is coming soon to all other copilots.

When your copilot hands off a conversation, it shares the full history of the conversation and all variables collected in the interaction. Your Omnichannel for 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 copilot conversations, see [Hand off to a live agent](#).

You can also enable single sign-on (SSO) to allow copilots to sign users in if they're logged in to the page where the copilot is deployed. For more information, see [Configure single sign-on](#) and [Pass authentication token to copilot during single sign-on in live chat](#).

Web app

Prerequisites

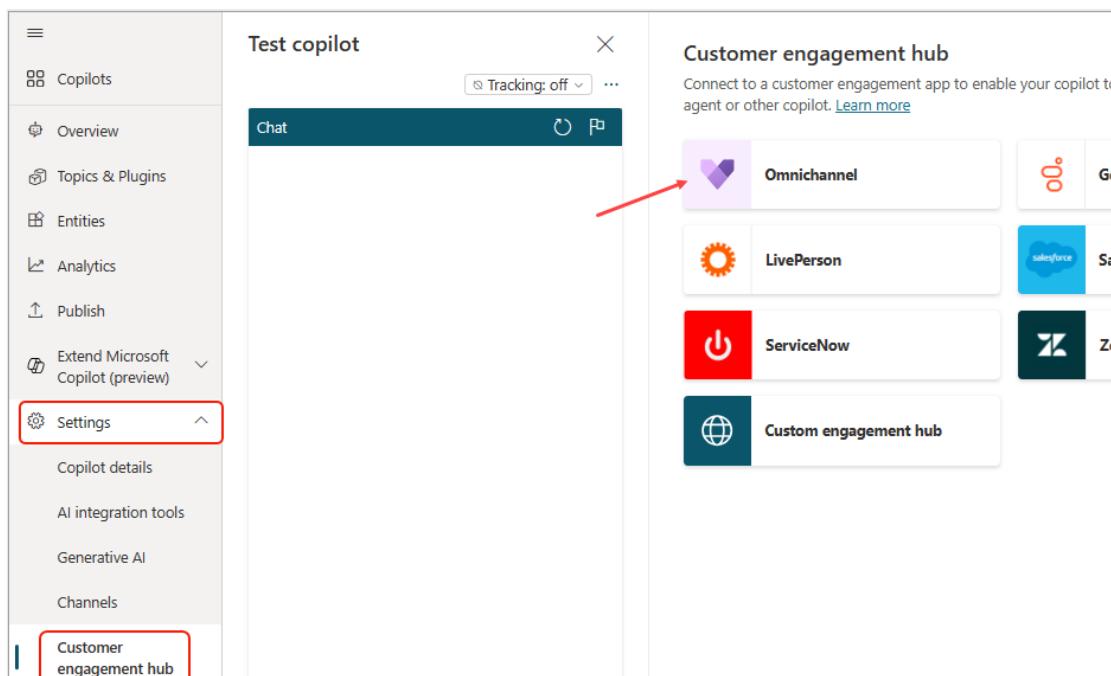
- Sign in with an account that has at least OC_Admin and Copilot Author roles.
- Have a [product license for Microsoft Copilot Studio](#) and a [product license for the Chat Add-in for Dynamics 365 Customer service](#).
- Your copilot and Omnichannel for Customer Service must be in the same environment.
- For end-to-end capabilities to work as expected, you must [publish](#) your copilot.

⚠ Warning

Microsoft Copilot Studio copilots with names that are longer than 30 characters will fail to connect when you follow the instructions in this article. Ensure your copilot's name contains less than 30 characters before you proceed.

Connect your copilot to Omnichannel for Customer Service

1. In Microsoft Copilot Studio, edit your copilot. In the navigation menu, select **Settings, Customer engagement hub**, and then select the **Omnichannel** tile.



2. Select Connect.

- Note that the OmniChannel installation must be in the same environment as the copilot.
- If you're using Application Lifecycle Management (ALM), you may see a message that we can't determine if Omnichannel for Customer Service integration is enabled for the environment. For more information, see [Bots with ALM](#).

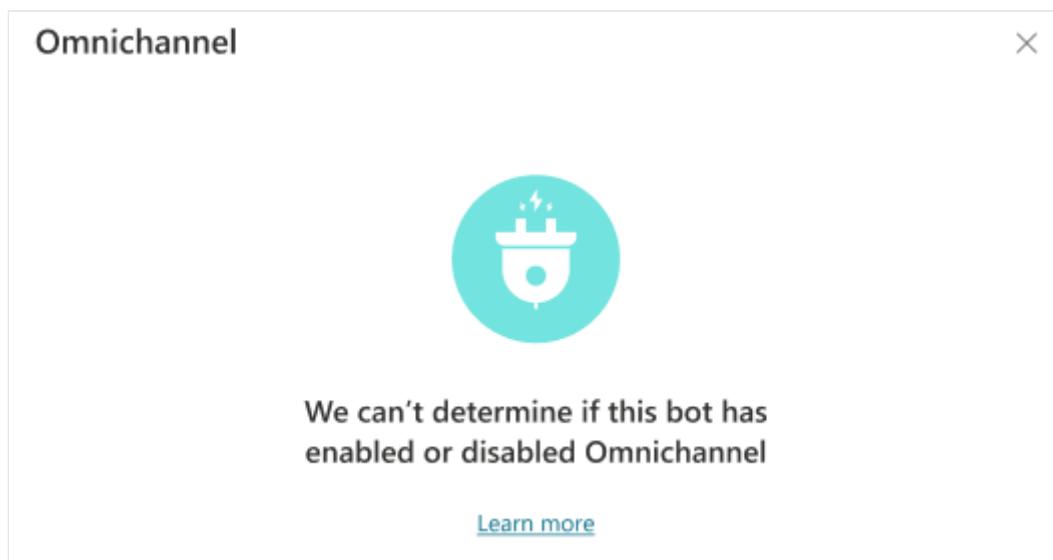
3. Click **View in Omnichannel** to [continue configuring the bot connection in Omnichannel for Customer Service](#).

i Important

To test the bot on your custom website, you must use the embed code that's specified in the chat widget you set up in Omnichannel for Customer Service. If you use the embed code from Microsoft Copilot Studio, handoff won't work. For more information, see [Embed chat widget in your website or portal](#).

Bots with Application Lifecycle Management

Suppose you've [set up Application Lifecycle Management \(ALM\) for your bots](#), and are exporting and importing bots between [development \(unmanaged\)](#) and [test or production \(managed\) environments](#). In that case, you might see a message that we can't determine if Omnichannel for Customer Service integration is enabled for the environment.

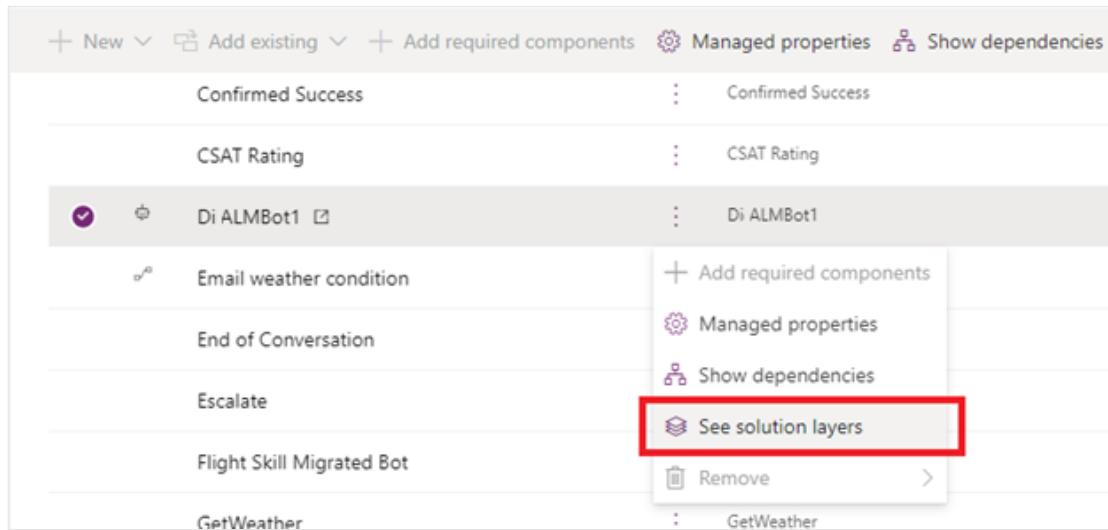


If the bot you exported has omnichannel capabilities enabled, you can ignore this message. The bot will still work properly. The message will disappear after you

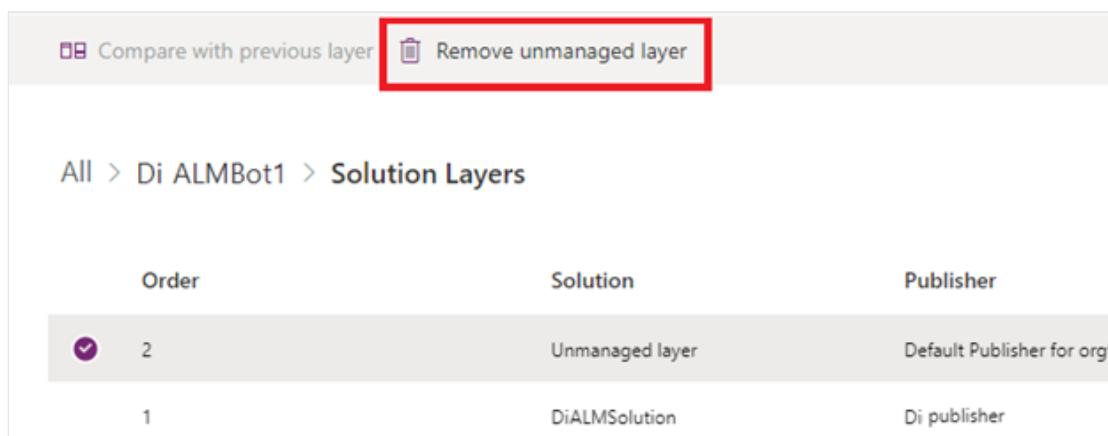
export the latest version of your bot from your development environment and then import it into a targeted test or production environment with managed bots.

If you continue to see the message after you export and import the latest version of your managed bot, make sure to remove any unmanaged layers:

1. Sign in to Power Apps and select the managed bot's environment.
2. In the navigation menu, select **Solutions**, and then select the solution that contains the bot with the unmanaged layer.
3. Next to the chatbot component in the solution, select **More commands (:)**, and then select **See solution layers**.



4. Select the unmanaged layer, and then select **Remove unmanaged layer**.



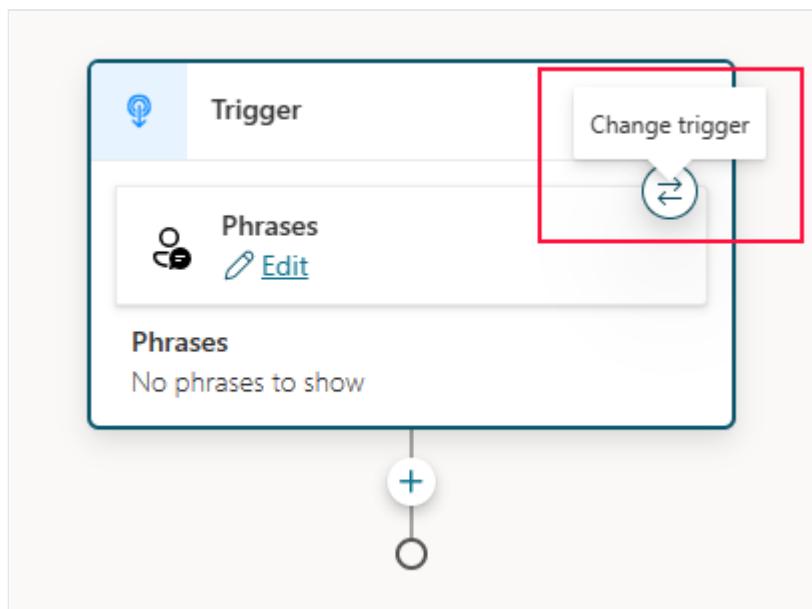
If your bot doesn't have omnichannel capabilities enabled, the message will always show.

Configure inactivity timeout to close conversation

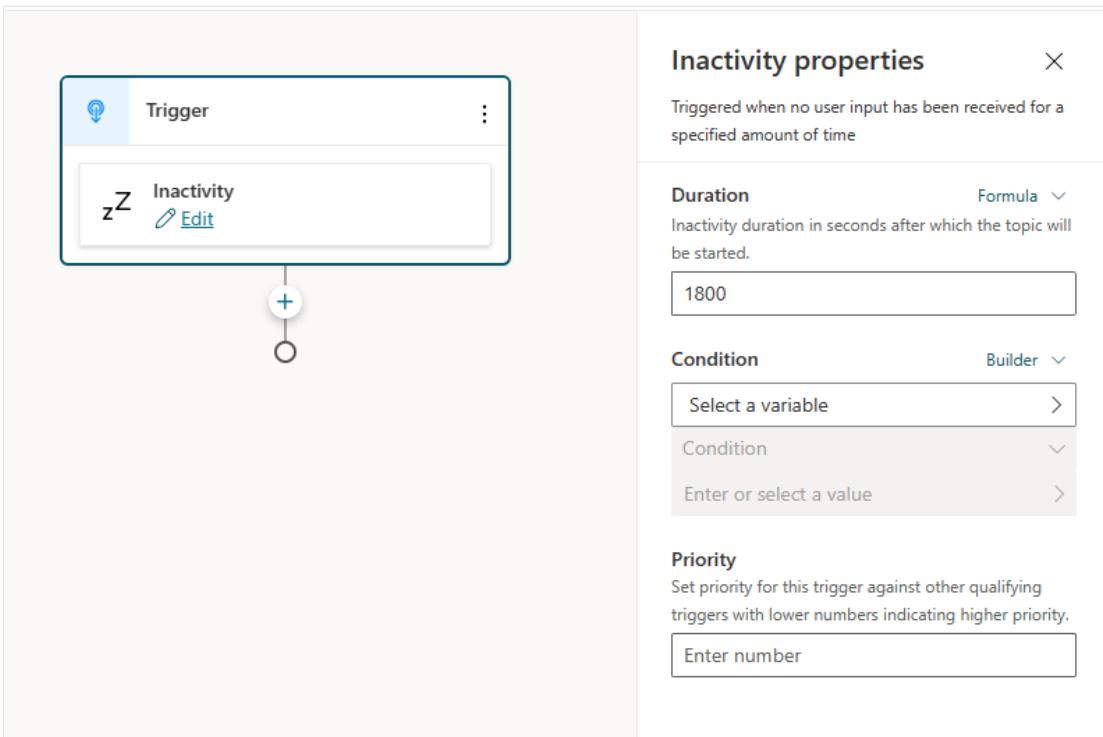
Dynamics 365 Customer Service expects a conversation to end after a period of inactivity. This feature is important to ensure the supervisor view displays the correct ongoing conversations and manages agent loads and metrics correctly. When you create a bot from Dynamics 365 Customer Service, the default bot 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 bot from the Microsoft Copilot Studio portal, those topics aren't included in the bot. To create them, follow the steps below:

Create a Session Timeout topic

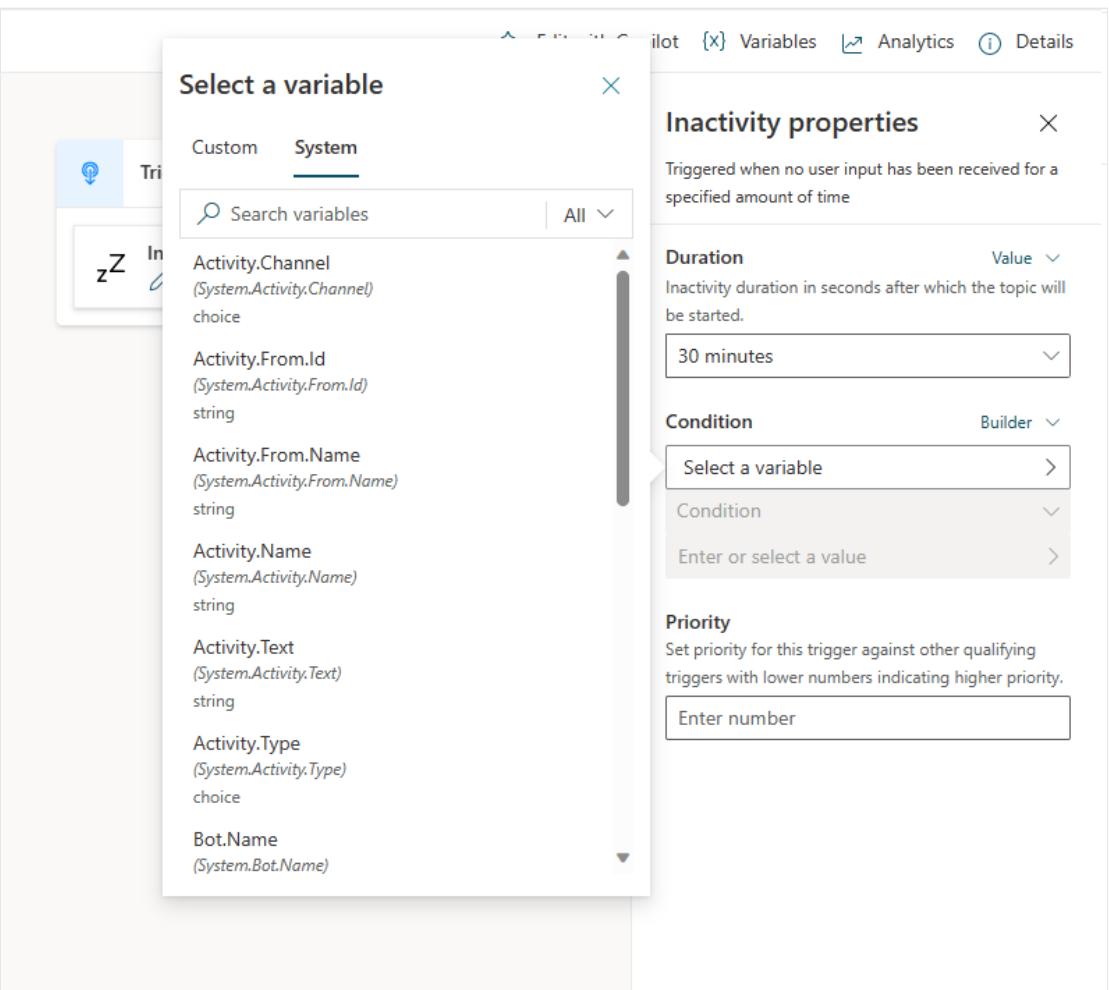
1. From the left navigation, select **Topics**.
2. From the top bar, select **New topic** and **From blank**.
3. Hover the trigger **Phrases** and select the **Change trigger** icon.



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 which channel it applies to by using the **Condition** option. For OmniChannel, click on **Select a variable** under the **Condition** block, select the **System** tab, and **Activity.Channel**.



8. Select **Omnichannel** from the dropdown.
9. Finally, add a message, and at the end, add an End Conversation node so the conversation ends. Click on the (+) sign, select **Topic management**, and then select **End conversation**.

Finally, save and publish your bot.

Manage your copilot's omnichannel capabilities

Select **Settings**, **Agent transfers**, and then select the **Omnichannel** tile. Here you can disconnect your copilot and find the link to go to the Omnichannel for Customer Service admin center to view the connection details.

Disconnect your copilot from Omnichannel for Customer Service or disable the connection

If you select **Disconnect**, the application user that represents the copilot in your Omnichannel for Customer Service instance is disabled. Your copilot effectively

disconnects from the Omnichannel for Customer Service environment and stops receiving any traffic from your Omnichannel for Customer Service instance.

To add your copilot back, you need to [connect it again](#), starting at Step 2.

Known limitations

See [limitations when using Microsoft Copilot Studio with the Chat Add-in for Dynamics 365 Customer Service](#).

Recommended extensions

The following extensions aren't required to hand off to Omnichannel for Customer Service, but they provide copilot authors with a better experience by providing [extra variables and actions](#).

Install [Omnichannel Power Virtual Agent extension](#).

Hand off to Genesys

Article • 06/04/2024

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 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 copilot connector documentation is for AWS (Amazon Web Services), equivalent steps are possible in Azure as well. Modify the two Python scripts accordingly.

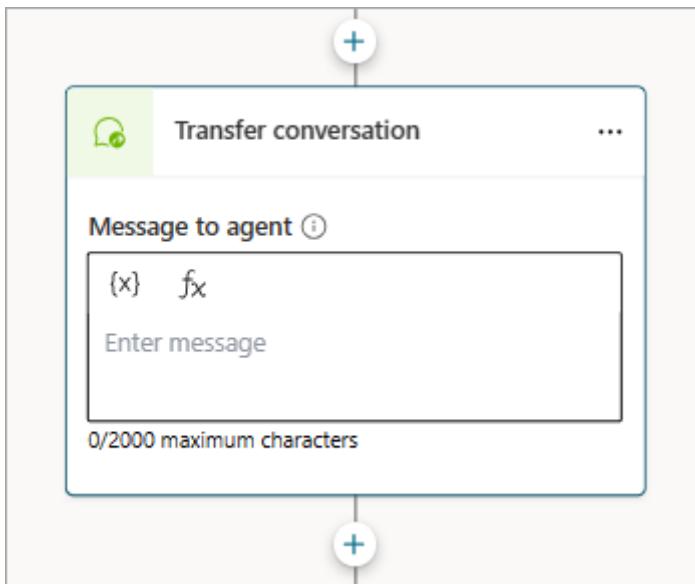
Ensure that the DynamoDB table name created matches the one in the `bot_sessions.py` file.

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 copilot 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 Copilot Studio interface with the following configuration:

- Intents**:
 - Integration**:

There are no slots defined for this intent.
 - Escalate**:
 - va_AgentMessage**:

A State.sAgentMessage
- Failure Outputs**:
 - errorType** ⓘ

No output value specified
 - errorMessage** ⓘ

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 copilot to your website](#) for detailed instructions on how to get a URL to your Copilot Studio copilot.

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?

 Yes

 No

[Provide product feedback ↗](#)

Hand off to LivePerson

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.

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 copilot.
3. Admin or developer access to both LivePerson and Power Platform.

Steps

Follow these steps to create and connect LivePerson to a Copilot Studio bot:

1. [Retrieve the Direct Line secret for your bot.](#)
2. Follow the instructions at the [LivePerson documentation site](#) ↗.
3. To set up handoff to LivePerson, see the [LivePerson documentation - Transfer to agent](#) ↗.

Hand off to Salesforce Einstein bot

Article • 03/26/2024

ⓘ 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 document provides step-by-step instructions on how to create and connect an Einstein Bot to a Copilot Studio bot. This integration allows you to leverage the capabilities of both platforms to enhance your chatbot experience.

Prerequisites

Before you begin, ensure that you have the following prerequisites in place:

- An active Salesforce account with Einstein Bots enabled.
- You'll need an account for Copilot Studio.
- Admin or developer access to both Salesforce and Microsoft Power Platform.

Step 1: Retrieve Direct Line secret for your bot

1. Retrieve your [Direct LinesSecret](#).

Step 2: Configure Salesforce remote site settings

1. Log 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, fill in **Remote Site Name** as *Copilot Studio* and **Remote Site URL** as <https://directline.botframework.com>
5. Click **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. Log 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: (String) conversationId, (Integer) responseCode, (String) errorMessage
- InvocableMethod getConversationID, use your DirectLine Secret to make an HTTP call to the DirectLine endpoint starting a conversation

DL_PostActivity

- Inputs: (String) conversationId, (String) userMessage
- Outputs: (Integer) responseCode, (String) errorMessage, (String) watermark
- InvocableMethod postActivity: Execute an HTTP call to the post activity DirectLine API using the conversationId in the URL and userMessage

DL_GetActivity

- Inputs: (String) conversationId, (String) watermark
- Outputs: (String) message, (String) watermark, (Integer) responseCode, (String) errorMessage
- InvocableMethod getActivity: Execute an HTTP call to the get activity 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. Log 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 additional 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 may 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 Handoff

When accessing Copilot from Einstein, Copilot will only be used in a request/response model: based on the user's query, Copilot will fetch the KB most appropriate and summarize the answer.

Because you can continue to configure your Einstein bot, you can implement any additional topic triggers or escalation triggers in Einstein. This will allow your users to

escalate to the appropriate Salesforce Live Agent queues.

For more information, see the Salesorce documentation [Route Conversations from an Enhanced Bot](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Hand off to ServiceNow

Article • 03/26/2024

This document provides step-by-step instructions on how to create and connect an ServiceNow Virtual Assistant to a Copilot bot and escalate to ServiceNow Live Agent. This integration allows you to leverage the capabilities of both platforms and generative answers to enhance your chatbot experience.

Prerequisites

Before you begin, ensure that you have the following prerequisites in place:

1. An active ServiceNow account with ServiceNow Virtual Agent enabled and ServiceNow Bot Interconnect enabled. (ServiceNow Vancouver or later required)
2. A Copilot Studio bot.
3. Admin or developer access to both ServiceNow and Power Platform.

Steps

Follow these steps to create and connect a ServiceNow Bot to a Power Virtual Agents (PVA) Bot:

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 Microsoft 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. Click 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 [Configuring handoff from Copilot Studio](#)

Step 7: Closing Virtual Agent and Agent Chat conversations [.](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback [.](#)

Conversational language understanding integration overview

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.

You can integrate your [conversational Language Understanding \(CLU\)](#) model with a Microsoft Copilot Studio bot. 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 Microsoft Copilot Studio) and the [Azure Language Studio](#) provides natural language processing (NLP) features for analyzing conversational text.

For projects created with this capability, bot creators can now access existing CLU model intents and entities directly in Microsoft Copilot Studio. Microsoft Copilot Studio users can now map CLU intents to trigger bot dialogs in the same way as native Microsoft Copilot Studio [utterance triggers](#). Entities imported from a CLU model appear on the Microsoft Copilot Studio [Entities](#) page and can be used alongside Microsoft 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 Microsoft 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 bots that you want to take advantage of CLU integration, you'll need to map the CLU model to Microsoft Copilot Studio, then you can update the bot's trigger phrases to bind each topic to a corresponding CLU intent. You can also

manually manage the relationship between the CLU model and Microsoft Copilot Studio.

Prerequisites

- A fully trained CLU model. This includes intents for each of [PVA'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
- [Microsoft Copilot Studio](#) account
- [Existing Microsoft Copilot Studio](#) bot
- 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 Microsoft Copilot Studio projects. For more information, see [Get started with language understanding](#).

CLU connectors

A *connector* is a wrapper around an API that allows the Azure Cognitive Service to talk to Microsoft 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, Microsoft Copilot Studio doesn't currently validate connectors.

Connections in Microsoft 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.

Microsoft Copilot Studio connections are environment specific. When you import a bot, 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 Microsoft Copilot Studio variables, including system variables, see [Use variables - Microsoft Copilot Studio](#).

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 Microsoft Copilot Studio [prebuilt entities](#) along with CLU entities.

Related topics

Other bot-building features that CLU integration supports include:

- System topics: [Use system topics in Microsoft Copilot Studio](#)
- Entities and slot filling: [Create and use entities in Microsoft Copilot Studio](#)
- 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 • 06/04/2024

ⓘ 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 includes the step-by-step procedures you need to get started with [conversational language understanding \(CLU\) integration](#). Let's get started.

To create a copilot 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 copilot](#).

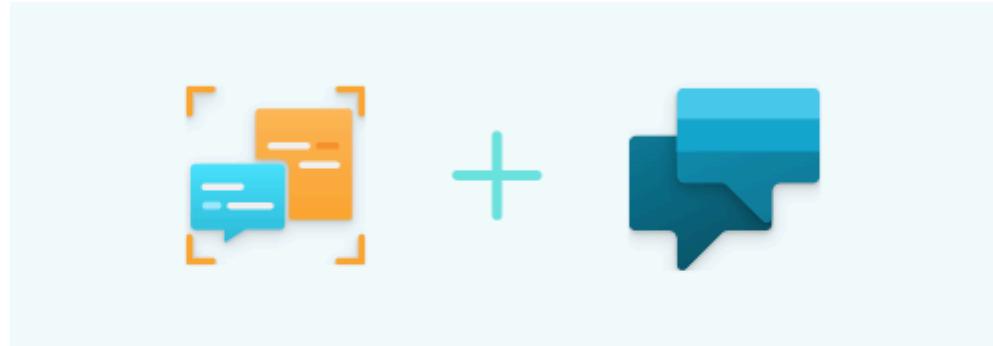
ⓘ Note

Ensure the [environment for your copilot](#) 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 copilot 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**.

 Save Refresh

Language

Customize your copilot's language settings. Review [supplemental terms and conditions](#) to learn more about features in preview.

Natural language understanding

Your copilot has been enabled to use an external natural language understanding (NLU) system. Select an external model or use Microsoft Copilot Studio built-in NLU. [Learn more](#)

NLU Resource *

my CLU connection

[Manage connections in Power Apps](#)

A window appears, prompting you to save a snapshot of your copilot.

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 copilot 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 copilot 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.

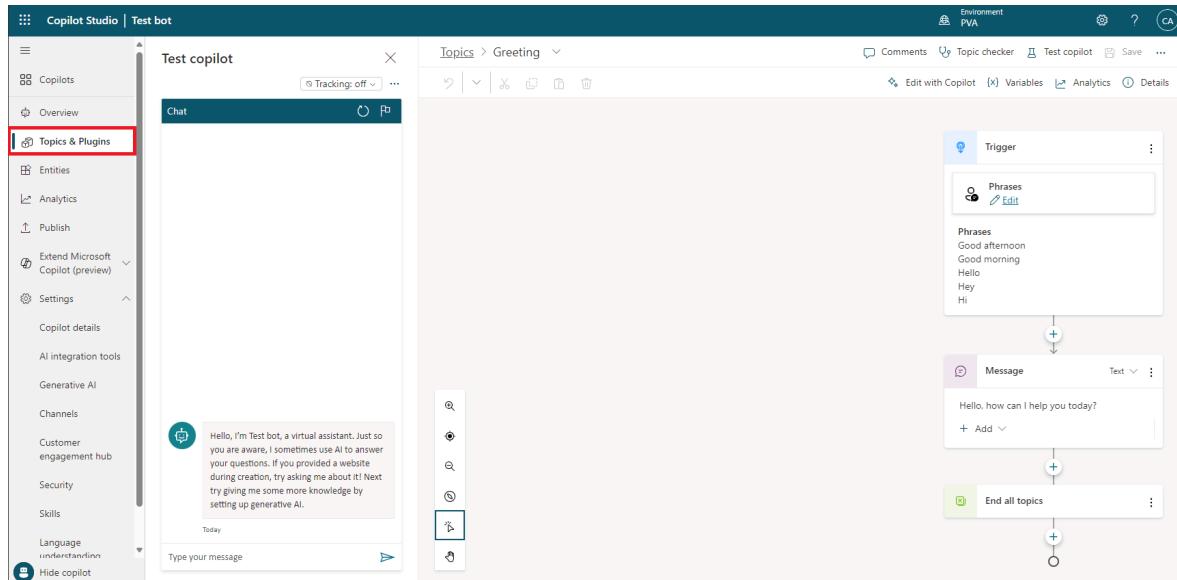
3. To ensure Copilot Studio functions correctly, you need to create intents for [Copilot Studio system topics](#) and for [any custom topics](#) you want to create.
4. Author [any custom entities](#) you want to use in your Copilot Studio project.
5. Train and deploy the project in the same region as your Copilot Studio resource.

Map CLU intents and entities to Copilot Studio prebuilt data types

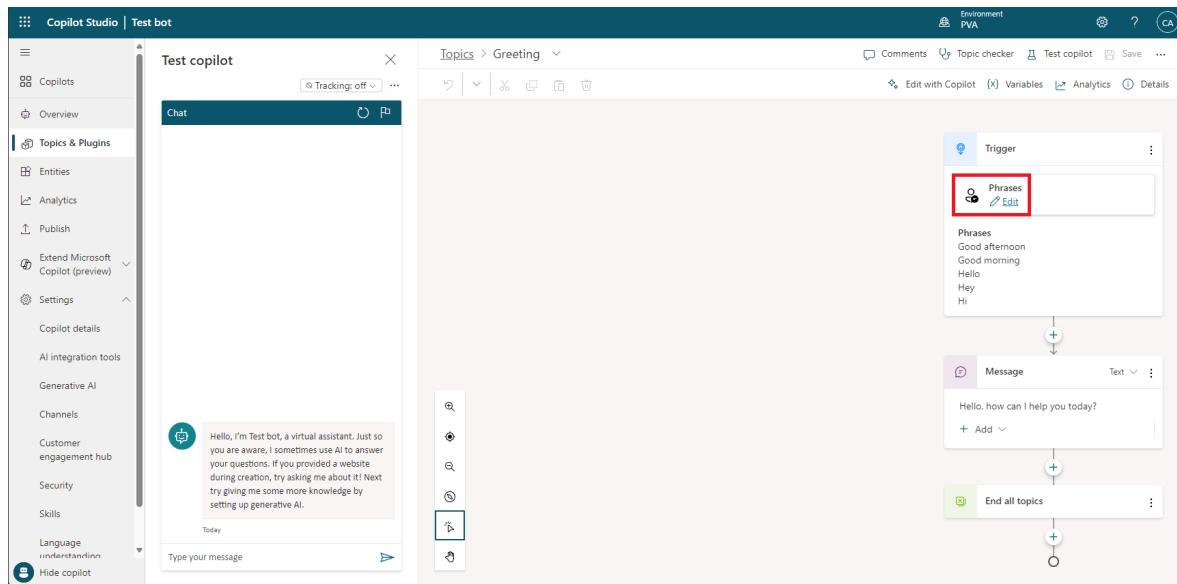
You create a new Copilot Studio copilot by mapping existing topics to CLU intents. You can manually map intents and entities, or for bulk mapping, see [Use the Bulk tool wizard](#).

Manually map intents

1. Open your Copilot Studio copilot.
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

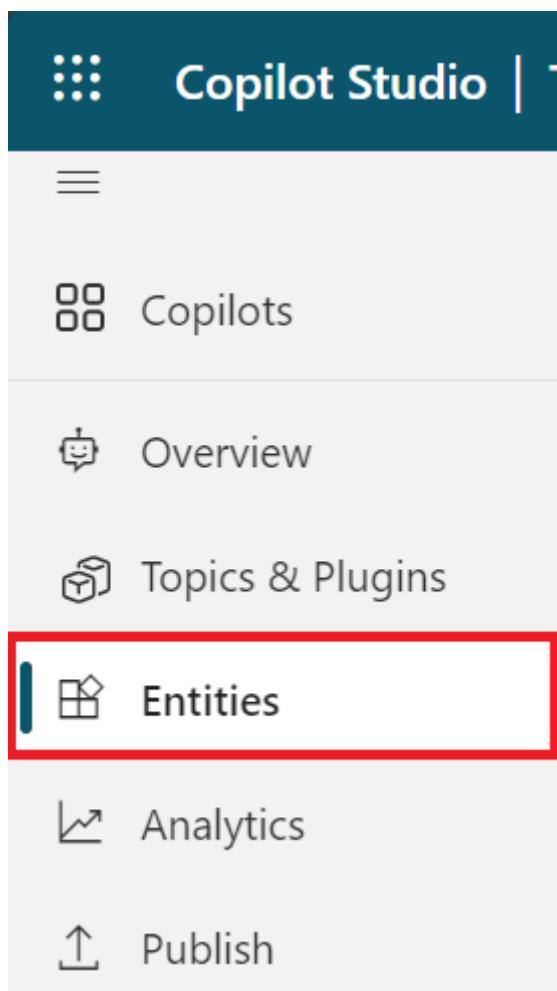
! Note

The CLU intent name must be typed exactly as stored in the CLU model, including matching case as uppercase and lowercase characters.

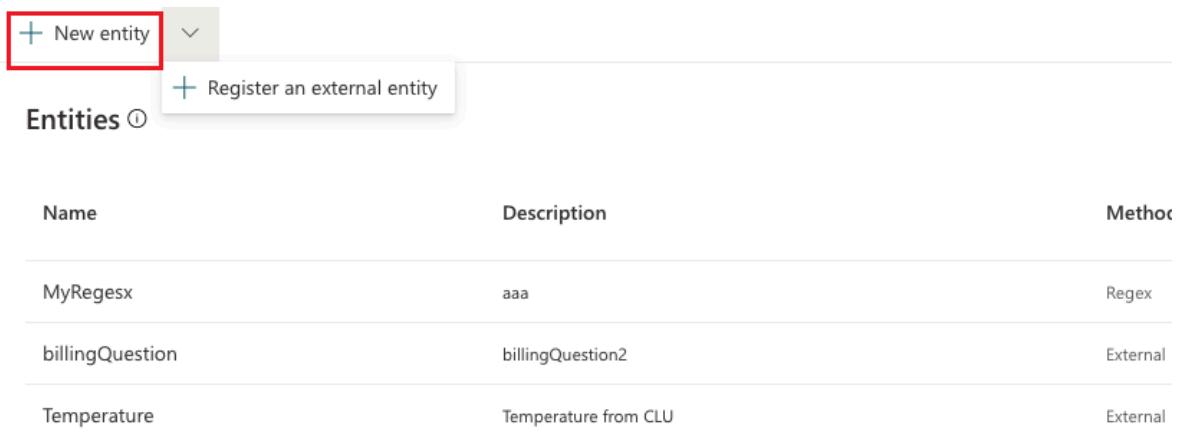
5. To save the intent, click the + (plus) sign to the right of the *Add phrases* box.
6. Repeat for the other intents.

Manually map entities

1. Open your copilot.
2. From the left navigation, select **Entities**.



3. Select **New entity** on the **Entities** page.



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

Name *

The display name of a closed list type cannot be empty

Description

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 copilot by bulk mapping existing topics to CLU intents using the **Map intents and entities** wizard. With the wizard, your copilot 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 copilot.

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, ...

Select an intent

Thank you

Select an intent

Lesson 2 - A simple topic with a con...

Select an intent

Lesson 1 - A simple topic

Select an intent

Goodbye

Select an intent

PurchaseTickets

Select an intent

Greeting

Select an intent

Start Over

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.

8. Once all CLU intents and entities are mapped to topics, select Done.

Add intents and entities

X

- Upload your model data file
- Map existing topics to CLU intents
- Create new topics for CLU intents
- Review

Success!

0 entities were imported

Done

Cancel

Once you are done with this procedure, your copilot topics now appear in the **Topics** page. For more information, see [Manage topics in Copilot Studio](#).

Create a CLU connection in Power Apps

1. In Power Apps, if the **Connections** page isn't already in focus, select **Connections** in the side navigation pane.
2. Select **New connection**.
3. Select **Azure Cognitive Service for Language** (use the search field to narrow down the list if needed). A window appears prompting you for authentication information.
4. For the authentication type, select **Api Key**, and enter the required account key.
5. Enter the appropriate root site URL, if any.
6. Select **Create**. The new connection, with the default name "Azure Cognitive Service for Language," appears in the list of connections for your environment.
7. Select the **More commands** icon (...) next to this connection and select **Edit**.
8. 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 • 06/04/2024

ⓘ 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 discusses adding conversational language understanding (CLU) entities to Copilot Studio copilots. 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 copilots, 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 copilots 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 copilot 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": "07:30:00Z"  
}
```

```
    "value": "07:30:00"
```

```
}
```

Related content

- Get started with conversational language understanding integration
-

Feedback

Was this page helpful?

 Yes

 No

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.

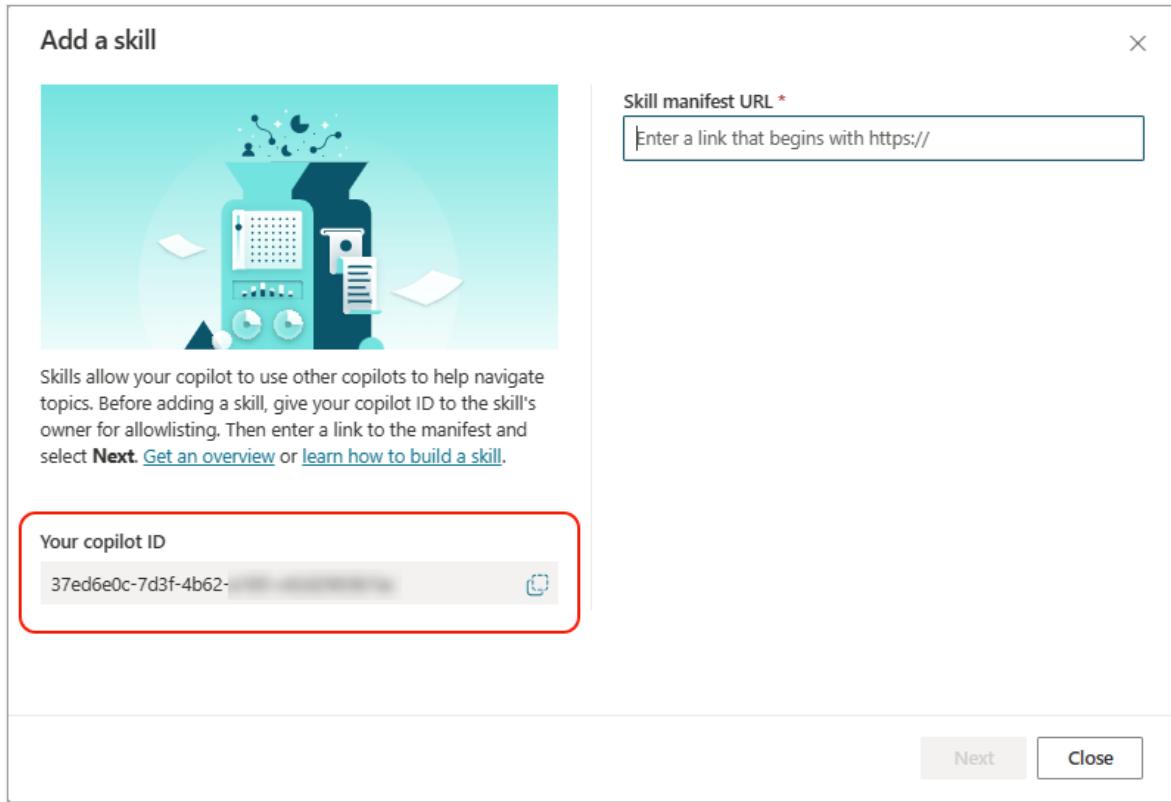
Configure a skill

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 • 06/06/2024

ⓘ 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.

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.

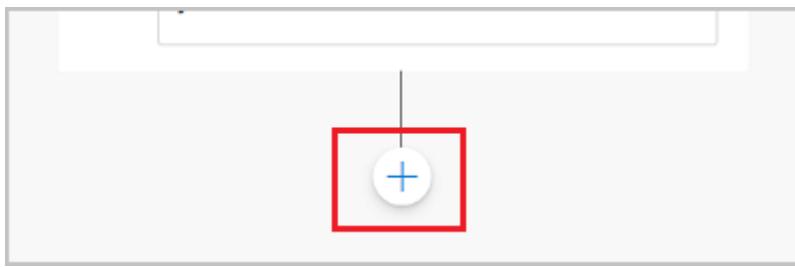
Add a skill to copilot conversations

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

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.

ⓘ Note

This article includes information applicable to Microsoft Copilot Studio classic chatbots only. This feature is not available in the new Microsoft Copilot Studio experience.

When you use a Microsoft Copilot Studio copilot as a [skill](#) with a Bot Framework bot, the Bot Framework bot determines if anything the user says matches the Microsoft Copilot Studio copilot's [trigger phrases](#). If there's a match, the Bot Framework bot [passes the conversation to the Microsoft Copilot Studio bot](#). The Microsoft Copilot Studio copilot extracts any [entities](#) and triggers the matching topic.

You can also [pass a variable to a Microsoft 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 Microsoft Copilot Studio license to use Microsoft Copilot Studio bots as skills. This capability isn't available with the [Teams Microsoft Copilot Studio license](#).

To set up your Microsoft Copilot Studio copilot as a skill:

1. [Add the Bot Framework bot to the allowlist for the Microsoft Copilot Studio copilot](#).
2. [Download the skill manifest for the Microsoft Copilot Studio copilot](#).

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 Microsoft Copilot Studio copilot

Add the Bot Framework bot's app ID to the allowlist for the Microsoft Copilot Studio copilot 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**.

Bot_Framework_bot

Configure your bot

Bot_Framework_bot (root)

Overview Development resources Connections Skill configuration Localization

Azure Language Understanding

Language Understanding (LUIS) is an Azure Cognitive Service that uses machine learning to understand natural language input and direct the conversation flow. [Learn more](#). Use an existing Language Understanding (LUIS) key from Azure or create a new key. [Learn more](#)

Application name [?](#)
Bot_Framework_bot

Language Understanding authoring key [?](#)
Type Language Understanding authoring key

Language Understanding region [?](#)
Select region

Set up Language Understanding

Azure QnA Maker

QnA Maker is an Azure Cognitive services that can extract question-and-answer pairs from a website FAQ. [Learn more](#). Use an existing key from Azure or create a new key. [Learn more](#)

QnA Maker Subscription key [?](#)
Type subscription key

Set up QnA Maker

Microsoft App ID

An App ID is used for communication between your bot and skills, services, websites or applications. Use an existing App ID or automatically generate an App ID when creating a publishing profile for this bot. [Learn more](#)

Microsoft App Id [?](#)
[REDACTED]

2. In Microsoft Copilot Studio, open the copilot you want to use as a skill.

3. In the navigation menu, go to **Settings** and select **Security**. Then select **Allowlist**.

Chatbots

Overview

Topics

Entities

Analytics

Publish

Settings

Details

Channels

Agent transfers

Security

AI capabilities

Sharing

Invite people to collaborate on your bot.

Authentication

Verify a user's identity during a chat.

Web channel security

Review other enhanced security options.

Allowlist

Let other bots call your bot as a skill.

4. Select Add allowed caller.

Manage allowlist for Test bot

Your chatbot is not yet published. In order to use your chatbot successfully go to the publish page and click publish. [Go to the publish page](#)

+ Add allowed caller

Description Chatbots on your allow list are authorized to use your chatbot as a Skill. That chatbot can then use your chatbots topics as actions when authoring a topic.

Allowed callers (0)

Name/ID	Added by	Added
---------	----------	-------

Download chatbot manifests To enable another chatbot to call yours as a Skill, download the .zip manifest, and give it to the other chatbots owner. [Learn more](#)

[Test manifest](#)

[Published manifest](#)

Microsoft Copilot Studio copilots aren't yet supported. At this time, only Bot Framework bots can be added to your allowlist.

Your chatbot doesn't have any allowed callers yet

5. Paste your Bot Framework bot's app ID and select Next.

Add allowed caller

Allowed caller's Microsoft app ID *

Enter a Microsoft app ID

Microsoft Copilot Studio copilots aren't yet supported. At this time, only Bot Framework bots can be added to your allowlist.

Description To authorize another copilot to call your copilot, enter the allowed caller's unique Azure app ID. (Note that some copilots might have multiple copilot IDs.)

After you've added the copilot to your allowlist, give a link to your copilot's manifest to the allowed caller's owner for the connection to work. [Learn more](#)

Next Cancel

⊗ Caution

A Microsoft Copilot Studio copilot can't act as a skill for other Microsoft Copilot Studio copilots. 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.

Microsoft 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.

Allowed caller details

Display name

My Bot Framework bot

Source

Bot Framework /Azure

Added

a few seconds ago

Calling bot's app ID

By clicking **Save** you acknowledge that your data may flow outside your organization's compliance and geo boundaries. This includes Government Cloud environments. Learn more about [where your data is located](#) and the [Microsoft Privacy Statement](#).

Delete

Save

Cancel

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.

Manage allowlist for Testbot

+ Add allowed caller

Description	Allowed callers (1)								
Chatbots on your allow list are authorized to use your chatbot as a Skill. That chatbot can then use your chatbots topics as actions when authoring a topic.	<table border="1"><thead><tr><th>Name/ID ↓</th><th>Delete</th><th>Added by</th><th>Added</th></tr></thead><tbody><tr><td>My Bot Framework bot</td><td>Delete</td><td>Bot Framework /Azure</td><td>a few seconds ago</td></tr></tbody></table>	Name/ID ↓	Delete	Added by	Added	My Bot Framework bot	Delete	Bot Framework /Azure	a few seconds ago
Name/ID ↓	Delete	Added by	Added						
My Bot Framework bot	Delete	Bot Framework /Azure	a few seconds ago						

Download chatbot manifests

To enable another chatbot to call yours as a Skill, download the .zip manifest, and give it to the other chatbots owner. [Learn more](#)

Test manifest

Published manifest

Microsoft Copilot Studio copilots aren't yet supported.
At this time, only Bot Framework bots can be added to your allowlist.

 **Note**

Bot Framework bots added to the allowlist aren't [exported as part of the bot content](#).

Download the copilot skill manifest for your Microsoft Copilot Studio copilot

Bot Framework bots can use a Microsoft Copilot Studio skill manifest to configure a connection to the copilot that produced the manifest.

All Microsoft Copilot Studio copilots 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 Microsoft Copilot Studio bot (for example, in response to something a user says to it).

Microsoft 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 a Microsoft Copilot Studio copilot should handle the user's request, based on the manifest data. It passes to the copilot everything the user said. Then, the Microsoft Copilot Studio copilot matches what the user said [to a Microsoft Copilot Studio topic](#), extracts any [entities that are needed for slot-filling](#), and triggers the Microsoft Copilot Studio topic.

Microsoft Copilot Studio skill manifests are automatically generated and updated. A Microsoft Copilot Studio copilot has two skill manifests:

- **Test manifest:** Allows the Bot Framework bot to connect to the test version of your Microsoft Copilot Studio copilot. Use the test manifest to validate changes to your skill before you publish it.
 - The test manifest is immediately available for every newly created Microsoft Copilot Studio bot.
 - It's automatically updated to reflect changes every time you [save](#) your copilot.
- **Published manifest:** Allows the Bot Framework bot to connect to the published version of your Microsoft Copilot Studio copilot.
 - The published manifest is only available for Microsoft Copilot Studio copilots that have been published at least once.

- It's automatically updated to reflect changes every time you [publish your copilot](#).

① Note

The Published manifest isn't available for Microsoft Copilot Studio copilots that have never been published.

To generate your bot's Published manifest, [publish your Microsoft Copilot Studio copilot](#).

Both skill manifests are shown on the **Manage allowlist** panel.

Manage allowlist for Testbot

[+ Add allowed caller](#)

Description	Allowed callers (1)
Chatbots on your allow list are authorized to use your chatbot as a Skill. That chatbot can then use your chatbots topics as actions when authoring a topic.	Name/ID ↓ My Bot Framework bot

Download chatbot manifests

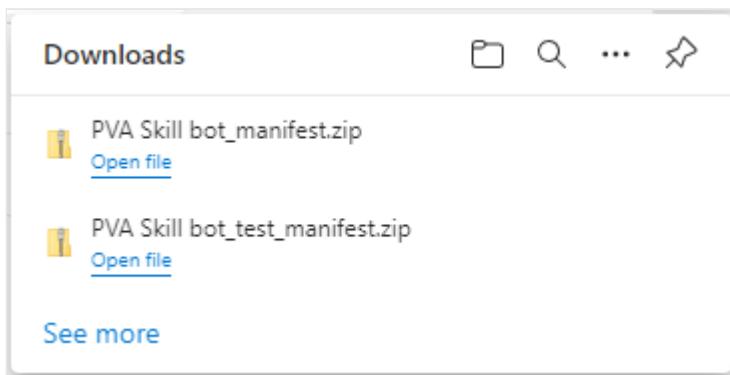
To enable another chatbot to call yours as a Skill, download the .zip manifest, and give it to the other chatbots owner. [Learn more](#)




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**.

The screenshot shows the Microsoft Copilot Studio interface for managing chatbots. On the left, a sidebar lists various sections: Chatbots, Overview, Topics, Entities, Analytics, Publish, Settings, and Details. The Details section is expanded, showing sub-options: AI integration tools, Channels, Customer engagement hub, Security, Skills, AI capabilities, and Test your chatbot. The main content area is titled 'Details' and has tabs for 'General' and 'Advanced', with 'Advanced' selected. A section titled 'Metadata' explains its purpose: 'Metadata holds all the information to uniquely identify your chatbot and its endpoints. It can be used for calling your chatbot or using your chatbot in external scenarios.' Below this are fields for 'Environment ID' (Default-c2983f0e-34ee-4b43-8abc-c2f460fd26be), 'Tenant ID' (c2983f0e-34ee-4b43-8abc-c2f460fd26be), 'Chatbot app ID' (f3c603a2-e881-49b6-9b27-870ada49d2e4), and 'Schema name' (new_bot_14ba34a0a0824f149cab89b9542e65fe). A red box highlights a section titled 'Download chatbot manifests' which contains instructions: 'To enable another bot to call your bot as a Skill, download the .zip manifest, and give it to the other bot's owner.' It includes two download buttons: 'Test manifest' and 'Published manifest'.

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**.



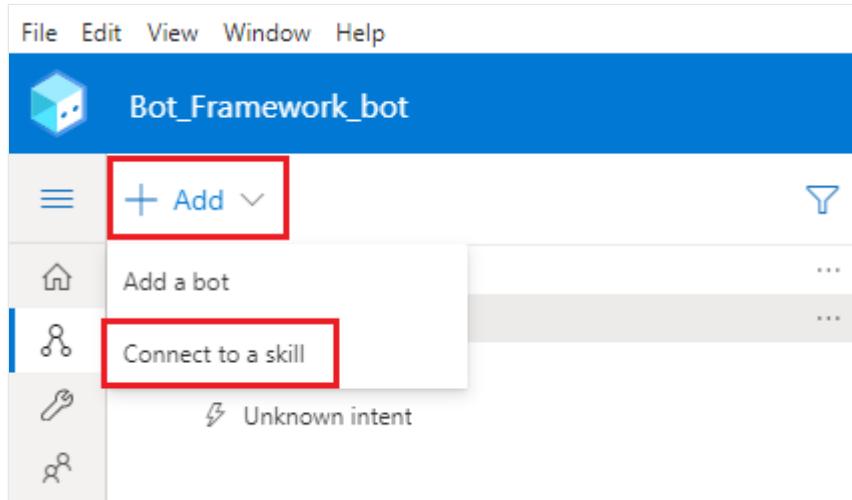
ⓘ Note

The skill manifest doesn't include system topics and Composer dialogs that are added to a Microsoft Copilot Studio copilot. Only topics that were created by copilot authors and Composer intent triggers that authors add to the bot are included.

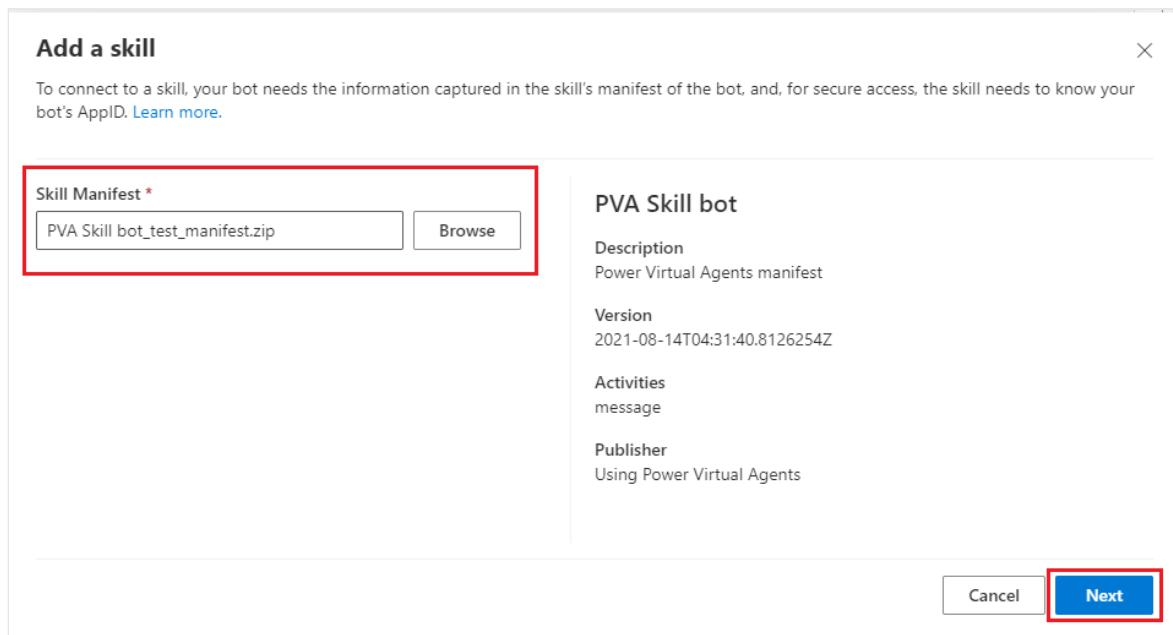
Connect to a Microsoft Copilot Studio skill in Composer

Use Microsoft 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 Microsoft Copilot Studio manifest .zip archive, and then select **Next**.



3. Select the topics, or intents, in your Microsoft Copilot Studio copilot that you want to add to your Bot Framework bot, and then select **Next**.

Select intents to trigger PVA Skill bot skill

X

These intents will trigger this skill from Bot_Framework_bot

Intents

<input checked="" type="checkbox"/>	Name
<input checked="" type="checkbox"/>	Lesson 1 - A simple topic-new_topic_c0fea4d80e66458ca429d6f19d81b4bf_33d5123770ba4daefae51b361ddc80a
<input checked="" type="checkbox"/>	Lesson 2 - A simple topic with a condition and variable-new_topic_c0fea4d80e66458ca429d6f19d81b4bf_21a34f1673a24096a99ba0effb63cc1d
<input checked="" type="checkbox"/>	Lesson 3 - A topic with a condition, variables and a pre-built entity-new_topic_c0fea4d80e66458ca429d6f19d81b4bf_dbafebd526364487911b63a2a774389f
<input checked="" type="checkbox"/>	Lesson 4 - A topic with a condition, variables and custom entity-new_topic_c0fea4d80e66458ca429d6f19d81b4bf_85df75ce08a44a9b82cc8f3fdb25f669

Back

Cancel

Next

4. Review and edit the trigger phrases for your Microsoft Copilot Studio copilot, and then select **Next**.

Add or edit phrases to trigger PVA Skill bot skill

X

These phrases will trigger this skill from Bot_Framework_bot

> Lesson 1 - A simple topic-new_topic_c0fea4d80e66458ca429d6f19d81b4bf_33d5123770ba4daefae51b361ddc80a
- When are you closed
- When are you open
- Store hours
- Daily open hours
> Lesson 2 - A simple topic with a condition and variable
> Lesson 2 - A simple topic with a condition and variable-new_topic_c0fea4d80e66458ca429d6f19d81b4bf_21a34f1673a2
- Are there any stores around the area
- Check store location
- Closest store near me
- Find me your nearest location
- Store near me
> Lesson 3 - A topic with a condition, variables and a pre-built entity
> Lesson 3 - A topic with a condition, variables and a pre-built entity-new_topic_c0fea4d80e66458ca429d6f19d81b4bf_dbafebd526364487911b63a2a774389f

Back

Cancel

Next

5. Make sure **Use Orchestrator for multi-bot projects** is selected, and then select **Continue**.

Enable Orchestrator

X

A bot that consists of multiple bots or connects to skills (multi-bot project) needs Orchestrator to detect and route user input to the appropriate bot or skill.

A bot that consists of multiple bots or connects to skills (multi-bot project) needs Orchestrator to detect and route user input to the appropriate bot or skill.

[Learn more about Orchestrator](#)

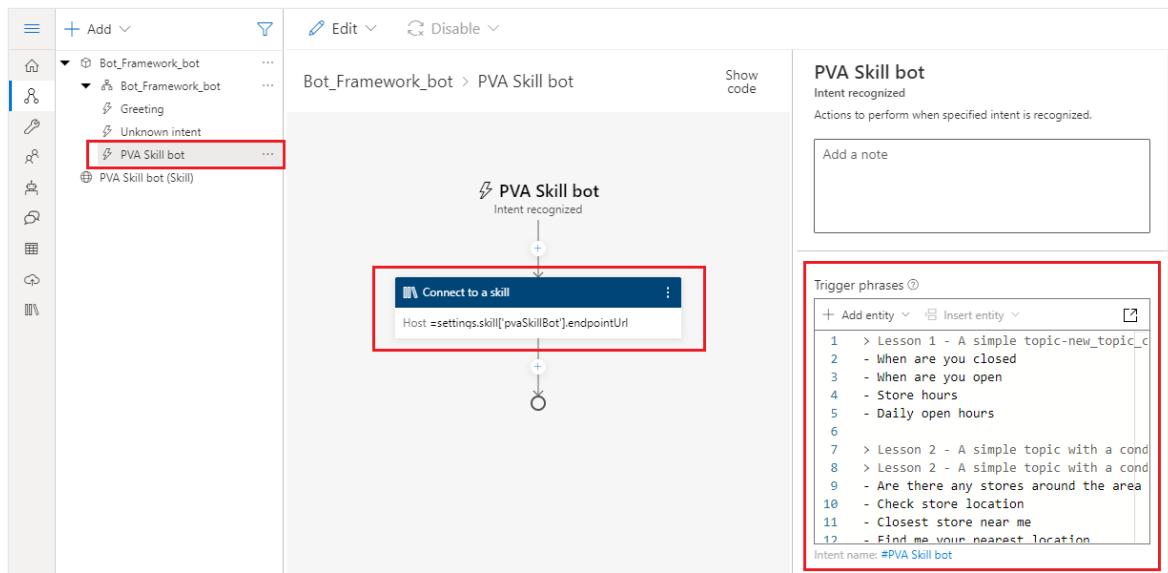
Use Orchestrator for multi-bot projects (bots that consist of multiple bots or connect to skills).

Back

Skip

Continue

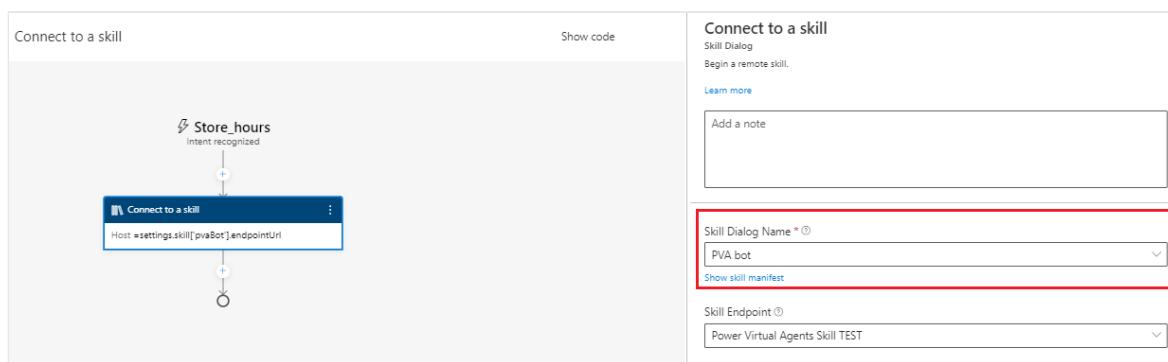
- Verify that the Microsoft Copilot Studio skill is added to your Bot Framework bot correctly, and that a new trigger with your Microsoft Copilot Studio copilot's name appears in the project in Composer.



Locate the topic in the skill manifest

Use an event to trigger a Microsoft 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.

- Select the **Connect to a skill** action.
- Under **Skill Dialog Name**, select **Show skill manifest**.



- Find **activities** in the manifest. This property contains nested properties that represent the topics available in your Microsoft Copilot Studio copilot.

PVA bot Manifest

```
1 {
2   "$schema": "https://schemas.botframework.com/schemas/skills/v2
3   "$id": "2021-10-08T20:22:47.8489574Z",
4   "name": "PVA bot",
5   "version": "2021-10-08T20:22:47.8489574Z",
6   "description": "Power Virtual Agents manifest",
7   "publisherName": "Using Power Virtual Agents",
8   "privacyUrl": "https://go.microsoft.com/fwlink/?linkid=2138950
9   "iconUrl": "https://bot-framework.azureedge.net/bot-icons-v1/8
10  "endpoints": [
11    {
12      "name": "Power Virtual Agents Skill TEST",
13      "protocol": "BotFrameworkV3",
14      "description": "TEST",
15      "endpointUrl": "https://powervaengine.us-il002.gateway.tes
16      "msAppId": 'REDACTED'
17    }
18  ],
19  "activities": {
20    "message": {
21      "type": "message",
22      "description": "Message activity types"
23    },
24    "Lesson 1 - A simple topic-new_topic_87609dabd86049f7bc6507c
25      "type": "event",
26      "name": "dispatchTo_new_topic_87609dabd86049f7bc6507c6f726
27      "description": "Lesson 1 - This lesson will show you how y
28    },
29    "Lesson 2 - A simple topic with a condition and variable-new_
30      "type": "event",
31      "name": "dispatchTo_new_topic_87609dabd86049f7bc6507c6f726
32      "description": "Lesson 2 - This lesson will show you how y
33    },
34    "Lesson 3 - A topic with a condition, variables and a pre-bu
35      "type": "event",
36      "name": "dispatchTo_new_topic_87609dabd86049f7bc6507c6f7263aba_33d
37  }
38}
```

Close

4. Find the topic you want to call and look for its **name** property. When the Microsoft Copilot Studio skill receives an event with this name, it triggers the topic.

In the following example, the event activity name is

`dispatchTo_new_topic_87609dabd86049f7bc6507c6f7263aba_33d`.

PVA bot Manifest

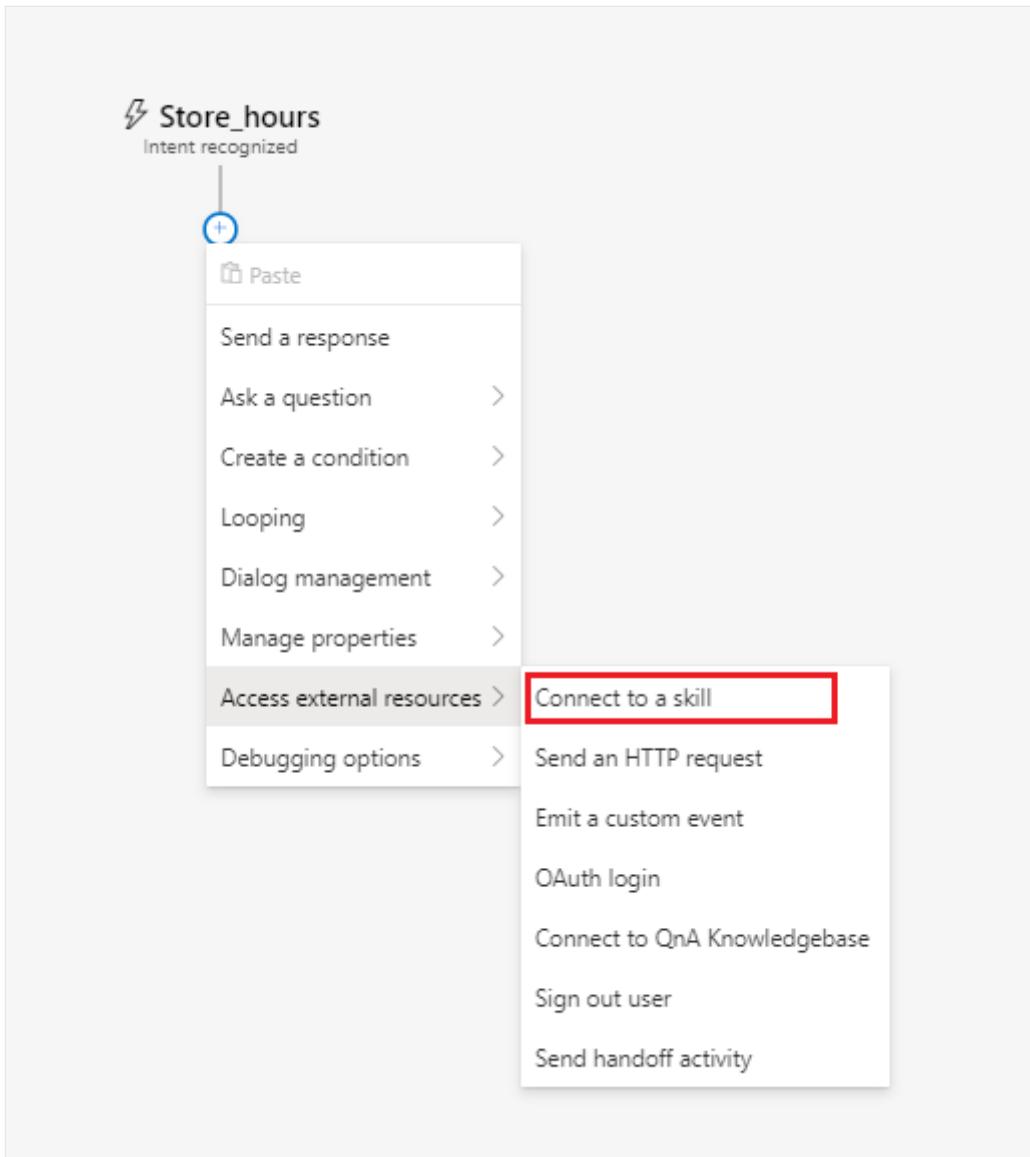
```
1 {
2   "$schema": "https://schemas.botframework.com/schemas/skills/v2"
3   "$id": "https://schemas.botframework.com/schemas/skills/v2",
4   "name": "PVA bot",
5   "version": "2021-10-08T20:22:47.8489574Z",
6   "description": "Power Virtual Agents manifest",
7   "publisherName": "Using Power Virtual Agents",
8   "privacyUrl": "https://go.microsoft.com/fwlink/?linkid=2138950"
9   "iconUrl": "https://bot-framework.azureedge.net/bot-icons-v1/8"
10  "endpoints": [
11    {
12      "name": "Power Virtual Agents Skill TEST",
13      "protocol": "BotFrameworkV3",
14      "description": "TEST",
15      "endpointUrl": "https://powervaengine.us-il002.gateway.tes"
16      "msAppId": "https://bot-framework.azureedge.net/bot-ic...  
[redacted]
17    }
18  ],
19  "activities": {
20    "message": {
21      "type": "message",
22      "description": "Message activity types"
23    },
24    "Lesson 1 - A simple topic-new_topic_87609dabd86049f7bc6507c6f726": {
25      "type": "event",
26      "name": "dispatchTo_new_topic_87609dabd86049f7bc6507c6f726"
27      "description": "Lesson 1 - This lesson will show you how yo"
28    },
29    "Lesson 2 - A simple topic with a condition and variable-new": {
30      "type": "event",
31      "name": "dispatchTo_new_topic_87609dabd86049f7bc6507c6f726"
32      "description": "Lesson 2 - This lesson will show you how yo"
33    },
34    "Lesson 3 - A topic with a condition, variables and a pre-bu": {
35      "type": "event",
36      "name": "dispatchTo_new_topic_87609dabd86049f7bc6507c6f726"
37      "description": "Lesson 3 - This lesson will show you how yo"
38    }
39  }
40}
```

Close

Call a Microsoft Copilot Studio skill topic

Instead of relying on what a user says to trigger a Microsoft Copilot Studio skill topic, you can call it directly. You can only directly call Microsoft 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 Microsoft Copilot Studio copilot.

The screenshot shows the 'Connect to a skill' configuration page. It includes sections for 'Add a note', 'Skill Dialog Name' (set to 'PVA bot'), and 'Skill Endpoint' (set to 'Power Virtual Agents Skill TEST').

3. Find the event activity's name for the Microsoft Copilot Studio bot 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.

Connect to a skill

Skill Dialog

Begin a remote skill.

[Learn more](#)

Add a note

Skill Dialog Name * ②

PVA bot



[Show skill manifest](#)

Skill Endpoint ②

Power Virtual Agents Skill TEST



OAuth connection name (SSO) ②

abc =settings.connectionName

Activity ②

[Show code](#)

Text +



Responses ②



[Add alternative](#)

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
]
```

Connect to a skill

Skill Dialog

Begin a remote skill.

[Learn more](#)

Add a note

Skill Dialog Name * ⓘ

PVA bot

[Show skill manifest](#)

Skill Endpoint ⓘ

Power Virtual Agents Skill TEST

OAuth connection name (SSO) ⓘ

abc =settings.connectionName

Activity ⓘ

[Show response editor](#)



{x} {/x}



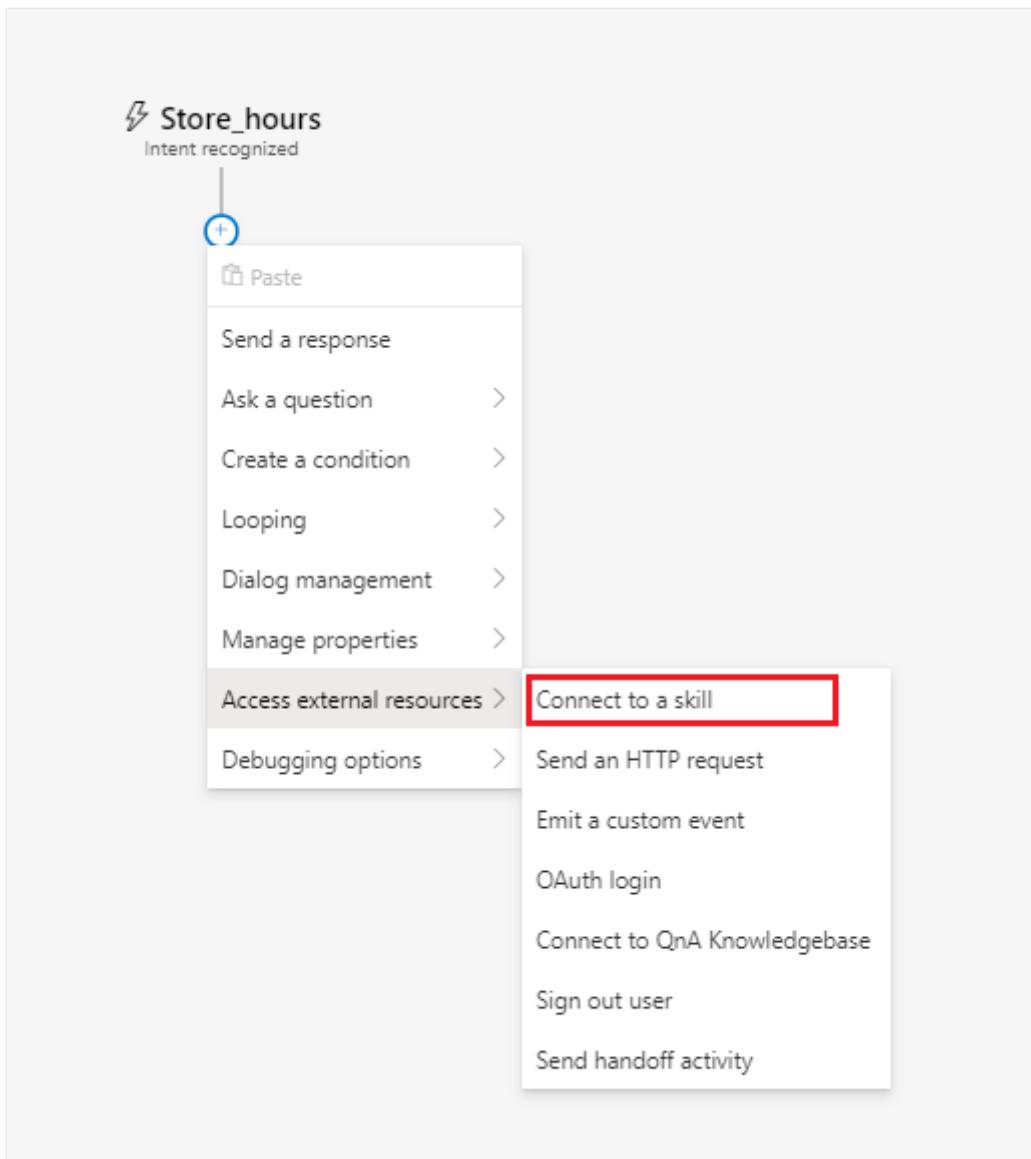
```
1 [Activity
2   type = event
3   name = dispatchTo_new_topic_87609dabd86049f7bc6507c6f7263aba_33d
4 ]
```

Pass an input variable to a Microsoft Copilot Studio skill topic

Bot Framework Composer can pass a variable to a Microsoft Copilot Studio skill topic as an input. To learn how to create Microsoft Copilot Studio topics that accept input variables, see [Passing variables between topics](#).

If a Microsoft 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 Microsoft Copilot Studio copilot.

A screenshot of the Microsoft Copilot Studio interface showing the 'Connect to a skill' configuration dialog. On the left, there's a preview of the dialog flow: 'Store_hours' intent leads to a 'Connect to a skill' activity, which then leads to a final step. On the right, the configuration pane shows: 'Skill Dialog' section with 'Begin a remote skill.', 'Learn more', and a note input field; 'Skill Dialog Name' dropdown set to 'PVA bot' (highlighted with a red box); 'Show skill manifest' link; 'Skill Endpoint' dropdown set to 'Power Virtual Agents Skill TEST'; and a 'Show code' button.

3. Find the event activity's name for the Microsoft Copilot Studio copilot 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 Microsoft Copilot Studio topic doesn't have inputs, it won't have a **value** property.

PVA bot Manifest

```
17    "activities": [
18      "message": {
19        "type": "message",
20        "description": "Message activity types"
21      },
22      "Lesson 1 - A simple topic-new_topic_127cdcdbbb4a480ea113c5101f30": {
23        "type": "event",
24        "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
25        "description": "Lesson 1 - This lesson will show you how you can dispatch to a new topic"
26      },
27      "Lesson 2 - A simple topic with a condition and variable-new_topic_127cdcdbbb4a480ea113c5101f30": {
28        "type": "event",
29        "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
30        "description": "Lesson 2 - This lesson will show you how you can add conditions and variables to your topics"
31      },
32      "Lesson 3 - A topic with a condition, variables and a pre-built event-new_topic_127cdcdbbb4a480ea113c5101f30": {
33        "value": {
34          "$ref": "#/definitions/Lesson+2++A+simple+topic+with+a+condition+and+variables"
35        }
36      },
37      "Lesson 4 - A topic with a condition, variables and custom event-new_topic_127cdcdbbb4a480ea113c5101f30": {
38        "type": "event",
39        "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
40        "description": "Lesson 4 - This lesson continues to show you how you can use custom events"
41      },
42      "Lesson 5 - A topic with a condition, variables and a pre-built event-new_topic_127cdcdbbb4a480ea113c5101f30": {
43        "type": "event",
44        "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
45        "description": "Lesson 5 - This lesson continues to show you how you can use pre-built events"
46      }
47    ],
48    "definitions": {
49      "Lesson+2++A+simple+topic+with+a+condition+and+variables-new": {
50        "type": "object",
51        "properties": {
52          "pva_StoreLocation": {
53            "type": "string"
54          }
55        }
56      }
57    }
58  ]
```

Close

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 Microsoft Copilot Studio topic's inputs. You'll use them in the next steps.

PVA bot Manifest

```
21     "type": "message",
22     "description": "Message activity types"
23 },
24     "Lesson 1 - A simple topic-new_topic_127cdcdbbb4a480ea113c5101f30":
25         "type": "event",
26         "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
27         "description": "Lesson 1 - This lesson will show you how yo
28     ],
29     "Lesson 2 - A simple topic with a condition and variable-new_":
30         "type": "event",
31         "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
32         "description": "Lesson 2 - This lesson will show you how yo
33         "value": {
34             "$ref": "#/definitions/Lesson+2---A+simple+topic+with+a+
35         }
36     ],
37     "Lesson 3 - A topic with a condition, variables and a pre-bu":
38         "type": "event",
39         "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
40         "description": "Lesson 3 - This lesson will show you how yo
41     ],
42     "Lesson 4 - A topic with a condition, variables and custom e":
43         "type": "event",
44         "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30"
45         "description": "Lesson 4 - This lesson continues to show yo
46     ]
47 },
48 "definitions": {
49     "Lesson+2---A+simple+topic+with+a+condition+and+variable-new_":
50         "type": "object",
51         "properties": {
52             "pva_StoreLocation": {
53                 "type": "string"
54             }
55         }
56 }
```

Close

6. Select **Close**.

7. Under the **Activity** section, select **Show code**.

Connect to a skill

Skill Dialog

Begin a remote skill.

[Learn more](#)

Add a note

Skill Dialog Name * ②

PVA bot

[Show skill manifest](#)

Skill Endpoint ②

Power Virtual Agents Skill TEST

OAuth connection name (SSO) ②

abc =settings.connectionName

Activity ②

[Show code](#)

Text +

Responses ②

[Add alternative](#)

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 the following the example, the Composer variable `dialog.storeLocation` provides a value to the input variable `pva_StoreLocation` in the Microsoft Copilot

Studio copilot topic

```
dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f309089_21a34f16.
```



The screenshot shows the Microsoft Bot Framework Composer interface. A node titled "Activity" is selected, indicated by a yellow border. The code within the node is as follows:

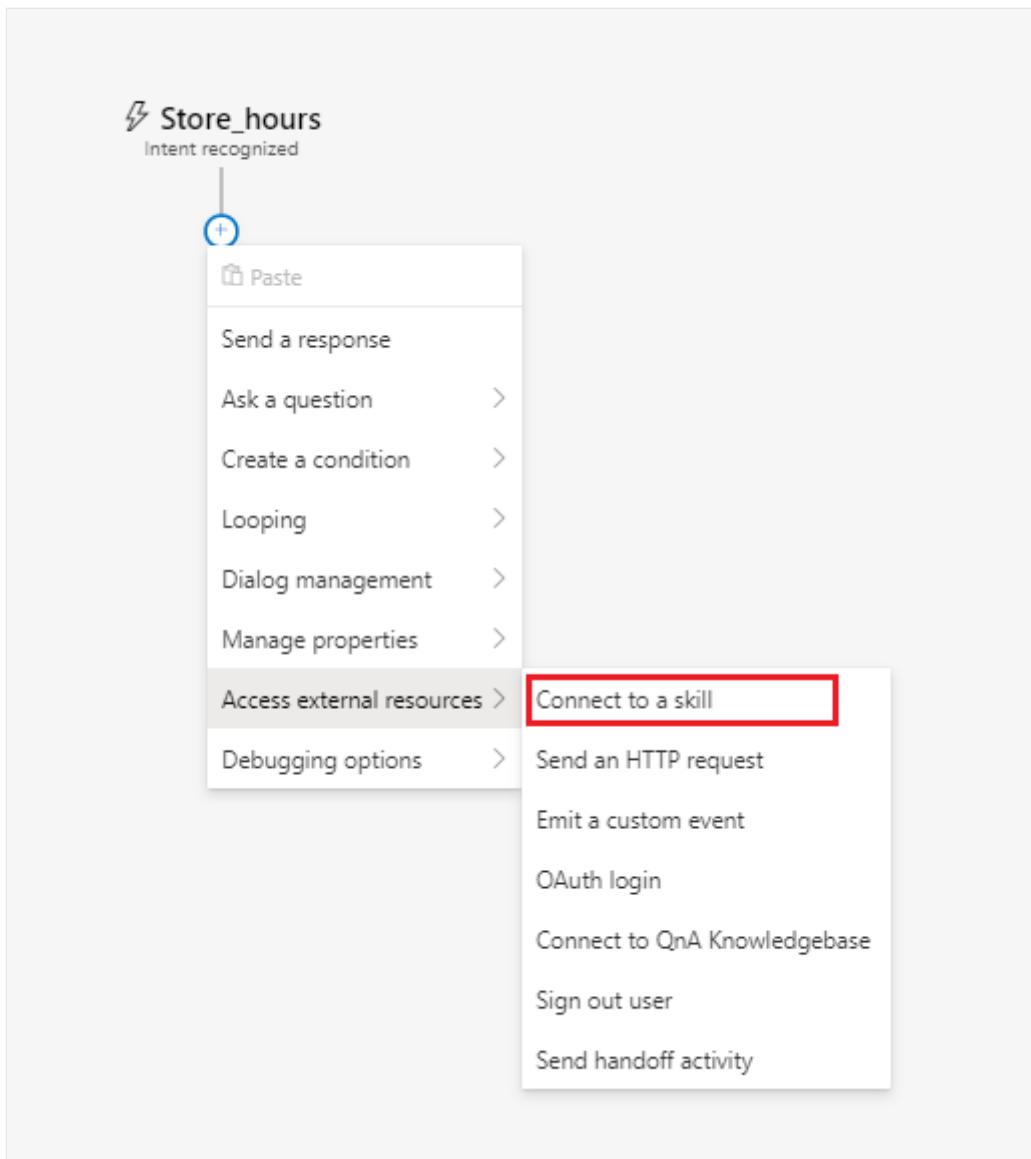
```
[Activity
  type = event
  name = dispatchTo_new_topic_bb76d062e0dc4d109ad9b9d588168c2d
  value = ${addProperty(json("{}"), 'pva_StoreLocation', dialog.storeLocation)}]
```

At the top right of the node, there is a "Show response editor" link.

Receive an output variable from a Microsoft Copilot Studio skill topic

It's possible to receive outputs from a Microsoft Copilot Studio skill topic in Composer. To learn how to create Microsoft Copilot Studio topics that return outputs, see [Passing variables between topics](#).

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 Microsoft Copilot Studio copilot.

A screenshot of the Microsoft Copilot Studio interface showing the 'Connect to a skill' configuration dialog. On the left, there's a preview of the dialog flow: 'Intent recognized' leads to 'Connect to a skill' (Host: settings.skill[pvaBot].endpointUrl). On the right, the configuration panel shows: 'Skill Dialog' (Begin a remote skill), 'Add a note' (text area), 'Skill Dialog Name' (dropdown set to 'PVA bot', highlighted with a red box), 'Show skill manifest' (button), 'Skill Endpoint' (dropdown set to 'Power Virtual Agents Skill TEST').

3. Find the event activity's name for your Microsoft Copilot Studio bot 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 Microsoft Copilot Studio topic doesn't have outputs, it won't have a **resultValue** property.



The screenshot shows a code editor window titled "PVA bot Manifest". The code is a JSON object representing a manifest. It contains several "Lesson" objects, each with a "type": "event", "name", and "description". The "Lesson 3" object includes a "resultValue" property, which is highlighted with a red box. This property contains a "\$ref" value pointing to a nested "definitions" object. The "definitions" object contains a "Lesson+2+-+A+simple+topic+with+a+condition+and+variable-new" entry, which is also highlighted with a red box. The code is numbered from 30 to 64 on the left side. A "Close" button is visible at the bottom right of the editor.

```
30  "type": "event",
31  "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30",
32  "description": "Lesson 2 - This lesson will show you how yo",
33  "value": {
34    "$ref": "#/definitions/Lesson+2+-+A+simple+topic+with+a+",
35  },
36 },
37 "Lesson 3 - A topic with a condition, variables and a pre-bu,
38   "type": "event",
39   "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30",
40   "description": "Lesson 3 - This lesson will show you how yo",
41   "resultValue": {
42     "$ref": "#/definitions/Lesson+3+-+A+topic+with+a+conditi",
43   },
44 },
45 "Lesson 4 - A topic with a condition, variables and custom e,
46   "type": "event",
47   "name": "dispatchTo_new_topic_127cdcdbbb4a480ea113c5101f30",
48   "description": "Lesson 4 - This lesson continues to show yo",
49 },
50 "Lesson 3 - A topic with a condition, variables and a pre-bu,
51   "type": "event",
52   "name": "dispatchTo_new_topic_b54284df1e2e4d15ac8ed4bbd8d2",
53   "description": "Lesson 3 - This lesson will show you how yo",
54   "resultValue": {
55     "$ref": "#/definitions/Lesson+3+-+A+topic+with+a+conditi",
56   },
57 },
58 },
59 "definitions": {
60   "Lesson+2+-+A+simple+topic+with+a+condition+and+variable-new": {
61     "type": "object",
62     "properties": {
63       "pva_StoreLocation": {
64         "type": "string"
65       }
66     }
67   }
68 }
```

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 Microsoft Copilot Studio topic's output variables. You'll use them in the next steps.

In the following example, the Microsoft Copilot Studio topic returns two outputs of type String, **pva_State** and **pva_Item**.

PVA bot Manifest

```
40     "description": "Lesson 4 - This lesson continues to show you how to use variables and conditions in your bot's logic. You will learn how to use loops and decisions to make your bot more intelligent and responsive."},  
41 },  
42 "Lesson 3 - A topic with a condition, variables and a pre-build message": {  
43     "type": "event",  
44     "name": "dispatchTo_new_topic_b54284df1e2e4d15ac8ed4bbd8d2",  
45     "description": "Lesson 3 - This lesson will show you how to use variables and conditions in your bot's logic. You will learn how to use loops and decisions to make your bot more intelligent and responsive.",  
46     "resultValue": {  
47         "$ref": "#/definitions/Lesson+3---A+topic+with+a+condition%2c+variables+and+a+pre-build+message"  
48     }  
49 },  
50 },  
51 "definitions": {  
52     "Lesson+2---A+simple+topic+with+a+condition+and+a+variable+new": {  
53         "type": "object",  
54         "properties": {  
55             "pva_StoreLocation": {  
56                 "type": "string"  
57             }  
58         }  
59     },  
60     "Lesson+3---A+topic+with+a+condition%2c+variables+and+a+pre-build+message": {  
61         "type": "object",  
62         "properties": {  
63             "pva_State": {  
64                 "type": "string"  
65             },  
66             "pva_Item": {  
67                 "type": "string"  
68             }  
69         }  
70     },  
71     "Lesson+3---A+topic+with+a+condition%2c+variables+and+a+pre-build+message": {  
72         "type": "object",  
73         "properties": {  
74             "pva_State": {  
75                 "type": "string"  
76             }  
77         }  
78     },  
79     "Lesson+3---A+topic+with+a+condition%2c+variables+and+a+pre-build+message": {  
80         "type": "object",  
81         "properties": {  
82             "pva_State": {  
83                 "type": "string"  
84             }  
85         }  
86     }  
87 }  
88 }  
89 }  
90 }
```

Close

6. Select **Close**.

7. In the **Activity** section, select **Show code**.

Connect to a skill

Skill Dialog

Begin a remote skill.

[Learn more](#)

Add a note

Skill Dialog Name * ②

PVA bot



[Show skill manifest](#)

Skill Endpoint ②

Power Virtual Agents Skill TEST



OAuth connection name (SSO) ②

abc =settings.connectionName

Activity ②

[Show code](#)

Text +

Responses ②



[Add alternative](#)

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
]
```

Activity ②

Show response editor

```
1 [Activity
2   Type = event
3   Name = dispatchTo_new_topic_b54284df1e2e4d15ac8
4 ]
```

9. Under the code area, set **Property** to a Composer variable to receive and store the output values from the Microsoft Copilot Studio skill topic.

Activity ②

Show response editor

```
1 [Activity
2   Type = event
3   Name = dispatchTo_new_topic_b54284df1e2e4d15ac8ed4
4 ]
```

Template name: #BeginSkill_Activity_C6Dhnd()

Allow interruptions ②

y/n	false
-----	-------

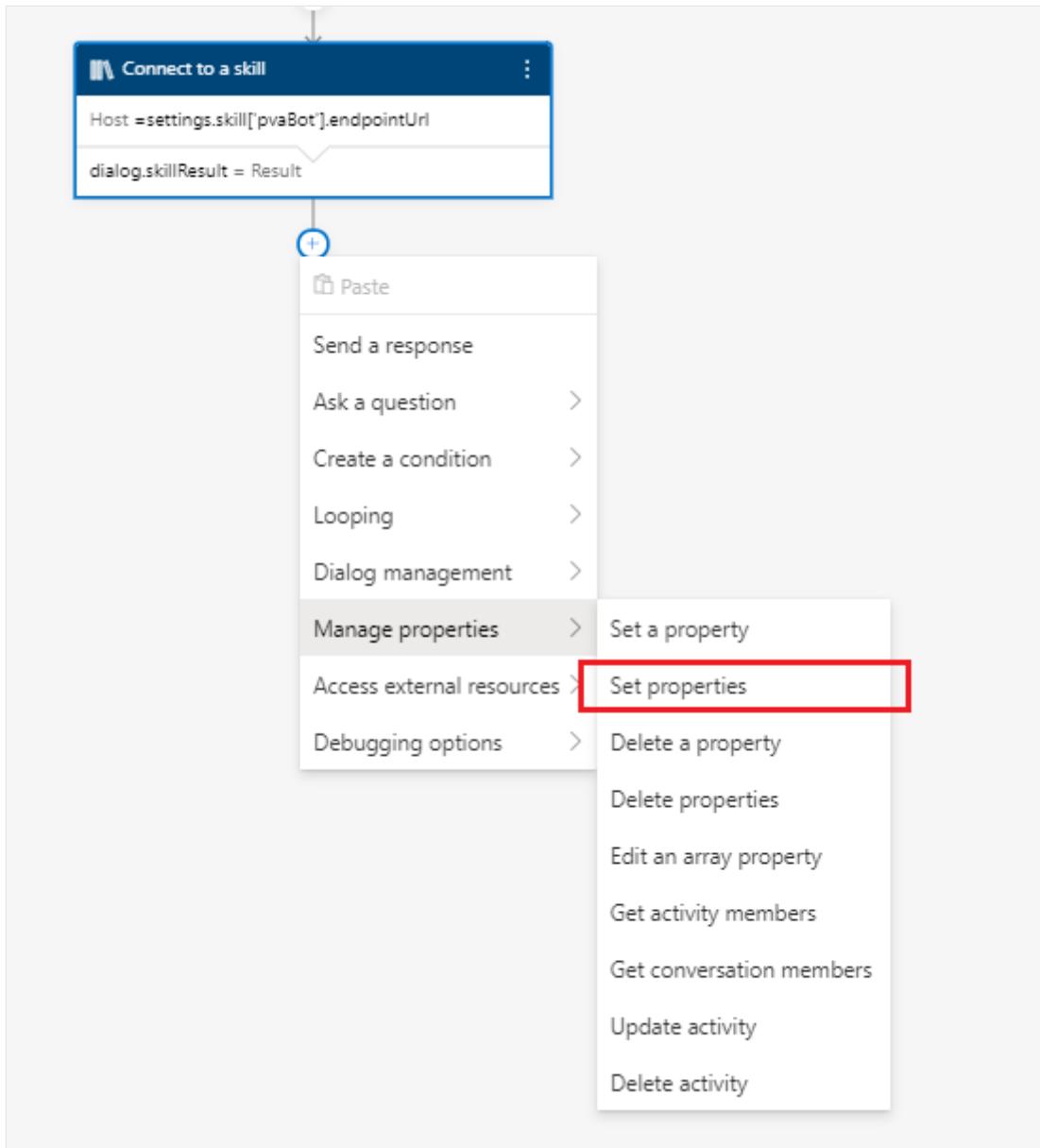
Property ②

abc	dialog.skillResult
-----	--------------------

Activity processed ②

y/n	true
-----	------

10. In the Composer authoring canvas, select **Add**, then select **Manage properties**, and then select **Set properties**.



11. 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 the following example, the output values from `dialog.skillResult.pvaState` and `dialog.skillResult.pvalItem` are extracted into new Composer variables `dialog.State` and `dialog.Item`.

Set properties

Set Properties

Set one or more property values.

[Learn more](#)

Add a note

Assignments * ②

Property ②

abc dialog.State

Value ②

abc dialog.skillResult.pva_State

Property ②

abc dialog.item

Value ②

abc dialog.skillResult.pva_Item

[Add new](#)

Test your Microsoft Copilot Studio copilot as a skill with a Bot Framework bot

Use the [Bot Framework Emulator](#) to test that your Bot Framework bot is properly calling a Microsoft Copilot Studio copilot as a skill.

ⓘ Important

Your Bot Framework bot must be added to the Microsoft Copilot Studio copilot'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

Overview of Bot Framework Composer

Article • 06/05/2024

ⓘ 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.

ⓘ Note

Microsoft Copilot Studio now [supports these features](#) without Composer.

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:

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 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.

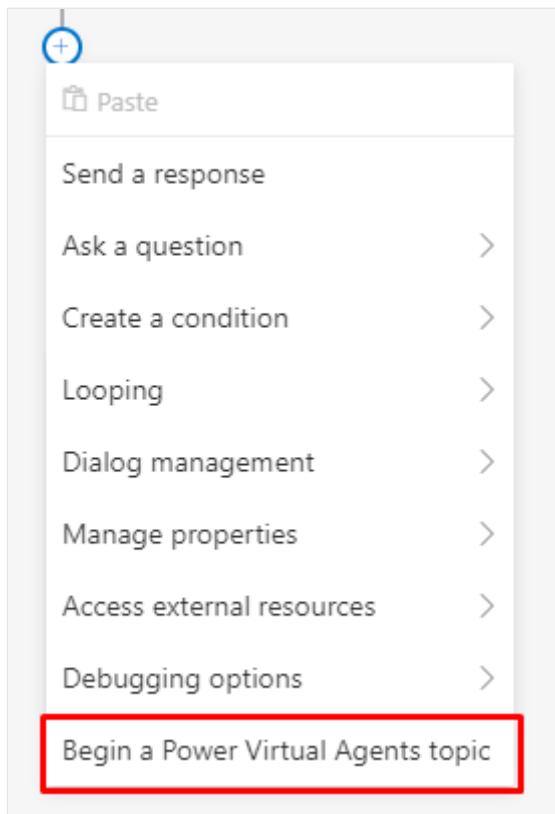
Guidelines for creating content for Copilot Studio in Composer

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

- [Getting started with Bot Framework Composer](#)
-

Feedback

Was this page helpful?



[Provide product feedback](#)

Getting started with Bot Framework Composer

Article • 05/01/2024

ⓘ 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.

ⓘ Note

Microsoft Copilot Studio now [supports these features](#) without Composer.

To use Bot Framework Composer to extend your Microsoft Copilot Studio bot, you'll need to first import your bot into a Composer project.

Prerequisites

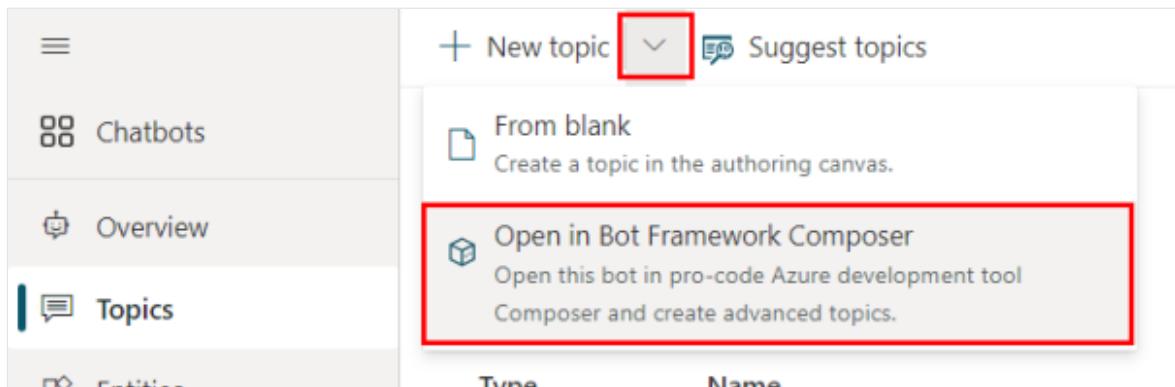
- Extend your bot with Bot Framework Composer

Open your bot in Composer

ⓘ Important

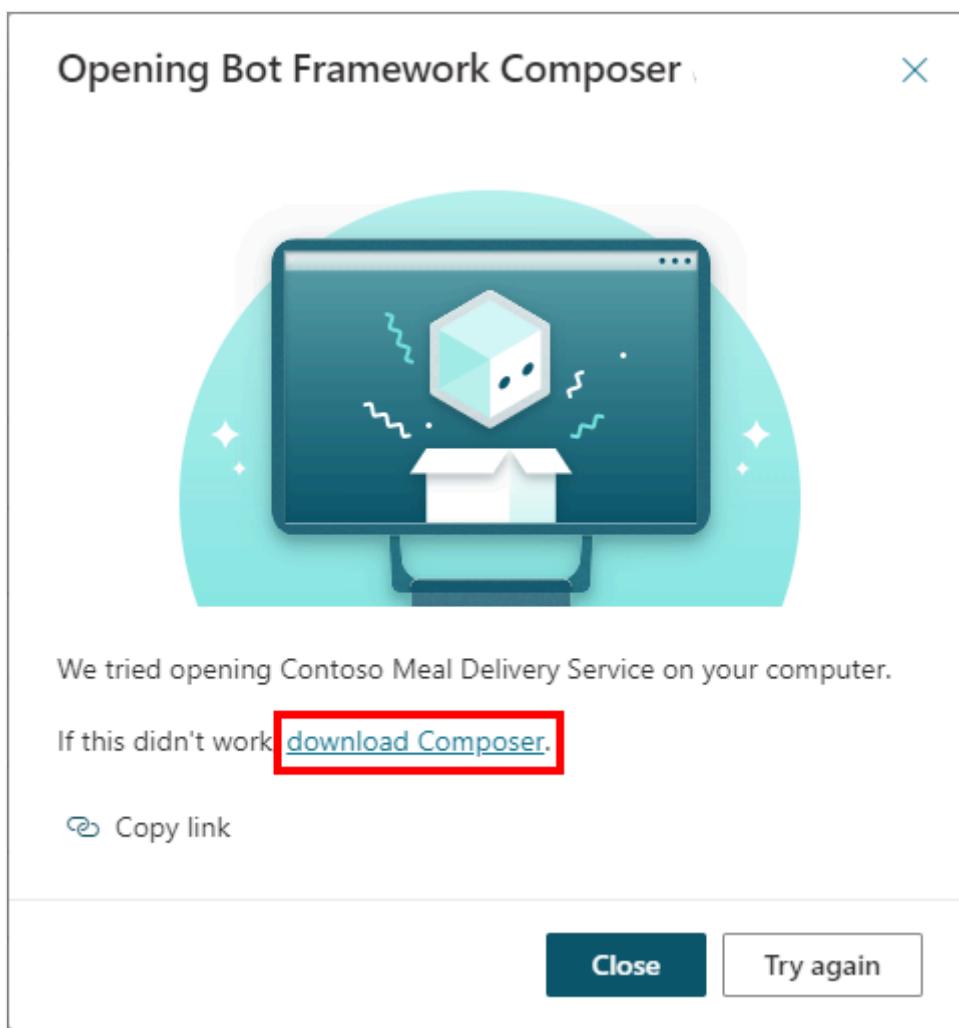
You should always open Composer from within Microsoft Copilot Studio. This will ensure that Composer has all the necessary plug-ins needed for integration with Microsoft Copilot Studio.

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:

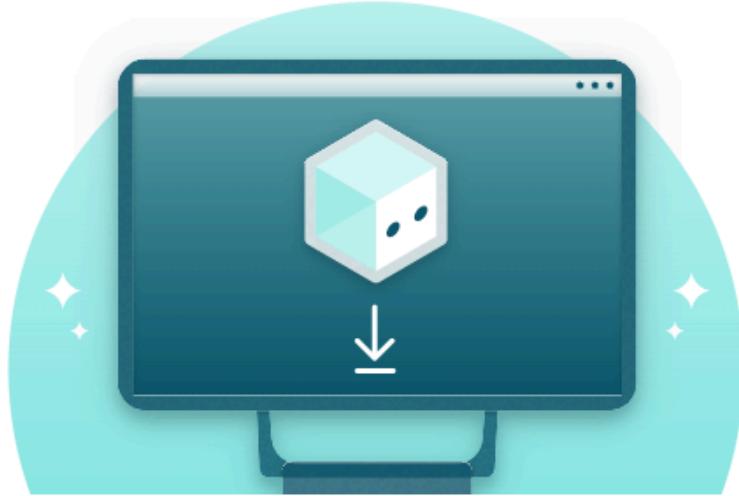
- Select **download Composer**.



- 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**.

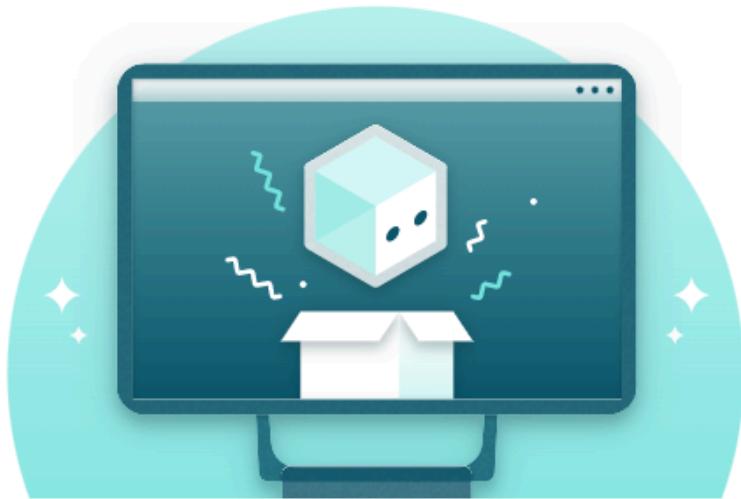
① Note

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

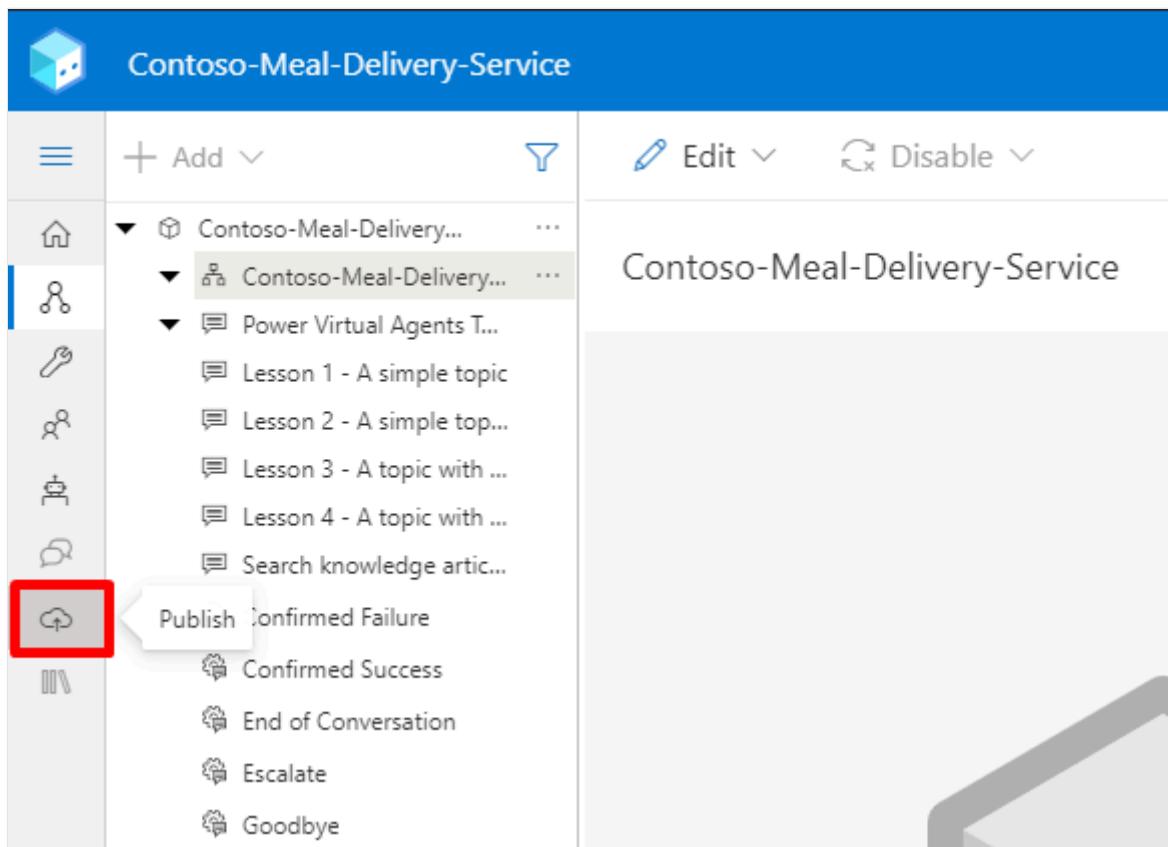
You can't use the Bot Framework Emulator to test Composer content created for Microsoft Copilot Studio bots. Instead you'll need to publish your Composer content to your Microsoft Copilot Studio bot, then use the [Test bot pane](#).

Important

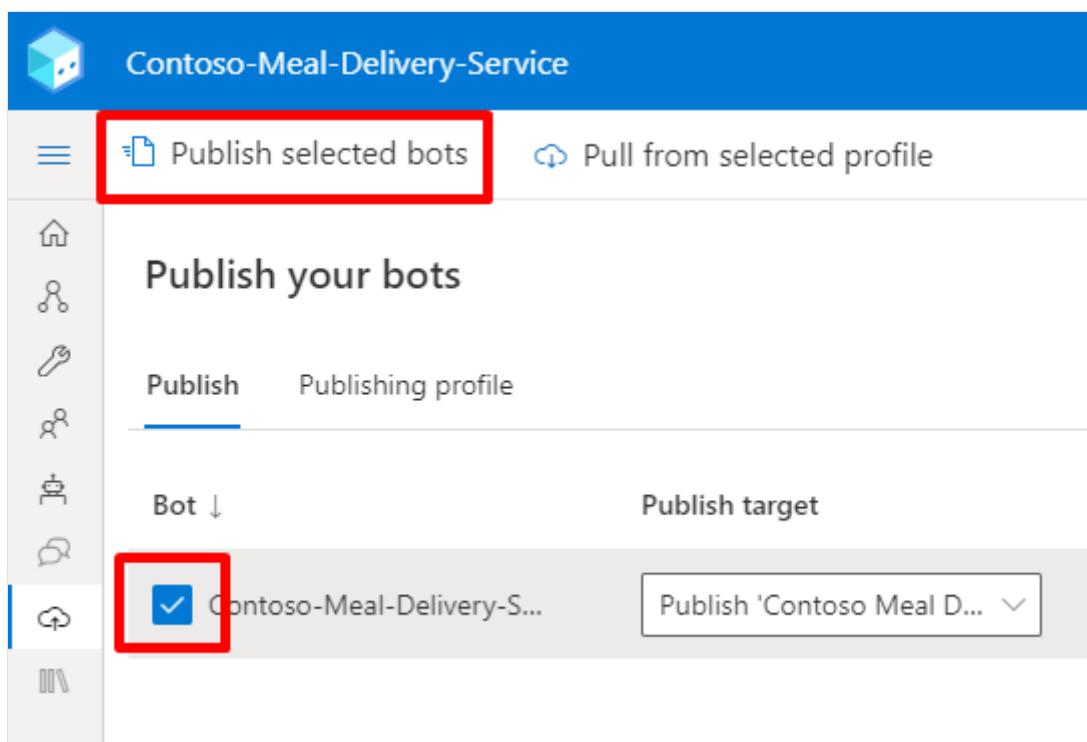
Selecting **Publish** in Composer makes the changes available for testing, but doesn't automatically publish your bot.

[Publish](#) your bot in Microsoft Copilot Studio to update your bot across all the channels it's connected to.

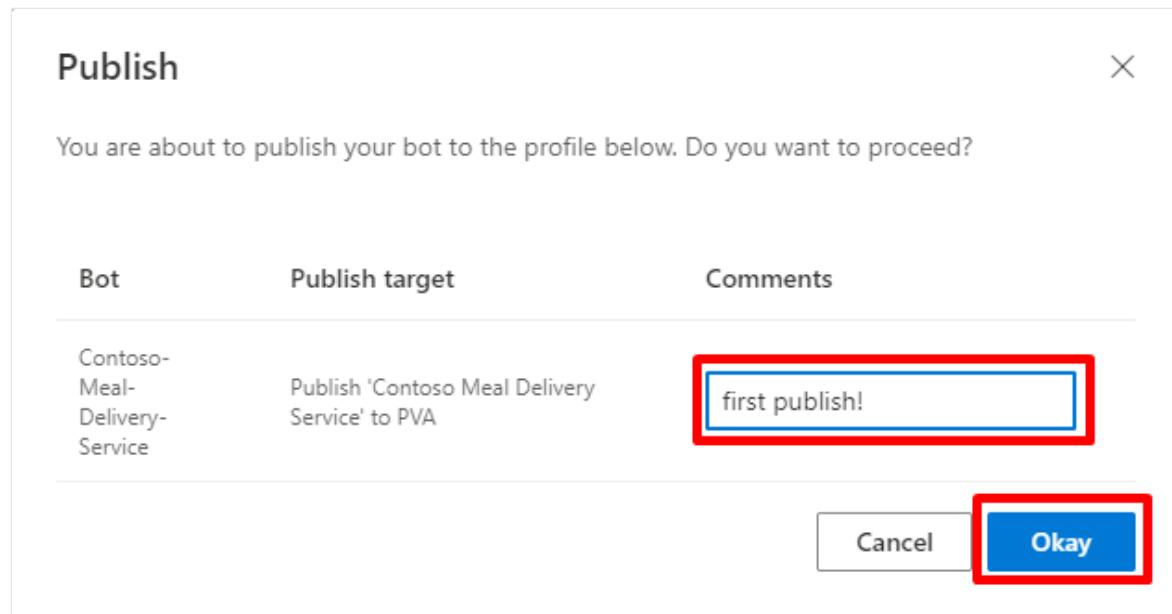
1. Go to the **Publish** page.



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 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 "Comment" column of the first row in the "Publish history" section.

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:

- Example 1 - Show an Adaptive Card in Microsoft Copilot Studio
 - Example 2 - Display a multi-select options list in Microsoft Copilot Studio
 - Example 3 - Use Bing Search as a fallback in Microsoft Copilot Studio
 - Example 4 - Display a form with a Submit button in Microsoft Copilot Studio
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Capture telemetry with Application Insights

Article • 06/04/2024

ⓘ 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 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 ↗](#)

Example 1 - Show an Adaptive Card in Microsoft 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.

ⓘ Note

Microsoft Copilot Studio now [supports these features](#) without Composer.

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](#).
- [Composer fundamentals](#).

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 Microsoft Bot Framework Composer interface. On the left, a card editor window is open, displaying a question card with the following fields:

- Question:** Ask a question
- Text:** What city are you in?
- Identify:** City
- Save response as:** {x} bot.user_city (city)

A vertical dashed line connects this card to a variable properties dialog on the right.

Variable properties

Name *
bot. user_city
Modifying the variable name will rename it in every topic.

Type
City

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 [Getting started with Bot Framework Composer](#).
2. Go to the **Create** page. In your bot, select **More options (...)** then select **+ Add a dialog**.

The screenshot shows the Microsoft Bot Framework's Bot Explorer interface. At the top, there's a blue header bar with the title "Contoso-Meal-Delivery-Service". Below the header, there's a navigation sidebar on the left with icons for Home, Bot, Power Virtual Agents, and Chat. The main area displays a tree structure of the bot's components. Under the root node "Contoso-Meal-Delivery-Service", there are three children: another "Contoso-Meal-Delivery-Service" node, "Power Virtual Agents Topics (13)", and a list of "Lesson" nodes. A context menu is open over the first "Contoso-Meal-Delivery-Service" node, with options like "...", "Edit", "Disable", "Add a dialog", "Export as .zip", and "Bot settings". The "Add a dialog" option is highlighted with a red box.

3. For **Name**, enter `Meals` then select **OK**.

The screenshot shows a modal dialog titled "Create a dialog" with a close button "X" in the top right corner. The main content area has a heading "Specify a name and description for your new dialog." Below it, there are two input fields: "Name *" which contains the value "Meals" and "Description" which is currently empty. At the bottom of the dialog are two buttons: "Cancel" and "OK". The "Name" input field and the "OK" button are both highlighted with red boxes.

4. Go to the **Bot Responses** page. In the bot explorer, select **Meals**, then select **Show code**.

The screenshot shows the Microsoft Bot Framework Composer interface. On the left, there's a sidebar with icons for Home, Create, Configure, User input, Bot responses (which is highlighted with a red box), Knowledge base, Publish, and Package manager. The main area is titled "Bot Responses". It has a tree view on the left with "Hello-World" expanded, showing "Common", "Hello-World", and "Meals" (which is also highlighted with a red box). To the right, there's a table with columns "Name" and "Responses". A button "Show code" is located in the top right corner of the main area.

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

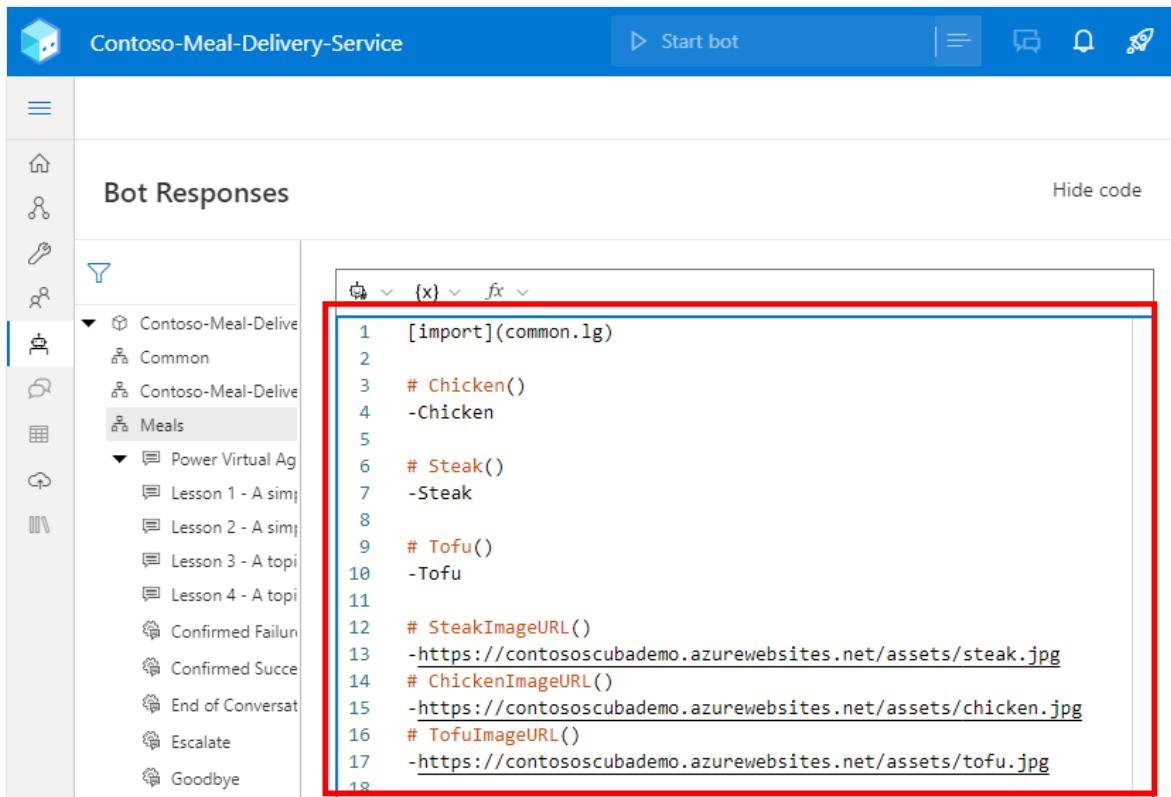
# Steak()
-Steak

# 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. In the top navigation bar, there's a blue header with the title "Contoso-Meal-Delivery-Service". To the right of the title are buttons for "Start bot", "More options", "Copy", "Share", and "Send". On the left side, there's a sidebar with various icons and a tree view of the bot's structure. The main area is titled "Bot Responses" and contains a code editor with the following content:

```
1 [import](common.lg)
2
3 # Chicken()
4 -Chicken
5
6 # Steak()
7 -Steak
8
9 # Tofu()
10 -Tofu
11
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
```

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. In the top navigation bar, there's a blue header with the title "Contoso-Meal-Delivery-Service". To the right of the title are buttons for "Start bot", "More options", "Copy", "Share", and "Send". On the left side, there's a sidebar with various icons and a tree view of the bot's structure. The main area is titled "Bot response" and contains the following code:

```
# adaptivecardjson_meals(location)
- ``
{
    "type": "AdaptiveCard",
    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
    "version": "1.1",
    "body": [
        {
            "type": "TextBlock",
            "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-Delivery-Service

Common

Contoso-Meal-Delivery-Service

Meals

Power Virtual Agent

- Lesson 1 - A simple meal
- Lesson 2 - A simple meal
- Lesson 3 - A topic
- Lesson 4 - A topic

Confirmed Failure

Confirmed Success

End of Conversation

Escalate

Goodbye

Greeting

Start over

Thank you

```
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()}",
"size": "Stretch",
43 }
44]
45 }
46]
47 }
48]
49 }
```

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. On the left, there's a sidebar with icons for Home, Bot Explorer, and other tools. The main area is titled "Bot Responses". In the "Bot Explorer" section, under "Contoso-Meal-Delivery-Service", there's a tree view with "Common", "Contoso-Meal-Delivery-Service", and "Meals" expanded. Under "Meals", there are several nodes: "Power Virtual Agents 1", "Lesson 1 - A simple topic", "Lesson 2 - A simple topic", "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". The right side is the "Code Editor" with the following JSON code:

```

93 "type": "TextBlock",
94 "text": "${Chicken()}",
95 "wrap": true,
96 "horizontalAlignment": "Center"
97 }
98],
99 {
100 "type": "Column",
101 "width": "stretch",
102 "items": [
103 {
104 "type": "TextBlock",
105 "text": "${Tofu()}",
106 "wrap": true,
107 "horizontalAlignment": "Center"
108 }
109]
110 }
111 }
112]
113 }
114 }
115 ...
116
117
118
119 # AdaptiveCardMeals(location)
120 [Activity
121 Attachments = ${json(adaptivecardjson_meals(location))}
122]

```

A red box highlights the last three lines of the code.

## 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, there's a navigation bar: Meals > BeginDialog > Send a response. To the right of the navigation is a "Show code" button. The main area displays a flowchart with a "BeginDialog" event at the top, followed by a "Send a response" step, and finally an end node at the bottom. The "Send a response" step is highlighted with a blue border. To the right of the flowchart is a panel titled "Send a response" with sub-sections for "Send Activity", "Learn more", "Add a note", "Bot responses", and "Responses". A red box highlights the "Show code" button in the "Responses" section.

### ⚠️ Warning

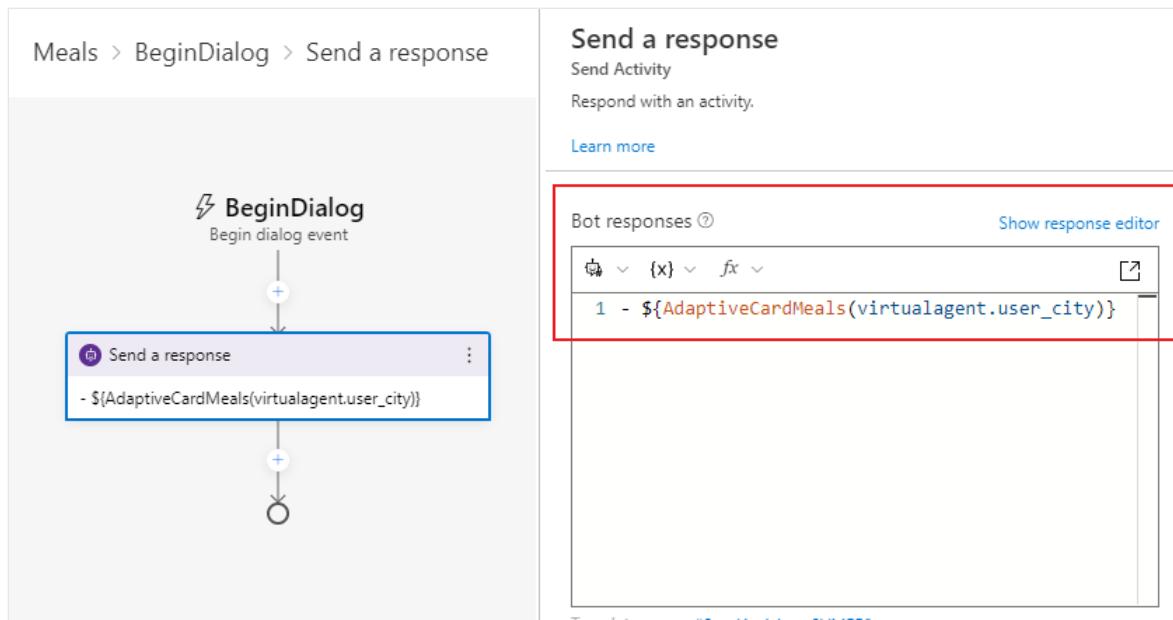
Adding the expression in the next step to the *response* editor instead of the *code* editor will result in the bot responding with raw JSON instead of an Adaptive Card.

4. Copy and paste the following expression into the code editor.

The screenshot shows a "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.

**ⓘ Important**

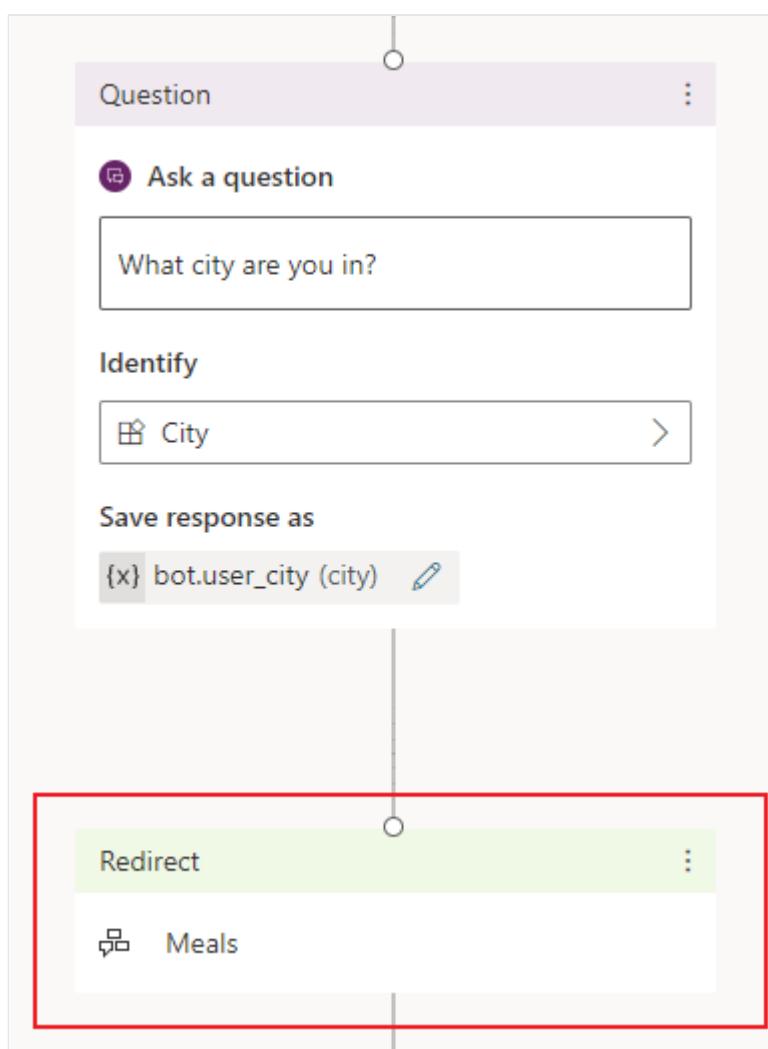
Selecting **Publish** in Composer makes the changes available for testing, but doesn't automatically publish your bot.

[\*\*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 Microsoft Bot Framework Composer interface. On the left, there's a sidebar with icons for Chatbots, Overview, Topics (which is selected and highlighted in blue), Entities, Analytics, Publish, Settings, and Test your bot. The main area is titled "Topics" with a circular info icon. It has two tabs: "Existing (14)" (underlined) and "Suggested (0)". Below the tabs is a table with three columns: "Type", "Name", and "Trigger phrases". There are two rows in the table. The first row, "Meals", is highlighted with a red border. Its details are shown in the table: Type is a document icon, Name is "Meals", and Trigger phrases are "No trigger phrases". The second row, "Meal delivery options", has a document icon, Name is "Meal delivery options", and Trigger phrases are "(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**.

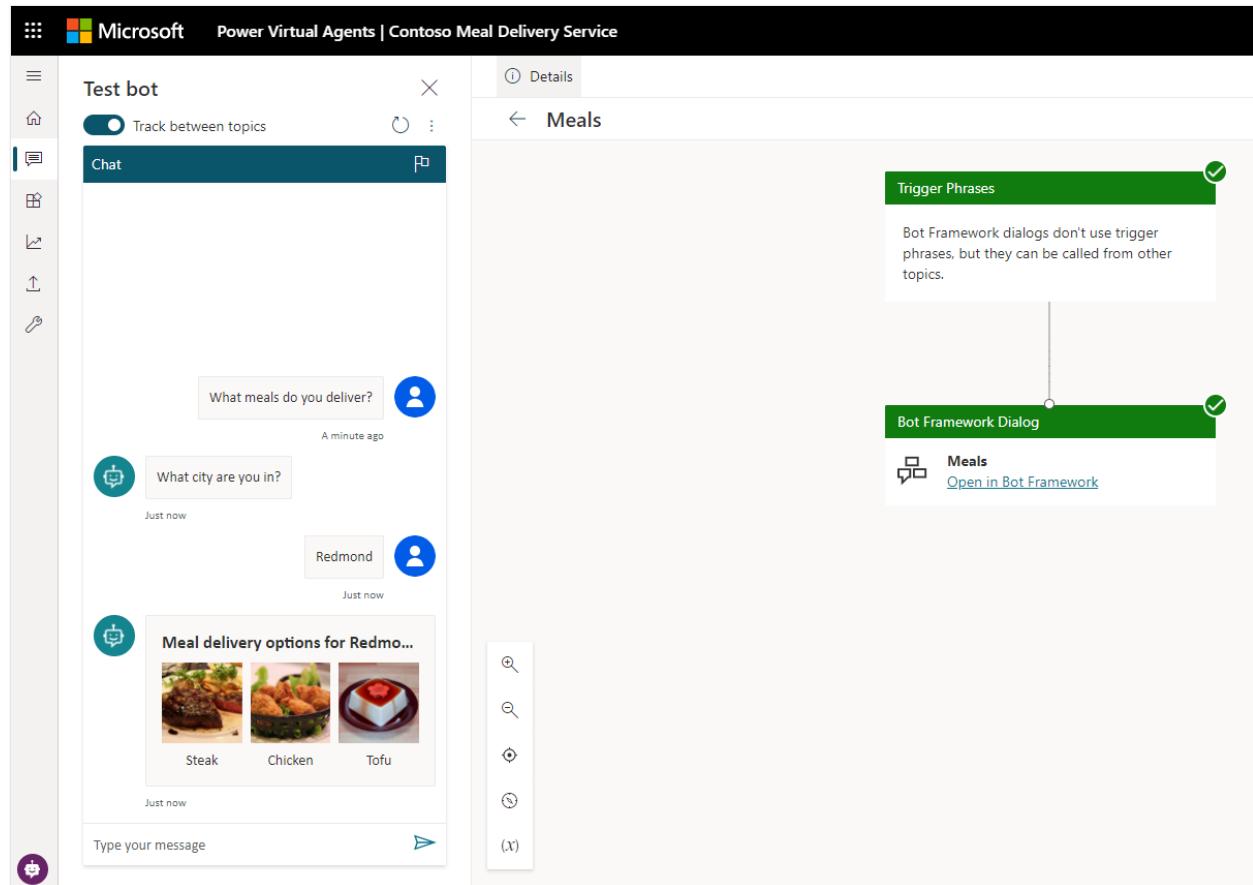


5. Select **Save** to save the changes to your topic.

# 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 Steps

[Example 2 - Display a multi-select options list in Microsoft Copilot Studio.](#)

# Example 2 - Display a multi-select options list in Microsoft Copilot Studio

Article • 05/01/2024

## ⓘ 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.

## ⓘ Note

Microsoft Copilot Studio now [supports these features](#) without Composer.

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

- Example 1 - Show an Adaptive Card in Microsoft Copilot Studio.

## Create a new dialog

1. In Microsoft Copilot Studio, open the bot you created in Example 1. If you haven't completed Example 1 yet, see [Show an Adaptive Card in Microsoft Copilot Studio](#).

2. Open your bot in Composer. For instructions on how to do so, see [Getting 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`.

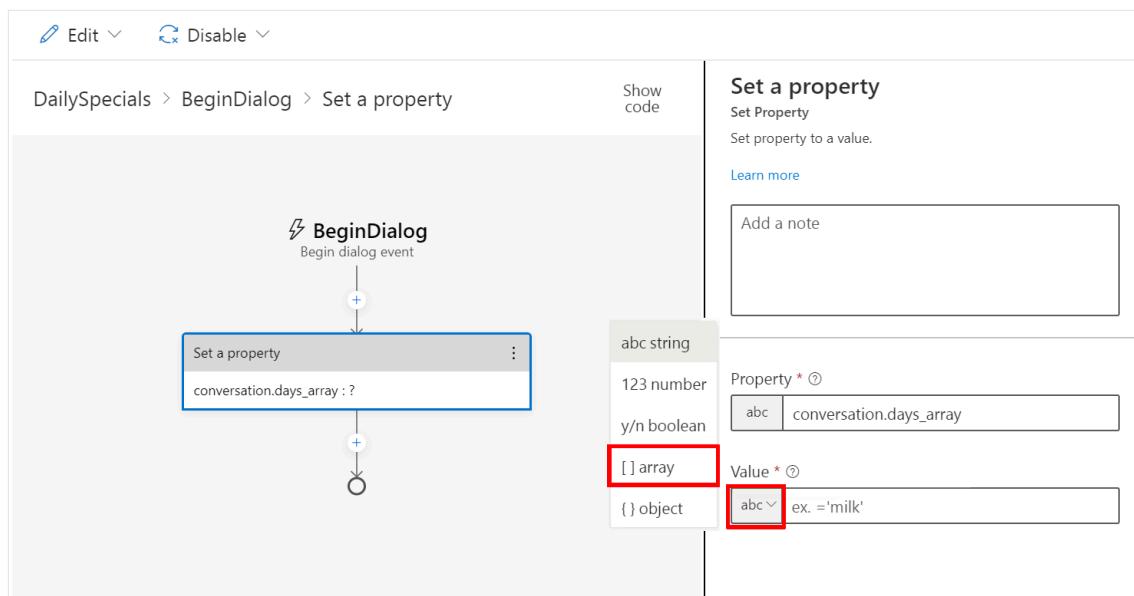
4. In your new **DailySpecials** dialog, select the **BeginDialog** trigger to open the authoring canvas.

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"]
```

The screenshot shows the Microsoft Bot Framework Composer interface. On the left, there's a navigation sidebar with icons for Home, Bot, Lessons, Power Virtual Agents Topics, and Settings. The main area shows a hierarchy: Contoso-Meal-Delivery-S... > Meals > DailySpecials > BeginDialog > Set a property. A tooltip for the 'BeginDialog' node says 'Begin dialog event'. The 'Set a property' node has the code: 'conversation.days\_array : ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]'. To the right, there's a panel titled 'Set a property' with sections for 'Set Property' (with a note input field), 'Property \*' (set to 'abc conversation.days\_array'), and 'Value \*' (set to '["Thursday", "Friday", "Saturday"]').

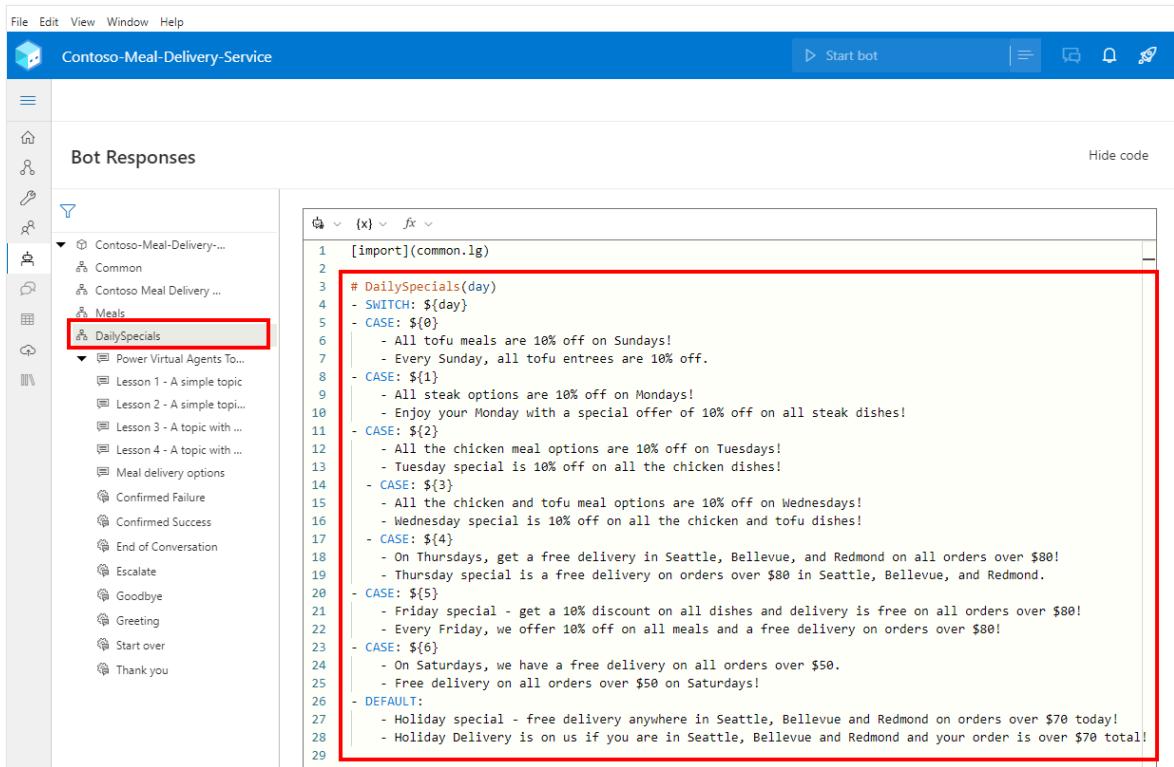
## Configure bot responses

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!



The screenshot shows the Microsoft Bot Framework Composer interface. The top navigation bar includes File, Edit, View, Window, Help, and a Start bot button. The left sidebar contains icons for Home, Bot, User, Power Virtual Agents, Lessons, Cloud, and Bot Explorer. The main area is titled "Bot Responses". A tree view on the left lists various dialogs: Contoso-Meal-Delivery..., Common, Contoso Meal Delivery ... (with a red box around it), Meals, DailySpecials (with a red box around it), Power Virtual Agents To..., Lesson 1 - A simple topic, Lesson 2 - A simple topic with ..., Lesson 3 - A topic with ..., Lesson 4 - A topic with ..., Meal delivery options, Confirmed Failure, Confirmed Success, End of Conversation, Escalate, Goodbye, Greeting, Start over, and Thank you. The "DailySpecials" dialog is selected and expanded, showing its code:

```

1 [import](common.1g)
2
3 # DailySpecials(day)
4 - SWITCH: ${day}
5 - CASE: ${0}
6 - All tofu meals are 10% off on Sundays!
7 - Every Sunday, all tofu entrees are 10% off.
8 - CASE: ${1}
9 - All steak options are 10% off on Mondays!
10 - Enjoy your Monday with a special offer of 10% off on all steak dishes!
11 - CASE: ${2}
12 - All the chicken meal options are 10% off on Tuesdays!
13 - Tuesday special is 10% off on all the chicken dishes!
14 - CASE: ${3}
15 - All the chicken and tofu meal options are 10% off on Wednesdays!
16 - Wednesday special is 10% off on all the chicken and tofu dishes!
17 - CASE: ${4}
18 - On Thursdays, get a free delivery in Seattle, Bellevue, and Redmond on all orders over $80!
19 - Thursday special is a free delivery on orders over $80 in Seattle, Bellevue, and Redmond.
20 - CASE: ${5}
21 - Friday special - get a 10% discount on all dishes and delivery is free on all orders over $80!
22 - Every Friday, we offer 10% off on all meals and a free delivery on orders over $80!
23 - CASE: ${6}
24 - On Saturdays, we have a free delivery on all orders over $50.
25 - Free delivery on all orders over $50 on Saturdays!
26 - DEFAULT:
27 - Holiday special - free delivery anywhere in Seattle, Bellevue and Redmond on orders over $70 today!
28 - Holiday Delivery is on us if you are in Seattle, Bellevue and Redmond and your order is over $70 total!
29

```

## 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:`

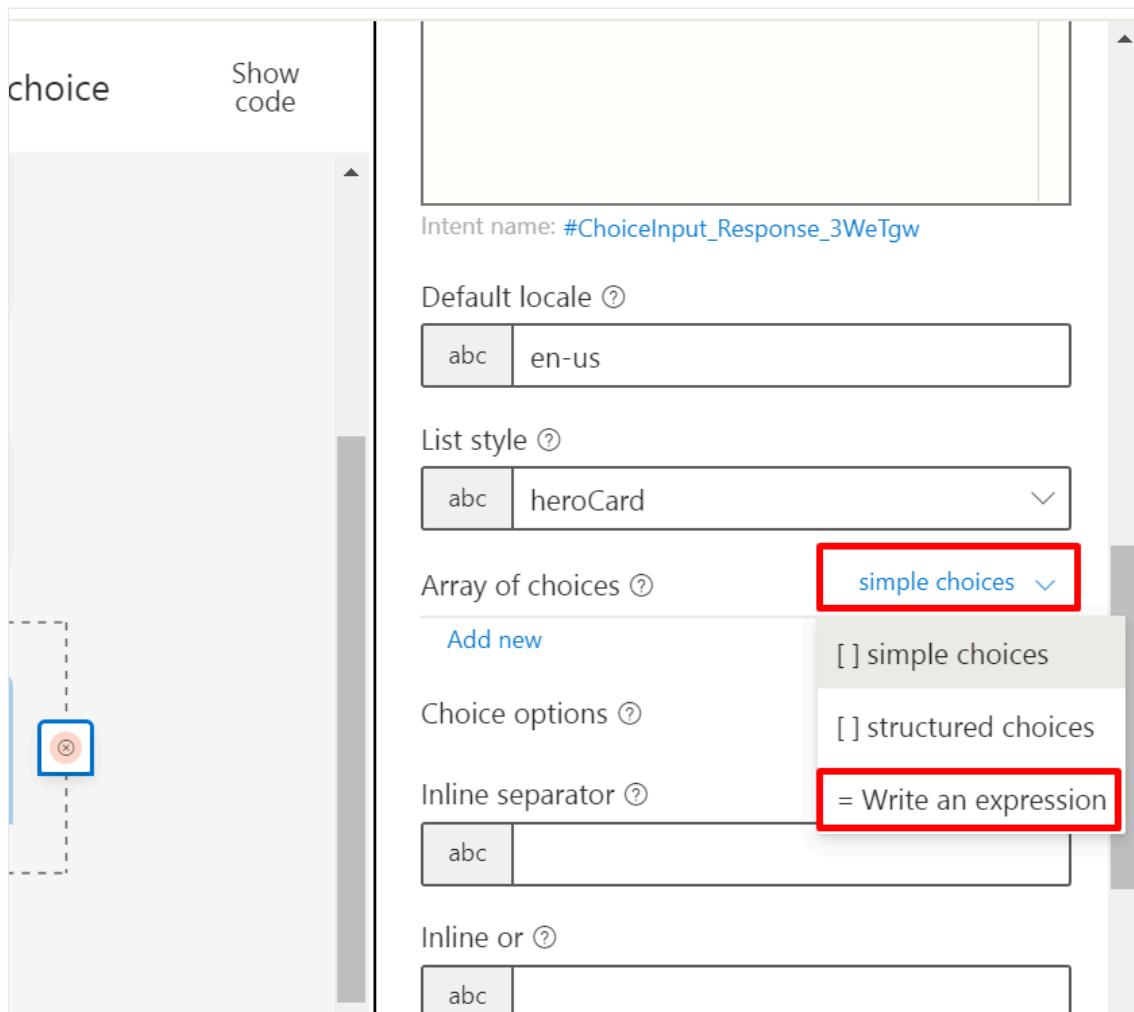
The screenshot shows a bot builder interface. On the left, a flowchart starts with a 'BeginDialog' event, followed by a 'Set a property' action setting 'conversation.days\_array' to an array of days, and finally a 'Prompt with multi-choice' action with the text 'Please select a day:'. On the right, the properties pane for the 'Prompt with multi-choice' step is open, showing the 'Bot response' tab selected. It displays the text 'Please select a day:'.

4. Select the **User Input** page in the properties pane, then do the following:

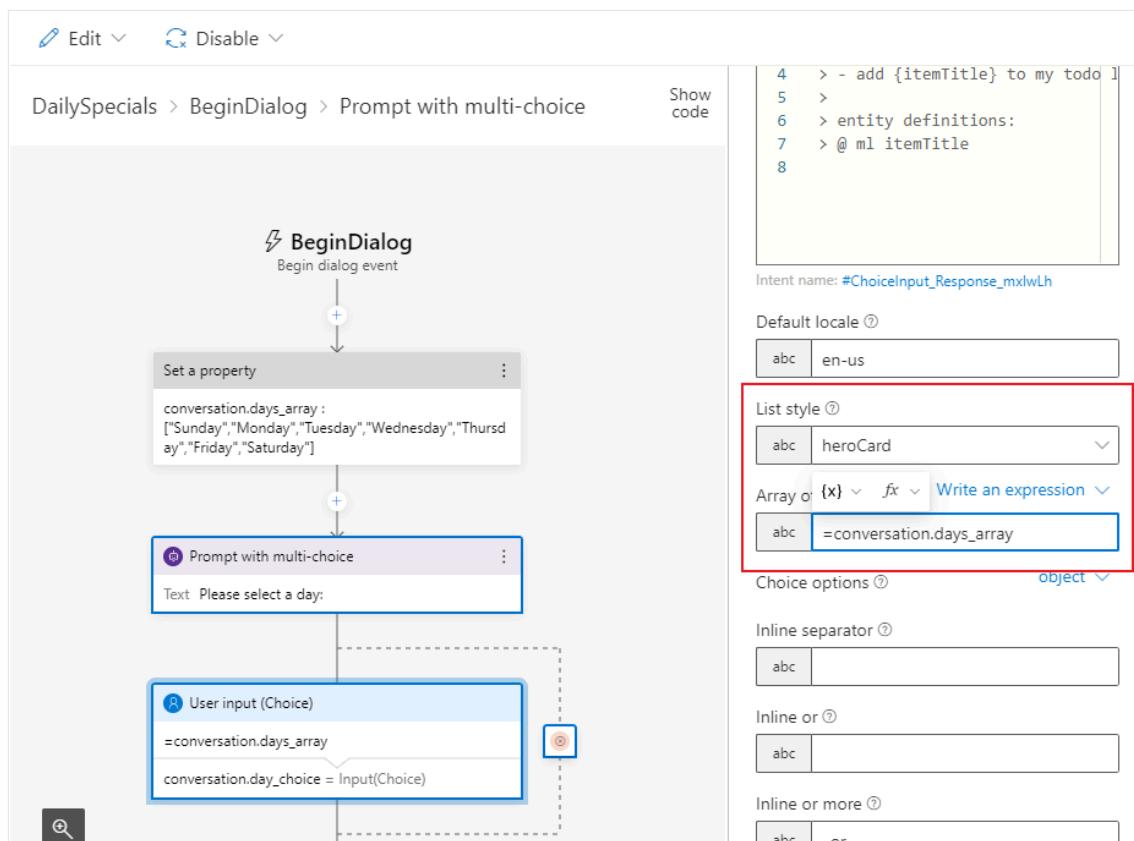
- For **Property**, enter `conversation.day_choice`.
- For **Output format**, choose **index** to return the index of the selected option instead of a value.

The screenshot shows the properties pane for the 'Prompt with multi-choice' step. The 'User input' tab is selected. It shows the 'Property' field set to `conversation.day_choice` and the 'Output format' field set to `index`. Below this, the 'Value' field contains the placeholder 'ex. hello world, Hello \${user.name}, ...'. On the right, the 'Expected responses' section lists several items, with the second item 'Please remind me to {itemTitle}' highlighted with a yellow background.

- For **List style**, choose **heroCard** to display our options list vertically.
- 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:

The screenshot shows the Microsoft Bot Framework Authoring Canvas. On the left, a dialog flow is displayed with the following steps:

- A **BeginDialog** event triggers a **Set a property** action. The property is named `conversation.days_array` and its value is set to `["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]`.
- This is followed by a **Prompt with multi-choice** step with the text `Please select a day:`. A **User input (Choice)** action is connected to it, setting `=conversation.days_array` and `conversation.day_choice = Input(Choice)`.
- The final step in the flow is a **Send a response** step, which contains the expression `- ${DailySpecials(conversation.day_choice)}`.

On the right side of the screen, the **Send a response** properties pane is open. Under **Bot responses**, the expression `- ${DailySpecials(conversation.day_choice)}` is listed. This expression is highlighted with a red box.

3. In the bot explorer, go to the main dialog. Select **More options (...)** then select **Add new trigger**.

The screenshot shows the Microsoft Bot Explorer interface. The main dialog tree on the left shows the following structure:

- Contoso-Meal-Delivery-Service** (selected)
- Contoso Meal Delivery Service**
- Meals**
  - BeginDialog**
- DailySpecials**

On the right, the **More options (...)** menu is open, and the **Add new trigger** option is highlighted with a red box.

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
```

## Create a trigger

What is the type of this trigger?

Intent recognized

What is the name of this trigger?

Specials

Trigger phrases

+ Add entity ▾   ➔ Insert entity ▾

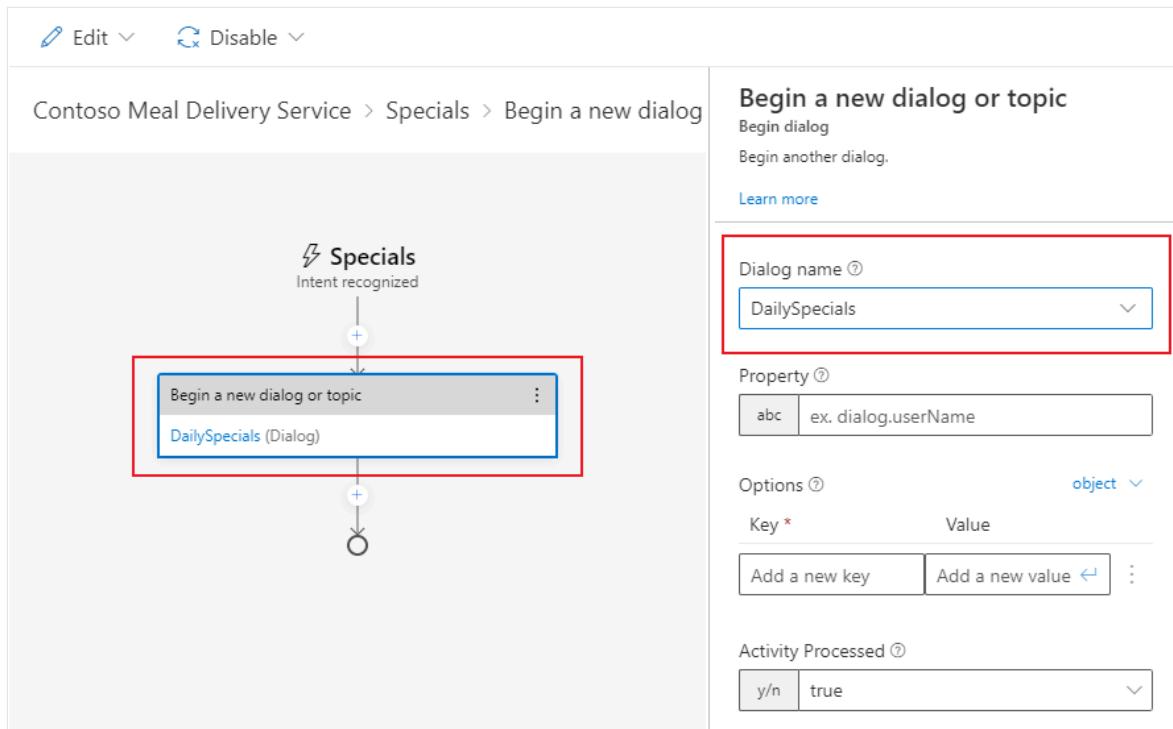
|   |                            |
|---|----------------------------|
| 1 | -what specials do you have |
| 2 | -any special deals         |
| 3 | -do you have discounts     |

Cancel   Submit

- 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 Microsoft Copilot Studio bot.

**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.

**Topics** <sup>①</sup>

Existing (16) Suggested (0)

| Type    | Name          | Trigger phrases               |
|---------|---------------|-------------------------------|
| Chatbot | Specials      | (3) what specials do you have |
| Dialog  | DailySpecials | No trigger phrases            |
| Dialog  | Meals         | No trigger phrases            |

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.

Microsoft Power Virtual Agents | Contoso Meal Delivery Service

Test bot X

Track between topics

Chat

Please select a day:

- Sunday
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday

A minute ago

Monday

A minute ago

Enjoy your Monday with a special offer of 10% off on all steak dishes!

A minute ago

Type your message ▶

Details ...

◀ DailySpecials

Trigger Phrases

Bot Framework dialogs don't use trigger phrases, but they can be called from other topics.

Bot Framework Dialog

DailySpecials [Open in Bot Framework](#)

## Next Steps

[Example 3 - Use Bing Search as a fallback in Microsoft Copilot Studio.](#)

## Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

# Example 3 - Use Bing Search as a fallback in Microsoft Copilot Studio

Article • 03/26/2024

## ⓘ 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.

## ⓘ Note

Microsoft Copilot Studio now [supports these features](#) without Composer.

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 set Bing search as a fallback answer in Microsoft Copilot Studio by using Composer.

## ⓘ Important

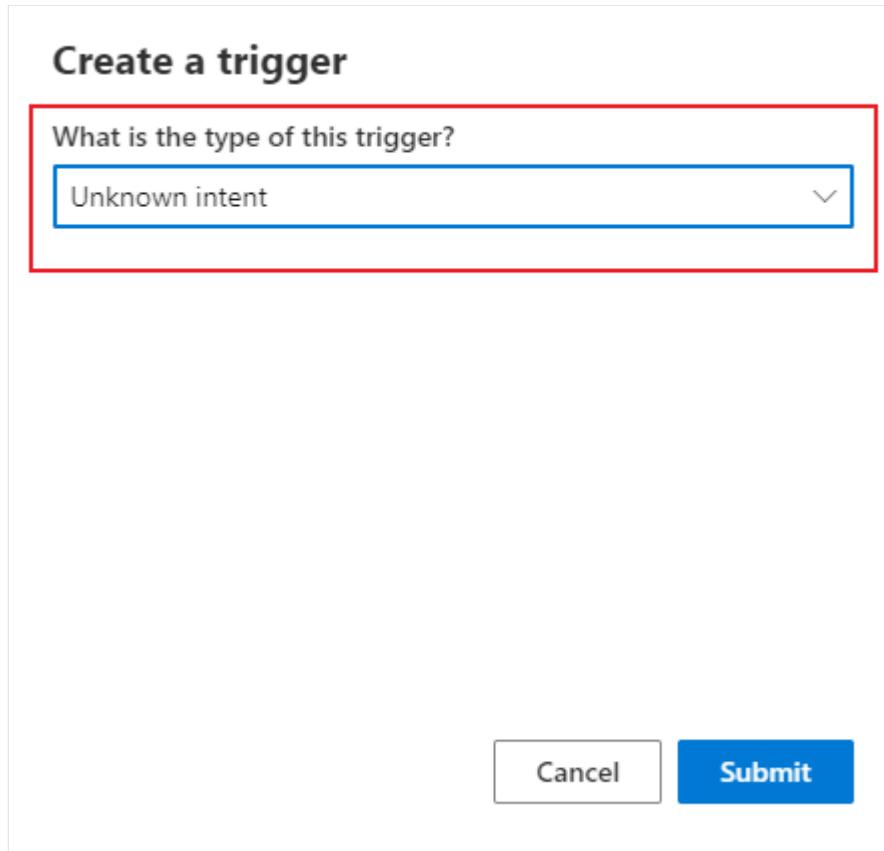
Composer integration isn't available to users who only have the [Teams Microsoft Copilot Studio license](#). You must have a [trial](#) or full Microsoft Copilot Studio license.

## Prerequisites

- Example 1 - Show an Adaptive Card in Microsoft Copilot Studio.
- Example 2 - Display a multi-select options list in Microsoft Copilot Studio.

## Create a fallback trigger

1. In Microsoft Copilot Studio, open the bot from Example 2. If you haven't completed Example 2 yet, see [Display a multi-select options list in Microsoft Copilot Studio](#).
2. Open your bot in Composer. For instructions on how to do so, see [Getting 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 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"
 }
]
}
```

```

File Edit View Window Help

Contoso-Meal-Delivery-Service

Start bot

Bot Responses

Hide code

Contoso-Meal-Delivery-Service

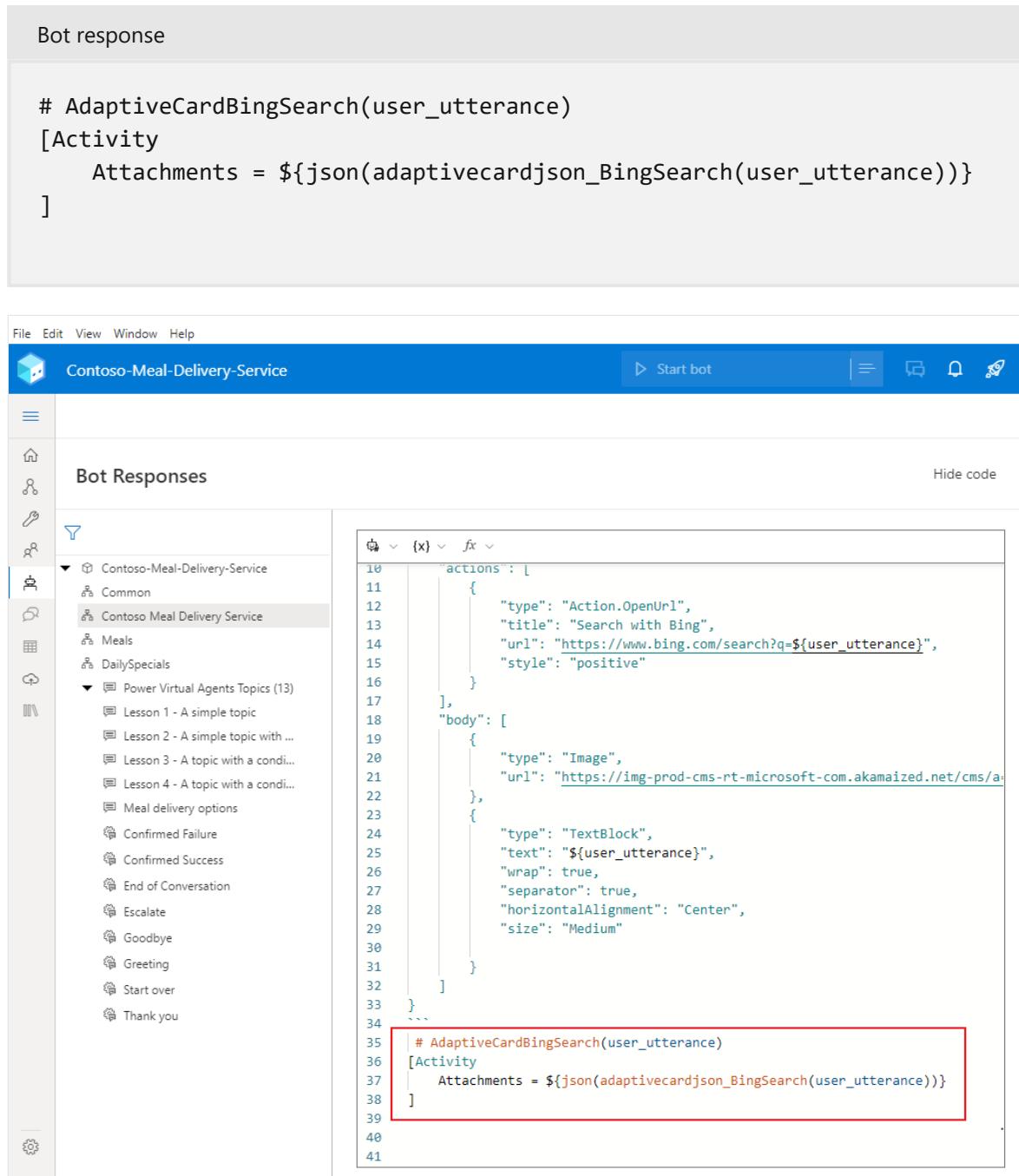
- Common
- Contoso Meal Delivery Service
- Meals
- DailySpecials
- Power Virtual Agents Topics (13)
 - Lesson 1 - A simple topic
 - Lesson 2 - A simple topic with ...
 - Lesson 3 - A topic with a condit...
 - Lesson 4 - A topic with a condit...
 - Meal delivery options
 - Confirmed Failure
 - Confirmed Success
 - End of Conversation
 - Escalate
 - Goodbye
 - Greeting
 - Start over
 - Thank you

```

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    "body": [
19      {
20        "type": "Image",
21        "url": "https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/"
22      },
23      {
24        "type": "TextBlock",
25        "text": "${user_utterance}",
26        "wrap": true,
27        "separator": true,
28        "horizontalAlignment": "Center",
29        "size": "Medium"
30      }
31    ]
32  }

```

7. Copy and paste the following code into the same code view:



The screenshot shows the Microsoft Bot Framework Emulator interface. At the top, there's a header bar with 'File', 'Edit', 'View', 'Window', and 'Help' options. Below the header is a title bar with the bot name 'Contoso-Meal-Delivery-Service'. On the left, there's a sidebar with various icons and a tree view of bot responses. The main area is titled 'Bot Responses' and contains a code editor. The code in the editor is:

```
# AdaptiveCardBingSearch(user_utterance)
[Activity
    Attachments = ${json(adaptivecardjson_BingSearch(user_utterance))}
]
```

The last two lines of the code are highlighted with a red box.

8. Go to the **Create** page. In the main dialog, select the **Unknown intent** trigger.

9. On the authoring canvas, select **Add (+)** then **Send a response**.

10. 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.

11. Copy and paste the following expression:

The screenshot shows the Microsoft Bot Framework Composer interface. On the left, there's a flowchart titled "Contoso Meal Delivery Service > Unknown intent > Send a response". A step labeled "Unknown intent" has a blue arrow pointing down to a "Send a response" step. This step contains the expression: "- \${AdaptiveCardBingSearch(turn.activity.text)}". To the right of the flowchart is a panel titled "Send a response" with the sub-section "Send Activity". It says "Respond with an activity." and "Learn more". Below this is a "Bot responses" section with a red border around it. It shows two entries: "1 - \${AdaptiveCardBingSearch(turn.activity.text)}" and "2".

Test your bot

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 **OnUnknownIntent** topic.

Topics

Existing (19) Suggested (0)

Type	Name	Trigger phrases
Unknown intent	Unknown intent	No trigger phrases
Specials	Specials	(3) what specials do you have

3. Open the **Test bot pane** and make sure **Track between topics** is turned on. Enter the message `Is tofu vegan?` to start your conversation.

Test bot

Track between topics

Chat

Is tofu vegan? 2 minutes ago

Bing

Is tofu vegan? 2 minutes ago

type your message ➤

Trigger Phrases

Bot Framework event triggers don't use trigger phrases. They are called when certain events happen in the bot.

Bot Framework Trigger

Unknown intent

Open in Bot Framework

Next Steps

[Example 4 - Display a form with a Submit button in Microsoft Copilot Studio.](#)

Feedback

Was this page helpful?

[Yes](#)

[No](#)

[Provide product feedback ↗](#)

Example 4 - Display a form with a Submit button in Microsoft Copilot Studio

Article • 03/26/2024

ⓘ 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.

ⓘ Note

Microsoft Copilot Studio now [supports these features](#) without Composer.

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 form with a Submit button 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

- Example 1 - Show an Adaptive Card in Microsoft Copilot Studio.
- Example 2 - Display a multi-select options list in Microsoft Copilot Studio.
- Example 3 - Use Bing Search as a fallback in Microsoft Copilot Studio.

Create a new trigger

1. In Microsoft Copilot Studio, open the bot from Example 3. If you haven't completed Example 3 yet, see [Use Bing Search as a fallback in Microsoft Copilot Studio](#).
2. Open your bot in Composer. For instructions on how to do so, see [Getting 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:

Language understanding

```
- start trial  
- sign up to try  
- register to try service
```

- d. Select **Submit**.

Create a trigger

What is the type of this trigger?

Intent recognized



What is the name of this trigger?

StartTrial

Trigger phrases

+ Add entity ▾ ↗ Insert entity ▾

- 1 - start trial
- 2 - sign up to try
- 3 - register to try service

Cancel

Submit

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**.

```

48
49 # adaptivecardjson_StartTrialForm()
50 - ``
51 {
52 "type": "AdaptiveCard",
53 "body": [
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 }
```

```

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 ]
96

```

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**.

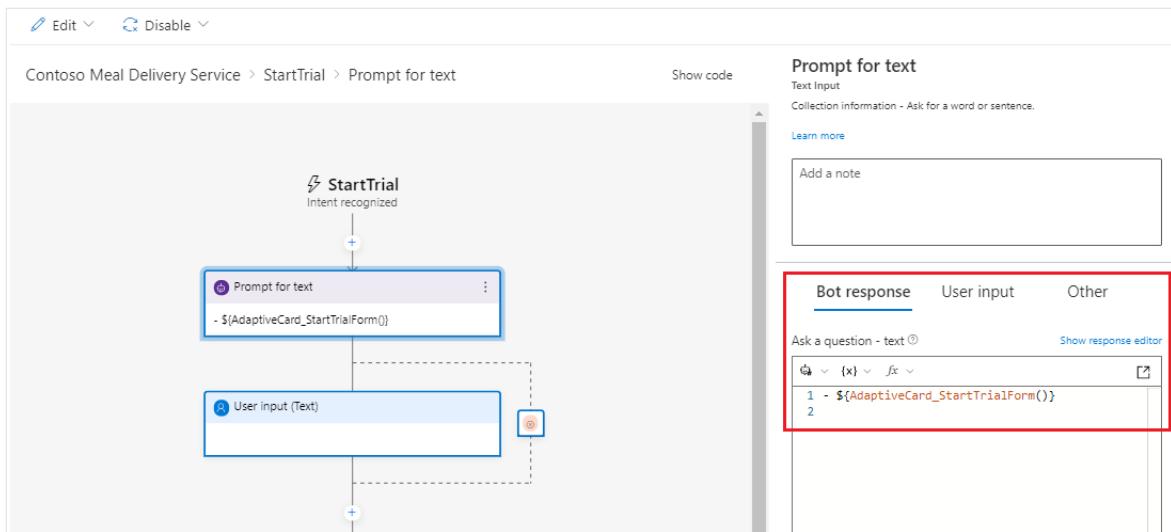
⚠ Warning

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.

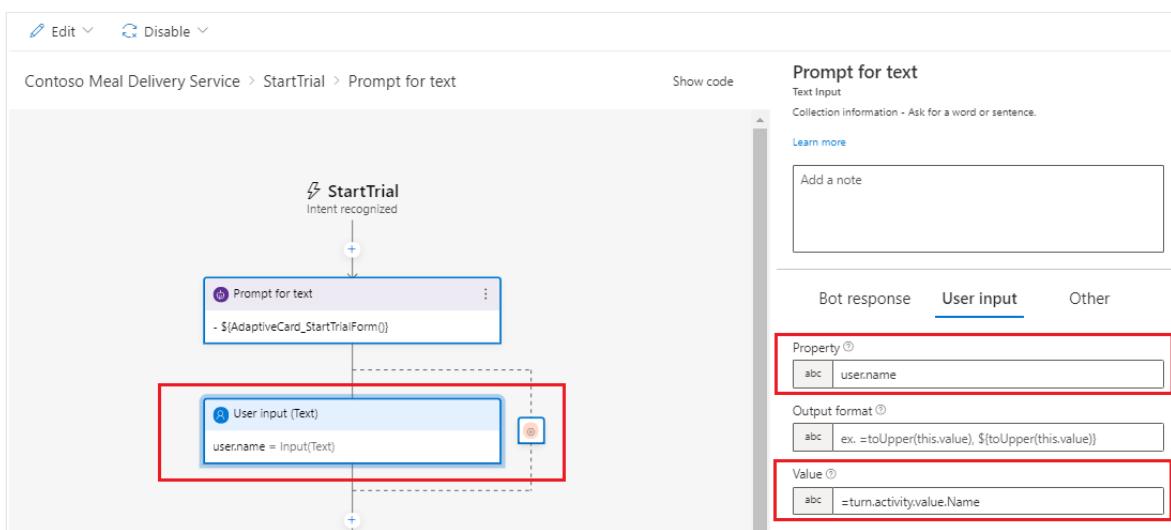
4. Copy and paste the following expression:

Bot response

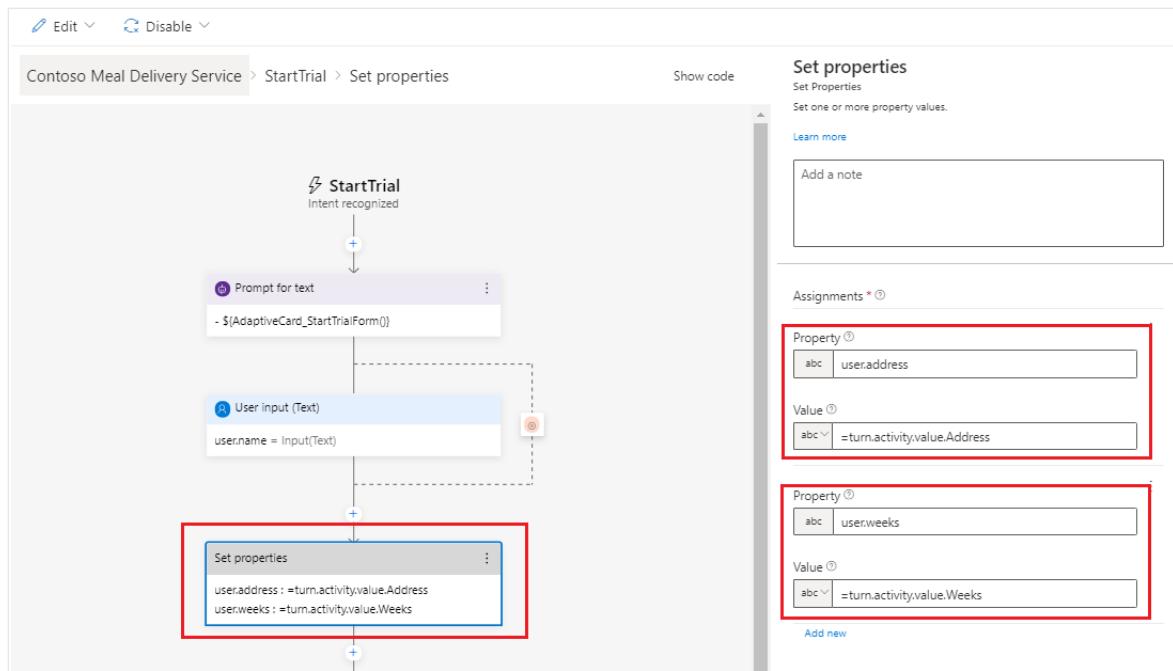
- \${AdaptiveCard_StartTrialForm()}



5. Select the **User Input** tab in the properties pane, then do the following:
 - a. For **Property**, enter `user.name`.
 - b. 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:
 - a. Under **Assignments**, select **Add new** to add a **Property** box and a **Value** box.
 - b. For **Property**, enter `user.address`
 - c. For **Value**, enter `=turn.activity.value.Address`.
 - d. Select **Add new** a second time to add another **Property** and **Value**.
 - e. For **Property**, enter `user.weeks`.
 - f. For **Value**, enter `=turn.activity.value.Weeks`.



8. On the authoring canvas, select Add (+) then Send a response.

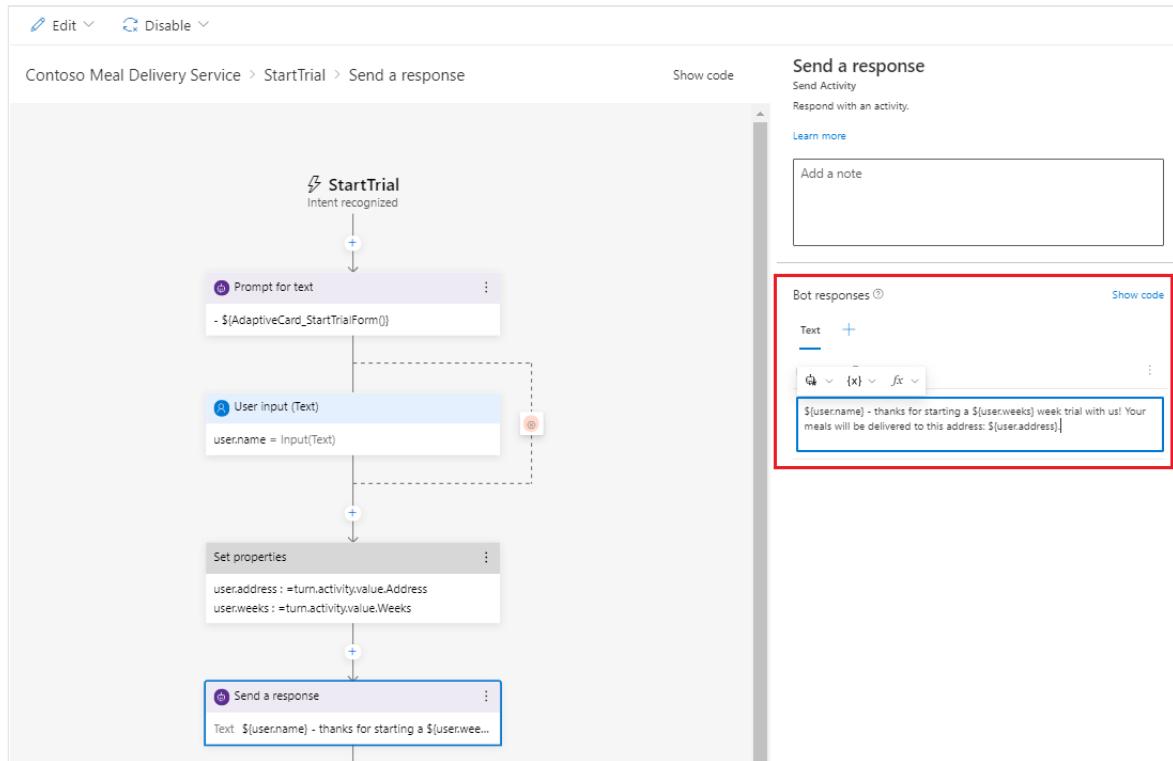
9. In the response editor, copy and paste the following:

Bot response

```

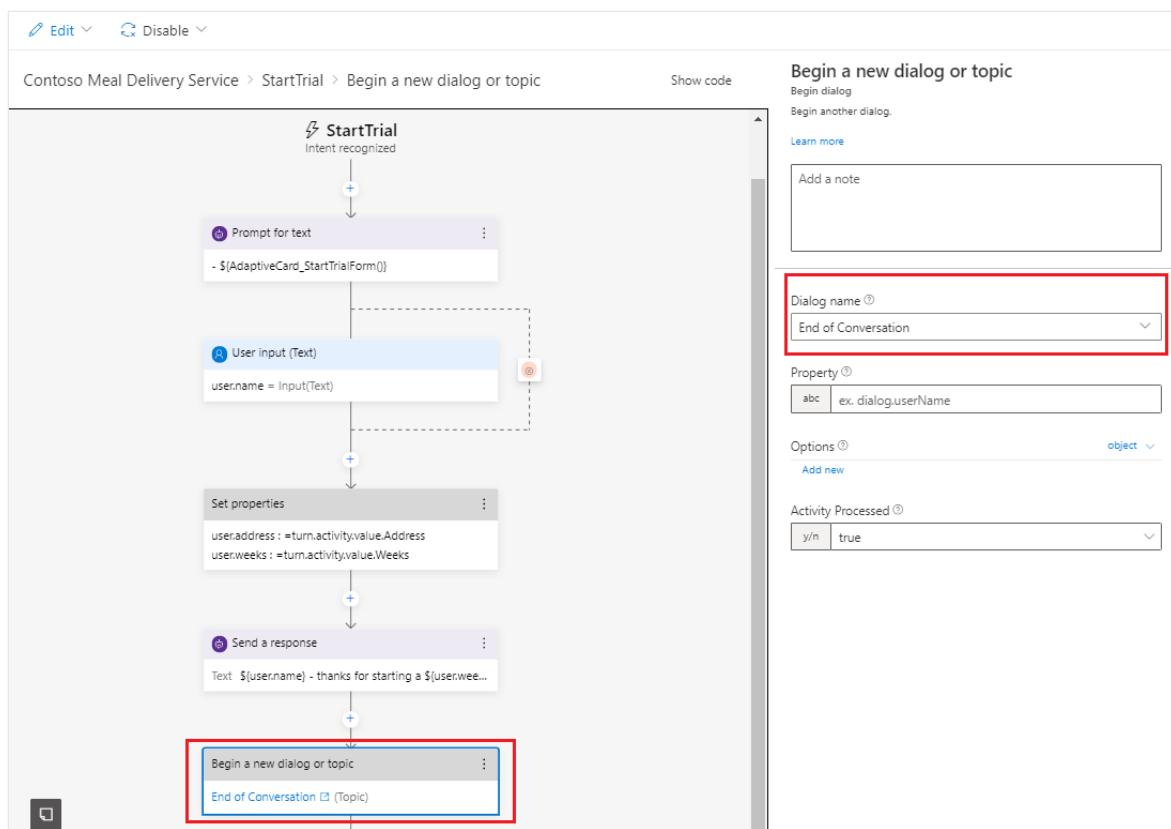
${user.name} - thanks for starting a ${user.weeks} week trial with us!
Your meals will be delivered to this address: ${user.address}.

```



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 Microsoft Copilot Studio bot.

ⓘ 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 **StartTrial** topic.

The screenshot shows the Microsoft Bot Framework portal interface. On the left, there's a sidebar with icons for Chatbots, Overview, Topics (which is selected and highlighted in blue), Entities, Analytics, Publish, Settings, and Test your bot. The main area is titled "Topics" with a subtitle "(18 Existing, 0 Suggested)". It lists two entries: "OnUnknownIntent" (Type: Entity, Name: OnUnknownIntent, Trigger phrases: No trigger phrases) and "StartTrial" (Type: Entity, Name: StartTrial, Trigger phrases: (3).start_trial). The "StartTrial" entry is highlighted with a red box.

Type	Name	Trigger phrases
Entity	OnUnknownIntent	No trigger phrases
Entity	StartTrial	(3).start_trial

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.

The image shows a split-screen view. On the left, a 'Test bot' window displays a conversation with a user named 'Jane Doe'. The user has registered for a meal plan trial, providing their name and address. The bot has responded with a confirmation message. The user then asks if the answer is correct, and the bot replies 'Yes'. Finally, the bot asks for a rating from 1 to 5 stars. On the right, the 'Confirmed Success' configuration screen shows the trigger phrases for the bot's response, the star rating scale, and the message text.

Feedback

Was this page helpful? Yes No

[Provide product feedback ↗](#)

Key concepts – Analytics 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 has a comprehensive set of analytics that show you the key performance indicators for your copilot.

ⓘ Note

Conversation transcripts in Dataverse are unavailable for download on the Microsoft Copilot Studio app in Teams. To review and export transcripts in Dataverse, you'll need to sign up for the [Microsoft Copilot Studio web app](#). Session chat transcripts can be downloaded via the Microsoft Copilot Studio app. For more details, refer to [Download copilot session transcripts](#).

Multiple charts show you trends and usage for your copilot's topics. These charts use AI to highlight the topics that have the greatest impact on your copilot's performance.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4mNOu?postJs||Msg=true> ↗

As a copilot maker, your job isn't done when you publish your copilot for use. You'll want to know how well your copilot is performing its intended duties. The comprehensive analytics in Microsoft Copilot Studio keep your copilot's key performance indicators front and center. Multiple charts show trends and usage for your copilot's topics. These charts use AI to highlight the topics that have the greatest effect on your copilot's performance.

Analytics in Microsoft Copilot Studio

In Microsoft Copilot Studio, in the navigation menu, select **Analytics**.

The analytics section is divided into a number of pages to give you multiple ways to understand copilot performance.

The following terms are used in this section.

- A *conversation* is an ongoing interaction between a specific user, or group of users, on a channel and your copilot. Conversations can pause and resume later, or be transferred to a live agent. The conversation may be one-way, either from the end user to the copilot, or from the copilot to the end user, but is more commonly a back-and-forth interaction between the end user and the copilot.
- *Analytics sessions* track user engagement for your copilot. Analytics sessions try to capture how well your copilot handles user tasks.
- *Billed sessions* are used to measure resource consumption. Billed sessions are independent of analytics sessions.

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 Microsoft Copilot Studio app. It can take up to an hour after the analytic session *ends* before the appearing in the analytics dashboard.

In the classic version of Microsoft Copilot Studio, conversation transcripts in Dataverse are available approximately 6 hours after the analytic session *starts*.

 **Note**

Copilot analytics features aren't yet available for Microsoft 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.

View analytics

In the side navigation pane, select **Analytics**.

Related topics

 Expand table

Topic	Description
Analyze copilot performance and usage	Review overall analytics for your copilots.
Analyze customer satisfaction	Identify overall trends in customer satisfaction, and where satisfaction is falling behind.
Analyze topic usage	See how individual topics are doing and how well they're working at resolving issues.
Analyze session information	Review and monitor how sessions are doing, including analytics from the topics triggered during a session.
Analyze billed session information	Understand how sessions are being billed, based on their usage.
Work with conversation transcripts	Export conversation transcripts from Dataverse and create custom reports.

Analyze copilot performance and usage in Copilot Studio

Article • 04/03/2024

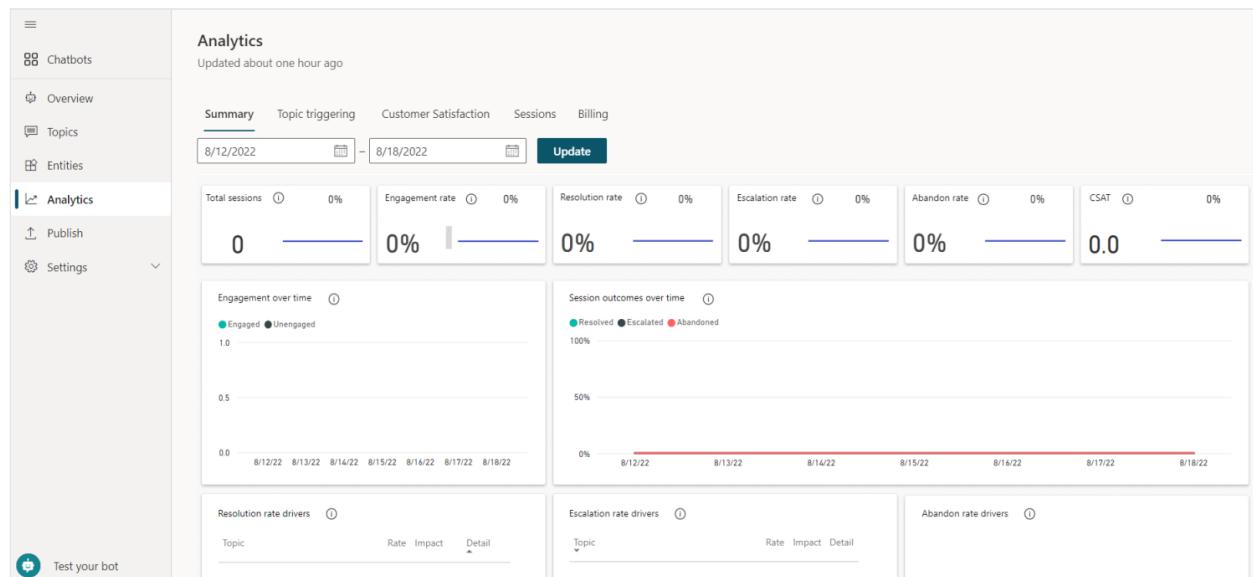
ⓘ Important

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.

The **Summary** tab on the **Analytics** page gives you a broad overview of your copilot's performance. It uses artificial intelligence (AI) technology to show you which topics are having the greatest impact on escalation rate, abandon rate, and resolution rate.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4mQpm?postJsIIMsg=true>



💡 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.

Analytics sessions track user engagement with your copilot and try to capture how well your copilot 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 copilot is doing and help you find the topics with the most impact on your copilot's performance. A session will time out after 30 minutes of inactivity.

- A session starts when a user interacts with your copilot or the copilot 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 will 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 copilot's key performance indicators. The remainder of this article describes each chart in more detail.

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.
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 Microsoft 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 copilot.

Increase the resolution rate of topics with high negative impact percentages to more quickly improve the overall resolution rate for your copilot.

To see additional information about each topic, select the **Detail** link to display the topic details page. For more information, see [Topic details page](#).

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 Microsoft Copilot Studio topic.

Description	Details
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 copilot.

Lower the escalation rate of topics with high impact percentages to more quickly decrease the escalation rate for your copilot.

To see additional information about each topic, select the **Detail** link to display the topic details page. For more information, see [Topic details page](#).

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 Microsoft 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 copilot.

Lower the abandon rate of topics with high impact percentages to more quickly decrease the abandon rate for your copilot.

To see additional information about each topic, select the **Detail** link to display the topic details page. For more information, see [Topic details page](#).

Analyze your copilot's customer satisfaction metrics

Article • 05/21/2024

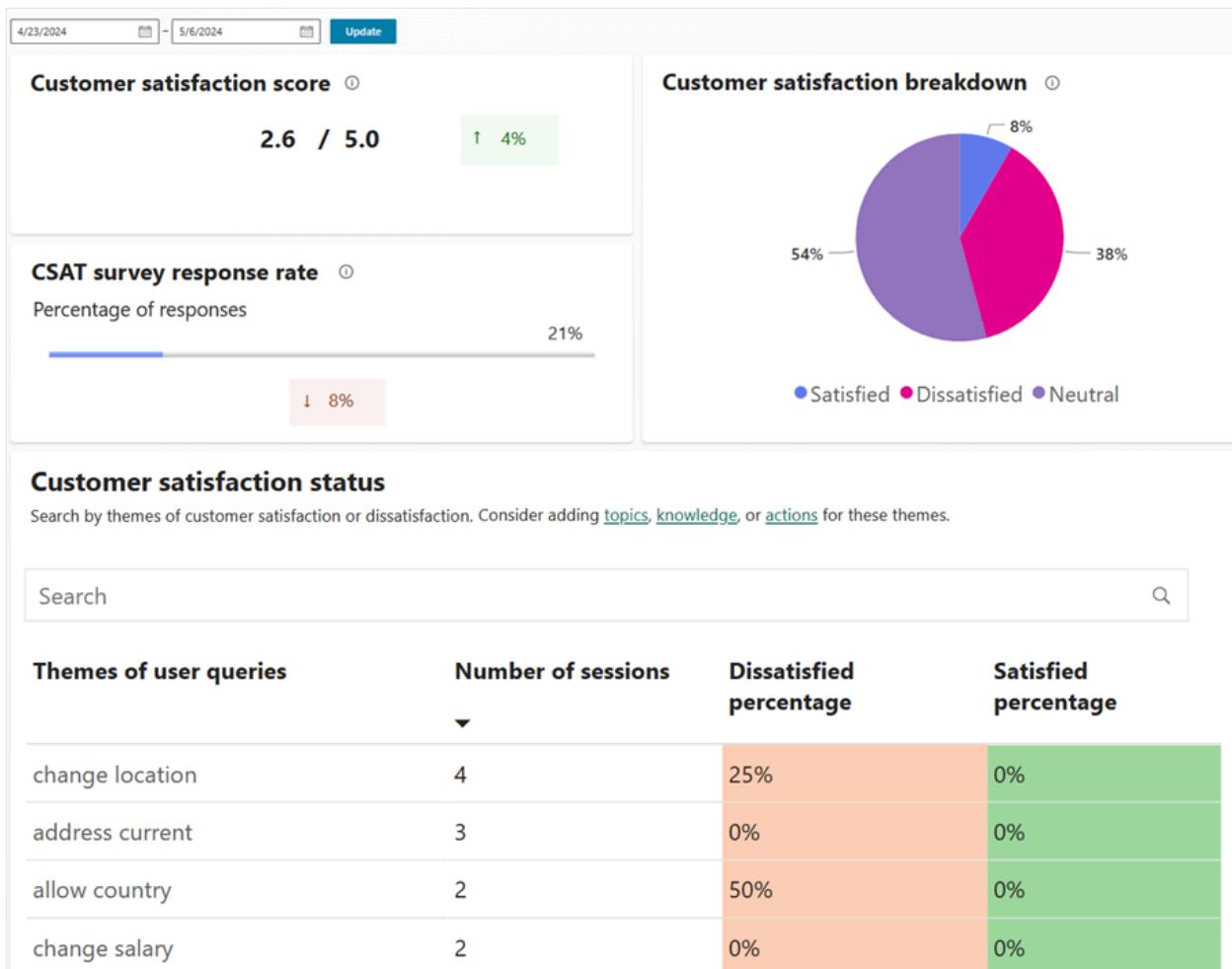
Important

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.

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 copilot's responses.

By default, the page shows you 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 an end user 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 prior to the selected period. Please note that the period-over-period indicator will only be shown if your copilot 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 will not be shown.

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 have searched for, and the satisfaction level of the users about the copilot'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 were neither satisfied nor 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 personally identifiable information (PII) or sensitive information, such as phone numbers. Additionally, if themes contain profane or harmful language, these themes are masked.

Hover over each segment of the chart to see the specific satisfaction or dissatisfaction drivers. 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 copilot is classified as negative. Sentiment is determined using a publicly available ML model finetuned 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 copilot 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 copilot 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 involves analyzing the text of the session to determine whether the overall sentiment is positive, negative, or neutral, and preprocessing 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 session health

Article • 05/21/2024

The **Session health** tab of the **Analytics** page provides a detailed view of how generative answers performed in response to user queries over the selected time period. It provides three key insights:

- Themes of user queries that didn't return any search results.
- Themes for which generative AI successfully returned an answer.
- Insights on user queries that triggered the content filtering system of Copilot Studio.

The screenshot shows the Microsoft Copilot Analytics interface. The left sidebar has links for Summary, Customer satisfaction, Session Health (which is selected), Recommendations, Sessions, and Billing. The main area has a date range from 4/1/2024 to 4/1/2024 with an 'Update' button. The 'Session Health' section is divided into three main panes: 'No search results', 'Answered themes', and 'Content moderation'. The 'No search results' pane shows a breakdown of reasons why AI generated responses had no search results, with a table for 'Themes with no search results' and a note about adding topics, knowledge, or actions. The 'Answered themes' pane shows a breakdown of most commonly answered questions, with a table for 'Themes of answered questions'. The 'Content moderation' pane shows statistics for harmful content, prompt injection, and ungrounded content, all at 0 (0%).

Prerequisites

You must turn on [generative answers](#) for your copilot, to obtain session health insights.

No search results

The **No search results** pane shows themes and associated number of sessions where generative answers were triggered but couldn't retrieve data from existing knowledge sources. Themes are generated by using an ML algorithm to extract the overall theme of an individual session. Similar sessions are grouped together into themes.

💡 Tip

Consider adding new topics or updating knowledge sources for themes that have no search results.

Answered themes

The **Answered themes** pane shows the user query themes for which generative AI successfully returned answers. This information can help you quickly gain a sense of the most common questions that end users are asking your copilot. To better understand whether customers are reacting positively or negatively to these answers, use the sentiment insights provided in the [Customer satisfaction](#) tab.

💡 Tip

You can review how your users react to the quality of your copilot's answers in the [Customer satisfaction](#) tab. If customers are dissatisfied with the quality of the answers, consider improving existing topics or adding additional knowledge sources.

Content moderation

The **Content moderation** pane shows detection rates of content and behaviors that suggest your copilot was used in a manner that might violate applicable product terms. Copilot Studio includes a content filtering system that detects and prevents (blocks) user-generated and AI-generated harmful content, to comply with Microsoft Responsible AI standard guidelines. If your copilot is configured to send telemetry data to Azure Applications Insights, follow the instructions in [Content blocked by content moderation](#) to identify the specific user messages that triggered content moderation.

The content filtering system works by running both the input prompt and the resulting [completion](#) provided by the Copilot Studio through an ensemble of classification models. These models are designed to identify and block the generation of harmful content. It acts on specific categories of potentially harmful content in input prompts and output completions. For more information on these categories and their definitions, see [Content filtering](#), in the Azure OpenAI Service documentation.

The following section provides information about the content filtering categories:

[+] [Expand table](#)

Category	Metric definition	Harm description
Harmful content	Block rate: the number of user requests unanswered because harmful content was detected divided by the total number of user requests in the selected period	User prompts with harmful content in these categories: Hate and Fairness, Sexual, Violence, and Self-harm.
Prompt injection	Block rate: the number of user requests unanswered because prompt injection was detected divided by the total number of user requests in the selected period.	User prompts designed to provoke the generative AI model into exhibiting behaviors it was trained to avoid or to break the rules set in the System Message.
Ungrounded content	Block rate: the number of user requests unanswered because ungrounded content was detected divided by the total number of user requests in the selected period.	A large language model (LLM) that generates information lacking factual basis, responses that aren't present in the provided input sources or can't be inferred from them, which might result in users being exposed to inaccurate or misleading content.

Related content

- [Analyze customer satisfaction with copilot responses](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Analyze session information 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.

By default, you can download up to seven days of copilot conversation transcripts from the past 30 days directly from the Copilot Studio portal.

You can also download and view conversation transcripts in the Power Apps portal.

See [Work with conversation transcripts](#) for more details and tips.

Prerequisites

- To view session transcripts, one needs to have the transcript viewer security role. Only admins can grant this role by [Assign the Copilot transcript viewer security role during copilot sharing](#).

Download copilot session transcripts

1. In the Microsoft Copilot Studio navigation menu, select **Analytics**.

2. Select the **Sessions** tab.

If your copilot has many sessions, they're displayed in multiple rows. Each row contains up to 2,500 sessions.

3. Select a row to download.

The screenshot shows the 'Analytics' page in the Microsoft Bot Framework portal. The left sidebar has a navigation menu with options: Chatbots, Overview, Topics, Entities, Analytics (which is selected and highlighted in blue), Publish, and Settings. The main content area is titled 'Analytics' and shows 'Updated about one hour ago'. Below this are tabs for Summary, Topic triggering, Customer Satisfaction, Sessions (which is underlined in blue to indicate it's active), and Billing. A date range selector shows '8/12/2022' to '8/18/2022' with a 'Update' button. The 'Sessions for download' section contains a red box around the text '3:53 PM 11/11/19 – 3:53 PM 11/11/19'.

The download starts immediately. The file is saved to your browser's default download location.

Work with session transcripts

The session transcript is a comma-separated values (CSV) file that contains the following information:

- **SessionID:** The session's unique identifier
- **StartTime:** The time the session started (by default, sessions are sorted by this column in descending order)
- **InitialUserMessage:** The first message the user typed
- **TopicName:** The name of the last authored topic that was triggered in the session
- **ChatTranscript:** The transcript of the session in the format **User says: ; Bot says: ;**
For example:

The screenshot shows a text editor with the word 'text' at the top. Below it is a code block containing a session transcript:
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.;

Bot says doesn't include the options presented to the user.

- **SessionOutcome:** The outcome of the session (Resolved, Escalated, Abandoned, Unengaged)
- **TopicId:** The unique identifier of the last authored topic triggered in the session

1. In the navigation menu, select **Analytics**. Go to the **Sessions** tab.

If your copilot had a high number of sessions, they'll be broken down into multiple rows. Each row contains 2500 sessions.

2. Select each row to download the session transcripts for the specified time frame.

The screenshot shows the Microsoft Copilot Analytics interface. On the left is a navigation sidebar with options: Chatbots, Overview, Topics, Entities, Analytics (which is selected and highlighted in blue), Publish, and Settings. The main area is titled 'Analytics' and shows it was 'Updated about one hour ago'. Below this are tabs: Summary, Topic triggering, Customer Satisfaction, Sessions (which is underlined in blue to indicate it's active), and Billing. Under the Sessions tab, there are date range controls showing '8/12/2022' to '8/18/2022' with an 'Update' button. Below these controls is a section titled 'Sessions for download' with a red box highlighting the text '3:53 PM 11/11/19 – 3:53 PM 11/11/19'.

The downloaded file contains the following information:

- SessionID: A unique identifier per session.
- StartDateTime: 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:
<message-text>; Bot says: <message-text>;`.
 - Conversation turns are separated by semicolons
 - *Bot says* 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.

 **Note**

The download will start when you select the time period. It will be downloaded into your default browser download's location.

Work with conversation transcripts

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.

You can view, export and download transcripts of customer interactions with your copilot in both Power Apps and Microsoft 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. [Download copilot session transcripts in Microsoft 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.

Prerequisites

- To view conversation transcripts, one needs to have the transcript viewer security role. Only admins can grant this role by [Assign the Copilot transcript viewer security role during copilot sharing](#).

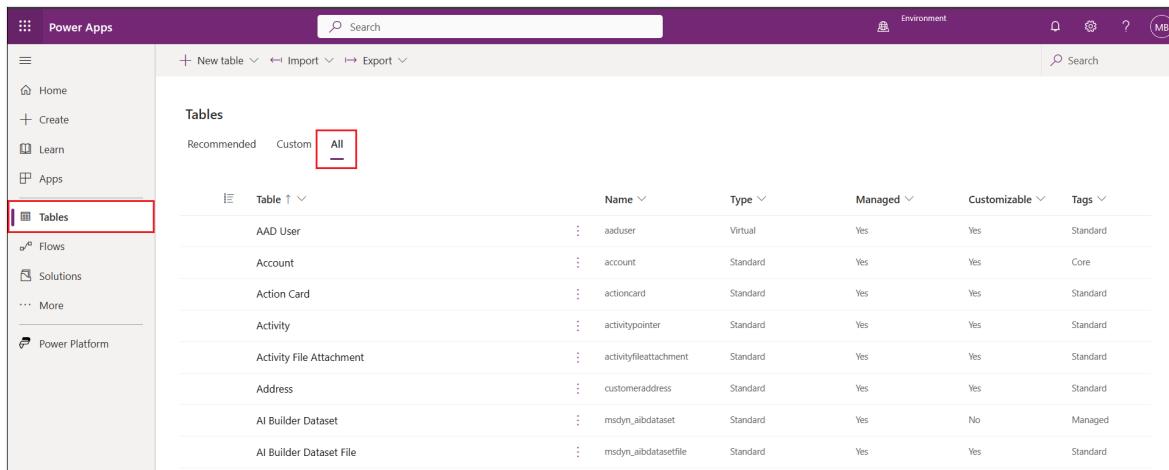
View and export copilot conversation transcripts from the Power Apps portal

You can also view and export copilot conversation transcripts from the Power Apps portal.

First, you'll need to sign in to <https://www.powerapps.com> with your credentials.

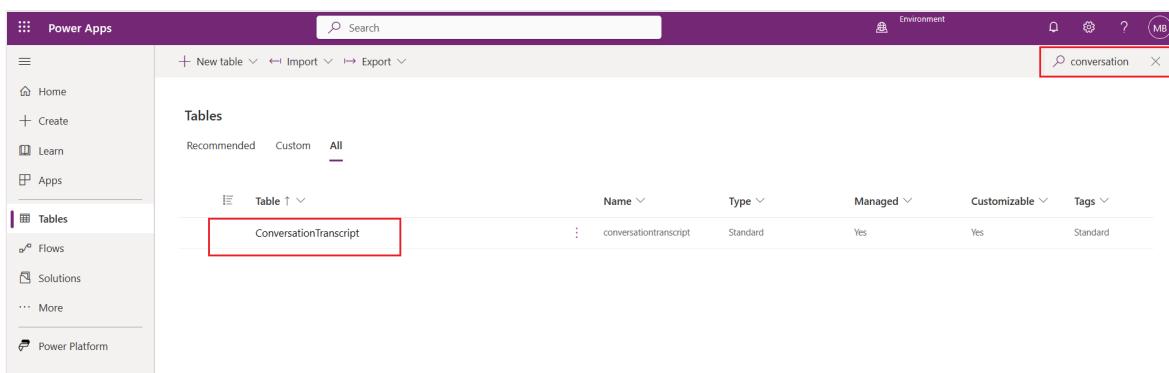
View conversation transcripts

1. In the side pane select the **Tables** node. At the top, select **All**.



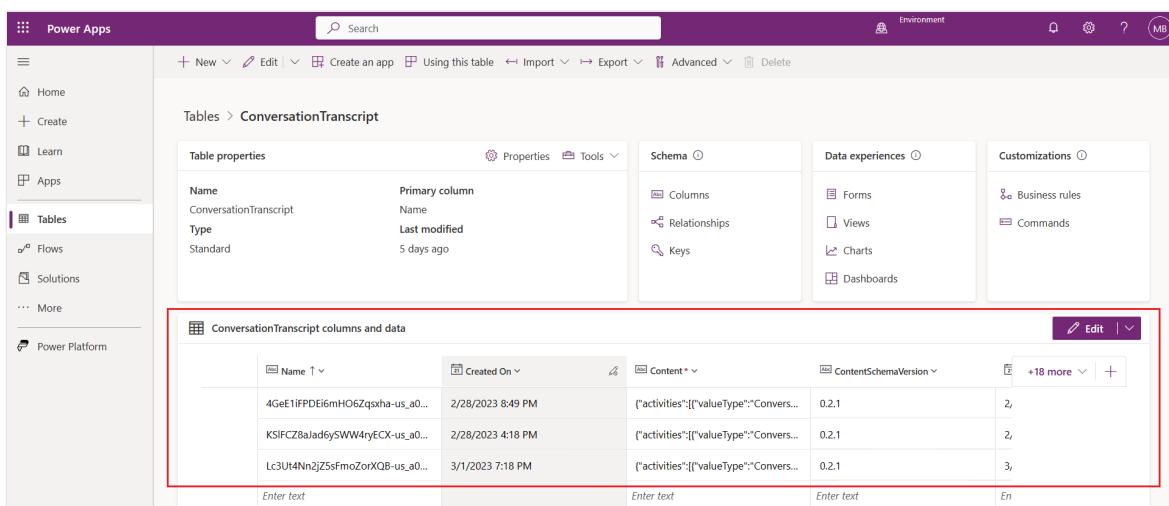
The screenshot shows the Power Apps interface with the 'Tables' node selected in the left sidebar. At the top, the 'All' filter is highlighted with a red box. The main area displays a list of tables, including AAD User, Account, Action Card, Activity, Activity File Attachment, Address, AI Builder Dataset, and AI Builder Dataset File. Each table entry includes columns for Name, Type, Managed, Customizable, and Tags.

2. Type **Conversation** in the **Search** textbox on the top right. Select **ConversationTranscript** under **Tables**.



The screenshot shows the Power Apps interface with the search bar containing 'conversation'. The results list 'ConversationTranscript' under the 'Tables' section, which is highlighted with a red box. The table details show it is a Standard type table with a primary column named 'conversationtranscript'.

3. A page for the **ConversationTranscript** table will open and show a list of data available to view.



The screenshot shows the 'ConversationTranscript' table properties and data view. The properties pane shows the table name, primary column (Name), type (Standard), and last modified date (5 days ago). The data view shows a list of three records with columns: Name, Created On, Content, and ContentSchemaVersion. The first record has a detailed view expanded, showing its JSON content and other details like valueType and version.

Name	Created On	Content	ContentSchemaVersion
4GeE1FPDEI6mHO6Zqsxha-us...	2/28/2023 8:49 PM	{"activities": [{"valueType": "Convers...}}	0.2.1
K5IFCZ8ajad6ySWW4IyfCX-us...	2/28/2023 4:18 PM	{"activities": [{"valueType": "Convers...}}	0.2.1
Lc3Ut4Nn2jZ5sFmoZorXQB-us...	3/1/2023 7:18 PM	{"activities": [{"valueType": "Convers...}}	0.2.1

Export conversation transcripts

1. In the side pane, select the **Tables** node. At the top, select **All**.

The screenshot shows the Power Apps Tables screen. On the left, there's a sidebar with options like Home, Create, Learn, Apps, and Tables. The Tables option is highlighted with a red box. At the top, there are buttons for New table, Import, Export, and a search bar. Below that, there are tabs for Recommended, Custom, and All, with All also highlighted with a red box. The main area is a table listing various tables with columns for Name, Type, Managed, Customizable, and Tags. Some rows have a small icon to the left of the table name.

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

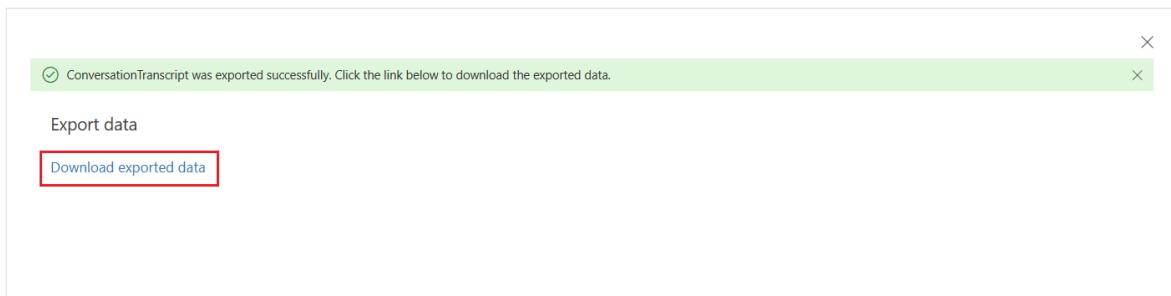
2. Type **Conversation** in the **Search** textbox at the top right. Select **ConversationTranscript** under **Tables**.

The screenshot shows the same Power Apps Tables screen as before, but now the search bar at the top right contains "conversation". The table list has a row for "ConversationTranscript" which is highlighted with a red box. The rest of the interface is identical to the first screenshot.

3. Expand the **Export** menu item at the top of the screen and select **Export data**. It may take a few minutes for the data to be compiled for export.

The screenshot shows the Power Apps Tables screen again. The search bar now contains "conversation". The "ConversationTranscript" table row is selected (indicated by a purple circle icon). At the top, the "Export" menu item is expanded, and the "Export data" option is highlighted with a red box. There's also a "Link to Azure Synapse" button below it.

4. Select **Download exported data** to download the content.



Download copilot conversation transcripts

1. In the Power Apps navigation menu, select **Dataverse**, and then select **Tables**.
2. Select **All**.
3. In the search box at the upper-right corner, type **conversation**.
4. Select the **ConversationTranscript** table.

The screenshot shows the Power Apps navigation menu on the left with 'Dataverse' selected. Under 'Dataverse', 'Tables' is also selected. The main area displays a list of tables with a search bar at the top right containing the text 'conversation'. The 'ConversationTranscript' table is highlighted with a red box. The table details show it is a Standard type, Managed by Yes, and Customizable by Yes. It has a single column named 'conversationtranscript'.

5. Select **Export**, and then select **Export data**.

The screenshot shows the 'Tables > ConversationTranscript' properties page. The 'Export' button is highlighted with a red box. Below it, the 'Export data' option is also highlighted with a red box. The table properties pane shows the following details: Name: ConversationTranscript, Primary column: Name, Description: Contains the transcripts of conversations between participants such as customers, Virtual Agents or Human agents. Type: Standard, Last modified: 1 month ago.

6. Allow a couple of minutes for Power Apps to prepare the data for export. 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 Content field 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 copilot ID, tenant ID, and copilot name	{"BotId": "198eca5f-xxxx-4ae6-8c08-835d884a8688", "AADTenantId": "72f988bf-xxxx-41af-91ab-2d7cd011db47", "BotName": "Test Bot"}
Name	The name of the custom row that's 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 do not include the batch number.	8YYe8iif49ZKkycZLe7HU0-o_198eca5f-xxxx-4ae6-8c08-835d884a8688_0
Bot_ConversationTranscript	The copilot ID	198eca5f-xxxx-4ae6-8c08-835d884a8688
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 copilot. Common activity types include message and event:

- Message activities represent the content that's shown in a conversation. Message activities may contain text, speech, interactive cards, and binary or unknown attachments.
- Event activities communicate programmatic information from a client or channel to the copilot.

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 that's stored in the activity; dictates what information the activity is providing (Common activity value types)
<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">• <code>id</code> - the ID of the invoker• <code>role</code> - holds 0 or 1<ul style="list-style-type: none">◦ 0 - the activity is coming from the copilot◦ 1 - the activity is coming from the user interacting with the copilot <p>Notes: The <code>id</code> can be used to calculate the number of active users that are interacting with the copilot 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. For security and privacy, the ID is hashed before it's written to the transcript.</p>
<code>channelId</code>	The ID of where the activity is coming from; for example, <code>directline</code> , <code>msteams</code> , or <code>facebook</code>
<code>textFormat</code>	The format of the text; for example, <code>plain</code> or <code>markdown</code>
<code>attachments</code>	Dynamic rich data associated with the activity; for example, <code>AdaptiveCards</code> , <code>HeroCards</code> , or <code>Carousel</code> data
<code>text</code>	The text for <code>message</code> activities
<code>value</code>	Fields specific to the activity based on the value type; this field is where most of the useful information exists
<code>channeldata</code>	<ul style="list-style-type: none">• Contains channel data:<ul style="list-style-type: none">◦ for messages:<ul style="list-style-type: none">◦ <code>DialogTraceDetail</code>◦ <code>DialogErrorDetail</code>◦ <code>VariableDetail</code> (contains the value assigned to a variable)

Key	Description
	<ul style="list-style-type: none"> ○ <code>CurrentMessageDetail</code> ○ for events: <ul style="list-style-type: none"> ○ <code>cci_trace_id</code> ○ <code>traceHistory</code> ○ <code>enableDiagnostics</code> ○ <code>clientTimestamp</code> ○ <code>clientActivityId</code>
<code>name</code>	The name of the event activity; for example, <code>SetPVAContext</code>

Common activity value types

[Expand table](#)

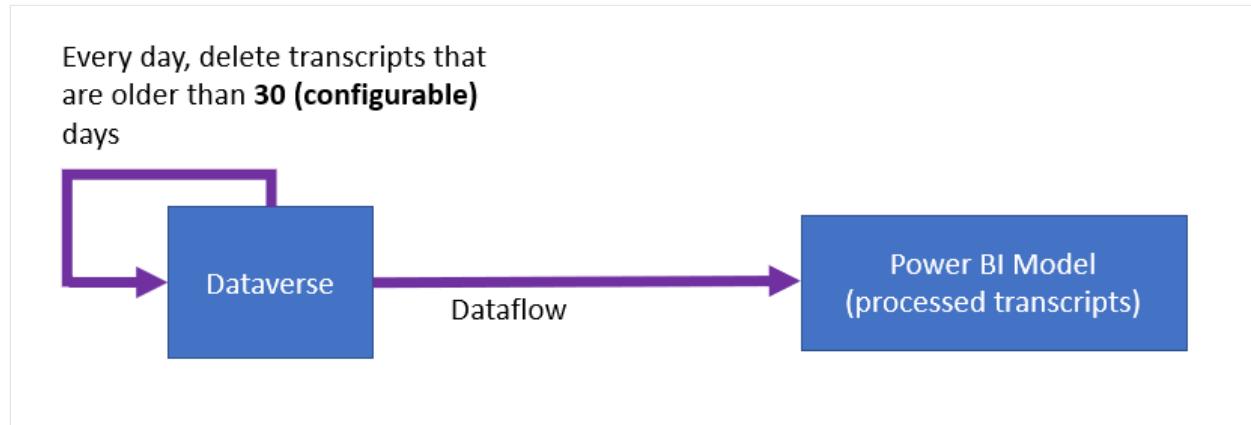
Activity value type	Description
<code>ConversationInfo</code>	Whether the conversation is from the Microsoft Copilot Studio test pane (<code>isDesignMode</code>) and the locale of the conversation
<code>CSATSurveyRequest</code>	The user is presented with a customer satisfaction (CSAT) survey
<code>CSATSurveyResponse</code>	The user responds to a CSAT survey
<code>DialogRedirect</code>	The user is redirected to another topic
<code>ImpliedSuccess</code>	The user has reached a question node in the topic, where one of the conditions points to the "Confirmed Success" CSAT system topic or calls the "End of Conversation" system topic
<code>IntentRecognition</code>	A topic was triggered by the user
<code>PRRSurveyRequest</code>	The user was asked if the topic answered their question from the "End of Conversation" topic
<code>PRRSurveyResponse</code>	The user's response to whether the topic answered their question from the "End of Conversation" topic
<code>SessionInfo</code>	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
<code>VariableAssignment</code>	A value has been 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](#)

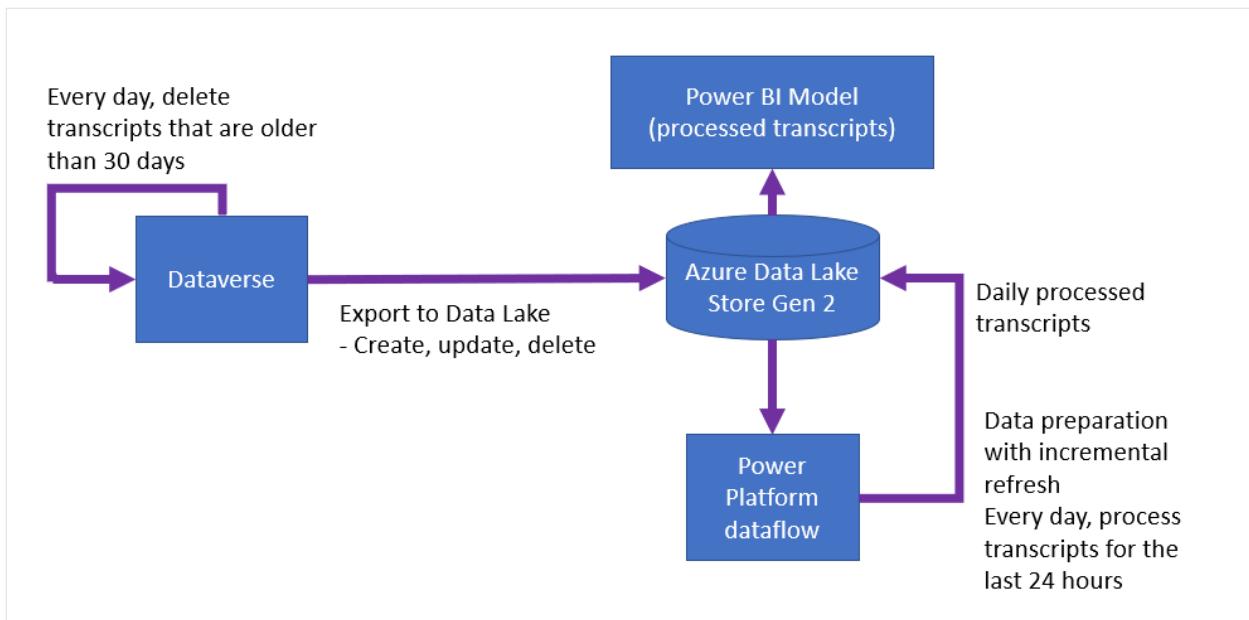
Pros	Cons
Easiest implementation.	Refresh time may increase as transcripts increase. High cost for storage in Dataverse.

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.

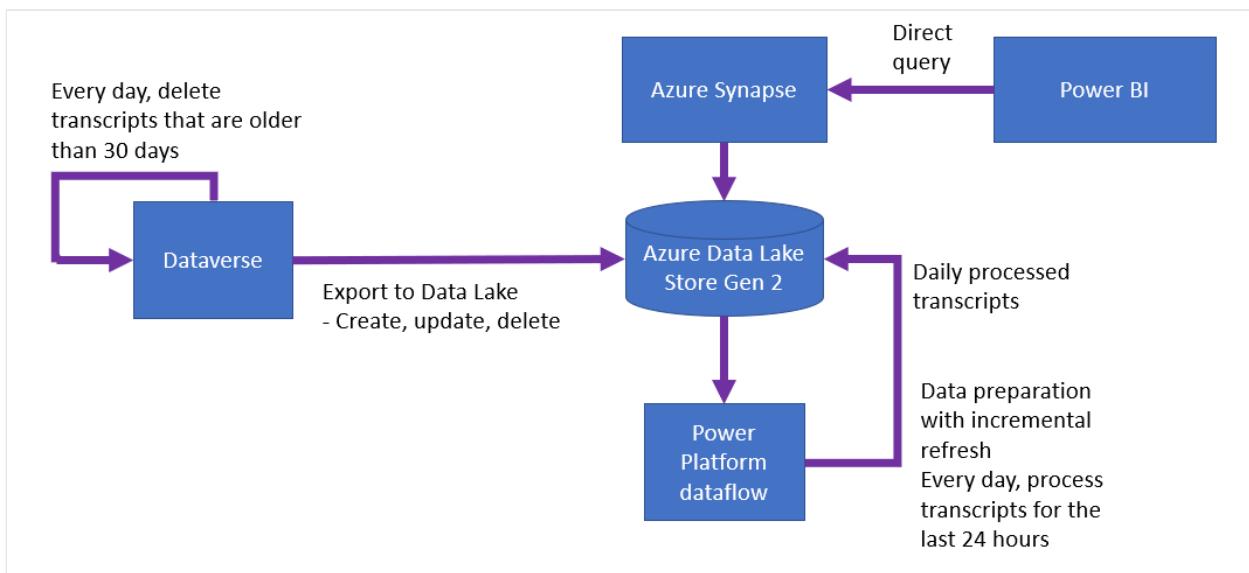


[+] [Expand table](#)

Pros	Cons
Low storage cost in Azure Data Lake Storage. Moderately simple implementation. No Azure Compute required.	Refresh time may increase as transcripts increase each day.

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.



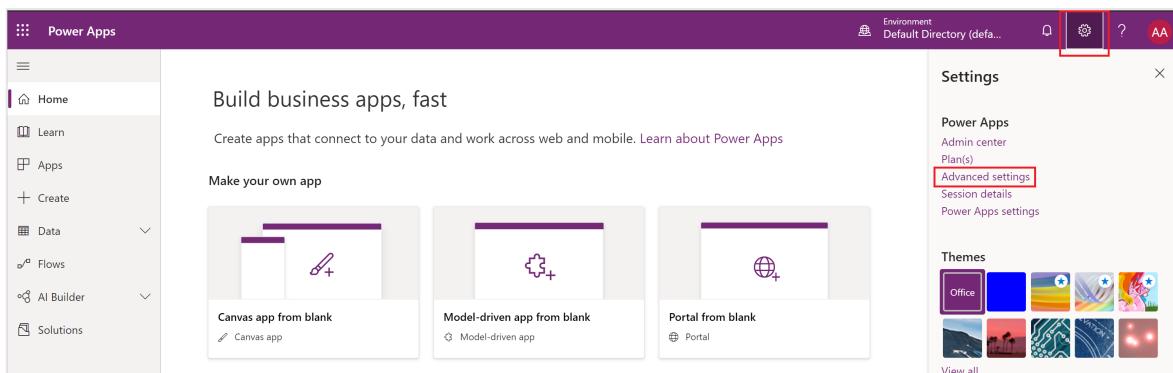
Pros	Cons
Refresh time is constant. Moderately simple implementation. Low storage cost in Azure Data Lake Storage.	Compute cost for Azure Synapse.

Change the default retention period

A Power Apps bulk delete job automatically removes copilot 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.

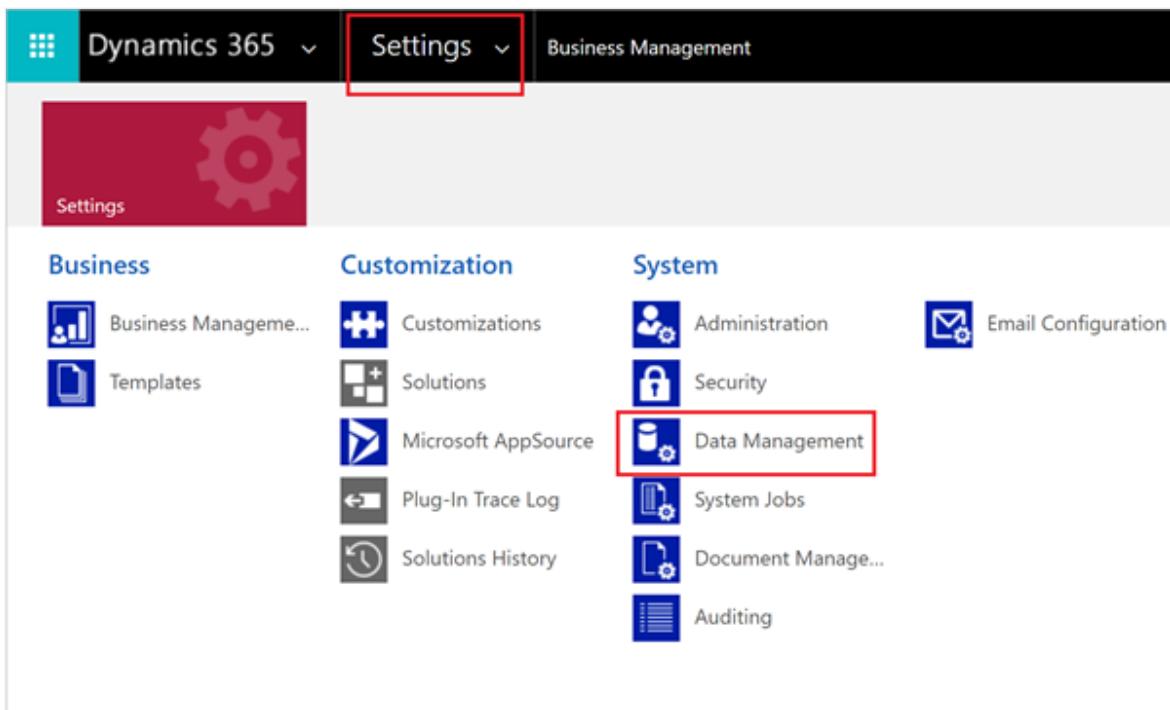
Cancel the existing bulk delete job

1. In Power Apps, in the upper-right corner, select the **Settings** cog icon, and then select **Advanced settings**.

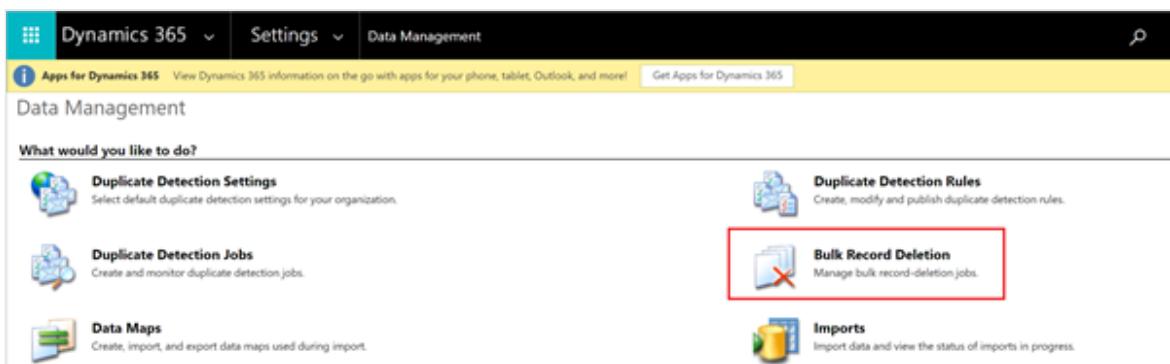


The Dynamics 365 portal opens in a new tab.

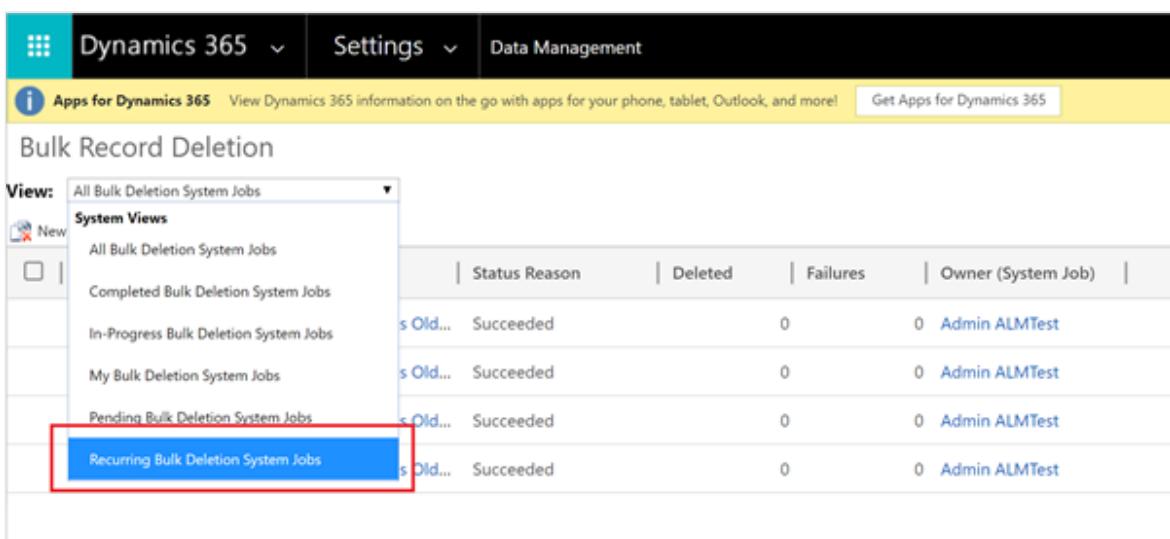
2. Select the caret next to **Settings**, and then under **System**, select **Data Management**.



3. Select Bulk Record Deletion.



4. In the View list, select Recurring Bulk Deletion System Jobs.



5. Select the Bulk Delete Conversation Transcript Records Older Than 1 Month job.

Dynamics 365	Settings	Data Management
Apps for Dynamics 365	View Dynamics 365 information on the go with apps for your phone, tablet, Outlook, and more!	Get Apps for Dynamics 365
Bulk Record Deletion		
View: Recurring Bulk Deletion System Jobs		
New		
		More Actions
<input type="checkbox"/> System Job Name ↑	Status Reason	Owner (System Job)
<input checked="" type="checkbox"/> Bulk Delete Conversation Transcript Records Older Than 1 Month	Waiting	Admin ALMTest
Delete completed pipeline events	Waiting	SYSTEM
Delete completed process sessions for Sync Workflows	Waiting	SYSTEM
	1/17/2020 7:15 PM	1/17/2020 11:18 AM
	1/17/2020 11:18 AM	

6. In the More Actions list, select Cancel.

Dynamics 365	Settings	Data Management
Apps for Dynamics 365	View Dynamics 365 information on the go with apps for your phone, tablet, Outlook, and more!	Get Apps for Dynamics 365
Bulk Record Deletion		
View: Recurring Bulk Deletion System Jobs		
New		
		More Actions
<input type="checkbox"/> System Job Name ↑	Set View as Default View	Status Reason
<input checked="" type="checkbox"/> Bulk Delete Conversation Transcript Records Older Than 1 Month	Modify Recurrence	Owner (System Job)
	Cancel	Next Run
Delete completed pipeline events	Resume	Waiting
Delete completed process sessions for Sync Workflows	Postpone	Waiting
	Pause	Waiting
Delete completed system jobs		Waiting
		SYSTEM
		1/17/2020 7:15 PM
		1/17/2020 11:18 AM
		1/17/2020 11:18 AM
		1/17/2020 11:18 AM

Create a new bulk delete job

1. Select New, and then select Next to start the Bulk Deletion Wizard.

2. Set the following search criteria:

- In the Look for list, select **ConversationTranscripts**.
- Select **ConversationStartTime**, select **Older Than X Months**, and type **12**. This setting keeps the transcripts for 12 months. Choose a different comparison and value if you like.
- Select **SchemaType**, leave **equals**, and type **powervirtualagents**.

Define Search Criteria

[Help](#)

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

[Help](#)

Specify the name of the bulk deletion system job, and scheduling and notification options.

Name:

Bulk Deletion Conversation Transcripts Older Than 12 Months

Bulk deletion job start time:

Immediately

At scheduled time:

1/16/2020

11:43 PM

Run this job after every:

1 days

Send an email to me (ccialmtestadmin@ccibotstest1outlook.onmicrosoft.com) when this job is finished.

Also notify:



[Back](#)

[Next](#)

[Cancel](#)

5. Select Next, and then select Submit.

Tips for getting the most out of your conversation transcripts

Variables can be used to store data relevant to your copilot content or copilot 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 copilot content.

Analyze topic usage in Copilot Studio

Article • 04/26/2024

ⓘ 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.

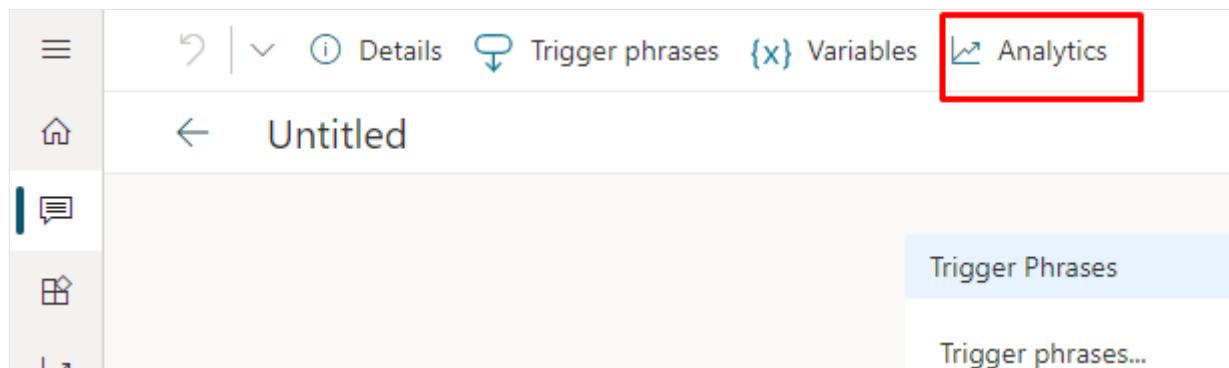
Some articles and screenshots may refer to Power Virtual Agents while we update documentation and training content.

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](#) tab:

- Escalation rate drivers (Summary tab)
- Abandon rate drivers (Summary tab)
- Resolution rate drivers (Summary tab)
- Customer satisfaction drivers (Customer Satisfaction tab)

Alternatively, open an individual topic from the [Topics page](#) and select **Analytics** at the top of the page.



The topic details page includes a variety of charts with graphical views of a topic's key performance indicators. For information about each chart, see:

- [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 impact 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 ↗](#)

Microsoft Copilot Studio messages and capacity management

Article • 03/26/2024

ⓘ Note

This article is for billed messages in the Microsoft Copilot Studio license, available for purchase starting December 1, 2023.

A *message* is an interaction between a customer and a copilot, representing one unit of consumption. A message is defined as any request or message sent to the copilot that triggers an action or a response from the copilot.

The purchase of a Microsoft 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 Microsoft Copilot Studio features for copilots in the environment.

Allocation happens in the [Power Platform Admin Center](#) when you select **Resources > Capacity > Summary** tab, and then select **Assign to an environment** in the upper-left menu. Additional information can be found in the [Power Platform capacity add-ons](#) article.

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	Copilot stored in
User logs vacation days in a conversation with the copilot in	Included with select Microsoft 365 subscriptions	Any topic type	Teams	Yes (only standard connectors)	Not used	Dataverse for Teams

Scenario	Is session billed?	Topic type	Channel	Power Automate flow	Bot Framework extensibility	Copilot stored in
Microsoft Teams.						
Copilot greets user on your website, user escalates to a human agent directly or doesn't interact with the copilot.	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
User interacts with the copilot 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

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 !\[\]\(ccfe38885c14d401d17de284bf94b899_img.jpg\)](#)

View billed sessions

Article • 04/29/2024

ⓘ 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.

A *billed session* is an interaction between a customer and a copilot. It represents one unit of consumption. The purchase of a Microsoft Copilot Studio license includes a specific number of billed sessions. Microsoft Copilot Studio pools this capacity across the entire tenant. How much of your capacity you've consumed isn't reported at the tenant level. However, you can view consumption for individual copilots.

ⓘ Note

This article is for billed sessions in the legacy Power Virtual Agents license. This legacy license is no longer available for purchase starting on January 1, 2024.

A billed session begins when a user topic or premium functionality is triggered. A billed session ends when one of the following conditions is met:

- The user ends the chat session.
When the copilot doesn't receive a new message for more than 30 minutes, the session is considered closed.
- The session is longer than 60 minutes.
The first message that occurs after 60 minutes starts a new session.
- The session has more than 100 *turns*.
A turn is defined as one exchange between a user and the copilot. The one-hundred-and-first turn starts a new session.

Sessions that use the [embedded test chat](#) don't count toward billed sessions.

View your copilot's billed sessions

1. In Microsoft Copilot Studio, in the navigation menu, select **Analytics**.
 2. Select the **Billing** tab.
- By default, the page displays billed sessions for the last seven days. To change the time period shown, select a period start date and (optionally) a period end date in the calendar controls.
- The Billed Session report doesn't distinguish between sessions that are included in the Microsoft Copilot Studio for Microsoft Teams plan and sessions that aren't included. The report might show an inflated number of billed sessions.

Common scenarios

The following table helps illustrate when sessions are billed and when sessions are 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	Copilot stored in
User logs vacation days in a conversation with the copilot in Microsoft Teams.	Included with select Microsoft 365 subscriptions	Any topic type	Teams	Yes (only standard connectors)	Not used	Dataverse for Teams
Copilot greets user on your website, user escalates to a human agent directly or doesn't interact with the copilot.	No	System	Any channel	Not used	Not used	Any environment type
Greeting topic redirects to Bot Framework	Yes	Any topic type	Any channel	Either used or not used	Yes	Any environment type

Scenario	Is session billed?	Topic type	Channel	Power Automate flow	Bot Framework extensibility	Copilot stored in
Composer or a skill topic for advanced operations.						
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
User interacts with the copilot 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

Frequently asked questions

What's the difference between total sessions, engaged sessions, and billed sessions?

- *Billed sessions* are used to measure usage against the capacity that is allocated when you purchase a license. For more information, see the [Microsoft Power Platform Licensing Guide](#).
- *Total sessions* and *engaged sessions* are metrics of *analytics sessions*, used to determine how efficient a copilot is in addressing a user's questions.
- *Billed sessions* and *analytics sessions* are distinct and independent.

For more information, see [Summarize your copilot's performance and usage](#).

Where can I see the total number of billed sessions for my organization?

You can only view billed sessions for individual copilots.

If I'm on a trial subscription but have some billed sessions, do I have to pay?

Not necessarily. The trial offer provides you with some billed sessions.

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Key concepts - Configure 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.

Copilot Studio has many advanced features that let you extend the capabilities of what copilot conversations can do.

These features allow you to personalize the look and feel of your copilot and give you more flexibility in what your users can ask your copilot to do.

ⓘ Note

With the Web app version, you can use single sign-in (SSO) to backend systems, seamlessly integrate between virtual support conversations, and customize the look and feel of your copilot.

Related articles

Web app

 Expand table

Article	Description
Assign access and change security options	Determine who can use the copilot in Microsoft Teams.
Configure end-user authentication	Allow users to sign in with any OAuth2 identity provider directly in a copilot conversation.
Configure single sign-on	Retain a user's authentication to use other signed-in products.

Article	Description
Manage end user credentials	Determine whether your copilot uses author supplied credentials, or prompts end users for credentials.
Configure handoff to a generic engagement hub	Hand off conversations to any engagement hub.
Configure handoff to Omnichannel for Customer Service	Seamlessly and contextually hand off copilot conversations to live (human) agents using Omnichannel for Customer Service.
Configure your copilot to automatically start the conversation	Make your copilot start the conversation automatically whenever the copilot is loaded.
Customize the look and feel of your copilot	Send and receive dynamic responses (such as adaptive cards and carousels) that you can custom render directly in the copilot conversation.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Share your copilot with other users

Article • 04/12/2024

ⓘ 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.

You can share your copilots with others to grant them special permissions. There are two different sharing methods:

- Grant security groups, or your whole organization, permission to chat with the copilot.
- Invite users to collaborate on your copilot project. Collaborators always have permission to chat with the copilot.

Prerequisites

- The copilot's [end user authentication setting](#) must be configured to **Only for Teams or Manual**, 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 copilot in your organization.

Web app

Share a copilot for chat

Copilot makers that you've [shared your copilot with for collaborative authoring](#) always have permission to chat with the copilot. However, you can also grant users permission to chat with the copilot without granting them authoring permissions.

To grant users permission to only chat with the copilot, you can either:

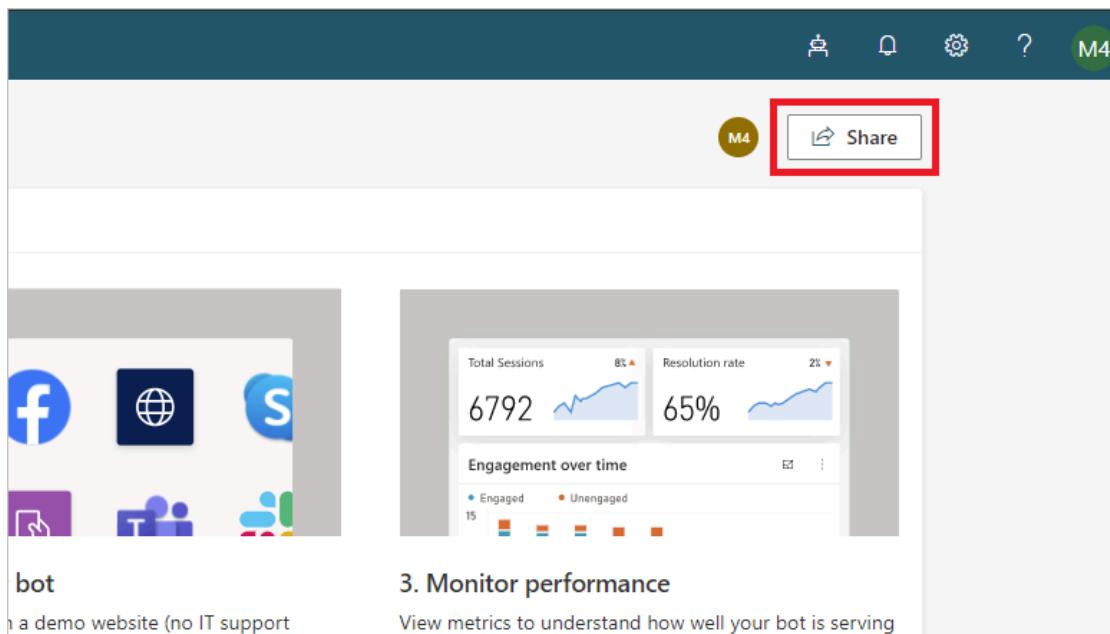
- Share your copilot with a security group.

- Share your copilot with everyone in your organization.

Share copilot with security groups

You can share your copilot with security groups so their members can chat with the copilot.

1. Select **Share** at the top of the copilot's overview page.



2. Specify the security group name that you would like to share the copilot with.

⚠ Note

When sharing a copilot for *chat*, you can only share a copilot with security groups. 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 bot

X

Share with users to collaborate or with security groups to use your bot. [Learn more](#)

-  PVASG1 Name
-  pvashg2 Owner, Manager, Power Automate user, Transc...

Select or add a user to set their permissions

My organization

-  Everyone in Test_Test_CCI Demos None

Send an email invitation to new users

Share

Cancel

3. Review the security group's permission.

Share bot

X

Share with users to collaborate or with security groups to use your bot. [Learn more](#)

PVASG1

New users

-  PVASG1 User

Shared with Sort by Name

-  Demo41 Owner, Manager, Power Automate user, Transc...

My organization

-  Everyone in Test_Test_CCI Demos None

Send an email invitation to new users

Bot permissions

The user's permissions for this bot.

-  User
Can use the bot

Data permissions ①

Make sure your users have access to the data used in Power Automate flows included in the bot. [Learn more](#)

Share

Cancel

4. If you want to let users know you've shared the copilot with them, select the **Send an email invitation to new users** checkbox.

⚠ Note

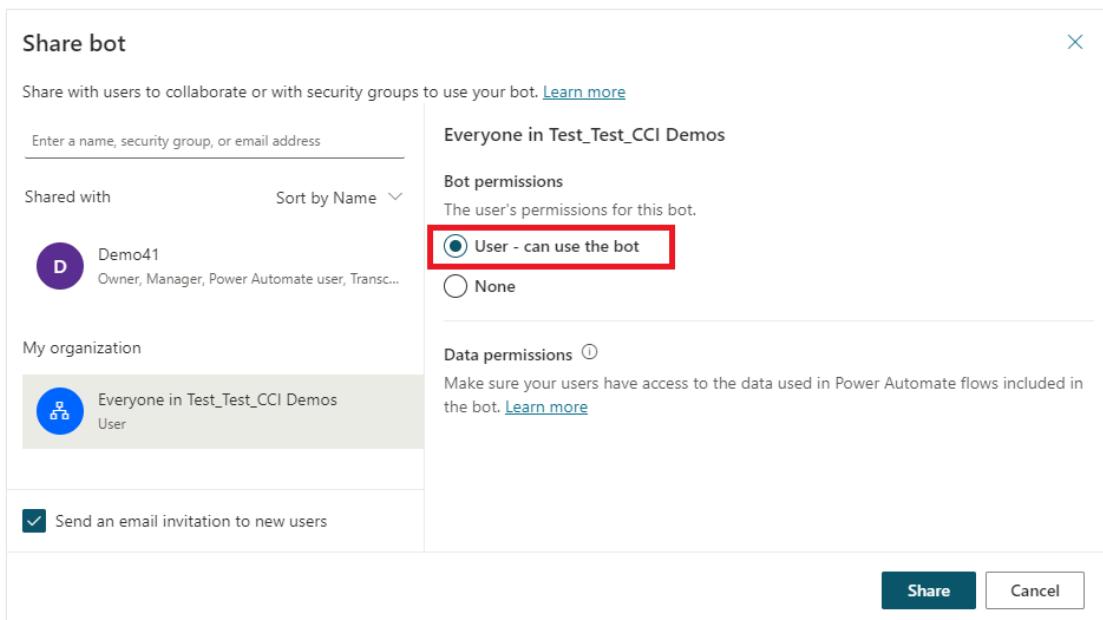
Only security groups with email enabled will receive an email invitation. You can [copy the link for the Demo website](#) and share it with users directly so they can chat with the copilot.

5. Select **Share** to share the copilot with the new security groups.

Share copilot with everyone in the organization

You can share your copilot to allow everyone in the same organization the copilot to chat with it.

1. Select **Share** at the top of the copilot's overview page.
2. Select **Everyone in your organization's name**.
3. Select **User - can use the copilot** option.



ⓘ Note

Microsoft Copilot Studio will not send email invitations to everyone in the organization. You can [copy the link for the Demo website](#) and share it with users directly so they can chat with the copilot.

4. Select **Share** to share the copilot with everyone in the organization.

Share a copilot for collaborative authoring

Individual users that you share a copilot with can view, edit, configure, share, and publish the copilot. They can't delete the copilot.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4DdcM?postJs||Msg=true>

ⓘ Note

You can only share a copilot 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. To share a copilot, sign in to Microsoft Copilot Studio and select **Share** at the top of the copilot's overview page.

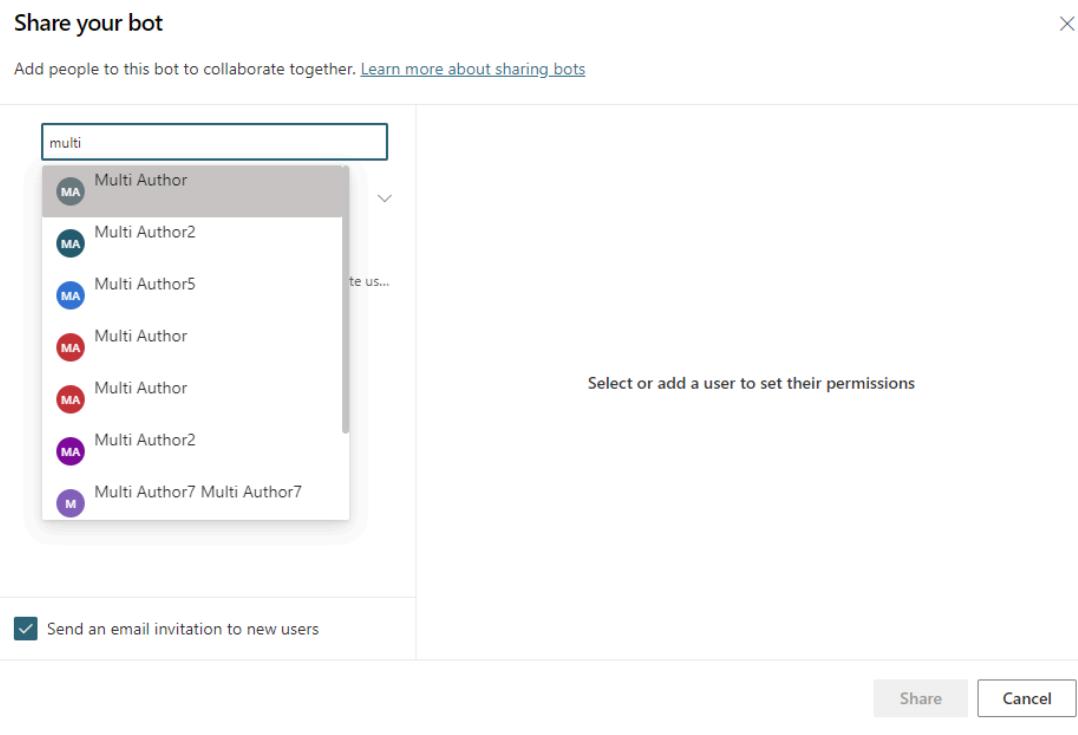
The screenshot shows the Microsoft Copilot Studio interface. At the top right, there is a 'Share' button with a red box drawn around it. Below the header, there are several icons: Facebook, a globe, and a blue 'S'. On the left, there is a section labeled 'bot' with a demo website link. On the right, there is a performance monitoring section titled '3. Monitor performance' which includes a chart showing Total Sessions (6792), Resolution rate (65%), and Engagement over time (Engaged: 15, Unengaged: 15).

2. Specify the name or email address of each user that you would like to share the copilot with.

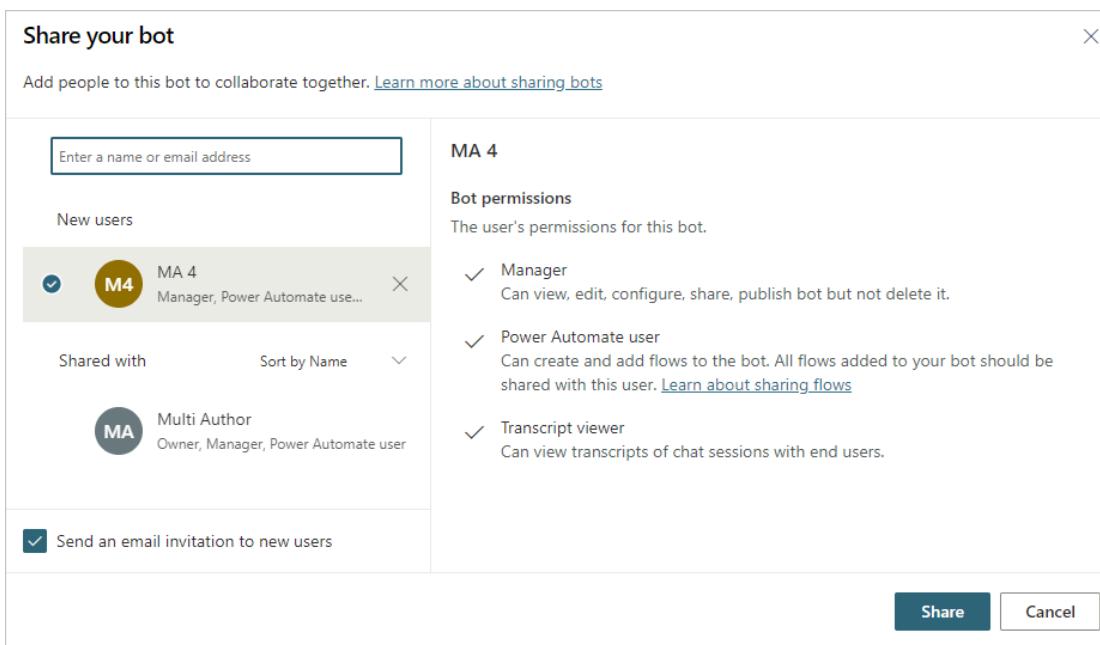
! Note

When sharing a copilot for *collaborative authoring*, you can only share it with individual users. You can't share it with:

- A security group in your organization.
- Distribution group in your organization.
- Users or groups outside of your organization.



3. Review the user's permissions.



4. If you want to let users know you've shared the copilot with them, select the **Send an email invitation to new users** checkbox.

5. Select **Share** to share the copilot with the new users.

ⓘ Important

It can take up to 10 minutes for the copilot to show up in the shared user's copilot list if the user was not previously part of the environment of the shared copilot.

Insufficient environment permissions

Users in the environment must have the **Environment maker** security role before a copilot can be shared with them.

The screenshot shows a 'Share your bot' dialog box. At the top, there's a search bar labeled 'Enter a name or email address'. Below it, under 'New users', there's a list with one item: 'Multi Author2' (represented by a purple circle with 'MA') with the note 'No permissions'. To the right of this list, a message states: 'You can't share the bot with this user because they don't have enough permissions. Contact your admin to give this user access to bots in this environment'. Below the list, there are buttons for 'Shared with' and 'Sort by Name'. Under 'Shared with', there's a list for 'Multi Author' (represented by a grey circle with 'MA') with the note 'Owner, Manager, Power Automate user'. At the bottom left, there's a checked checkbox for 'Send an email invitation to new users'. On the far right, there are 'Share' and 'Cancel' buttons.

System administrators of the environment need to assign the **Environment maker** security role to the user before you share the copilot. If you're a **System administrator**, you can [assign the Environment maker role when you share the copilot](#).

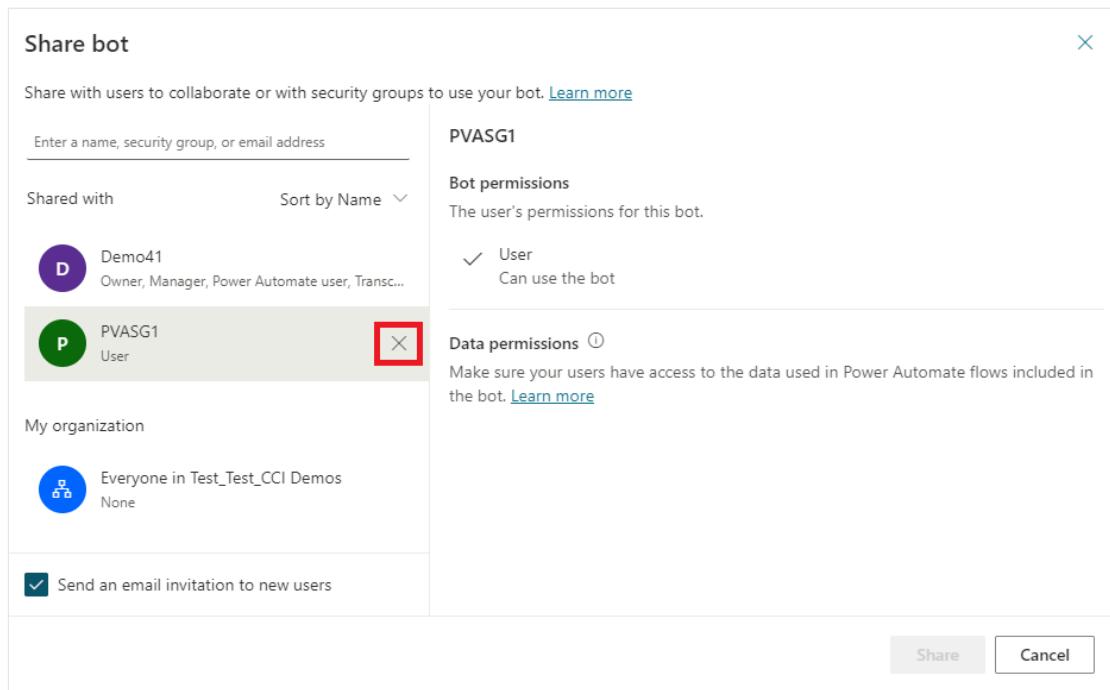
Learn more about [security roles](#) and [predefined security roles](#).

Stop sharing a copilot

You can stop sharing a copilot with a security group or everyone in your organization.

Stop sharing with a security group

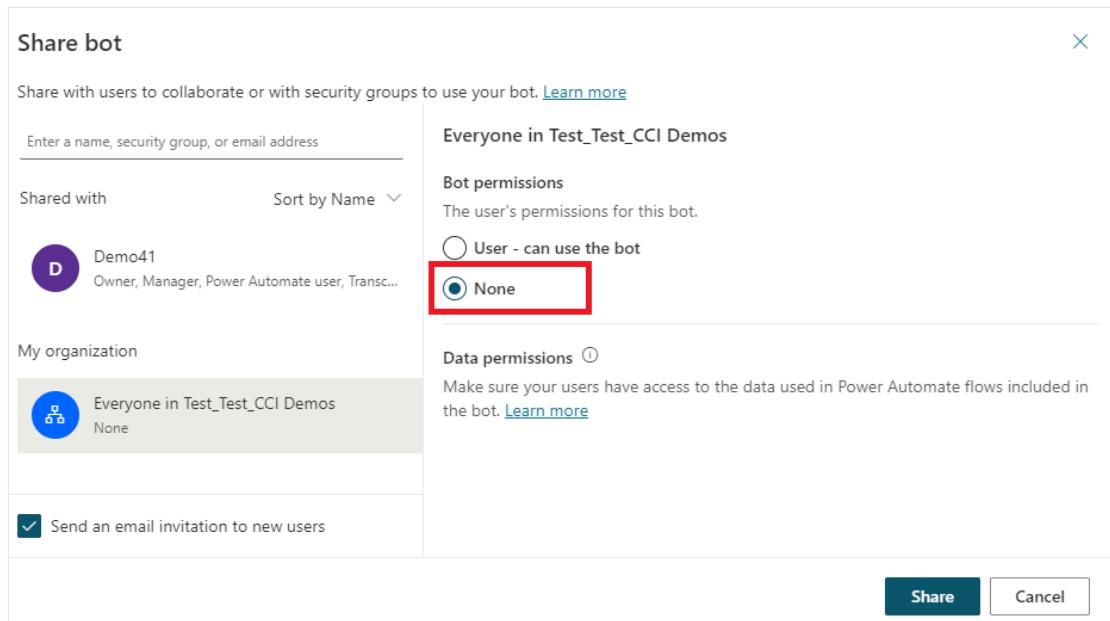
1. Select **Share** at the top of the copilot's overview page.
2. Select **X** next to the security groups that you want to stop sharing the copilot with.



3. Select **Share** to stop sharing the copilot with the security groups.

Stop sharing with everyone in the organization

1. Select **Share** on the copilot's overview page.
2. Select **Everyone in your organization's name**.
3. Select **None** option.

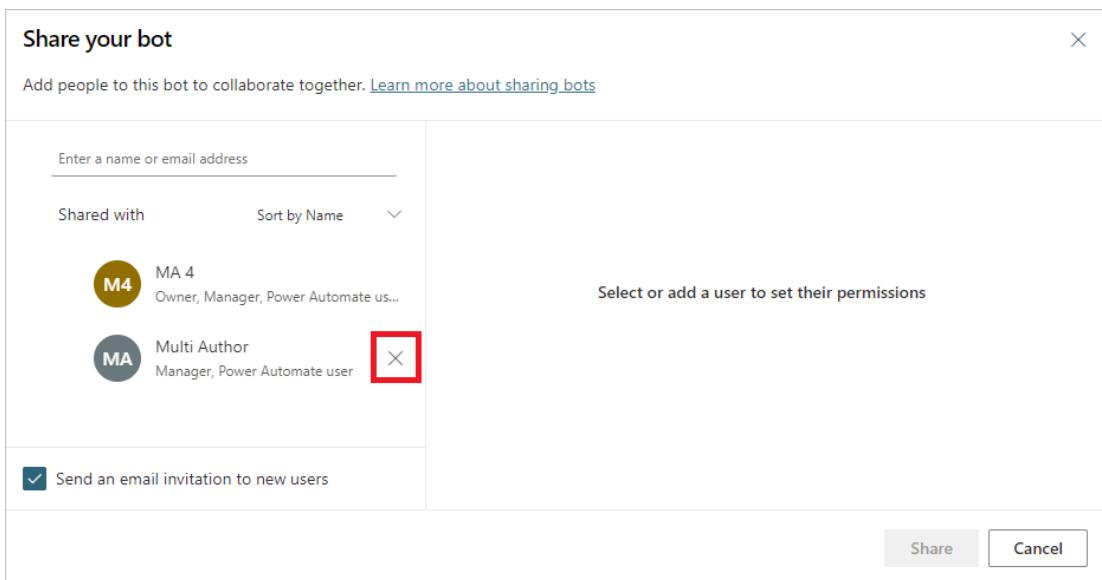


4. Select **Share** to stop sharing the copilot with everyone in the organization.

Stop sharing a copilot with individual users

You can stop sharing a copilot with a user, and any shared user can stop the copilot from being shared with other users, except for the owner. The owner always has access to the copilot.

1. Sign in to Microsoft Copilot Studio and select **Share** at the top of the copilot's overview page.
2. Select **X** next to the users that you want to stop sharing the copilot with.



3. Select **Share** to stop sharing the copilot with the selected users.

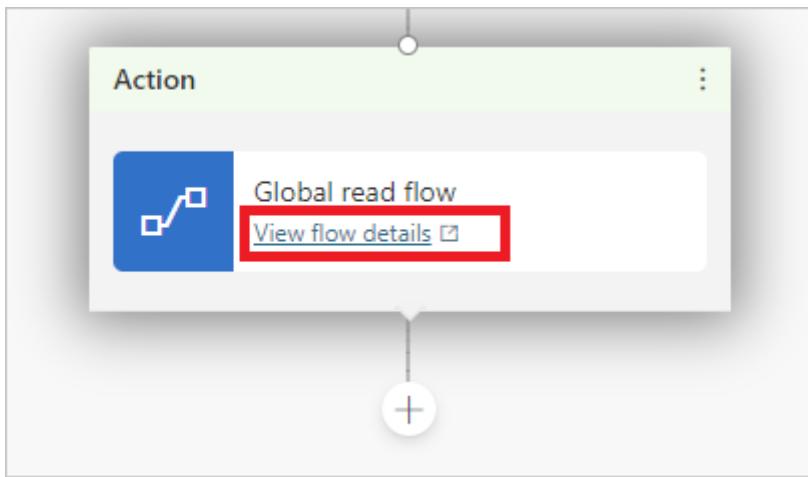
Share Power Automate flows used in a copilot

You can [add actions to a copilot using flows in Power Automate](#); however flows in a copilot aren't automatically shared with other users when sharing a copilot.

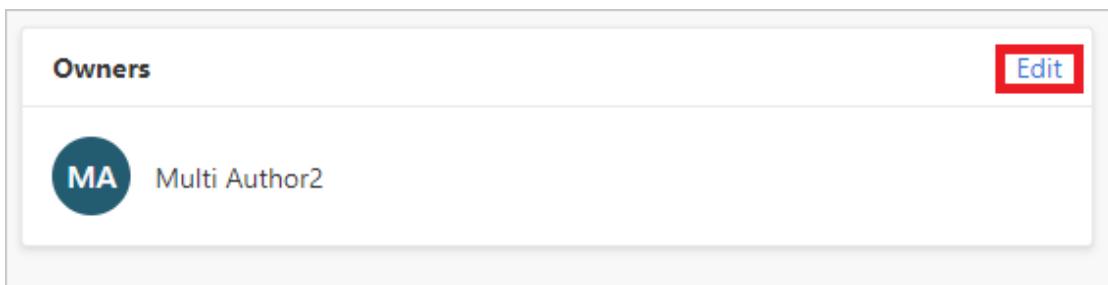
Users who don't have access to the shared flow can still run it by using the test copilot canvas.

To let other users edit or add flows, you need to 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 a copilot.

The **Environment security roles** section shows when you share a copilot and only if you're a **System administrator**. It lets you share copilots with users who don't have sufficient environment permissions to use Microsoft Copilot Studio.

You must be a **System administrator** of the environment where the copilot is located to view and add security roles.

Note

You can only *assign* security roles when sharing a copilot. 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](#).

Assign the Environment maker security role during copilot sharing

You can assign the **Environment maker** security role when sharing a copilot with a user who doesn't have sufficient environment permissions to run Microsoft Copilot Studio.

When [sharing the copilot for chat](#), if the specified user doesn't have sufficient permissions to use Microsoft Copilot Studio in the environment, you are notified that the **Environment maker** security role is assigned to the user so they can use the copilot.

The screenshot shows the 'Share your bot' dialog box. On the left, there's a search bar labeled 'Enter a name or email address'. Below it, under 'New users', a user named 'Multi Author' (Manager, Power Automate user) is selected. Under 'Shared with', another user named 'MA 4' (Owner, Manager, Power Automate us...) is listed. A checkbox 'Send an email invitation to new users' is checked. On the right, the 'Multi Author' section shows 'Bot permissions' with 'Manager' and 'Power Automate user' checked, and 'Transcript viewer' unchecked. The 'Environment security roles' section contains a note: 'This user needs environment security roles to work with bots in Power Virtual Agents. By sharing the bot the user will be assigned the selected security roles.' It lists 'Environment maker' (checked) and 'Bot transcript viewer' (unchecked). At the bottom are 'Manage security roles', 'Share', and 'Cancel' buttons.

Assign the Copilot transcript viewer security role during copilot sharing

You can assign the **Copilot transcript viewer** security role to users who don't have conversation transcript access when sharing a copilot.

Based on the content and target audience of the copilot, you may want to give transcript access only to users who have appropriate privacy training.

ⓘ Important

Conversation transcript access is managed by environment security roles. After assigning the **Copilot transcript viewer** security role to a user, that user can access conversation transcripts for all copilots that they create or are shared with in the environment.

By default, only admins have the **Copilot transcript viewer** or **Bot transcript viewer** role assigned. We recommend you create an environment for copilots that controls which user can view conversation transcripts. Learn more about how to [Create a new environment for your copilots](#)

When [sharing the copilot for chat](#), you can assign the **Copilot transcript viewer** role by selecting the checkbox.

Share your bot X

Add people to this bot to collaborate together. [Learn more about sharing bots](#)

Enter a name or email address	Multi Author
Shared with	Sort by Name
 MA 4 Owner, Manager, Power Automate us...	Bot permissions The user's permissions for this bot. <ul style="list-style-type: none">✓ Manager Can view, edit, configure, share, publish bot but not delete it.✓ Power Automate user Can create and add flows to the bot. All flows added to your bot should be shared with this user. Learn about sharing flows✓ Transcript viewer Can view transcripts of chat sessions with end users.
 MA Multi Author Manager, Power Automate user X	Environment security roles Security roles allow a user to work with bots in Power Virtual Agent in this environment <ul style="list-style-type: none">✓ Environment maker Can create bots, can be a bot Manager, and can use Power Automate<input checked="" type="checkbox"/> Bot transcript viewer Can view transcripts of chat sessions with end users.
Manage security roles	
<input checked="" type="checkbox"/> Send an email invitation to new users	
Share Cancel	

Manage security roles

You can [manage environment security roles](#) at the Power Platform admin center.

Data permissions

To make sure users who chat with your copilot can run [Power Automate](#) flows correctly, you should test your copilot to make sure their permissions with the integrated systems are configured correctly.

Collaborate on copilots

After you've shared a copilot, everyone can edit the copilot's topics.

In the **Topics** list, you can see who's working on a topic under the **Editing** column. You can select the person's icon to quickly chat with them in Teams or send them an email.

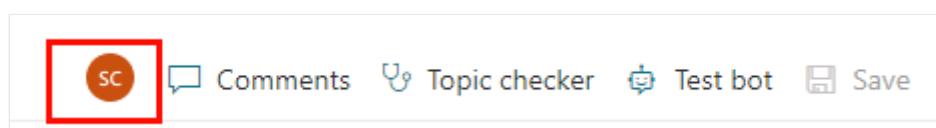
This list can help prevent conflicts if multiple people are working on the same topic.

Topics ①						<input type="text"/> Search existing topics
Existing (12)		Suggested (0)				
Type	Name	Trigger phrases	Status	Errors	Editing	Modified by ↓
Lesson	Lesson 1 - A simple topic	(4) When are you closed	<input checked="" type="button"/> On			Kristine Love 30 min ago
Lesson	Lesson 2 - A simple topic with a condition an...	(5) Are there any stores arou...	<input checked="" type="button"/> On			Samuel Carlson 1 h ago
Lesson	Lesson 3 - A topic with a condition, variables...	(5) Buy items	<input checked="" type="button"/> On			Therese Bray 3 h ago
Lesson	Lesson 4 - A topic with a condition, variables...	(5) What is the best product f...	<input checked="" type="button"/> On		KS 2+	Keith Strong 1 d ago
Greeting	Greeting	(52) Good afternoon	Always on			01/14/2019
Escalate	Escalate	(65) Talk to agent	Always on			01/14/2019
End of Conversation	End of Conversation	No trigger phrases	Always on			01/14/2019

① Note

The list of authors in the **Currently editing** column is only refreshed when the page is loaded.

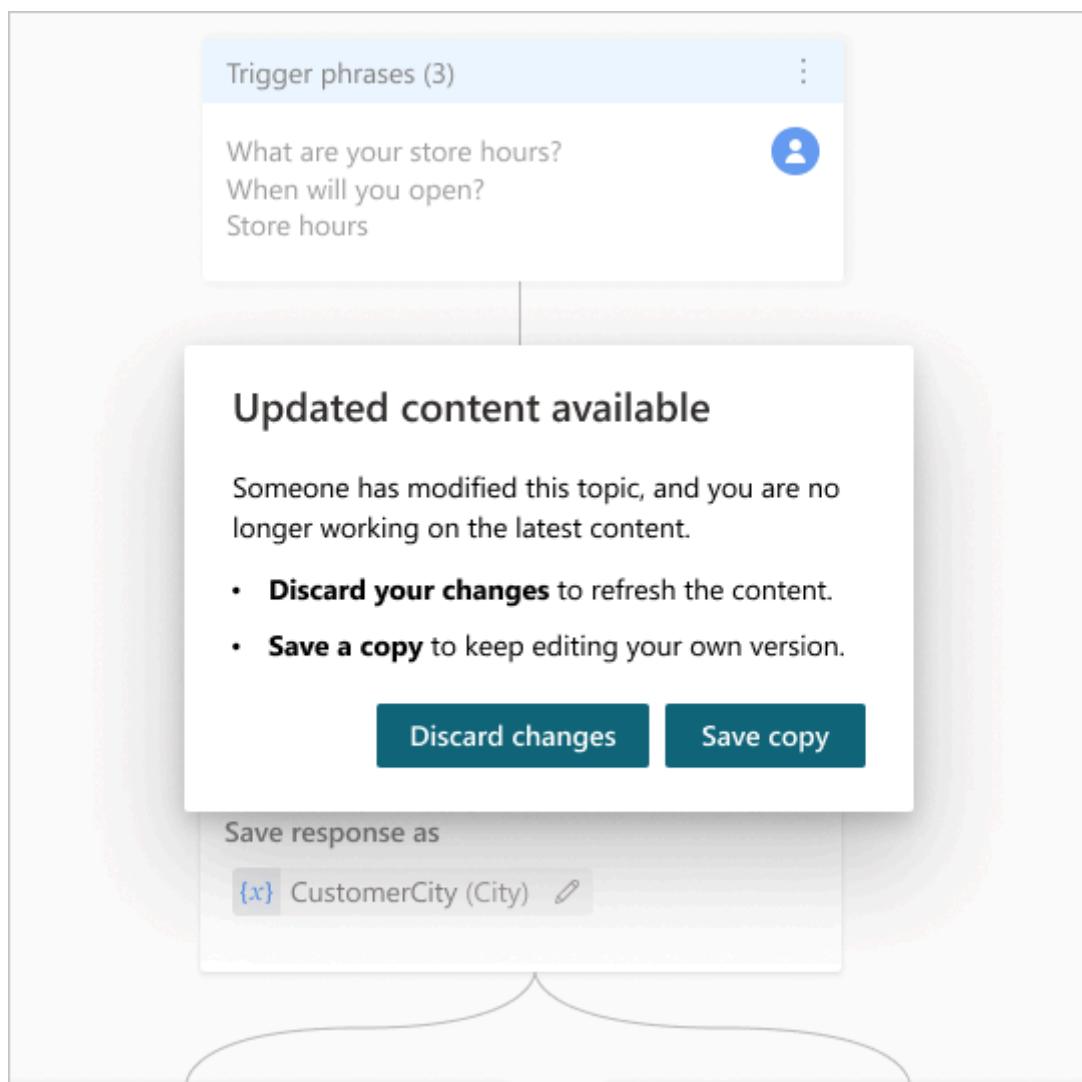
A list of who's editing the page also appears as user icons on the top of the authoring canvas when you've opened a topic for editing.



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 people 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, you go to save it, only to find yourself in a conflict state.

Microsoft Copilot Studio detects this and prevents you from overwriting your coworker by prompting you with an option: you can reload the content with the latest changes (discarding your work), or you can save a copy of the topic (keeping your changes in a new, duplicate copy of the topic).



If you save your changes to a new topic, you can then review your coworker's changes and merge the two topics, deleting the copy of the topic once you've finished.

Configure user authentication in Microsoft 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.

Authentication allows users to sign in, giving your copilot 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 Microsoft Copilot Studio copilot 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 your copilot](#) when you edit a topic.

Microsoft Copilot Studio supports the following authentication providers:

- Azure Active Directory v1
- Microsoft Entra ID
- Any identity provider that complies with the [OAuth2 standard](#)

ⓘ Important

Changes to the authentication configuration only take effect after you publish your copilot. Make sure to plan ahead before you make authentication changes to your copilot.

<https://www.microsoft.com/en-us/videoplayer/embed/RE4n4G2?postJs||Msg=true> ↗

Choose an authentication option

Microsoft Copilot Studio supports several authentication options. Choose the one that meets your needs.

To change your copilot's authentication settings, in the navigation menu under **Settings**, go to the **Security** tab and select the **Authentication** card.

The screenshot shows the Microsoft Copilot Studio Settings interface. On the left, a sidebar lists various settings categories: Copilot details, AI integration tools, Generative AI, **Security** (which is selected and highlighted with a red box), Entities, Skills, Languages, and Language understandings. The main content area is titled "Security" and contains four cards: "Sharing" (invite people to collaborate), "Authentication" (verify a user's identity during a chat, which is also highlighted with a red box), "Web channel security" (review other enhanced security options), and "Allowlist" (let other copilots call your copilot as a skill).

The following authentication options are available:

- No authentication
- Authenticate with Microsoft
- Authenticate manually

Authentication

X

Verify a user's identity during a conversation. The bot 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

Basic bot setup with no authentication action or authentication variables.

Only for Teams and Power Apps

User ID and User Display Name authentication variables available. Automatically sets up Azure Active Directory (AAD) authentication for Teams and Power Apps. All other channels will be disabled. [Learn more](#)

Manual (for custom website)

Support AAD or any OAuth2 identity provider. Authentication variables are available including authentication token.

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](#)

No authentication

No authentication means your copilot doesn't require your users to sign in when interacting with the copilot. An unauthenticated configuration means your copilot can only access public information and resources.

 Caution

Selecting the **No authentication** option will allow anyone who has the link to chat and interact with your bot or copilot.

We recommend you apply authentication, especially if you are using your bot or copilot 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 will be disabled.

Additionally, the **Authenticate with Microsoft** option is not available if your copilot is integrated with [Dynamics 365 Customer Service](#).

Teams and Power Apps authentication is enabled by default for copilots and copilots that you create in Microsoft Copilot Studio.

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 copilot needs expanded scope.

Only the Teams channel is available if you select this option. If you need other channels but still want authentication for your copilot (such as when [using generative AI features](#)), choose **Authenticate manually**.

If you select the **Authenticate with Microsoft** option, the following variables are available in the authoring canvas:

- `UserID`
- `UserDisplayName`

For more information about these variables and how to use them, see [Add user authentication to a Microsoft Copilot Studio copilot](#).

`AuthToken` and `IsLoggedIn` variables aren't available with this option. If you need an authentication token, use the **Manual** option.

If you change from **Authenticate manually** to **Authenticate with Microsoft** authentication, and your topics contain the variables `AuthToken` or `IsLoggedIn`, they're displayed as **Unknown** variables after the change. Make sure to correct any topics with errors before you publish your copilot.

Authenticate manually

You can configure any Microsoft Entra ID v1, Microsoft Entra ID, or OAuth2-compatible identity provider with this option. The following variables are available in the authoring canvas after you configure manual authentication:

- `UserID`
- `UserDisplayName`
- `AuthToken`
- `IsLoggedIn`

For more information about these variables and how to use them, see [Add user authentication to a Microsoft Copilot Studio copilot](#).

Once the configuration is saved, make sure to publish your copilot so the changes take effect.

 **Note**

- Authentication changes only take effect after the copilot is published.
- This setting can be controlled by the corresponding admin control in Power Platform. When the control is enabled, it prevents the **Manual** setting from being enabled or disabled within Copilot Studio. The control is always enabled, and the **Manual** setting can't be modified in Copilot Studio.

Required user sign in and copilot sharing

Require users to sign in determines whether a user needs to sign in before talking with the copilot. We highly recommend that you turn on this setting when your copilot needs to access sensitive or restricted information.

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

This option isn't available when the **No authentication** option is chosen.

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 [Configure DLP to require authentication in the Power Platform admin center](#).

If you turn off this option, your copilot 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 copilot 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](#).

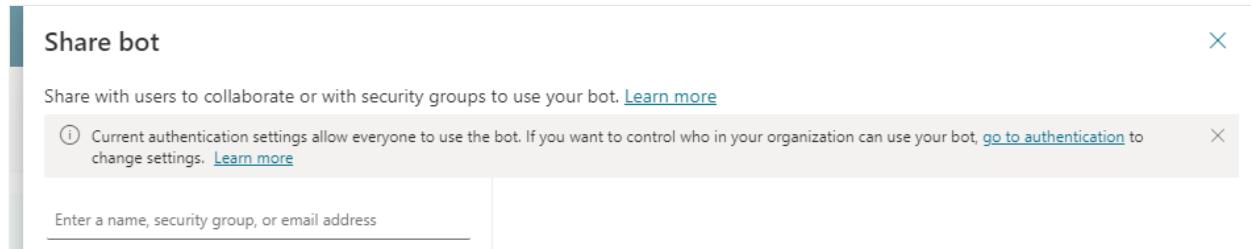
Control who can chat with the copilot in the organization

Your copilot's authentication and **Require user to sign in** setting in combination determines whether you can [share the copilot](#) to control who in your organization can chat with it. The authentication setting doesn't affect sharing a copilot for collaboration.

- **No authentication:** Any user who has a link to the copilot (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 copilot.
- **Authenticate with Microsoft:** The copilot 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 copilot sharing to control who in your organization can chat with the copilot.
- **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 copilot using copilot 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 copilot. You can't control which specific users in your organization can chat with the copilot using copilot sharing.

When a copilot's authentication setting can't control who can chat with it, if you select **Share** on the copilot's overview page a message informs you that anyone can chat with your copilot.



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.
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.
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> .

Field name	Description
Scope list delimiter	The separator character for the scope list. Empty spaces aren't supported in this field. ¹
Scopes	The list of scopes that you want users to have after they've signed in. Use the Scope list delimiter to separate multiple scopes. ¹ Only set necessary scopes and follow the least privilege access control principle .
Service provider	The service provider you want to use for authentication. For more information, see OAuth generic providers .
Tenant ID	Your Microsoft Entra ID tenant ID. Refer to Use an existing Microsoft Entra ID tenant to learn how to find your tenant ID.
Token body template	The template for the token body.
Token exchange URL (required for SSO)	This is an optional field used when you're configuring single sign-on .
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 (?).

¹ 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.

Remove the authentication configuration

1. In the navigation menu, under **Settings**, select **Security**. Then select the **Authentication** card.
2. Select **No authentication**.
3. Publish the copilot.

If authentication variables are being used in a topic, they become **Unknown** variables. Go to the Topics page to see which topics have errors and fix them before publishing.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Disable default authentication for Microsoft Copilot Studio copilots

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.

You can configure anyone to chat with your copilot by removing or changing the default authentication configuration, which is automatically enabled when copilots are created.

ⓘ Note

Copilots created in the web app and in Microsoft Teams come with authentication pre-configured to use **Only for Teams and Power Apps** authentication.

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

Read more about authentication options for when people interact with your copilot in the [Configure end-user authentication](#) article.

Web app

The copilot comes with the **Only for Teams and Power Apps** authentication option turned on. The copilot automatically uses Microsoft Entra ID authentication for Teams and Power Apps without requiring any manual setup, and only lets you chat with your copilot on Teams.

If you want to allow anyone to chat with your copilot, select **No authentication** in the **Security** settings for your copilot. If you want to use other channels and still have authentication for your copilot, select **Manual** authentication.

⊗ Caution

Selecting the **No authentication** option will allow anyone who has the link to chat and interact with your bot or copilot.

We recommend you apply authentication, especially if you are using your bot or copilot within your organization or for specific users, along with [other security and governance controls](#).

Authentication



Verify a user's identity during a conversation. The bot 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

Basic bot setup with no authentication action or authentication variables.

Only for Teams and Power Apps

User ID and User Display Name authentication variables available. Automatically sets up Azure Active Directory (AAD) authentication for Teams and Power Apps. All other channels will be disabled. [Learn more](#)

Manual (for custom website)

Support AAD or any OAuth2 identity provider. Authentication variables are available including authentication token.

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](#)

You can further configure user authentication within a topic to [allow your copilot users 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.

Warning Using actions configured to use [end user credentials](#) will not work in this type of copilot

Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#) ↗

Configure user authentication with Microsoft Entra ID

Article • 04/15/2024

ⓘ 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.

Adding authentication to your copilot allows users to sign in, giving your copilot 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](#).

If you have tenant administration rights, you can [configure API permissions](#). Otherwise, you need to ask a tenant administrator to do it for you.

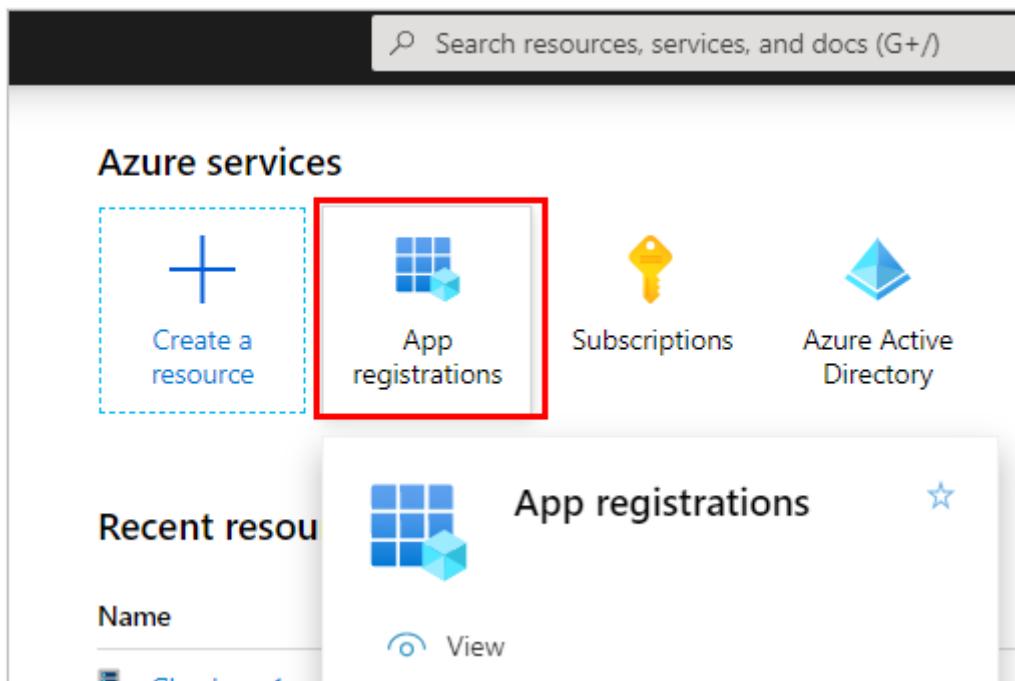
Prerequisites

- [Learn how to use user authentication in 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 copilot.
2. Go to [App registrations](#), either by selecting the icon or searching in the top search bar.



3. Select **New registration** and enter a name for the registration.

It can be helpful later to use the name of your copilot. For example, if your copilot is called "Contoso sales help," you might name the app registration "ContosoSalesReg."

4. Under **Supported account types**, select **Accounts in any organizational directory (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. Go to **Authentication**, and then select **Add a platform**.

Microsoft Azure

Home > App registrations > GBapp2reg | Authentication



GBapp2reg | Authentication

Search (Ctrl+ /)



Save



Discard



Overview

Quickstart

Integration assistant (preview)

Manage

Branding

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

Owners

Roles and administrators (Prev...)

Manifest

Platform configurations

Depending on the platform or redirect URIs, specific authentication

+ Add a platform

Supported account types

Who can use this application or a

Accounts in any organization or accounts (e.g. Skype, Xbox)

All users with a work or school Office 365 subscribers.

To change the supported ac properties may cause errors

Advanced settings

2. Under Platform configurations, select Add a platform, and then select Web.

Configure platforms

Web applications

 **Web**
Build, host, and deploy a web server application. .NET, Java, Python

 **Single-page**
Configure browser progressive web ap

Mobile and desktop applications

 **iOS / macOS**
Objective-C, Swift, Xamarini

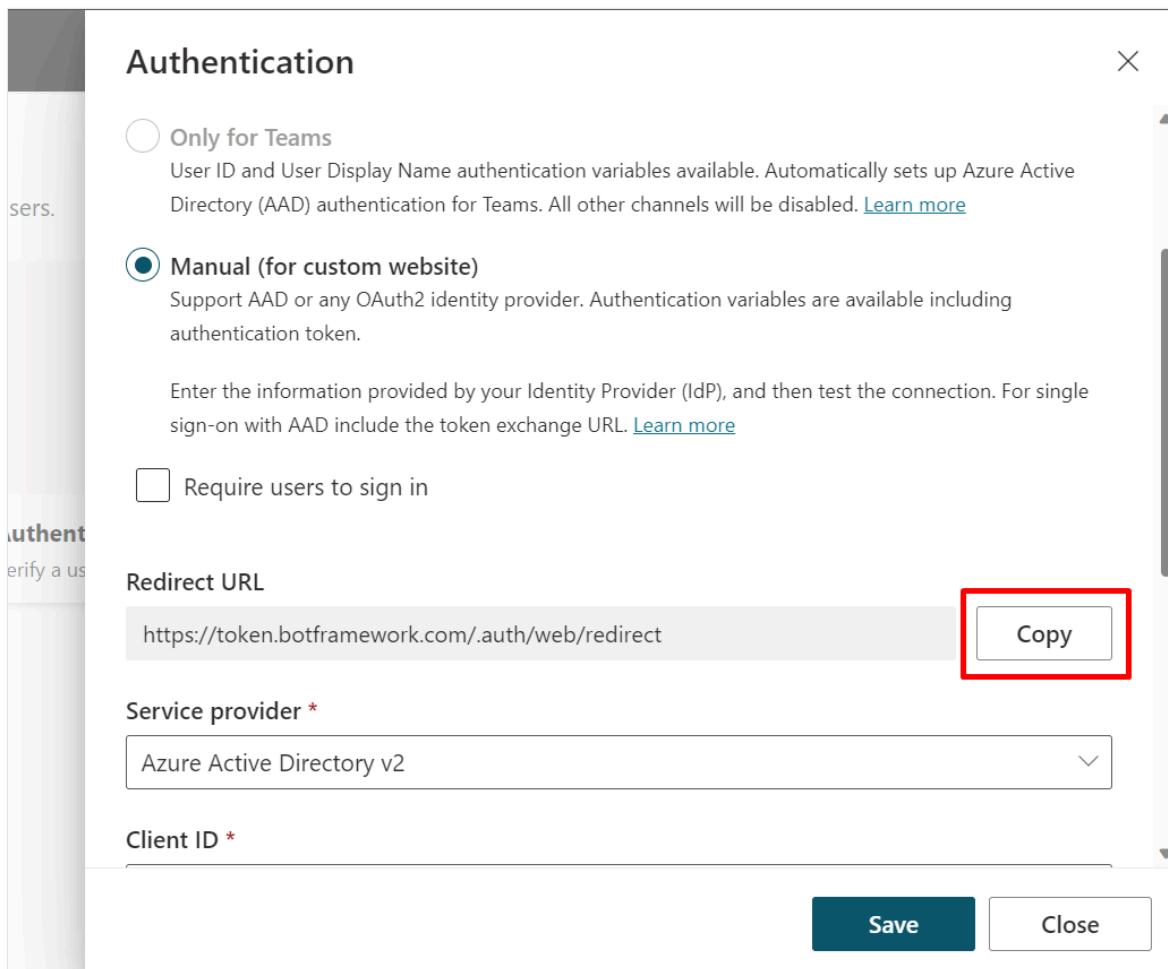
 **Android**
Java, Kotlin, Xamarin

3. Under **Redirect URIs**, enter `https://token.botframework.com/.auth/web/redirect` and `https://europe.token.botframework.com/.auth/web/redirect`.

ⓘ 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.

1. 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)**.



2. Select **Configure**.

Generate a client secret

1. Go to **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 copilot.
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 copilot's authentication later on.

Tip

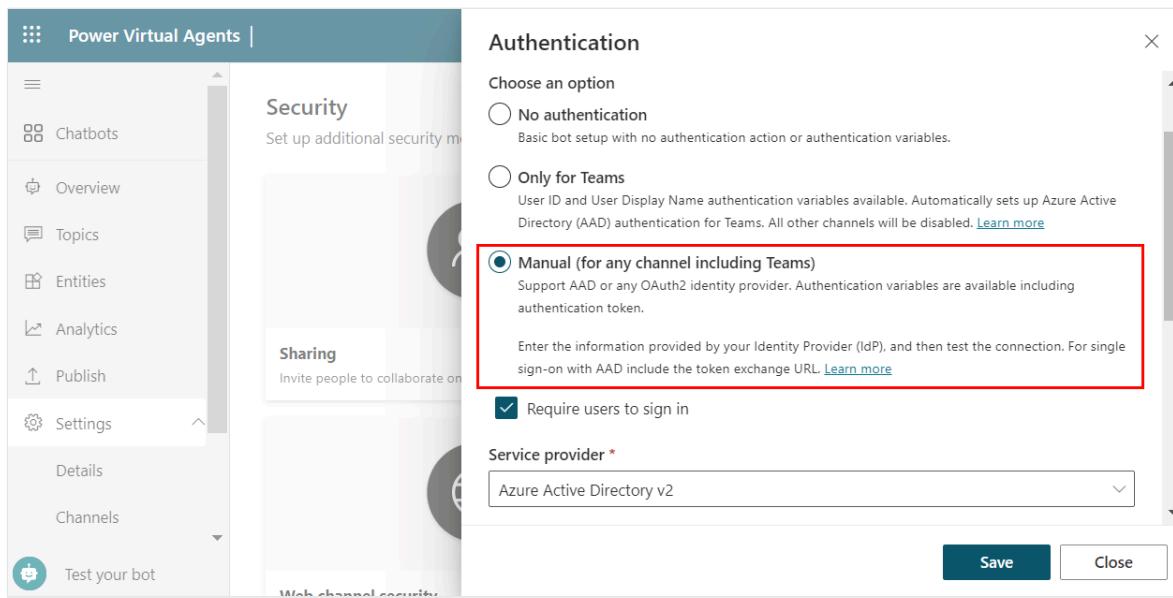
Don't leave the page before you copy the value of the client secret. If you do, the value is obfuscated and you must generate a new client secret.

Configure manual authentication

1. In Copilot Studio, in the navigation menu under **Settings**, select **Security**. Then select the **Authentication** card.

The screenshot shows the Copilot Studio navigation menu on the left with 'Settings' selected. The main area is the 'Security' page, which includes sections for 'Sharing', 'Authentication', 'Web channel security', and 'Allowlist'. The 'Authentication' section is highlighted with a red box, indicating it is the target for configuration. The 'Sharing' section also has a red box around its icon.

2. Select **Manual** (for any channel including Teams) then turn on **Require users to sign in**.



3. Enter the following values for the properties:

- **Service provider:** Select **Microsoft Entra ID**.
- **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`.

4. 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 may 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

! Note

To avoid users from having to consent to each application, a Global Administrator, Application Administrator, or Cloud Application Administrator can grant tenant-wide consent to your app 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 ⓘ View users' email address	No
<input type="checkbox"/> offline_access ⓘ Maintain access to data you have given it access to	No
<input checked="" type="checkbox"/> openid ⓘ Sign users in	No
<input checked="" type="checkbox"/> profile ⓘ View users' basic profile	No

6. Select Add permissions.

Define a custom scope for your copilot

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 sidebar with links like Overview, Quickstart, Integration assistant (preview), Manage (Branding, Authentication, Certificates & secrets, Token configuration, API permissions), and Owners. The 'Expose an API' link is highlighted with a red box. The main area has a search bar at the top. Below it, there's a section for 'Application ID URI' with a 'Set' button. Underneath, there's a 'Scopes defined by this API' section with a note about defining custom scopes. A large red box highlights the '+ Add a scope' button. Below that, there's a 'Scopes' section stating 'No scopes have been defined'. At the bottom, there's an 'Authorized client applications' section with a note about authorizing client applications.

2. Set the following properties. You can leave the other properties blank.

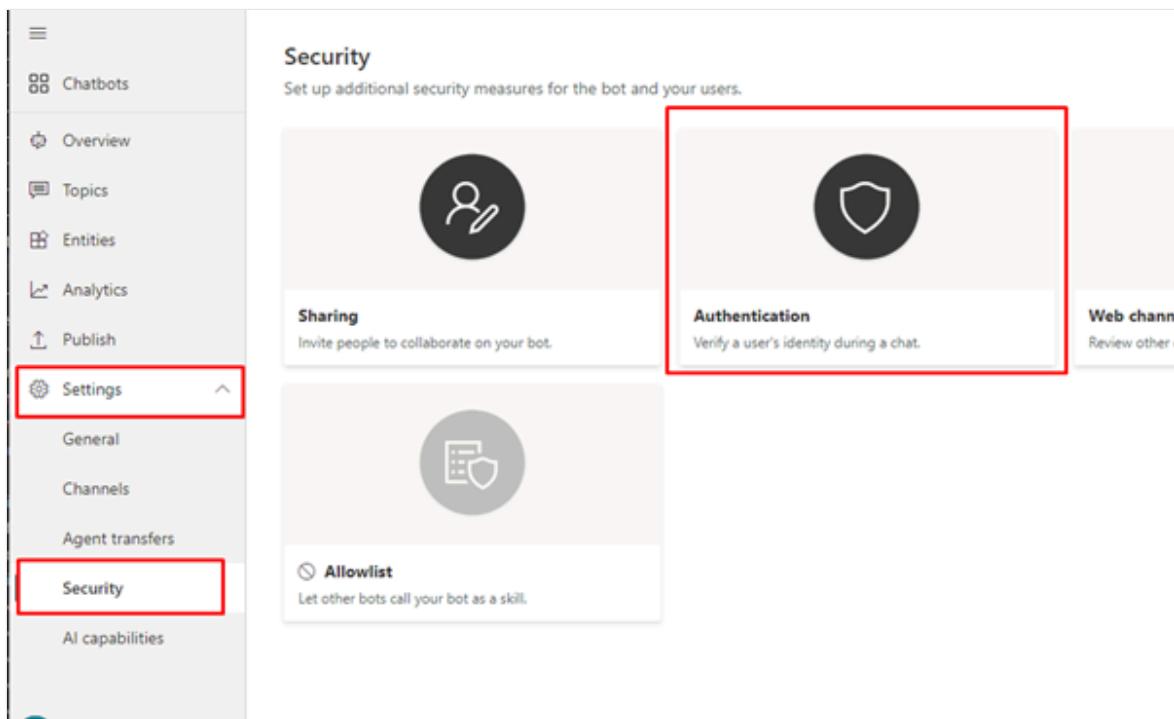
[\[\] Expand table](#)

Property	Value
Scope name	Enter a name that makes sense in your environment, such as Test.Read
Who can consent?	Select Admins and users
Admin consent display name	Enter a name that makes sense in your environment, such as Test.Read
Admin consent description	Enter Allows the app to sign the user in.
State	Select Enabled

3. Select **Add scope**.

Configure authentication in Microsoft Copilot Studio

1. In Copilot Studio, under **Settings**, select **Security**, and then select **Authentication**.



2. Select **Manual (for custom website)**.

3. Turn on **Require users to sign in**.

4. Set the following properties.

[\[\] Expand table](#)

Property	Value
Service provider	Select Microsoft Entra ID
Client ID	Enter the application (client) ID that you copied earlier in the Azure portal
Client secret	Enter the client secret you generated earlier in the Azure portal
Scopes	Enter <code>profile openid</code>

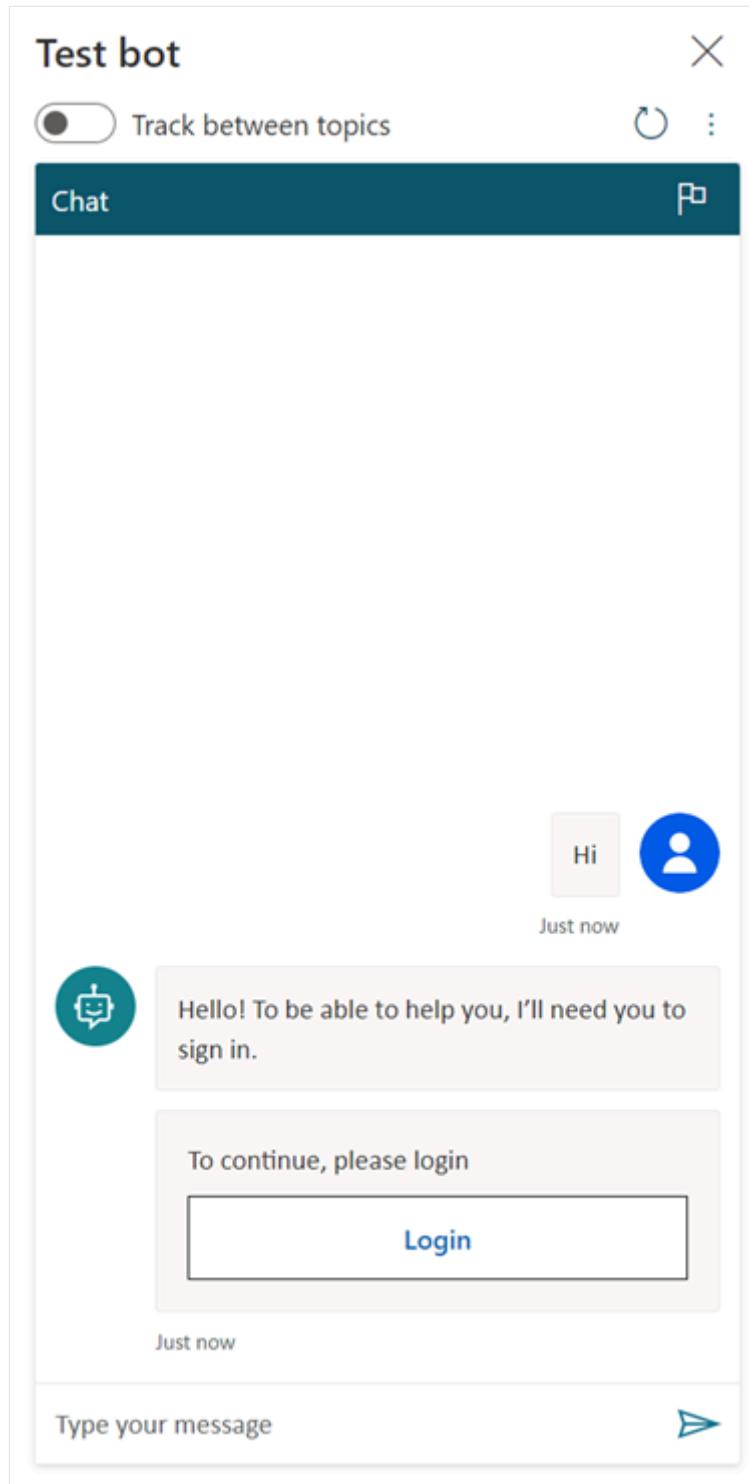
5. Select **Save**.

💡 **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 for your custom website](#).

Test your copilot

1. Publish your copilot.
2. In the **Test copilot** pane, send a message to your copilot.
3. When the copilot 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 copilot chat to complete the sign-in process.

171764

Just now



You're now logged in.

Hi! I'm a virtual agent. I can help with account questions, orders, store information, and more.

If you'd like to speak to a human agent, let me know at any time.

So, what can I help you with today?

Just now

Configure web and Direct Line channel security

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.

When you create a Microsoft Copilot Studio copilot, it's immediately available in the **Demo website** and **Custom website** channels to anyone who knows the copilot 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 copilot ID directly from within Microsoft Copilot Studio or by receiving it from someone. But, depending on the copilot'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

Microsoft Copilot Studio uses the [Bot Framework Direct Line channel](#) to connect your web page or app to the copilot.

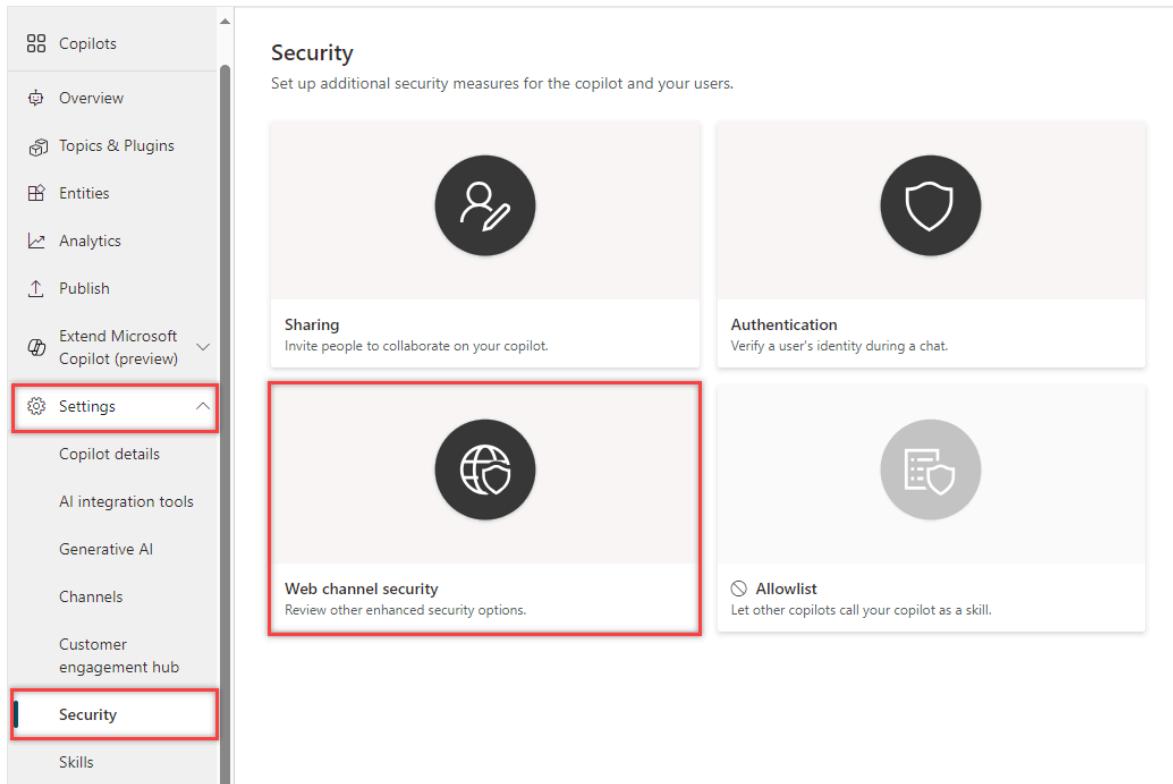
Enable or disable web channel security

You can enforce the use of secrets and tokens for each individual copilot.

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 copilot that doesn't provide this security measure won't work.

1. Under your copilot, under **Settings**, select **Security**. Then select the **Web channel security** tile.



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 copilot for this change to take effect.

You should plan ahead to avoid exposing your copilot 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 Microsoft Copilot Studio copilot.

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 copilot, you can do so without any downtime or interruption.

Microsoft 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 copilot conversation. For more information, see the section **Get Direct Line token** in the [Add copilot to mobile and custom apps](#) article.

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 the [Direct Line API - Authentication](#) article.

Related articles

- [Key Concepts - Publish your copilot](#)
- [Add copilot to mobile and custom apps \(web-based or native\)](#)

- Add copilot to Azure Bot Service channels
 - Manage end user credentials -
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Configure single sign-on with Microsoft Entra ID

Article • 04/15/2024

ⓘ 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 supports single sign-on (SSO). SSO allows copilots on your website to sign customers in if they're already signed in to the page or app where the copilot is deployed.

Prerequisites

- [Enable end-user authentication with Microsoft Entra ID](#).
SSO is only supported for Microsoft Entra ID. Other account types such as Microsoft Account or other OAuth accounts aren't supported.
- [Add an authentication topic to your copilot](#).
- [Use a custom canvas](#).

For example, the copilot 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 copilot.
3. Configure authentication in Copilot Studio to enable SSO.
4. Configure your custom canvas HTML code to enable SSO.

ⓘ Important

SSO is currently not supported when a copilot has been either:

- Published to a [SharePoint website](#).
- Published to a [Power Apps portal](#).

Supported channels

The following table details the [channels](#) that currently support SSO. You can suggest support for extra channels [at the Microsoft 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 ¹	Supported
Mobile App	Not supported
Omnichannel for Customer Service ²	Supported

¹ If you also have the Teams channel enabled, you need to follow the configuration instructions on the [Configure SSO for Teams channel](#) 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.

² Only the live chat channel is supported. For more information, see [Configure handoff to Dynamics 365 Customer Service](#).

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 copilot
- A *canvas app registration*, which enables SSO for your custom web page

We don't recommend reusing the same app registration for both your copilot 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 same instructions again to create a second app registration, which serves as your canvas app registration.
3. Return to this article.

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 copilot's token endpoint URL

1. In Copilot Studio, go to **Settings**, 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

`https://11f42229f782ef0483e2a3528d3aac0.0.environment.api.preprod.powerplatf`

[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 instructions below.

1. Go to the **Overview** page in Azure portal and copy the **Application (client) ID** and **Directory (tenant) ID** from your canvas app registration.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, the breadcrumb navigation shows 'Home > App registrations > ContosoSalesCanvas'. The main content area displays the details for the 'ContosoSalesCanvas' application. On the left, there's a sidebar with links: 'Overview' (which is highlighted with a red box), 'Quickstart', 'Integration assistant (preview)', 'Manage', and 'Branding'. The main panel shows the following details:

Display name	: ContosoSalesCanvas
Application (client) ID	: 03d95def-84cd-4e7d-aa
Directory (tenant) ID	: c5c89db8-aa3d-4a63-8a
Object ID	: bcaff820-a91b-48bb-81

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: '692e92c7-xxxx-4060-76d3-b381798f4d9c',
            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://1c0.0.environment.api.powerplatform.com/powervirtualagents
/bots/5a099fd/directline/token?api-version=2022-03-01-preview"
});
```

4. Edit the value of `userId` to include a custom prefix. For example:

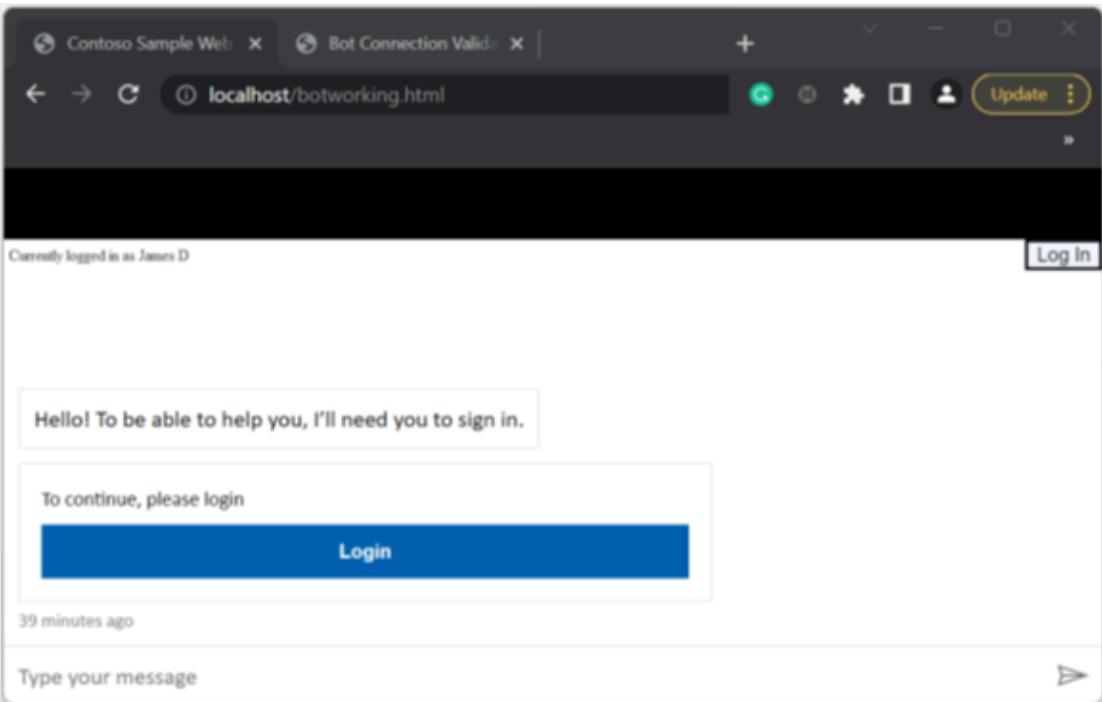
C#

```
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 copilot 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 copilot, and paste the validation code into the copilot conversation.

Related articles

- [Azure App Registration](#)

Configure single sign-on with Microsoft Entra ID for Copilot Studio copilots in Microsoft Teams

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.

The Copilot Studio app in Microsoft Teams supports single sign-on (SSO), which means copilots can automatically sign in users with their Microsoft Teams credentials.

SSO in the Microsoft Copilot Studio app in Teams is only supported when using Microsoft Entra ID. Other service providers, such as Azure AD v1, don't support SSO in the Microsoft Copilot Studio app in Microsoft Teams.

ⓘ Important

SSO is not supported when your copilot is integrated with [Dynamics 365 Customer Service](#).

ⓘ Note

If you are using Teams SSO authentication with the manual authentication option, and also using the copilot on custom websites at the same time, you must deploy the Teams app using the app manifest.

For more information, see [Download the copilot's Teams app manifest](#).

Other configurations such as authentications 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.](#)
- [Add your copilot to Microsoft Teams.](#)

Configure an app registration

Before configuring SSO for Teams, you'll first need to configure user authentication with Microsoft Entra ID. During this process, you'll create an app registration which you'll use to setup 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.

Follow the instructions in [Configure user authentication with Microsoft Entra ID](#) then return to this article.

Locate your Microsoft Teams channel app ID

1. In Microsoft Copilot Studio, open the copilot that you want to configure SSO for.
2. Under the settings for the copilot, select **Channels**. Select the **Microsoft Teams** tile.
3. Select **Edit details**, expand **More**, and then copy the **App ID** by selecting **Copy**.

The screenshot shows the 'Edit details' blade in the Power Virtual Agents interface. On the left, there's a sidebar with 'Manage' selected, and under 'Channels', 'Microsoft Teams' is chosen. The main area shows configuration for the Microsoft Teams channel, including fields for MPN ID (0000000), Teams channel SSO, AAD application's client ID (00000000-0000-0000-000000000000), Resource URI, and App ID. A red box highlights the 'Copy' button next to the App ID field. At the bottom are 'Save' and 'Cancel' buttons.

ⓘ Note

If you haven't already turned on the Microsoft Teams channel, you'll be notified that it's currently turned off. Select **Turn on Teams** and then re-try step 1 to get the app ID.

The screenshot shows a modal dialog titled 'Microsoft Teams'. It contains instructions: 'Nothing can stop a team, and now your bot can help you achieve more together. To open the lines of communication, select **Turn on Teams**. After a quick installation, your users and bot can start chatting.' Below this, a note says: 'Note that certain bot content may not appear the same on Microsoft Teams as it was authored in Power Virtual Agents. For details, refer to our article on [supported channel content](#)'. At the bottom are 'Turn on Teams' and 'Cancel' buttons, with 'Turn on Teams' highlighted by a red box.

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 copilot.
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. At the top right, there is a 'Set' button for the 'Application ID URI' which is highlighted with a red box. Below it, there is a section titled 'Scopes defined by this API'.

3. Enter `api://botid-{teamsbotid}` and replace `{teamsbotid}` with your Teams channel app ID that you found earlier.

The screenshot shows a modal dialog titled 'Set the App ID URI'. It contains a single input field labeled 'Application ID URI' with the value 'api://botid-'. This input field is also highlighted with a red box. At the bottom of the dialog are two buttons: 'Save' and 'Discard'.

4. Select Save.

Grant admin consent

Applications are authorized to call APIs when they are 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](#)

The admin consent option may be greyed out if it's not enabled by your tenant administrator. However, if it's available, you'll need to 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 **Yes**.

The screenshot shows the Azure portal's 'App registrations' section. On the left, a sidebar lists options like Overview, Quickstart, Integration assistant, and Manage (with sub-options: Branding & properties, Authentication, Certificates & secrets, and Token configuration). The main area is titled 'API permissions' and contains a search bar, refresh button, and feedback link. A warning message states: 'Starting November 9th, 2020 end users will no longer be able to grant consent to the publisher'. Below this, a note says: 'The "Admin consent required" column shows the default value for an organization. It may not reflect the value in your organization, or in organizations where this app will be used'. A section titled 'Configured permissions' includes a 'Grant admin consent for' button, which is highlighted with a red box.

Tip

To avoid users from having to consent to each application, a Global Administrator, Application Administrator, or a Cloud Application Administrator 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 will appear below. Expand **OpenId permissions** and turn on **openid** and **profile**.
4. Select **Add permissions**.

The screenshot shows the Azure portal interface for managing API permissions. On the left, there's a sidebar with navigation links like Overview, Quickstart, Integration assistant, and Manage (which is expanded). Under Manage, the 'API permissions' link is selected. The main area is titled 'Request API permissions' and shows a list of 'OpenId permissions'. Two permissions are listed with checkboxes: 'email' and 'offline_access', both of which have red boxes drawn around them. Below this list is a blue button labeled 'Add permissions' and a white button labeled 'Discard', with a red box drawn around the 'Add permissions' button.

Home > [REDACTED] | API perm

Search

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

Request API permissions

OpenId permissions (2)

email ⓘ View users' email address

offline_access ⓘ Maintain access to data you have given it access to

openid ⓘ Sign users in

profile ⓘ View users' basic profile

> AccessReview

Add permissions Discard

Define a custom scope for your copilot

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. At the top right, there is a search bar labeled 'Search (Ctrl+ /)' and a link 'Got feedback?'. Below the sidebar, the main area has a heading 'Application ID URI' with the value 'api://botid-[REDACTED]'. To the right of this, under the heading 'Scopes defined by this API', it says 'Define custom scopes to restrict access to c' and 'API can request that a user or admin conser'. Below this, it says '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 Admins and users
Admin consent display name	Enter <code>Test.Read</code>
Admin consent description	Enter <code>Allows the app to log in the user.</code>
State	Select Enabled

! 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, a button labeled '+ Add a client application' 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 steps above, 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.

Client Id	Scopes
1fec8e78-bce4-4aaf-ab1b-5451cc387264	1
5e3ce6c0-2b1f-4285-8d4b-75ee78787346	1

Add token exchange URL

To update the Microsoft Entra ID authentication settings in Copilot Studio, you'll need to 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 **Copy to clipboard**.

Home > App registrations > [REDACTED] | Expose an API

Search Got feedback?

Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protected by this API. An application can request that a user or admin consent to one or more of these.

Adding a scope here creates only delegated permissions. If you are looking to create application permissions, see [here](#). These are assignable to application type. [Go to App roles](#).

+ Add a scope

Scopes

Copy to clipboard

api://... /asdsas Admins only

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.

Power Virtual Agents | Authentication

Client ID *

Client secret *

Token exchange URL (required for SSO) [Learn more about SSO](#)

api://botid-/Test.Read

Tenant ID

Scopes ⓘ

profile openid

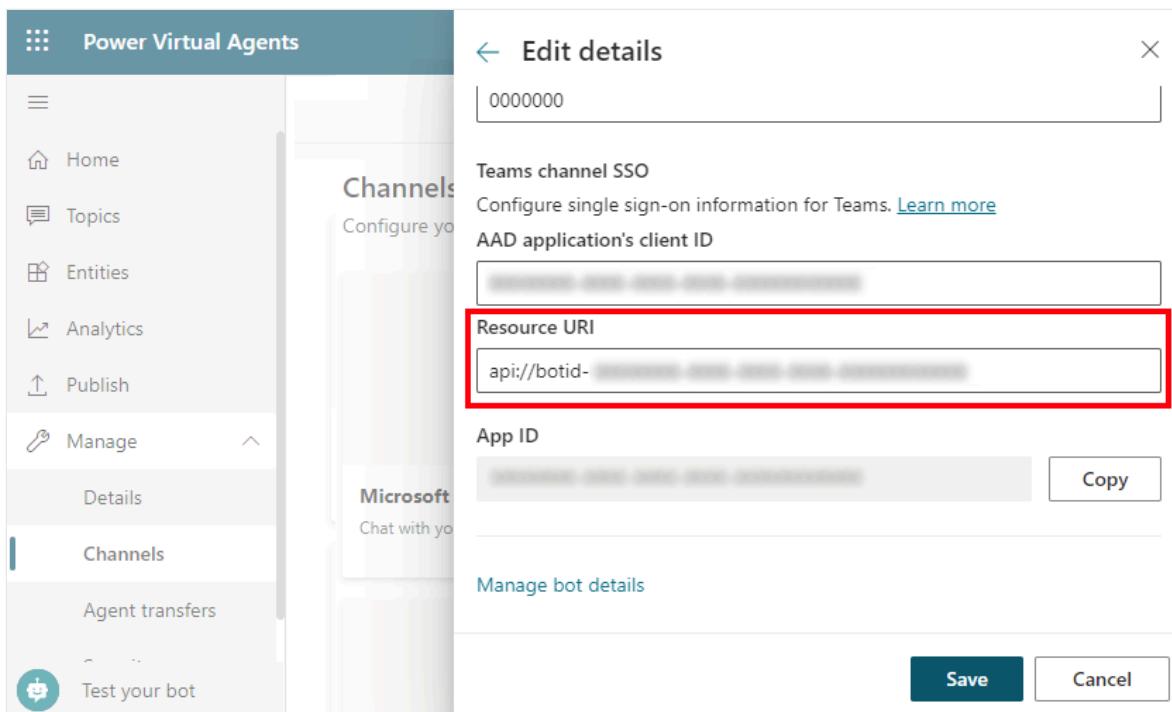
Save Close

5. Select **Save**.

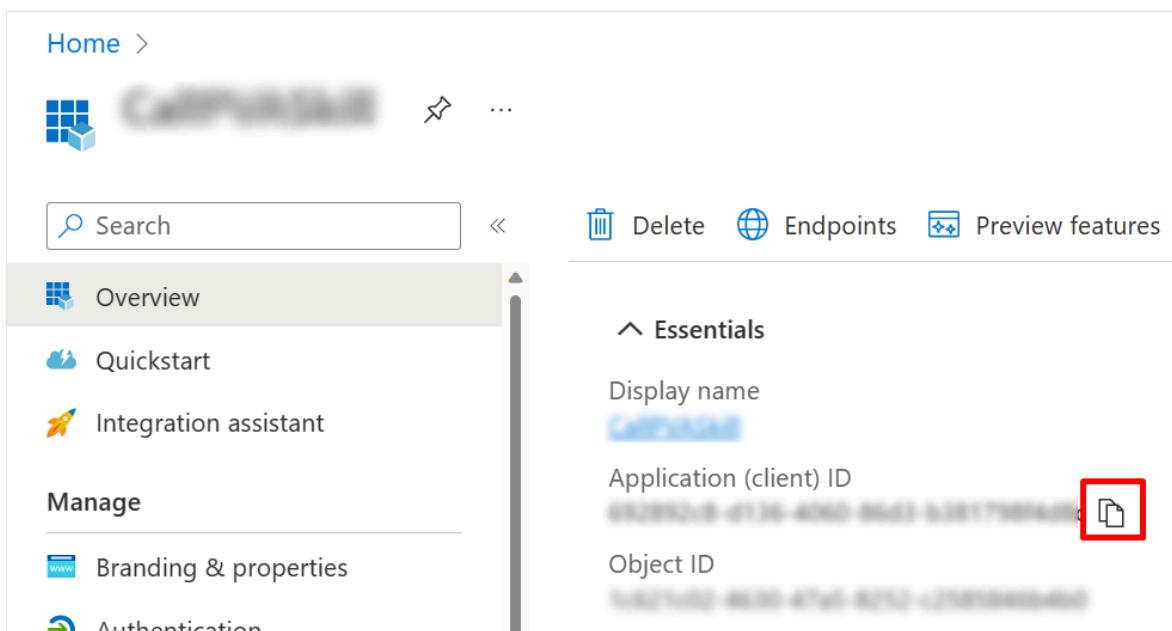
Add SSO to your copilot's Microsoft Teams channel

1. In Copilot Studio, under the settings for the copilot, select **Channels**.
2. Select the **Microsoft Teams** tile. Select **Edit details** and expand **More**.

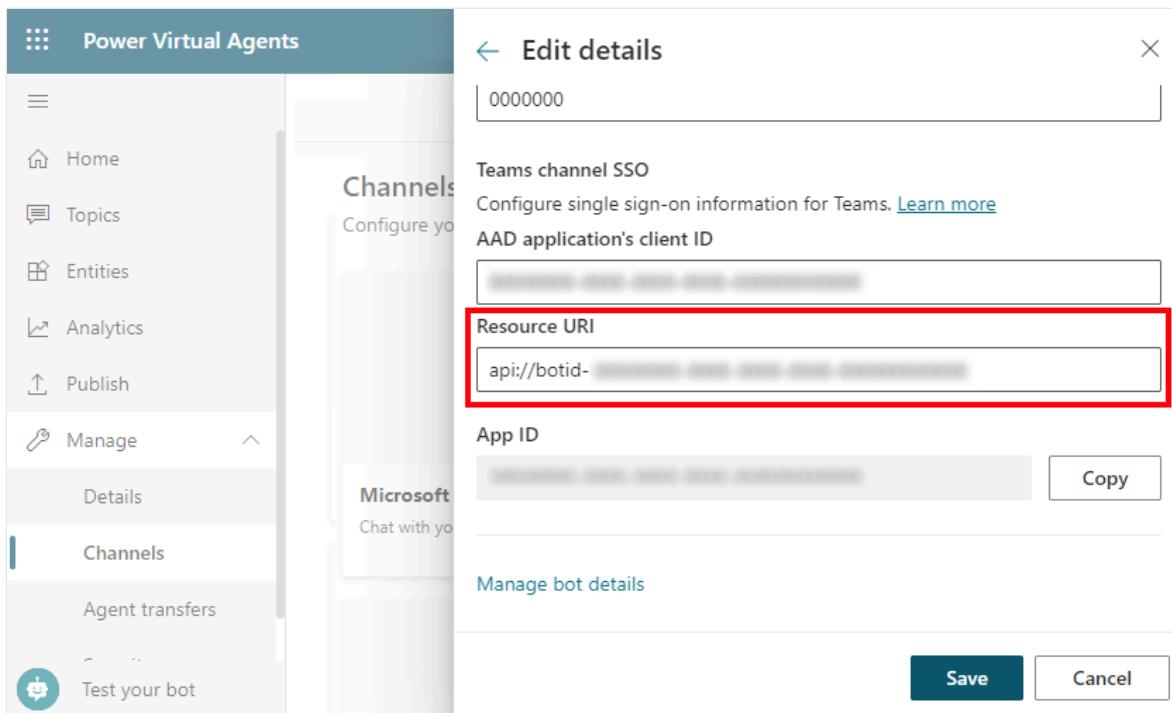
3. For AAD application's client ID, enter the Application (client) ID from your app registration.



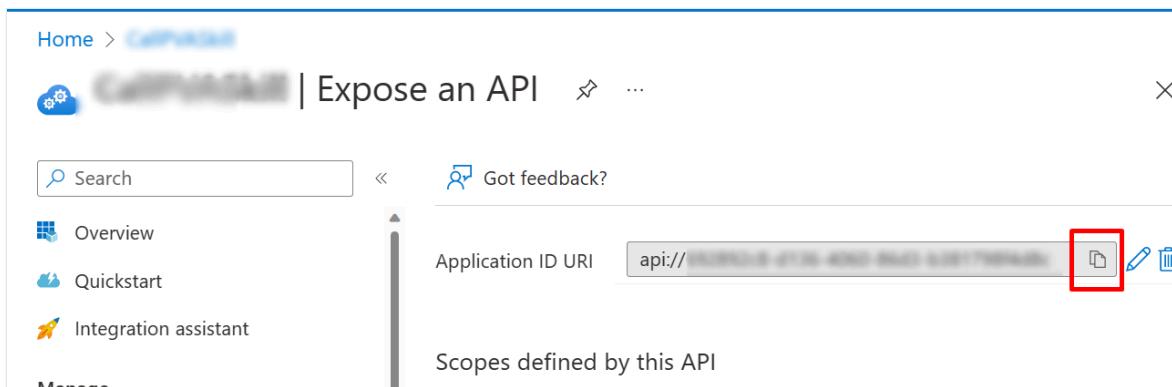
To obtain this value, open the Azure portal. Then on your app registration blade, go to **Overview**. Copy the value in the **Application (client) ID** box.



4. 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.



5. Select **Save**, and then **Close**.
6. Under the settings for the copilot, select **Publish**.
7. Select **Publish** to make the latest copilot content available to your customers.
8. In the navigation menu, under **Settings**, select **Channels**.
9. Select **Microsoft Teams**, then select **Open copilot**.
10. Start a new conversation with your copilot in Microsoft Teams to test if it automatically signs you in.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Configuring end user authentication

Article • 05/24/2024

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.

When creating a [Copilot Studio action](#) for an [authenticated Copilot Studio project](#), you can enable end-user authentication, or supply a set of credentials for the copilot to use on behalf of the user.

- Select **Copilot 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 copilot is meant to retrieve data that only the end 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 copilot](#).

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 copilot allows your end users to "retry" the action. It then completes with the end user's data access.

About data access and permission management

Copilot Studio does not store any credentials and reprompts end 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 copilot.

Supported channels

The following table details the [channels](#) that currently support end 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 ¹	Supported
Mobile App	Not supported
Omnichannel for Customer Service ²	Supported

¹ If you also have the Teams channel enabled, you need to follow the configuration instructions on the [Configure SSO for Teams channel](#) 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.

² 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 ↗

Automatically start a conversation

Article • 04/30/2024

ⓘ 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.

You can configure your copilot to start a conversation conversation with a user. You can also combine the customized greeting with [customization to the look and feel of the bot](#).

ⓘ Important

Having the copilot start the conversation will show up in your [analytics](#) and will increase your session count.

If the user of your copilot doesn't engage with the copilot (for example, they load the page but don't ask the copilot anything), the session will be [marked as an unengaged session](#). This might impact your analytics.

Web app

By default, copilots created with Microsoft 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 copilot start the conversation automatically when the copilot loads. For example, you could have your copilot say, "Hi, I'm Botty, a virtual agent" as soon as the copilot 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

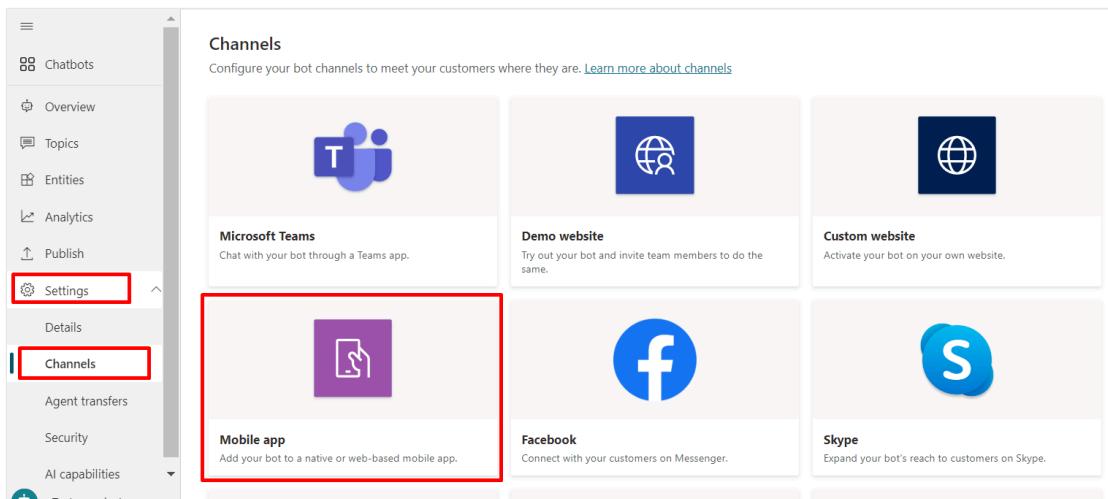
You may install and use the sample code included in this documentation only for use with the Microsoft Copilot Studio product. The sample code is licensed "as is" and is excluded from any service level agreements or support services. You bear the risk of using it.

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 copilot details.

1. In the navigation menu under **Settings**, select **Channels**.
2. Select **Mobile app**.



3. Next to **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

https://[REDACTED].com 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 a bot](#).

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')));
    </script>

```

```

        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 copilot in the custom canvas.
5. Test the copilot to ensure you are receiving responses from your copilot and that the copilot is working correctly.

If you encounter problems, make sure you've published your copilot, 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 copilot's default greeting

The code in the `index.html` file causes a topic to be called automatically when the copilot 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 a copilot (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 Microsoft Bot Framework's Topics management interface. On the left is a navigation sidebar with options like Chatbots, Overview, and Topics (which is highlighted with a red box). The main area is titled 'Topics' and shows a list of existing topics. The 'Greeting' topic is highlighted with a red box. The table below lists the topics:

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 copilot](#).

You can now test your copilot by going to the webpage where you deployed your copilot'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 a copilot and represents one unit of consumption. The billed session begins when a user topic is triggered. For more information, see the topic [Analyze billed session information](#).

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 copilot 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 copilot](#).

You can now test your copilot by going to the webpage where you deployed your copilot's custom canvas. You can see the copilot start the conversation by automatically showing the new topic.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Customize the look and feel of a copilot

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.

Your copilot's look and feel is defined by its canvas. To customize your copilot, you must customize its canvas. You can customize the canvas in two ways, depending on the complexity of the customizations:

- [Customize the default canvas](#) with JavaScript styling in the HTML code of the website where you deploy your copilot.
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 Microsoft 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 a copilot](#), your customers can [use the copilot's Web Chat canvas to interact with it](#).

You can also combine the customized canvas with [configuring your copilot to automatically start the conversation](#).

Lastly, you can [change the name and icon](#) of your copilot (when it's [shared in Microsoft Teams](#)) directly from the portal.

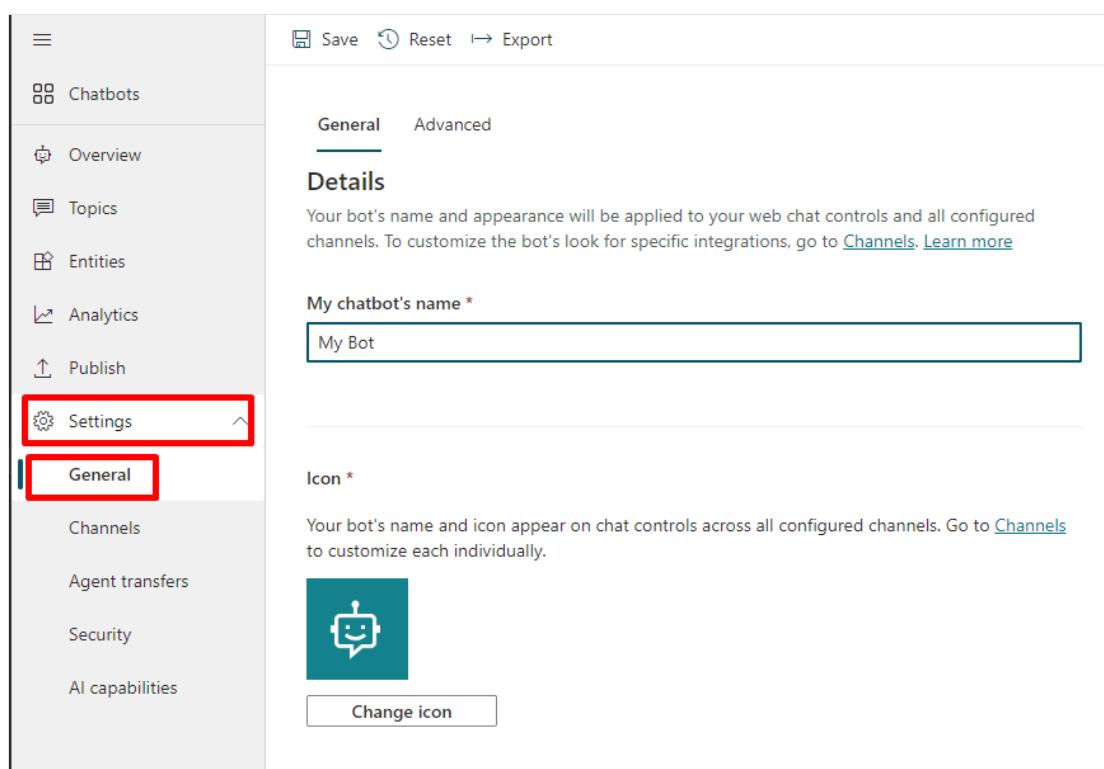
Change the copilot name and icon

ⓘ Important

If your copilot 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 copilot's name and icon. This will affect the icon in all channels where you publish your copilot.

1. In the navigation menu, under **Settings**, select **Details**.
2. Change the copilot's name and icon. [Review the recommendations on Microsoft Teams icon formats](#)
3. Select **Save** to commit your changes.



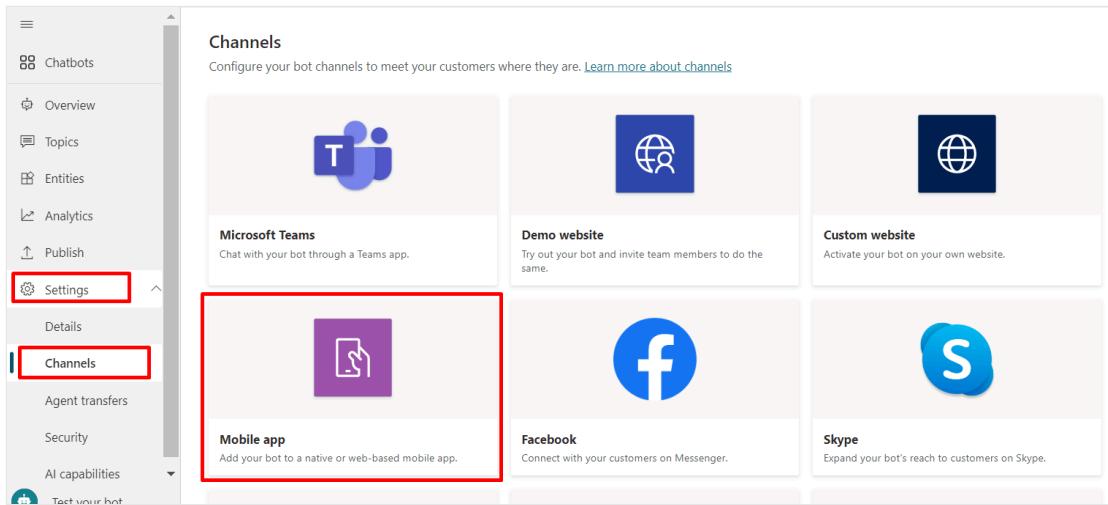
ⓘ Important

After updating your copilot's icon, it may take up to 24 hours for the new icon to appear everywhere.

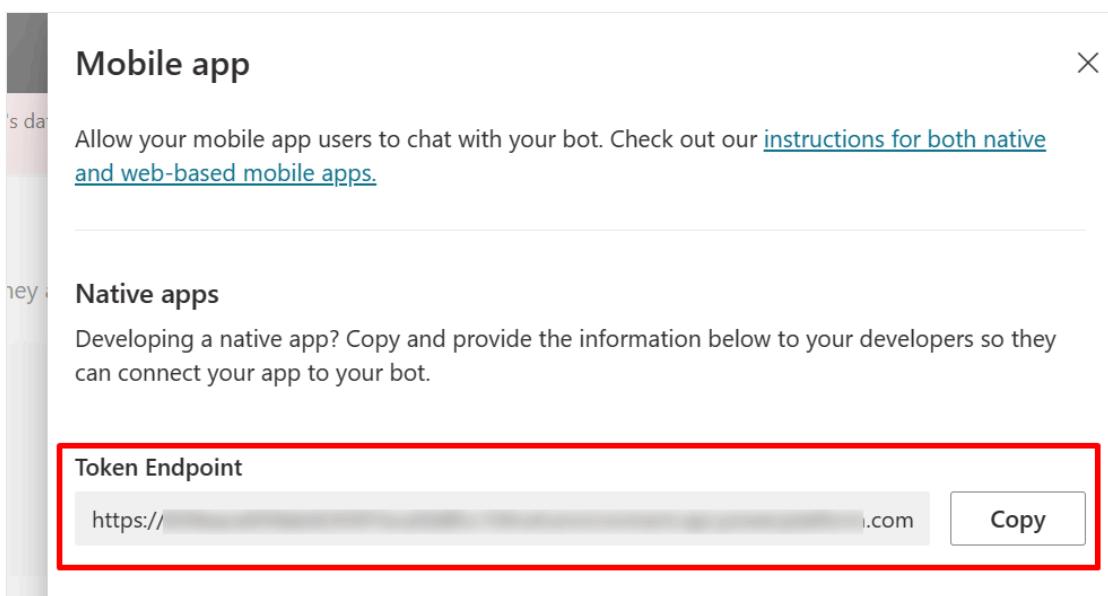
Retrieve token endpoint

To customize your canvas, whether it's the default canvas or a custom one you connect to, you need to retrieve your copilot 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 copilot canvas.

1. [Create and publish a copilot](#).

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.

HTML

```
<!doctype html>
<html lang="en">
  <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.

      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;
      }
    </style>
  </head>
  <body>
    <div id="banner"></div>
    <div id="webchat"></div>
  </body>
</html>
```

```
        width: 100%;  
    }  
  </style>  
</head>  
<body>  
  <div>  
    <div id="banner">  
      <h1>Contoso copilot 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('<COPILOT TOKEN  
ENDPOINT>');  
  
      // Specifies the language the copilot 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.  
    })();  
  </script>
```

```

        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/regionalsearchchannelsettings?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 copilot in the custom canvas.
5. Test the copilot to ensure you are receiving responses from it and that it's working correctly.

If you encounter problems, make sure you've published your copilot, 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 copilot icon, background color, and name

Once you get the customized canvas working with your copilot, you can make changes to it.

You can use the JavaScript `styleOptions` options to configure a number of pre-defined 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 copilot icon

1. Update the *index.html* file with the following sample code:

JavaScript

```
const styleOptions = {
  accent: '#00809d',
  botAvatarBackgroundColor: '#FFFFFF',
  botAvatarImage: 'https://learn.microsoft.com/azure/bot-
service/v4sdk/media/logo_bot.svg',
  botAvatarInitials: 'BT',
  userAvatarImage: 'https://avatars.githubusercontent.com/u/661465'
};
```

2. Replace the copilot and user avatar images with your company images.

If you don't have an image URL, you can use a Base64-encoded image string instead.

Change the background color

1. Update the *index.html* file with following sample code:

JavaScript

```
const styleOptions = {
  backgroundColor: 'lightgray'
};
```

2. Change `backgroundColor` to any color you wish. You can use standard CSS color names, RGB values, or HEX.

Change the copilot name

1. Update the `<h1>` text in the *index.html* file with the following:

HTML

```
<body>
<div id="banner">
  <h1> Contoso copilot
```

```
name</h1>
</div>
```

2. Change the text to whatever you want to call the copilot. 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 copilot 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:

card flight tracking

22 minutes ago

Showing flight, tracking

 Flight Status
DELAYED

Passengers	Seat
Sarah Hum	14A
Jeremy Goldberg	14B
Evan Litvak	14C

Flight	Departs	Arrives
KL0605	10:10 ...	12:00 AM

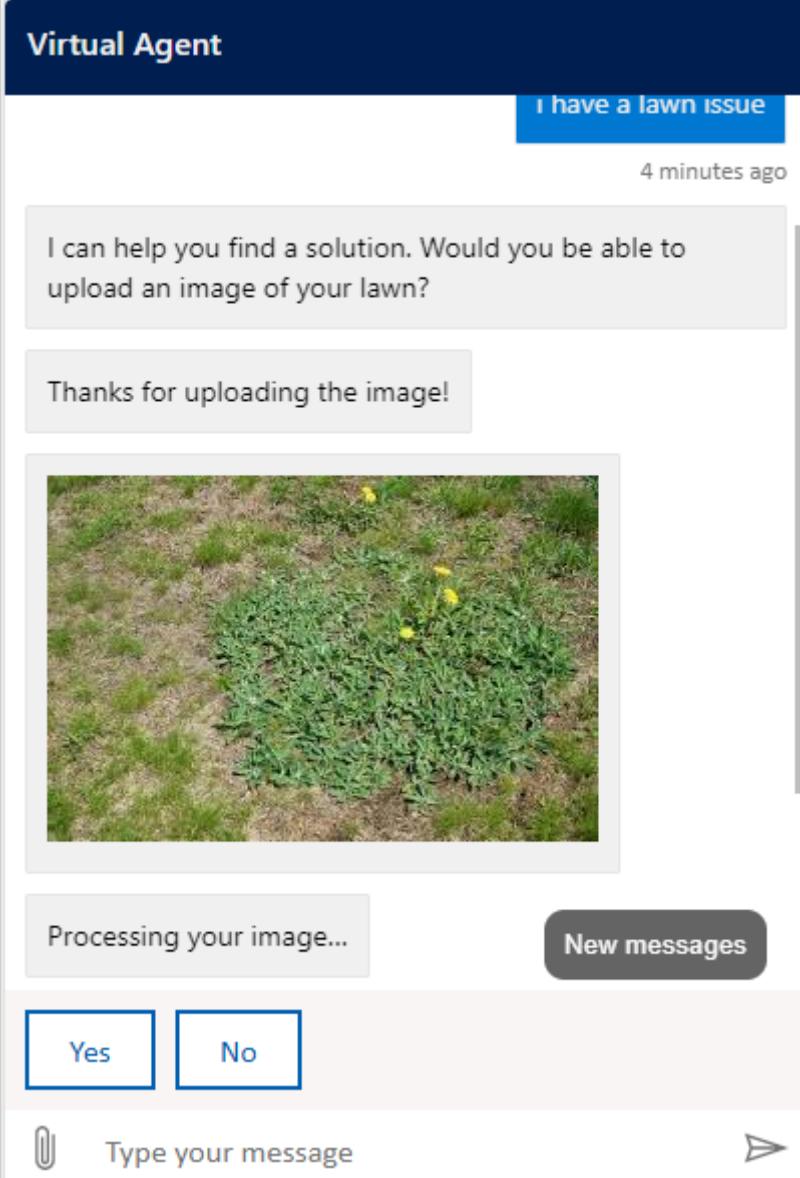
Amsterdam  San Francisco

AMS **SFO**

22 minutes ago

 Type your message 

- [Location and file uploading ↗](#) is a custom canvas capable of getting a user's location and sending it to a Copilot Studio copilot. 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 topics

- [Bot Framework Web Chat ↗](#)
 - [Generating a Direct Line token](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Personal data requests for Copilot Studio

Article • 03/26/2024

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 for the CDPR and CCPA](#) guide.

ⓘ Note

Because access to your copilot is managed by your Microsoft Entra ID tenant administrator, other users with admin permissions have access to your copilot content.

A note about 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

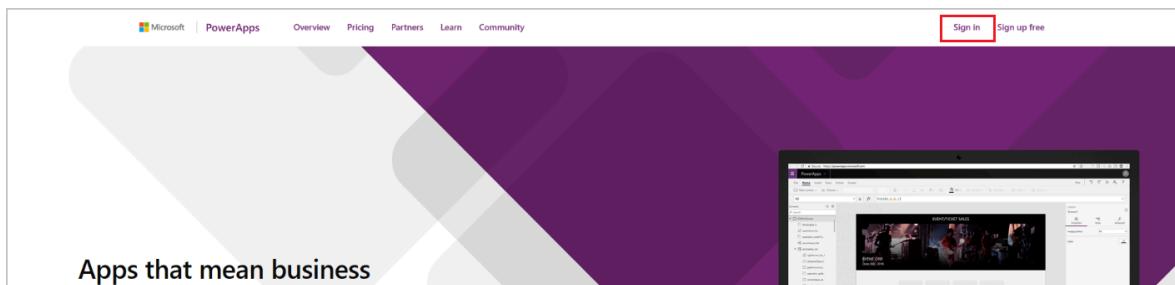
Article • 04/15/2024

This article discusses the Copilot Studio capabilities to find or export personal data for a specific user.

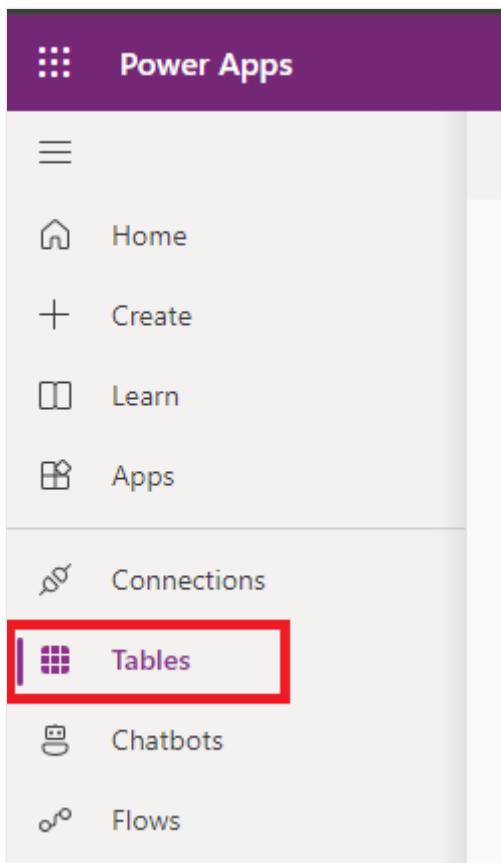
Export customer data

Copilot 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.

A screenshot of the Power Apps Tables page. The left sidebar shows "Tables" selected. The main area displays four buttons: "Describe the new table", "Upload an Excel file", "Start with a blank table", and "Create a virtual table". Below these are three buttons: "Recommended", "Custom", and "All". A table list follows, with the first two rows selected and highlighted with red boxes: "Chatbot" (Name: bot, Type: Standard, Managed: Yes, Customizable: Yes, Tags: Standard) and "Chatbot Approval BPF" (Name: new_chatbotapprovalbpf, Type: Standard, Managed: Yes, Customizable: Yes, Tags: Standard). The third row, "Chatbot subcomponent", is also partially selected.

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.

A screenshot of the Power Apps Table details page for the "Chatbot" table. The table has columns for Name and Type. The "Chatbot" row is selected. To the right of the table, there is a context menu with options: Open, Edit, Import, Export (which is highlighted with a red box), and Analyze.

5. When your export is ready, select Download exported data to download the content.

The screenshot shows the Microsoft Power Apps studio interface. In the top left, there's a navigation bar with 'Power Apps' and icons for Home, Create, Learn, and Apps. The main area has a 'Tables' section with a 'New table' button and a 'Put' icon. A green banner at the top right says 'Chatbot was exported successfully. Click the link below to download the exported data.' Below the banner are links for 'Export data' and 'Download exported data'. There's also a 'Describe the...' button.

Copilot sessions (as copilot 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 Azure User Privacy Overview page. At the top, there's a search bar and a navigation bar with 'Home > User privacy | Overview'. The main content area has a heading 'Welcome to Microsoft Cloud Services User Privacy Experience'. On the left, there's a sidebar with 'Overview' and 'Manage User Requests'. The main content includes a section titled 'Request to export user data' with a red box around it, which contains a 'Submit and manage user requests to export stored data' message and a 'Learn more about user privacy' link. Below this is a blue 'Add export request' button. To the right, there's a 'Delete user in Active Directory' section with a 'Delete users from the Azure environment by deleting the user in Active Directory' message and a 'Learn more about user privacy' link. At the bottom right, there's a 'Go to Azure AD to delete user' button.

Responding to personal data delete requests from data subjects

Article • 04/15/2024

This article discusses the experiences that Copilot Studio offers when deleting personal data for a specific user.

Delete Copilot Studio copilot chat logs

All copilot chat logs are deleted when the copilot is deleted. For more information, see [Delete a copilot](#).

Delete Copilot Studio copilot including all content

All copilot content is deleted when the copilot is deleted. For more information, see [Delete a copilot](#).

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 copilot. For more information, see [Delete a copilot](#).

Delete Copilot Studio system telemetry

All copilot 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.

Responding to personal data discovery requests from data subjects

Article • 04/15/2024

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 copilot settings.
User settings	Settings specific to the user of the copilot.
Conversation logs	Used to calculate copilot metrics and analytics.
Aggregated metrics	Used to determine copilot behavior.

Use interactive voice response in your copilots

Article • 05/31/2024

Copilot Studio supports interactive voice response 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 copilots 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 [Quickstart: Configure voice-enabled copilot with a phone number](#).

Key concepts for voice authoring

With the growing trend toward self-service applications, voice-enabled copilots are making a huge difference for businesses. Voice-enabled copilots are used in various applications, such as call centers, mobile apps, and messaging platforms.

Voice-enabled copilots can collect user input through speech and Dual-Tone Multi-Frequency (DTMF).

Supported voice features

After your copilot 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.

Feature	Description
Silence detection and timeouts	Detects when the user stops speaking, allowing the system to respond appropriately.
Speech recognition improvement	Speak naturally, without a script-a user's spoken command or question is translated for the voice-enabled copilot to process.
Speech Synthesis Markup Language (SSML)	Control how your copilot's voice sounds and behaves with end-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 copilot's voice sounds and behaves with end-users
 - [Barge-in](#)
 - [Latency message](#)
 - [Speech synthesis markup language \(SSML\)](#)
- Control the call flow by transferring calls or hanging up
 - [Transfer a call to an agent or external phone number](#)
 - [Hang up call at the end of a conversation](#)
- Authoring capabilities when building a voice-enabled copilot
 - [Build a voice-enabled copilot from a template](#)
 - [Voice authoring modality](#)
 - [Use voice variables](#)
 - [Test your voice-enabled copilot in chat](#)

Known limitations

These tips and limitations help you successfully integrate voice into your copilot.

[\[\] Expand table](#)

Feature	Tip or limitation
Channel order	Enable the Voice channel first and then connect with Omnichannel. The sequence is for channel reconnection.
Language/Locale	For a full list of supported languages and locales, see Supported languages for voice-enabled copilots . 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 dialpad	Pressing a key on the dialpad 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 copilots	Voice support is unavailable for secondary languages, and is only available for the copilot's primary language.
Customer engagement hub	<p>Apart from Omnichannel, all the other customer engagement channels only work with chat-based copilots. The following aren't supported for voice-enabled copilots:</p> <ul style="list-style-type: none"> - Genesys - Live person - Salesforce - ServiceNow
Generative AI for voice-enabled copilots	<ul style="list-style-type: none"> - Generative answers aren't supported for voice-enabled copilots. - Creating and editing topics using Copilot isn't supported for voice-enabled copilots. No messages will be created for Speech & DTMF, and DTMF aren't configured by Copilot. - Generative mode to orchestrate a copilot's topics and actions with generative AI isn't supported for voice-enabled copilots.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Supported languages for voice-enabled copilots

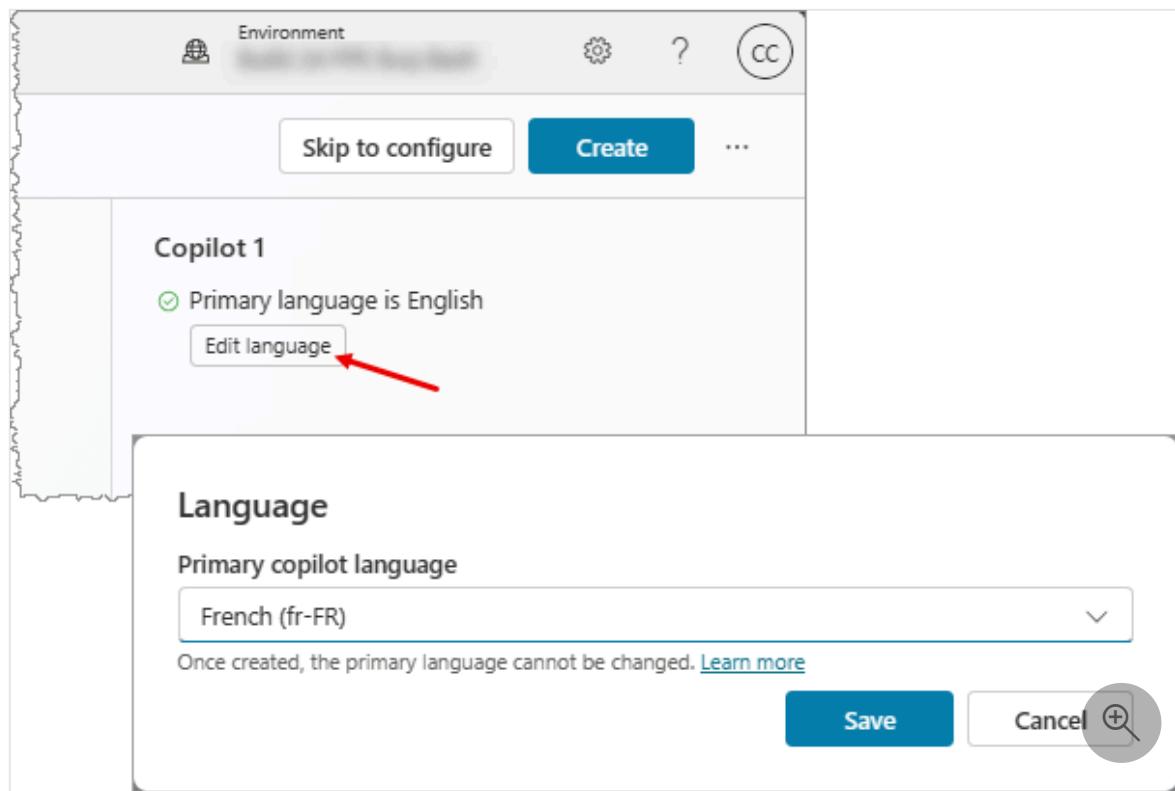
Article • 06/26/2024

You can create voice-enabled copilots in Copilot Studio in many languages. Your copilots can reach a broader audience and engage with more markets around the world.

Specify a language for your copilot

You specify the target language when you create a copilot in your [Copilot Studio canvas](#).

1. From the Home page, select **Create > New custom copilot**.
2. Select **Language**, then select a language from **Primary copilot language**, and then **Save**.



3. Select **Create** at the top of the page.

List of supported languages

[] Expand table

Language	Language and locale	Language code	Voice support
Chinese (Simplified)	Chinese-China-Mandarin (Simplified)	zh-CN	✓
Chinese (Traditional)	Chinese-Taiwan (Traditional)	zh-TW	✓
Czech	Czech-Czech Republic	cs-CZ	✓
Danish	Danish-Denmark	da-DK	✓
Dutch	Dutch-Netherlands	nl-NL	✓
English	English-Australia	en-AU	Preview
English	English-United Kingdom	en-GB	Preview
English	English-United States	en-US	✓
Finnish	Finnish-Finland	fi-FI	✓
French	French-Canada	fr-CA	Preview
French	French-France	fr-FR	✓
German	German-Germany	de-DE	✓
Greek	Greek-Greece	el-GR	✓
Hindi	Hindi-India	hi-IN	✓
Indonesian	Indonesian-Indonesia	id-ID	✓
Italian	Italian-Italy	it-IT	✓
Japanese	Japanese-Japan	ja-JP	✓
Korean	Korean-Korea	ko-KR	✓
Norwegian	Norwegian Bokmal-Norway	nb-NO	✓
Polish	Polish-Poland	pl-PL	✓
Portuguese	Portuguese-Brazil	pt-BR	✓
Russian	Russian-Russia	ru-RU	✓
Spanish	Spanish-Spain	es-ES	✓
Spanish	Spanish-United States	es-US	Preview
Swedish	Swedish-Sweden	sv-SE	✓
Thai	Thai-Thailand	th-TH	✓

Language	Language and locale	Language code	Voice support
Turkish	Turkish-Turkey	tr-TR	✓

Related content

- Chat-enabled copilot language support
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Voice-enabled copilot integration with Omnichannel for Customer Service

Article • 06/26/2024

To create and test the voice features and integration between Copilot Studio and Omnichannel for Customer Service applications:

1. Configure your Omnichannel for Customer Service app. For more information, see [Integrate a Copilot Studio bot](#).
2. Create a copilot in your new environment. For more information, see [Build a voice copilot from a template](#).
3. Add, configure, and edit the voice-related settings for the copilot. For more information about each feature and setting, see [Configure Voice capabilities](#).
4. Connect your copilot to Omnichannel for Customer Service. For more information, see [Configure hand-off to Dynamics 365 Customer Service](#).
5. Publish your copilot. For more information, see [Key concepts - Publish your copilot](#).
6. Configure voice functionality in Omnichannel for Customer Service.

Feedback

Was this page helpful?

 Yes

 No

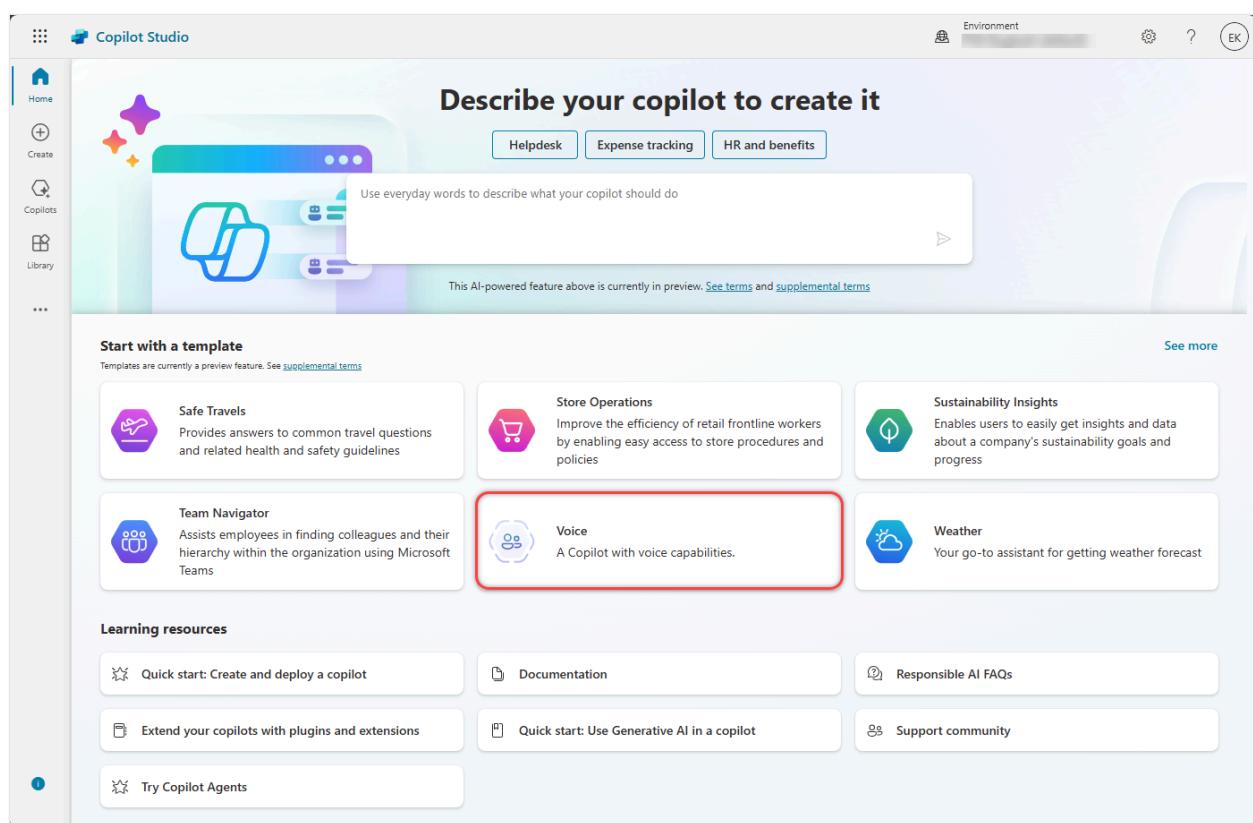
[Provide product feedback ↗](#)

Build a voice-enabled copilot from a template

Article • 05/31/2024

Building a voice-enabled copilot can provide an effective self-service, hands-free solution from a phone to reduce call volume and increase agent productivity. Customers can interact with a copilot by using copilot natural language and choosing the options(touch-tone) from a menu to resolve the issues faster.

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



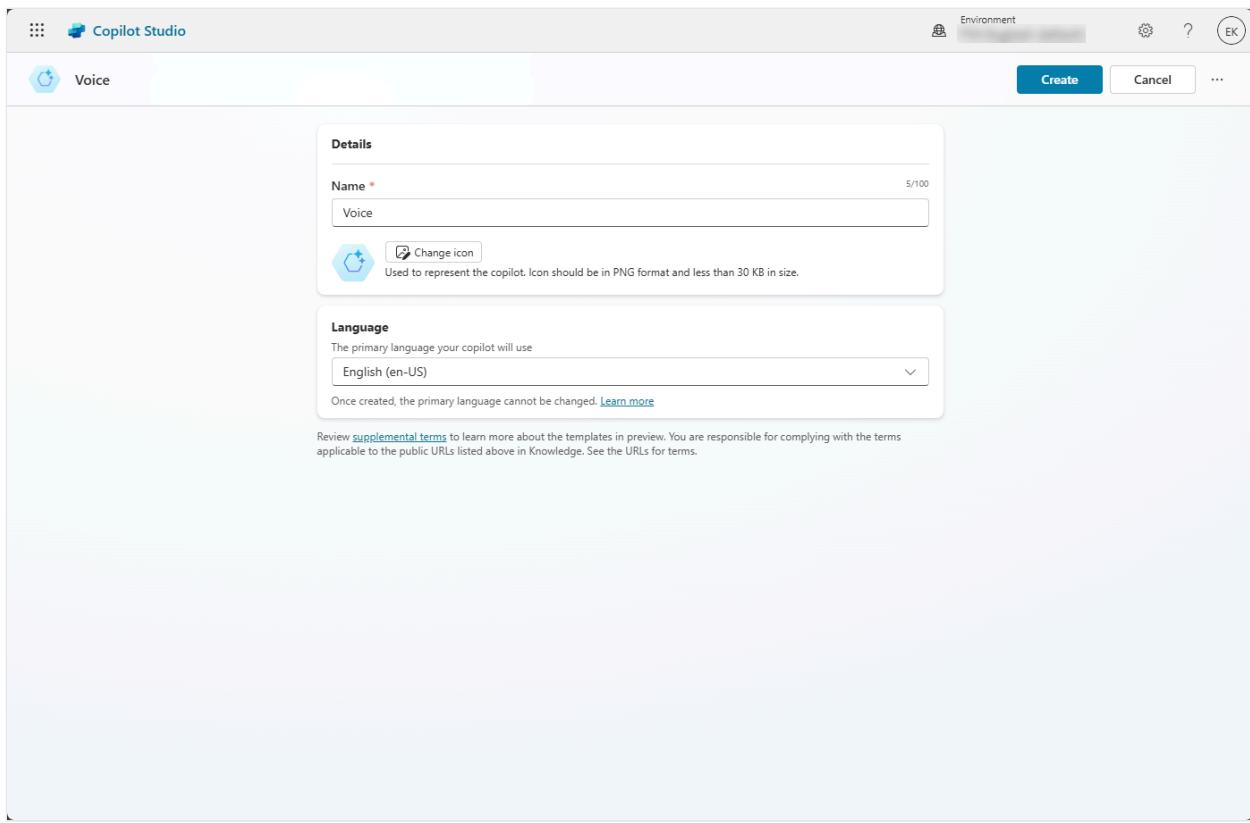
The screenshot shows the Microsoft Copilot Studio interface. At the top, there's a search bar with the placeholder "Describe your copilot to create it" and three categories: "Helpdesk", "Expense tracking", and "HR and benefits". Below this, a large input field says "Use everyday words to describe what your copilot should do". A note at the bottom of this field states: "This AI-powered feature above is currently in preview. See terms and supplemental terms".

In the "Start with a template" section, there are several cards:

- Safe Travels**: Provides answers to common travel questions and related health and safety guidelines.
- Store Operations**: Improve the efficiency of retail frontline workers by enabling easy access to store procedures and policies.
- Sustainability Insights**: Enables users to easily get insights and data about a company's sustainability goals and progress.
- Team Navigator**: Assists employees in finding colleagues and their hierarchy within the organization using Microsoft Teams.
- Voice**: A Copilot with voice capabilities. This card is highlighted with a red border.
- Weather**: Your go-to assistant for getting weather forecast.

Below this section is a "Learning resources" area with links to quick starts, documentation, responsible AI FAQs, and support community.

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



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

Voice-enabled copilot capabilities

When you select the voice-enabled copilot template option, the copilot is created from a template.

Voice features are included with the template:

- A *start conversation* topic (Main Menu) is added to your greeting, which highlights the voice menu with the DTMF global trigger mapped.

[+ Add a topic](#)[All](#)[Custom \(8\)](#)[System \(11\)](#)

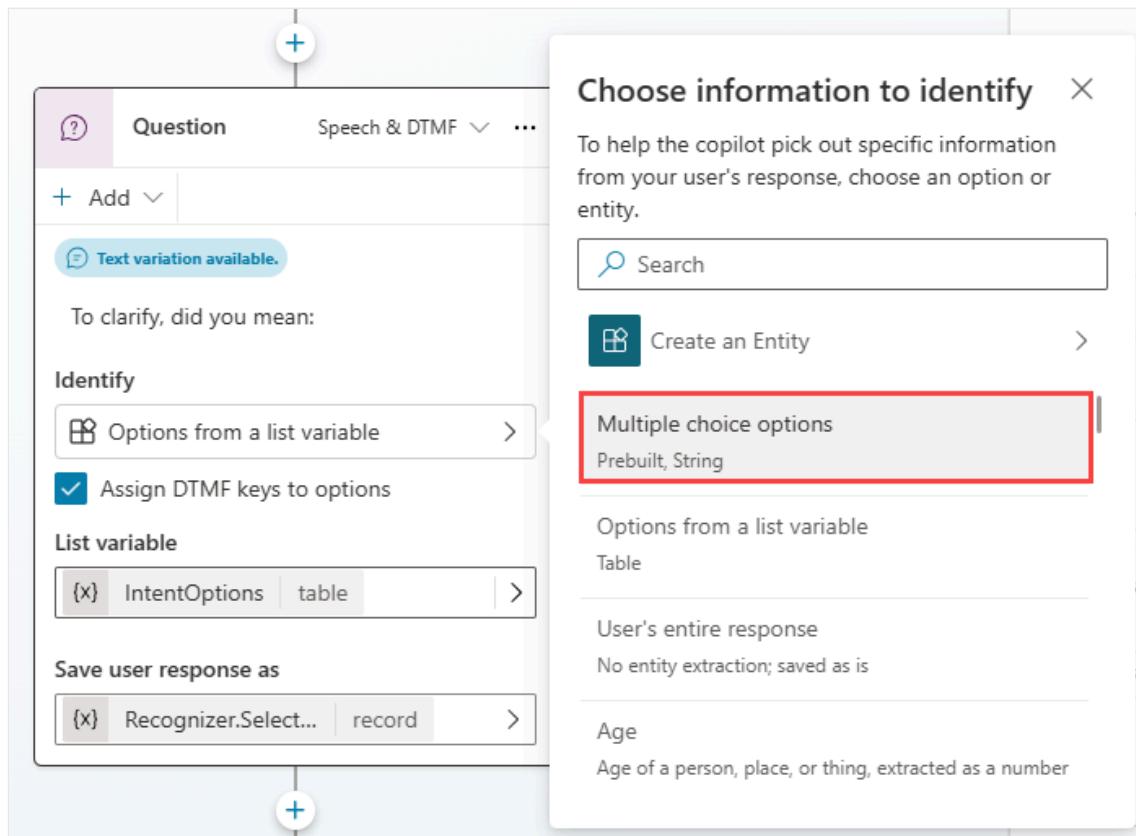
Name	Trigger	Description	Editing
Goodbye	Phrase	This topic triggers ...	
Greeting	Phrase	This topic is trigger...	
Lesson 1 - A simple t...	Phrase		
Lesson 2 - A simple t...	Phrase		
Lesson 3 - A topic wi...	Phrase		
Main Menu (preview)	DTMF received	This topic is the mai...	
Start Over	Phrase		
Thank you	Phrase	This topic triggers ...	

- Voice system topics include:
 - Silence detection (voice)
 - Speech unrecognized (voice)
 - Unknown dial pad press (voice) handles speech-related scenarios correctly

- DTMF-enabled multiple choice options in the question node

The screenshot shows the 'Topics' section of the Voice interface. At the top, there are navigation tabs: 'Voice' (with a microphone icon), 'Overview', 'Knowledge', 'Topics' (which is underlined, indicating it's the active tab), 'Actions', 'Analytics', and 'Channels'. Below the tabs is a button labeled '+ Add a topic'. Underneath are three filter buttons: 'All', 'Custom (8)', and 'System (11)' (which is also highlighted with a blue border). A note below the filters 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.' The main area is a table with columns: 'Name', 'Trigger', 'Description', and 'Editing'. The table lists the following system topics:

Name	Trigger	Description	Editing
Conversation Start	On Conversation Start	This system topic triggers when a conversation starts.	
End of Conversation	On Redirect	This system topic is triggered when a conversation ends.	
Escalate	On Talk to Agent	This system topic is triggered when a conversation is escalated.	
Fallback	On Unknown Input	This system topic triggers when a user provides an unknown input.	
Multiple Topics Matched	On Select Intent	This system topic triggers when multiple topics are matched.	
On Error	On Error	This system topic triggers when an error occurs.	
Reset Conversation	On Redirect		
Sign in	On Sign In	This system topic triggers when a user signs in.	
Silence detection	Silence detected	Triggered when a dial pad press is detected.	
Speech unrecognized	Unrecognized Input	Triggered when the speech is unrecognized.	
Unknown dial pad press	On Unknown Input	Triggered when a user presses an unknown dial pad key.	



- The voice authoring modality is enabled by default, and the copilot author can use voice authoring capabilities.

Copilot details

General Voice Advanced

Make sure your copilot works for you with voice-first features like advanced speech recognition and dual-tone multi-frequency (DTMF) input. Voice optimization is temporarily only supported in telephony channels. [Learn more](#)

Optimize for voice

Primary language
English (United States) (en-US) (Primary)

Secondary languages (0)
[Add and configure languages](#)

Use voice as primary authoring mode

Customized automatic speech recognition

Voice

DTMF

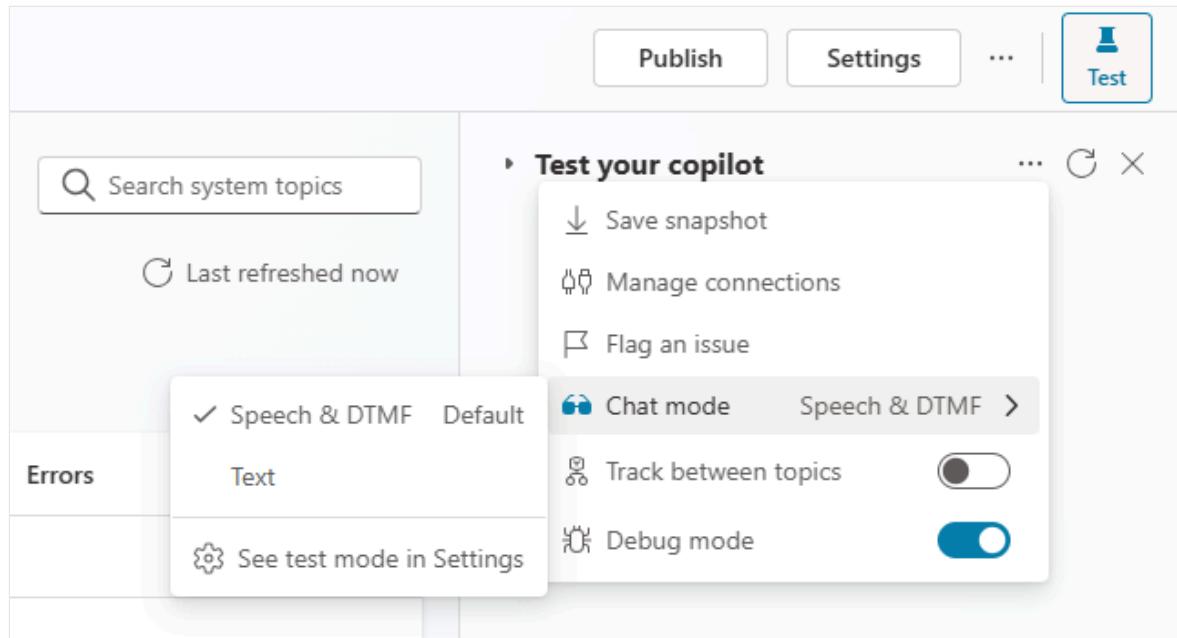
Silence detection

Speech collection

Latency messaging

- The **Test copilot** chat mode is automatically 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 copilot in text output.



- Voice channel is enabled by default.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Configure voice capabilities

Article • 06/12/2024

This article describes the features available in Copilot Studio for interactive voice response with Omnichannel for Customer Service.

To get your copilot ready for voice services, see [Quickstart: Configure voice-enabled copilot with a phone number](#).

For an overview of the voice services, see [Use interactive voice response in your copilots](#).

Voice authoring modality

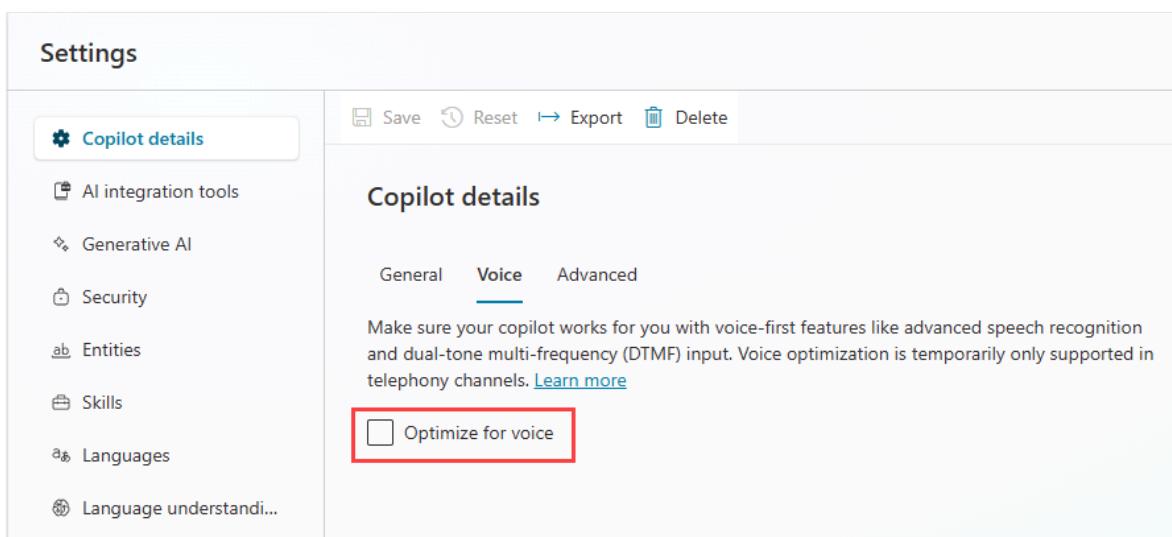
A voice-enabled copilot is different from a chat copilot. The voice-enabled copilot includes specific voice system topics for handling voice scenarios. A chat copilot uses the text modality as the default. A voice-enabled copilot uses the Speech & DTMF modality. The two modalities aren't compatible with each other.

Optimize for voice allows you to author voice-enabled copilots across different modalities and ensures speech-related features are authored correctly.

Optimize for voice

If you didn't [start your copilot with the Voice template](#), you must enable the **Optimize for voice** option in the copilot's **Settings**.

1. With a copilot open, go to **Settings** > **Copilot details**.
2. Go to the **Voice** tab and select **Optimize for voice**.



The **Use voice as primary authoring mode** option is also set by default.

Settings

Copilot details

General Voice Advanced

Make sure your copilot works for you with voice-first features like advanced speech recognition and dual-tone multi-frequency (DTMF) input. Voice optimization is temporarily only supported in telephony channels. [Learn more](#)

Optimize for voice

A¹ Primary language
English (United States) (en-US) (Primary)

> Secondary languages (0)
[Add and configure languages](#)

Use voice as primary authoring mode

> Customized automatic speech recognition

> Voice

> DTMF

> Silence detection

> Speech collection

> Latency messaging

Your copilot 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 copilot data* (on by default), which improves speech recognition accuracy.
- There's no change to the existing copilot 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 copilot.

- In addition, setting **Optimize for voice** on a copilot that wasn't originally configured for voice features means that the copilot 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 copilot authoring if you don't enable the voice channel. After you disable **Optimize for voice**, you get the following changes:

- No copilot 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 copilot flow and channel setting, since disabling optimization doesn't turn off the voice channel.
- Enabling or disabling the *optimize for voice* option doesn't take effect until you publish your copilot. If the copilot is enabled or disabled accidentally and switches between modalities, you have time to fix it.

ⓘ Important

If your voice channels are enabled, disabling **Optimize for voice** can break your copilot, 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 copilot 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	Copilot text & speech channel
Message available	Message empty	Message available
Message empty	Message available	Message not available

Enable barge-in

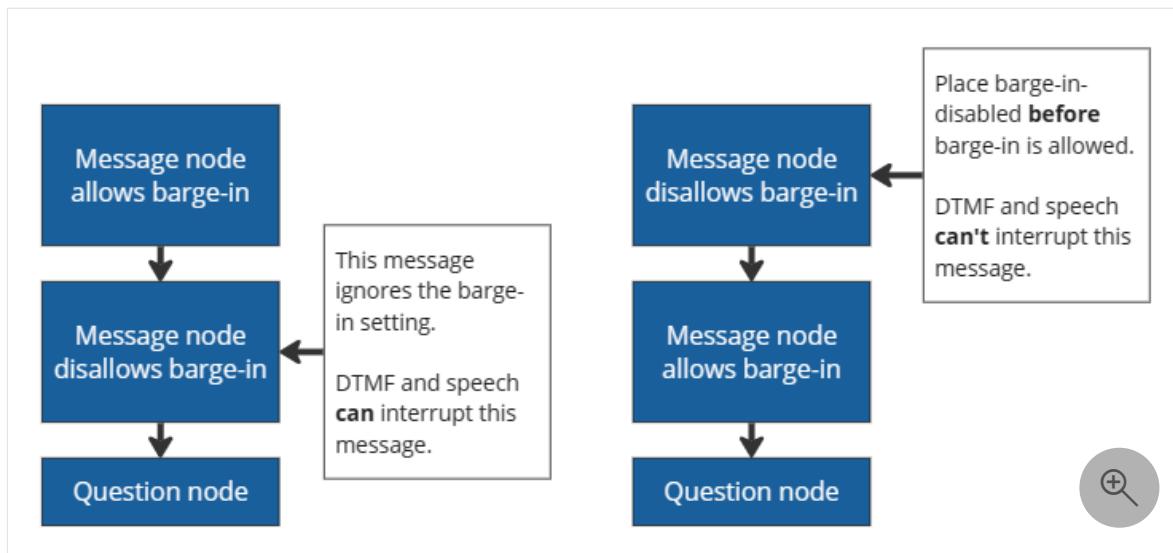
Enabling barge-in allows your copilot users to interrupt your copilot. This feature can be useful when you don't need the copilot user to hear the entire message. For example, callers might already know the menu options, because they heard them in the past. With barge-in, the copilot user can enter the option they want, even if the copilot isn't finished listing all the options.

Barge-in disable scenarios

- Disable barge-in if you recently updated a copilot message or if the compliance message shouldn't be interrupted.
- Disable barge-in for the first copilot message to ensure copilot users are aware of new or essential information.

Specifications

- Barge-in supports DTMF-based and voice-based interruptions from the copilot 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.

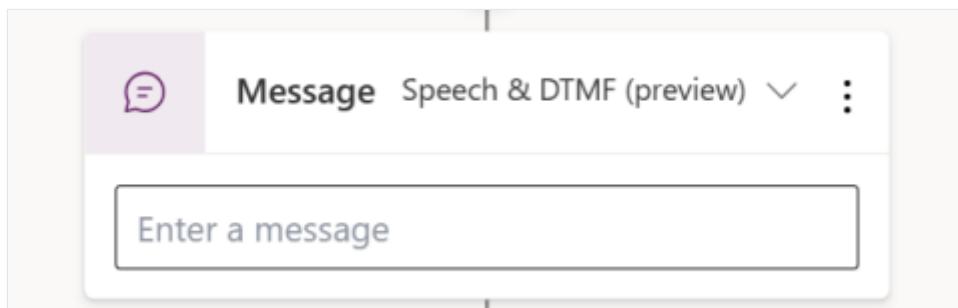
Tip

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 copilot users to be interacting with the copilot often. If your copilot user already knows the menu options, let them self-service where they want to go.

Set up barge-in

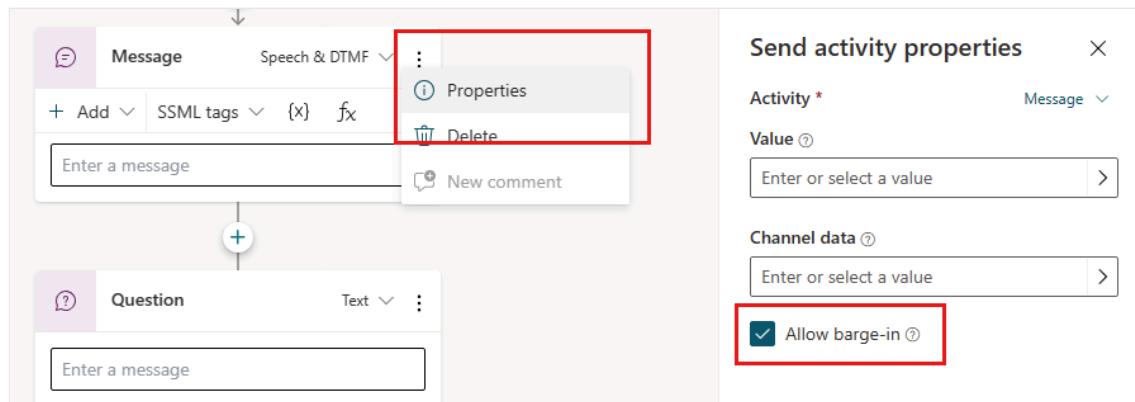
- With a **Message** or **Question** node selected, set the desired modality to **Speech & DTMF**.



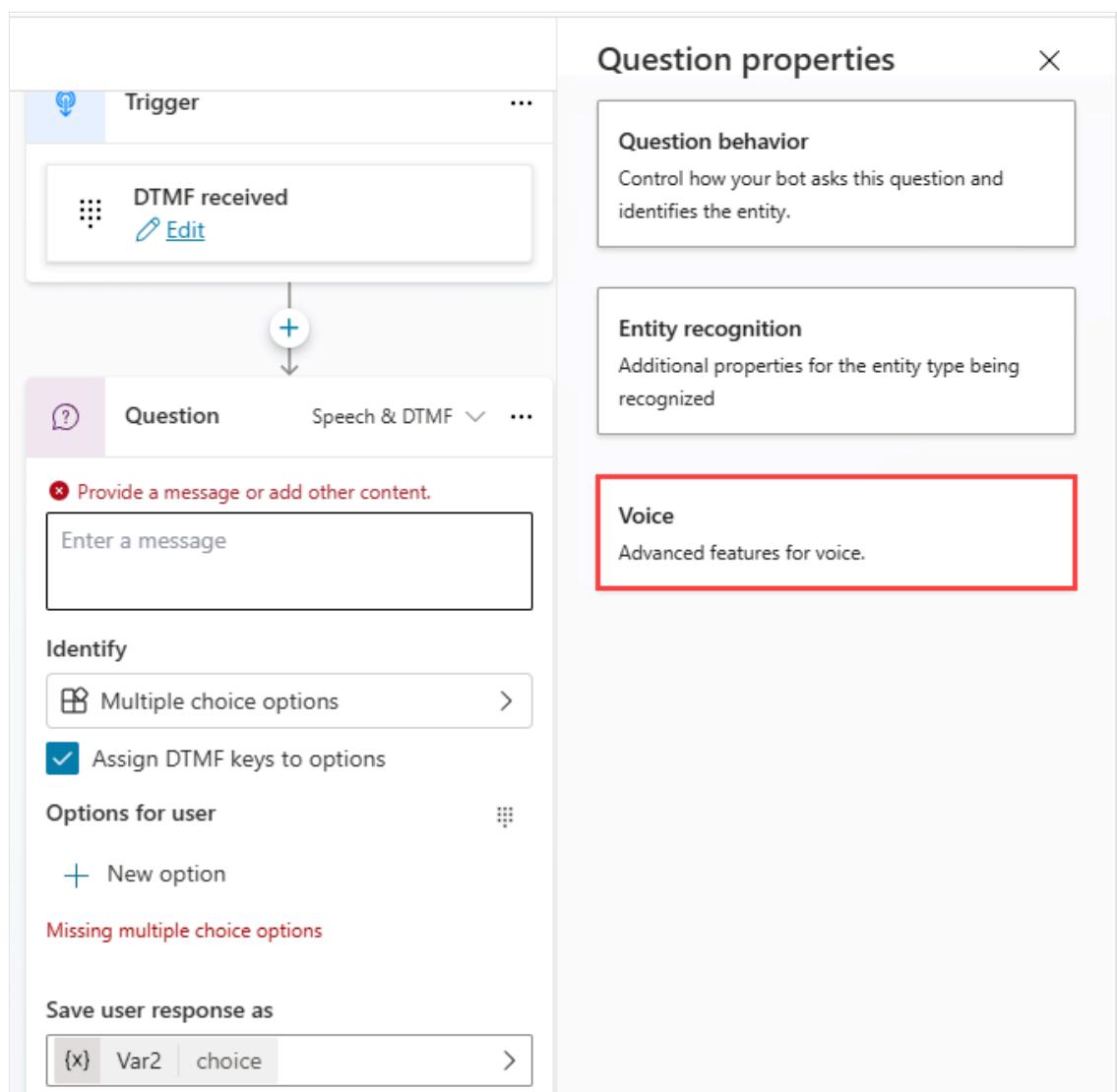
- Select the **More** icon (...) of the node, and then select **Properties**.

- 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**.

Barge in
Users can interrupt bot during audio play back
 Allow barge-in

Silence detection
Choose how your bot responds if it doesn't detect any input from your end user

Silence detection timeout ⓘ
 Use copilot setting (off)
 Disable for this node
 Customize in milliseconds

How many reprompts on silence ⓘ Manual input ▾
If the bot continues detects silence, it can ask the question again or send a new message.
Repeat up to 2 times

Reprompt message ⓘ
 Customize reprompt message

Fallback action ⓘ
Hang Up

> Advanced

3. Save the topic to commit your changes.

Configure silence detection and timeouts

Silence detection lets you configure how long the copilot 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 copilot 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:

Silence detection
Choose how your bot responds if it doesn't detect any input from your end user

Silence detection timeout ⓘ

Use copilot setting (off)

Disable for this node

Customize in milliseconds

[+] Expand table

Silence detection timeout option	Description
Use copilot setting	Node uses the global setting for silence detection.
Disable for this node	The copilot waits indefinitely for a response.
Customize in milliseconds	The copilot waits for a specified time before repeating the question.

Fallback action

You can configure some behaviors as a fallback action:

How many reprompts on silence [Manual input](#) ⓘ

If the bot continues detects silence, it can ask the question again or send a new message.

Repeat up to 2 times

Reprompt message ⓘ

Customize reprompt message

Fallback action ⓘ

Hang Up

- How many times the copilot should repeat a question

- What the reprompt message should say
- What the copilot should do after a specified number of repeats

Speech input

For speech input you can specify:

▼ Advanced

Utterance end timeout ⓘ

Use copilot setting (1500 ms)

Customize in milliseconds

Speech recognition timeout ⓘ

Use copilot setting (12000 ms)

Disable for this node

- **Utterance end timeout:** How long the copilot waits after the user finishes speaking
- **Speech recognition timeout:** How much time the copilot gives to the user once they start responding

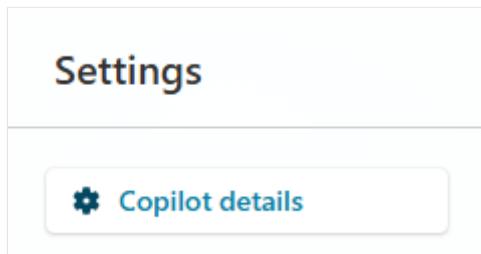
To configure silence detection behavior when your copilot waits for a trigger phrase, adjust the settings in the **On silence** system topic.

Copilot-level voice options reference

The **Copilot 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 copilot.

To make changes to the copilot-level timeout options:

1. With a copilot open, select **Settings > Copilot details**.



2. Go to the **Voice** tab, expand the DTMF settings and adjust the copilot's default timeout settings.

Copilot details

General **Voice** Advanced

Make sure your copilot works for you with voice-first features like advanced speech recognition and dual-tone multi-frequency (DTMF) input. Voice optimization is temporarily only supported in telephony channels. [Learn more](#)

Optimize for voice

Primary language
 English (United States) (en-US) (Primary)

Secondary languages (0)
 [Add and configure languages](#)

Use voice as primary authoring mode

Customized automatic speech recognition

Voice

DTMF

Choose how your copilot receives input from a dial pad. [Learn more](#)

 Interdigit timeout ⓘ
 ms

 Termination timeout ⓘ
 Continue without waiting
 Termination timeout
 ms



3. Select **Save** at the top of the page to commit your changes.

Copilot-level settings

The following table lists each option and how it relates to node-level settings.

 [Expand table](#)

Voice-enabled copilot-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 press termination key. Applies only to multi-digit DTMF input. After the limit times out and terminating DTMF key doesn't arrive, copilot ends the recognition and returns the result up to that point. If set to "continue without waiting," the copilot doesn't wait for termination key. Copilot returns immediately after user inputs the max length.	2000 ms	Question node with voice properties for Multi-digit DTMF input
Silence detection	Silence detection timeout	Maximum silence (milliseconds) allowed while waiting for user input. Limit applies when copilot doesn't detect any user input. The default is "no silence timeout." Copilot waits infinitely for user's input. Silence detection for voice times the period after the voice finishes speaking.	No silence timeout	Question node with voice properties for Multi-digit DTMF input System Topic (silence detection trigger properties) for Configure silence detection and timeouts
Speech collection	Utterance end timeout	Limit applies when user pauses during or after speech. If pause is longer than timeout limit, copilot presumes user finished speaking.	1500 ms	Question node with voice properties

Voice-enabled copilot-level section	Setting	Description	Default value	Node-level override
		The maximum value for utterance end timeout is 3000 milliseconds. Anything above 3000 ms reduces to 3000 milliseconds.		
Speech collection	Speech recognition timeout	Determines how much time the copilot 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. Copilot reprompts the question. If no response, the voice is beyond <i>Speech recognition timeout</i> .	12,000 ms	Question node with voice properties
Long-running operation alert	Send message delay	Determines how long the copilot 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
Long-running operation alert	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

Add a latency message for long running operations

For long backend operations, your copilot can send a message to users to notify them of the longer processes. Copilots 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.

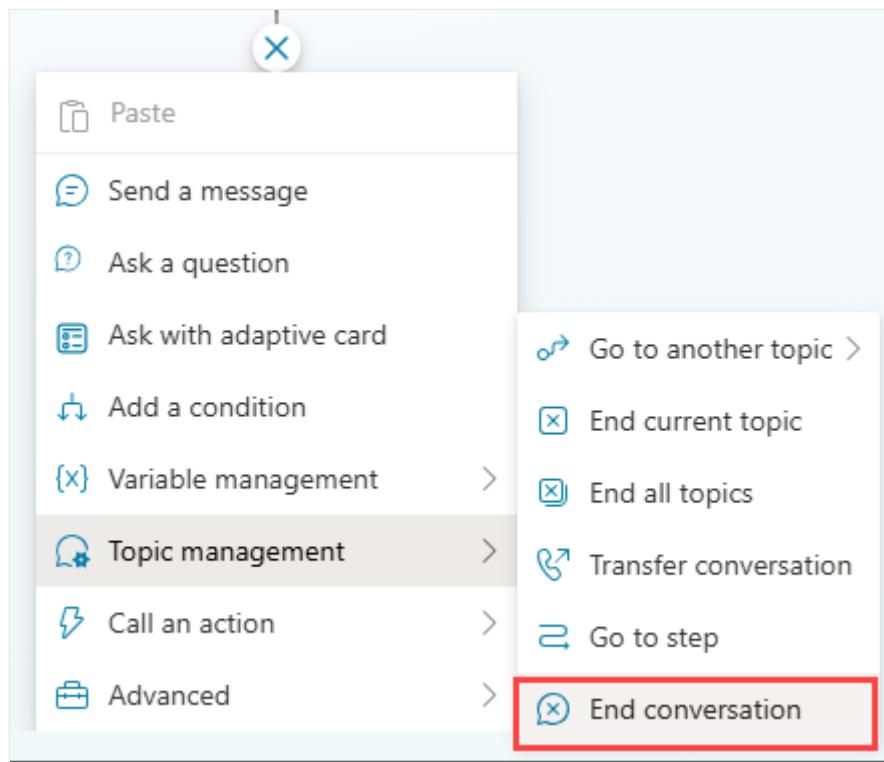
In Copilot Studio, your copilot can repeat a message after triggering a Power Automate flow:

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 copilot to say. You can use SSML to modify the sound of the message. The copilot repeats the message until the flow is complete.

You can adjust how long the copilot 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 copilot to end the call and hang up, add a new node (+) then select **Topic management > End conversation**.



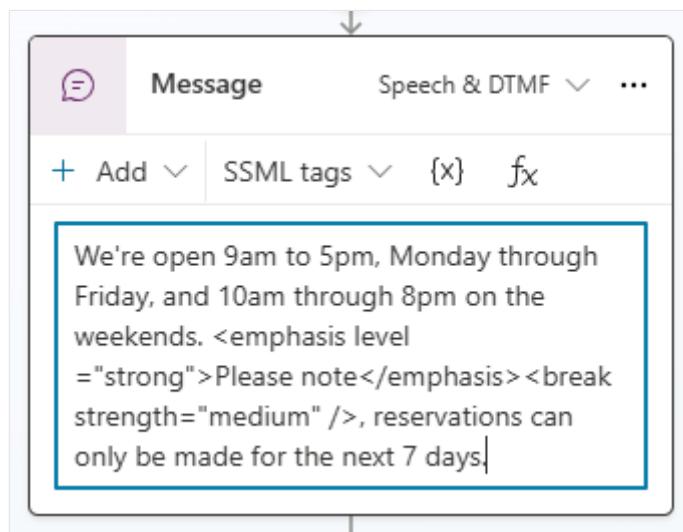
Format speech synthesis with SSML

You can use speech synthesis markup language (SSML) to change how the copilot 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 Power Virtual Agents:

[+] 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 copilot user.	Add recorded audio
<code><break /></code>	Insert pauses or breaks between words. Insert break options within the tag.	Add a break
<code><emphasis> Text you want to modify </emphasis></code>	Add levels of stress to words or phrases. Add emphasis options in the opening tag. Add the closing tag after the text you want to modify.	Adjust emphasis options
<code><prosody> Text you want to modify </prosody></code>	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



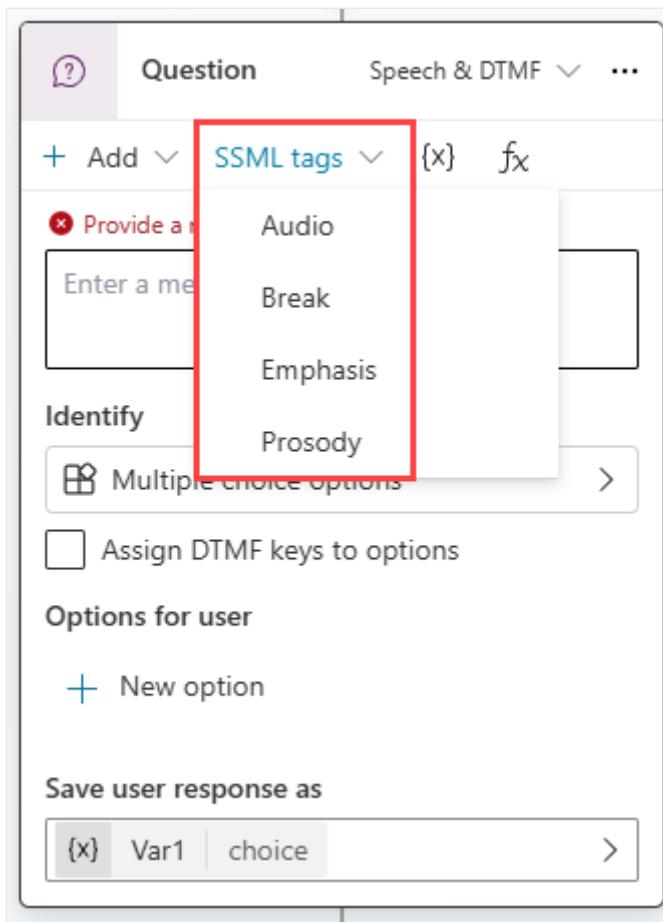
Find and use a tag

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

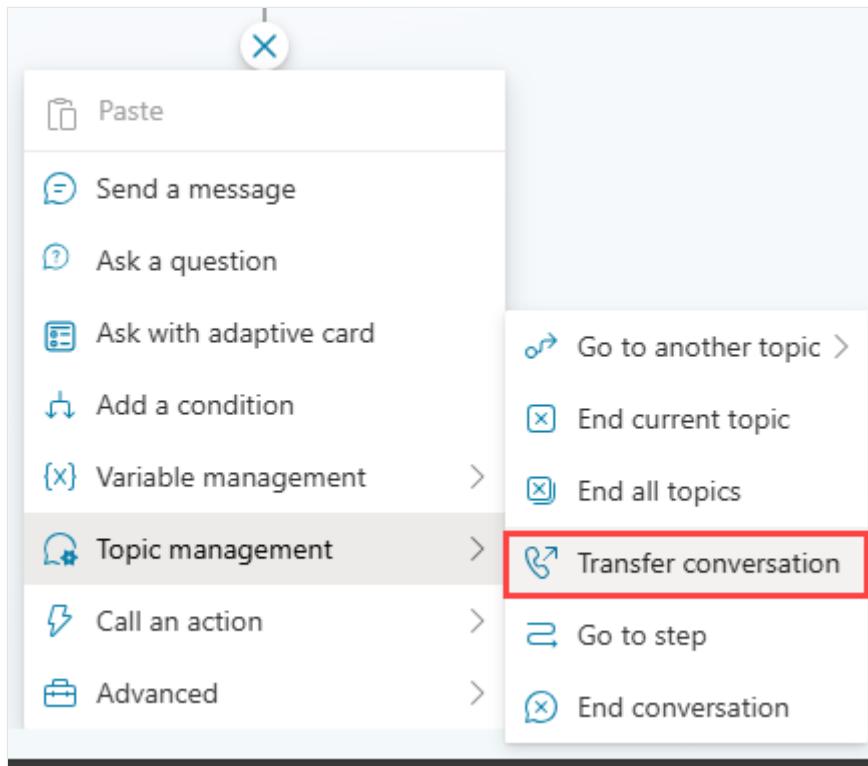
You can manually enter SSML tags that don't appear in the helper menu. To learn more about other tags you can use, see [Improve synthesis with Speech Synthesis Markup Language](#).

Transfer a call to an agent or external phone number

You can have the copilot transfer the call to an external phone number. In Copilot Studio, we support 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 an agent, see [Trigger hand-off to a live agent](#)-Explicit triggers.

Use voice variables

Copilot Studio supports the population of variables. You can use predefined variables, or create custom ones. See [Use variables](#) to learn how to use and create variables in copilot Studio.

A voice-enabled copilot in Copilot Studio supports context variables. These variables help you integrate your copilot conversations with Omnichannel for Customer Service when transferring a call.

For more information about context variables in Omnichannel for Customer Service, see [Manage context variables](#).

This integration supports these scenarios with variables when you transfer:

[+] [Expand table](#)

Variable	Type	Description
System.Activity.From.Name	String	The copilot user's caller ID
System.Activity.Recipient.Name	String	The number used to call or connect to the copilot
System.Conversation.SipUiHeaderValue	String	SIP header value when transferring through a direct routing phone number
System.Activity.UserInputType	String	Whether the copilot user used DTMF or speech in the conversation

Variable	Type	Description
<code>System.Activity.InputDTMFKey</code>	String	The copilot 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

Speech recognition improvement

Voice-enabled copilots 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 copilot to convert from speech to text.

To ensure the speech input is recognized accurately, you can improve speech recognition:

1. With your copilot open, select **Settings > Copilot details**.
2. Go to the **Voice** tab and adjust the copilot's default customized automatic speech recognition settings.

Settings

Save Reset Export Delete

Copilot details

General Voice Advanced

Make sure your copilot works for you with voice-first features like advanced speech recognition and dual-tone multi-frequency (DTMF) input. Voice optimization is temporarily only supported in telephony channels. [Learn more](#)

Optimize for voice

A¹ Primary language
English (United States) (en-US) (Primary)

> Secondary languages (0)
[Add and configure languages](#)

Use voice as primary authoring mode

▼ Customized automatic speech recognition

Add voice-specific data based on your copilot's content to your speech recognition model. Your copilot will take longer to publish. [Learn more](#)

Increase accuracy with copilot data

3. Select **Save** at the top to commit your changes.

4. Publish your copilot to see the new changes.

ⓘ Note

- A copilot with large trigger phrases and entity sizing takes longer to publish.
- If multiple users publish the same copilot at the same time, your publish action is blocked. You need to republish the copilot after others finish their existing copilot edits.

To learn more about the fundamentals of publishing, see [Key concepts-Publish your copilot](#).

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

DTMF support for your voice-enabled copilot

Article • 06/12/2024

Instead of relying on voice responses, dual-tone multi-frequency (DTMF) allows you to ask your copilot 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 copilot 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 copilot should wait before parsing the user's input](#).

If the key input by the copilot's user isn't mapped to the node's options, the copilot repeats the question or message and encourages the user to try again.

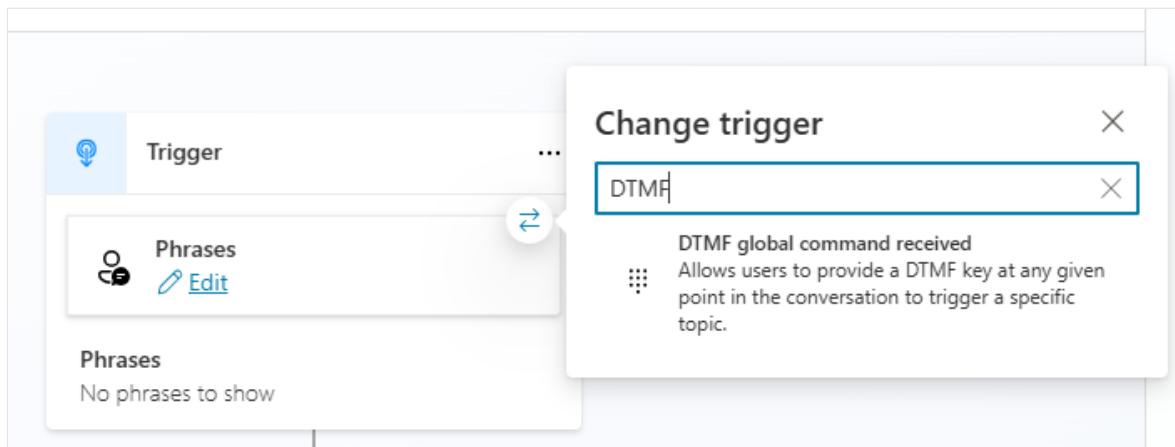
If the key input by the copilot's user isn't mapped to the DTMF global triggers, the copilot triggers the System topic, *Unknown dialpad press trigger (voice)*. You can customize it for your business logic.

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.

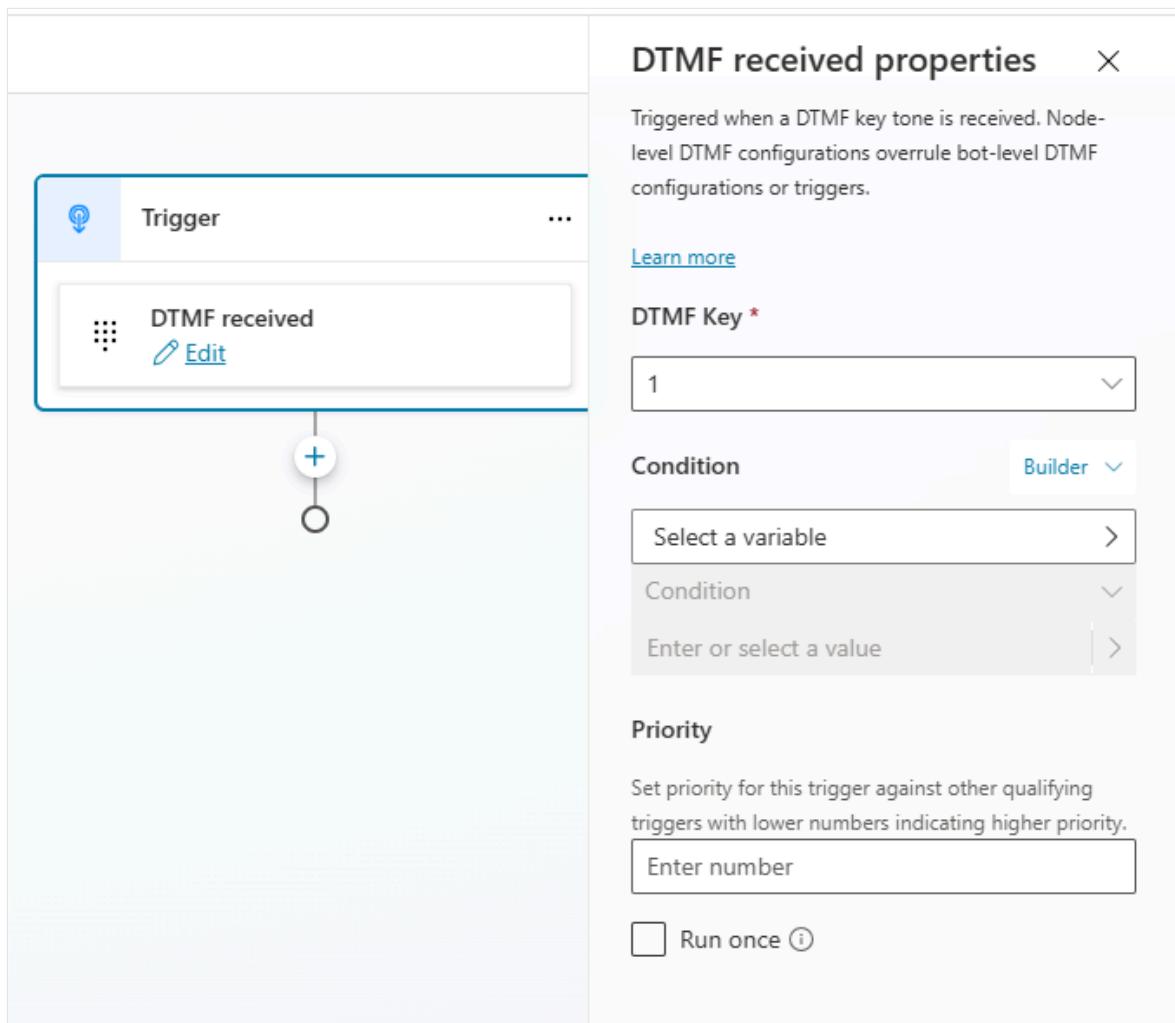
Name	Trigger	Description	Editing
Conversation Start	On Conversation Start	This system topic triggers when the b...	
End of Conversation	On Redirect	This system topic is only triggered by ...	
Escalate	On Talk to Agent	This system topic is triggered when t...	
Fallback	On Unknown Intent	This system topic 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 copilot 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

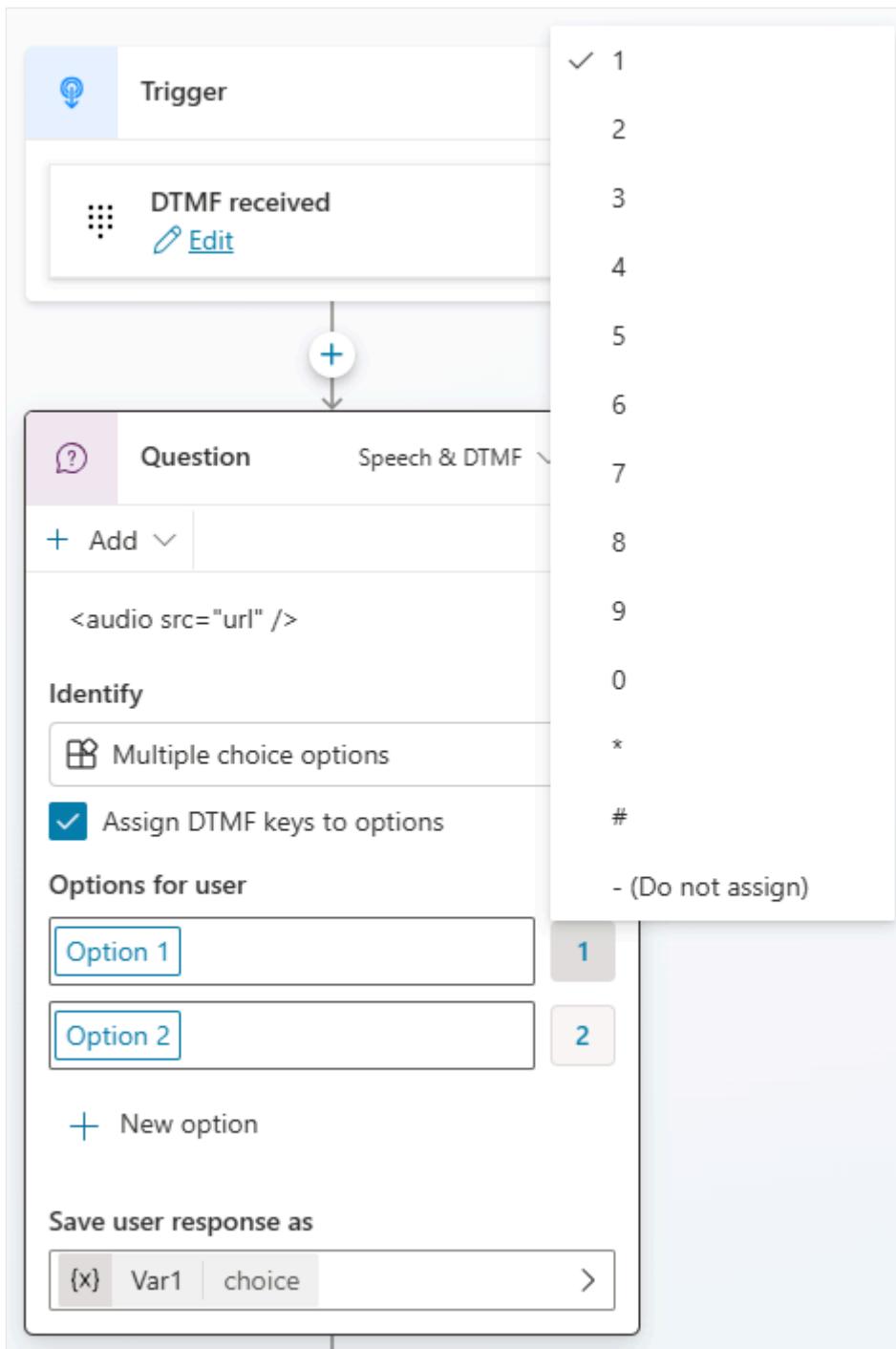
Entity (under Identify)	DTMF support	Details
Point of interest	✗	DTMF not supported
Speed		Can accept multi-digit DTMF input
State	✗	DTMF not supported
Street address	✗	DTMF not supported
Temperature		Can accept multi-digit DTMF input
URL	✗	DTMF not supported
Weight		Can accept multi-digit DTMF input
Zip code		Can accept multi-digit DTMF input

Assign DTMF keys

For some entity types, you can specify if the options provided by the copilot 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.

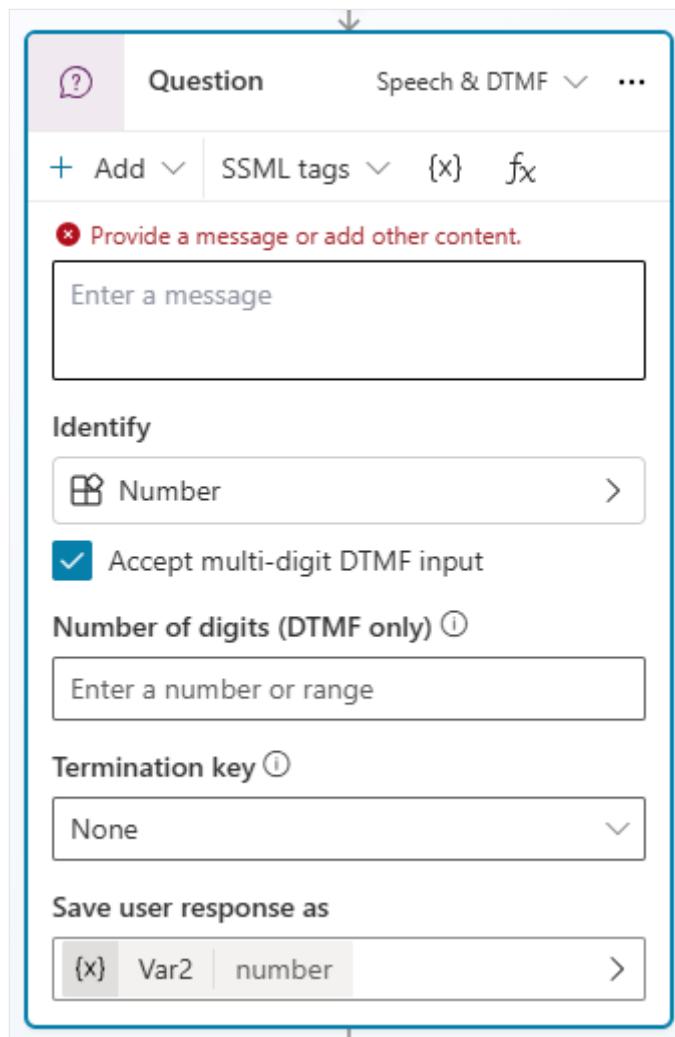
The copilot assigns keys in the order that the options are in, from 0 to 9.

You can have the copilot 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 copilot should read out the options to the copilot 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 copilot user's valid response. These multi-digit DTMF input entities include most entity types that extract a number from the copilot 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 copilot user has fewer or more digits than specified, then the copilot repeats the question.

If you don't specify the number of digits, then the copilot accepts any length of digits and continues the topic's flow.

You can also specify if the copilot should wait for a specific "termination" key to be pressed to indicate the copilot 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.

Copilot-level DTMF recognition

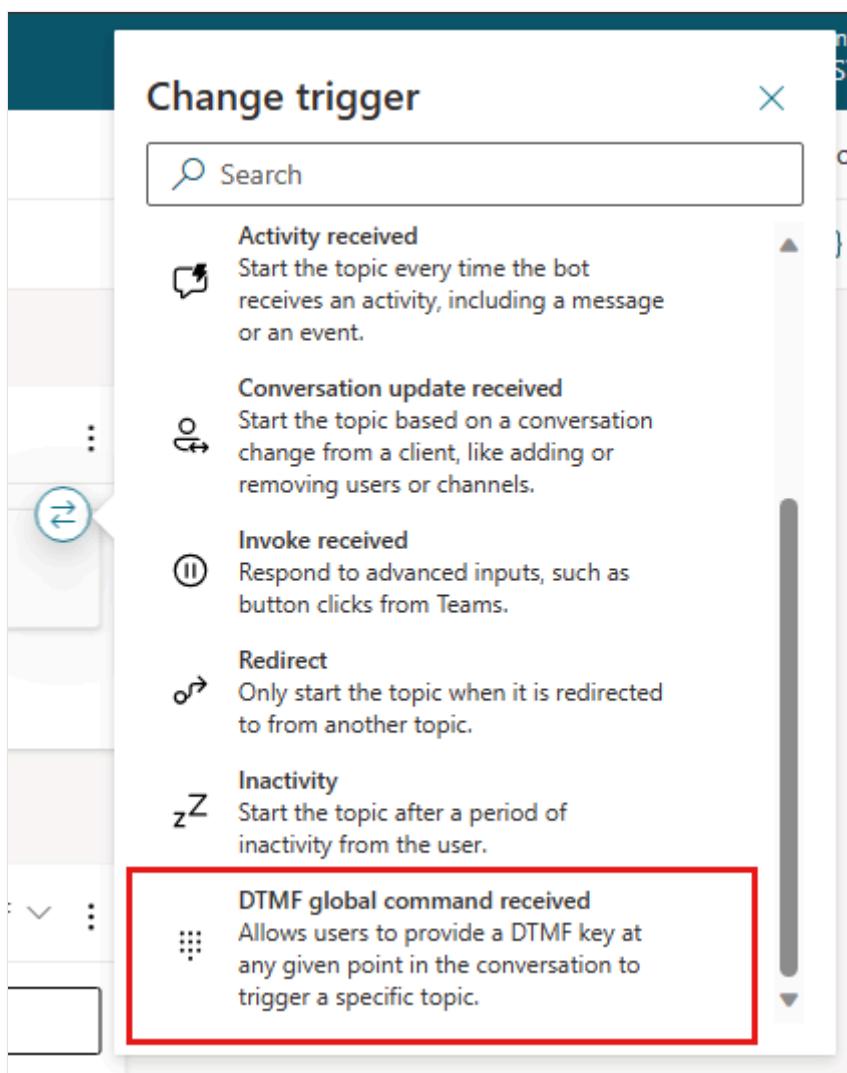
You can configure what key should be pressed for each option that the user can input and how long the copilot should wait before parsing the user's input.

ⓘ Important

The **DTMF received** trigger can only accept single DTMF keys. The trigger won't be called if the copilot 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 a copilot user to input keys without waiting for each question or prompt from the copilot. For example, caching is useful for copilot users who are familiar with the copilot'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 copilot 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
Copilot 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.
Copilot 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 copilot reprompts the question.

Scenario	Digit length	Condition	User input	Outcome
User doesn't hit the max length (valid length). The key input after the interdigit timer is cached for the next turn.	2-5	Doesn't matter if a term key is defined	1, 2, 3, then after the timeout, 1	The first two digits wait for the interdigit timer before 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 end user if the user doesn't input a termination key).

Feedback

Was this page helpful?

 Yes

 No

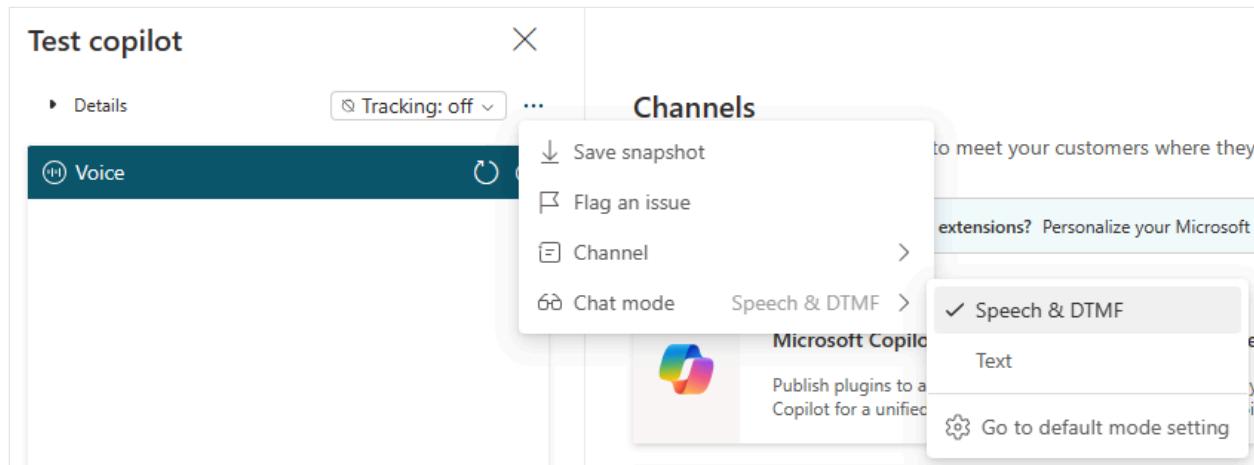
Provide product feedback ↗

Test your voice-enabled copilot

Article • 05/31/2024

You can test your copilot with text inputs that simulate the end 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 copilot

When you use **Test copilot** in chat in **Speech & DTMF** mode, you can get the copilot'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 copilot number
Conversation.SipUiHeaderValue	String	SIP header value for transfer-in context
va_CustomerLocale	String	Locale value for routing to another voice that supports the desired language. This value requires you to install the OC extension solutions.
va_SurveyConsent	Boolean (true/false)	<ul style="list-style-type: none"> - If the customer consents, the call is automatically transferred to the post-call survey copilot at the end of the personal digital assistant call. - If set to false, the call ends when the copilot hangs up, which requires you to install the OC extension solutions.

- If you created an SSML message for your voice-enabled copilot, 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'. It has a message input field ('Enter a message') and settings for 'Multiple choice options' (selected), 'Assign DTMF keys to options' (selected), and 'Save user response as' ('{x} Var2 choice').

If options accept DTMF input, you see the output for each option as *Press 1 or say book a flight* or *Press 2 or say check flight status*.

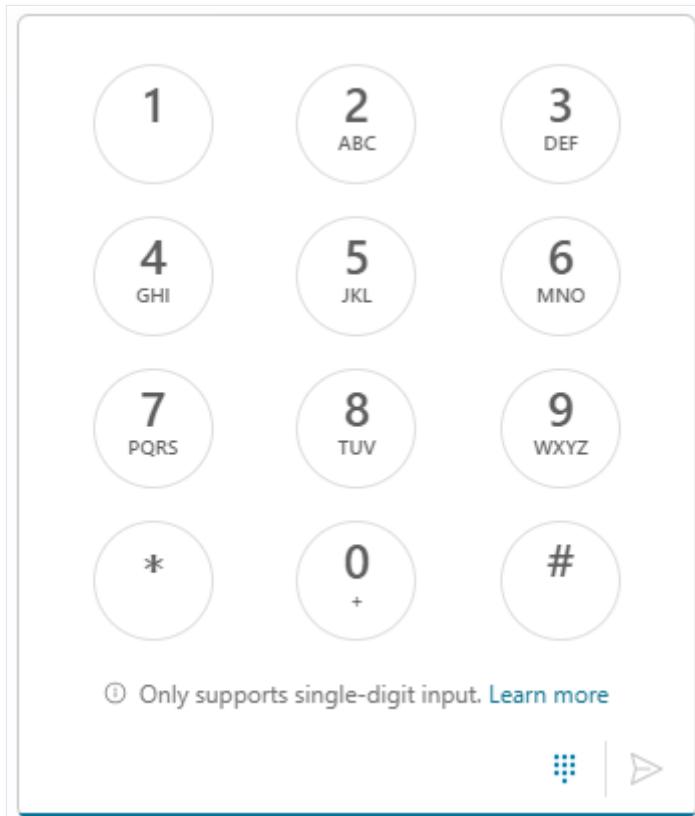
- For messages with barge-in disabled, you see **Barge in is not allowed**.

The screenshot shows a Twilio message preview. The message content is: "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 copilot message once. Instead of looping, the message displays until the operation

completes when it plays back in the voice channel.

- For single DTMF, you can press a key in dialpad, 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 copilot 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.

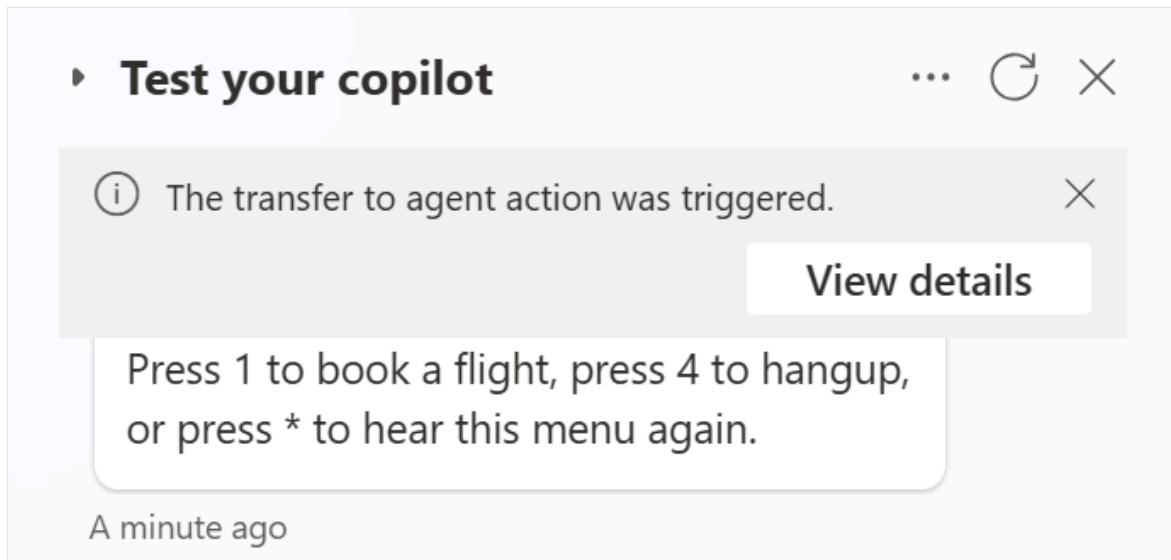
- For Multi DTMF, you can only use the command "/DTMFkey **DTMF keys**." For example, /DTMFkey 123, to simulate batched, multi-digit DTMF.

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.



Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Troubleshoot voice-enabled copilots and diagnose runtime error codes

Article • 06/26/2024

This article covers troubleshooting guidance for possible issues with your voice-enabled copilot, along with runtime error codes you might encounter.

Troubleshooting tips

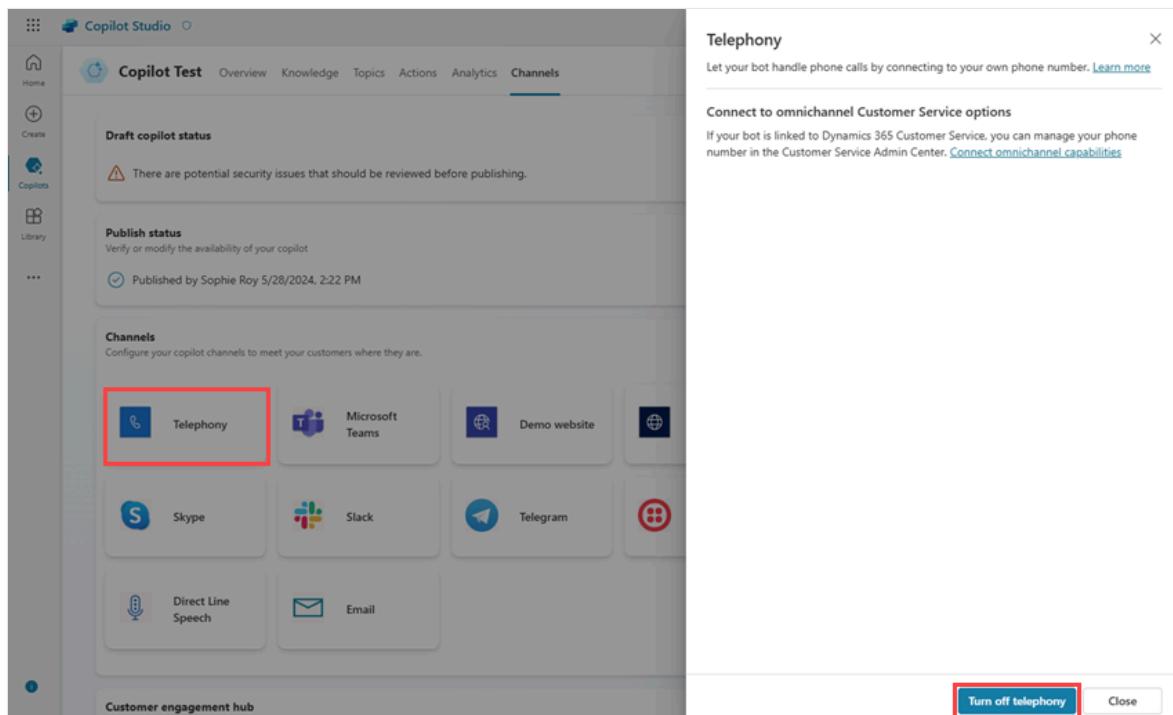
Callers can't engage with a published copilot

Issue: After a maker configures a phone number and workstream for a published copilot through Dynamics 365 Customer Service, callers only hear hold music or silence, and can't engage with the copilot.

Suggested solution: Disconnect and reconnect the Telephony channel and Dynamics 365 Customer Service customer engagement hub in Copilot Studio.

Steps:

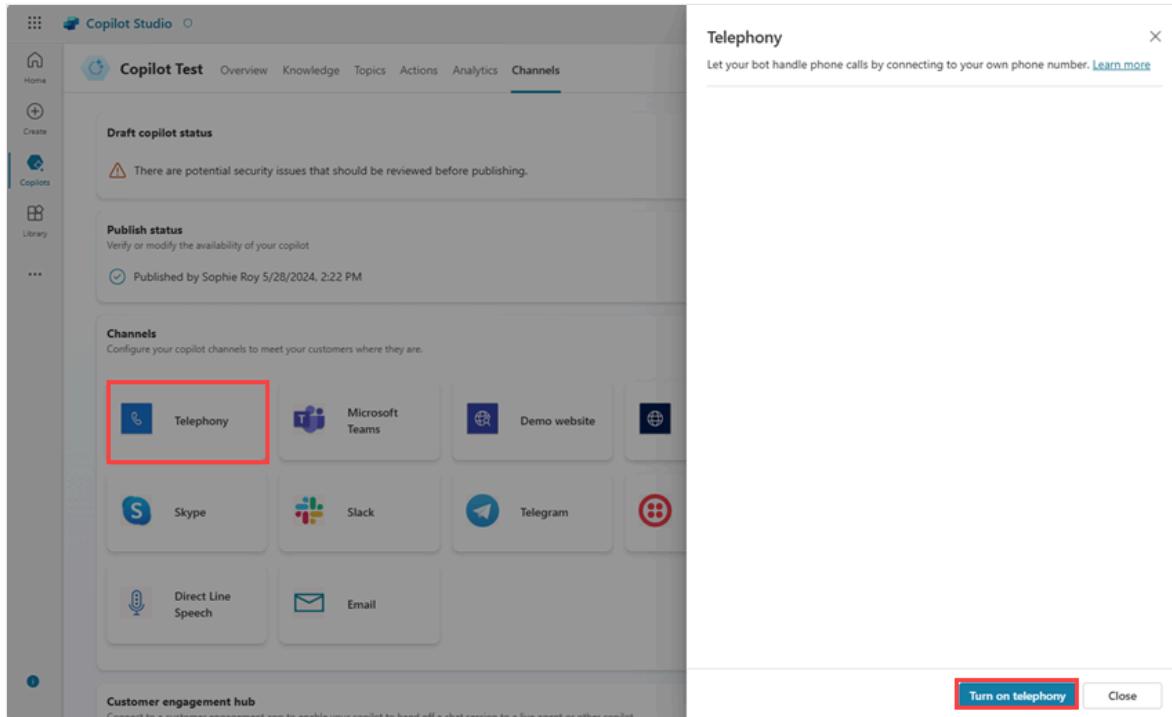
1. In Copilot Studio, go to **Channels > Telephony** and select **Turn off telephony**.



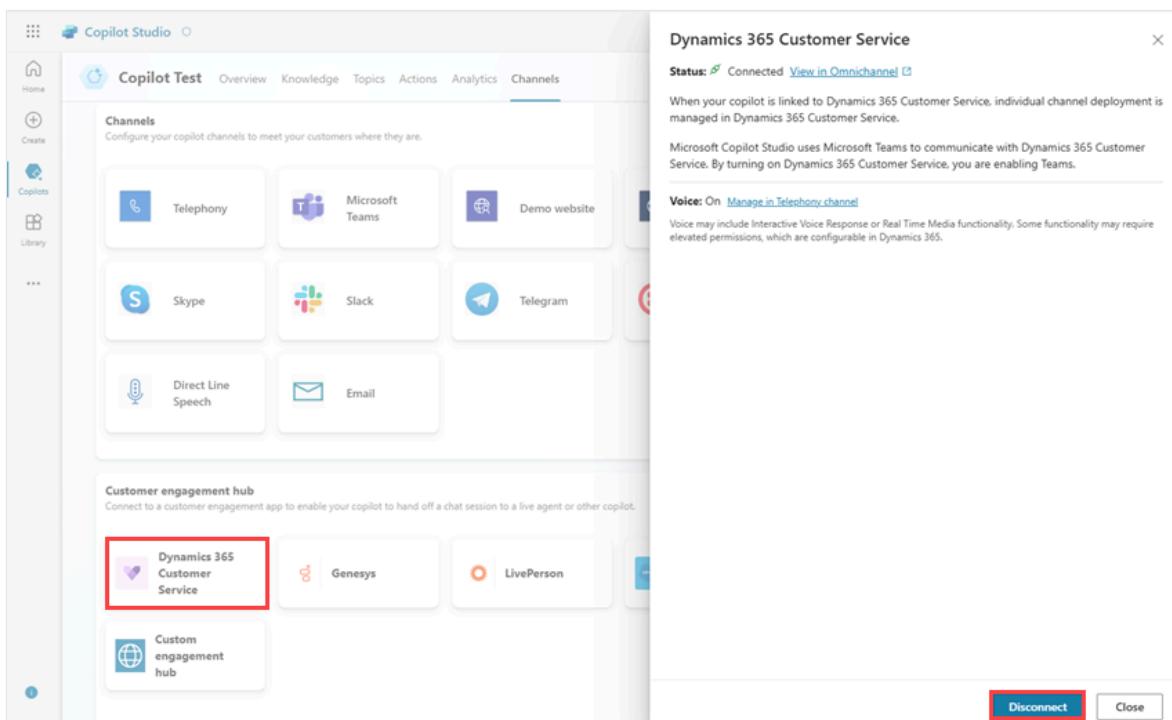
ⓘ Note

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

Genesys

LivePerson

Custom engagement hub

Connect **Close**

Can't publish a copilot or configure the Telephony channel

Issue: Maker can't publish a copilot 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.admins@powerplatform.com](#)). [Learn more](#)

✖ 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) Publish plugins to a Microsoft Copilot for a unified experience.	Telephony Let your users converse with the copilot via a phone number.	Microsoft Teams Chat with your copilot through a Teams app.	Den... Try o... the s...
Skype Expand your copilot's reach to customers on Skype.	Slack Expand your copilot's reach to customers on Slack.	Telegram Expand your copilot's reach to customers on Telegram.	Twilio Exp...
Email Expand your copilot's reach to customers on Email.			

Suggested solution: If you're unable to publish your copilot, 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 [Use DLP to block channels to disable copilot publish](#).

Voice-enabled copilot runtime error codes

Voice: CopilotNotResponseWithMessageBack

Error Message: The copilot 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 [Error Codes](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

Responsible AI FAQs for Microsoft Copilot Studio

Article • 03/26/2024

ⓘ 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.

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](#)

Feedback

Was this page helpful?



[Provide product feedback](#)

FAQ for generative answers

Article • 04/18/2024

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.

These frequently asked questions (FAQ) describe the AI impact of the generative answers feature in Microsoft Copilot Studio.

What are generative answers?

Generative answers make your copilot valuable out-of-the-box and increase the number of topics your copilot is conversational in, without requiring any manual dialog tree authoring.

What are generative answers capabilities?

When a user asks the copilot a question that doesn't contain a configured topic, the copilot can optionally search for relevant content from a source of your choosing. This search includes public websites, SharePoint, or your own custom data sources. The copilot uses generative AI to summarize that information into a response returned to the copilot user.

What are generative answers intended use?

Generative answers can be used as primary information sources in your copilot, 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 copilot.
- See the [Supported languages](#) section for the list of supported languages by this feature and their respective stage. You may be able to use other languages, but the answers generated might be inconsistent and the copilot might not respond properly or as you expect.
- This capability may 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 may 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 your copilot users. We've 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 copilots before publishing them, and consider collecting feedback from your copilot's users.

- Your admin can [disable the ability to publish copilots with generative answers](#) for your tenant in the **Power Platform admin center**.

The screenshot shows the 'Power Platform admin center' interface with the URL admin.local.powerplatform.microsoft.com/tenantsettings?setting=boostedConversations. The left sidebar has a 'Settings' section selected. The main area is titled 'Tenant settings' with a sub-section 'Publish bots with boosted conversations'. It describes allowing Power Virtual Agents to publish bots that can create real-time responses from a knowledge source chosen by canvas editors. A note states this feature is currently in preview. A toggle switch is set to 'Disabled'. Below it, a link leads to 'supplemental preview terms'. At the bottom right are 'Save' and 'Cancel' buttons.

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 may be your company website, for example www.microsoft.com. All webpages that belong to this domain would be searched for a match against the user's question.

Feedback you provide on your satisfaction with generated responses is used 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.

See also

- [Generative answers](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

FAQ for Copilot

Article • 03/26/2024

ⓘ 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.

These frequently asked questions (FAQ) describe the AI impact of the Copilot feature in Microsoft Copilot Studio.

What is Copilot?

The system is designed to generate a single bot topic from a human-written description.

What are Copilot's capabilities?

It combines the natural language understanding models already in Microsoft Copilot Studio with Azure OpenAI to:

- Understand what the bot 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 bot maker's request

What is Copilot's intended use?

The **Create with Copilot** option in the Microsoft Copilot Studio canvas lets bot 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, and offensive content, as well as 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 may be returned when users converse with the system in languages other than English.

- Your bot must be created in the US region. Other regions, and languages other than English, aren't currently supported.
- This capability may be subject to usage limits or capacity throttling.
- Topics generated by the capability are not always perfect, and may 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 bot users provide to continually improve our content filters. These filters and mitigations are not foolproof.

ⓘ Note

You should always test and review your bots 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 is not 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 on your satisfaction with generated topics will be used to improve system quality. You can provide feedback by selecting thumbs up or thumbs down icon for responses generated through Copilot. You can also include more feedback in free text.

See also

- [Create \(and edit\) with Copilot](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)