

# Power Platform Insights

## June 2023

# Microsoft Copilot: A Game Changer for Low-code Developers



**Featured Resource: Check out this on-demand webinar with our Power Platform and Dynamics 365 experts and featuring Copilot and Viva Sales:**

[AI and the Way We Work: Power Platform Toolbox Primer](#)

# INTRODUCTION

---

Microsoft recently introduced capabilities across most of their platforms to leverage the power of AI to automatically perform actions, and in the Power Platform, that means creating apps, flows, webpages, and chatbot topics from a prompt. This has the potential to significantly change what the future of business applications looks like (both pro code and low code), and in this issue of **Power Platform INSIGHTS**, we want to bring a realistic perspective to Microsoft Copilot — what is it good for now, how to maximize the effectiveness of Copilot, and what might it mean for the future?

Despite its name, Copilot does not commandeer the development process. Instead, it reduces barriers to app creation and automation, making these tasks more accessible to a larger audience. As the technology evolves, it may eventually tackle more complex scenarios, but the role of humans remains indispensable in understanding the business process, crafting the prompt, and refining the output of the Copilot-generated apps and flows.



Donald Panek, director of the Hitachi Solutions Rapid Solution Development (RSD) practice, underscores the importance of human involvement in the AI process: “While it is extremely helpful in increasing productivity across many areas, it still requires human scrutiny to ensure accuracy, which means smart humans still always be necessary. From a developer perspective, I see the need for developers and makers to create the conversational AI and train it for a host of specific industry and application needs.



Looking to the future, business applications are set to undergo a significant transformation. Generative AI models, such as ChatGPT, are already being used in conjunction with plug-ins to connect AI with a variety of services. The [recent announcement of Copilot plugins for Microsoft 365](#) signals a potential shift and expansion of much of what is currently accomplished with the Microsoft Power Platform to other interfaces. This transition is set to democratize digital solution building further, making it accessible to a broader audience, and significantly changing the whole concept of what defines an app. The plugin store may be the app store of the future.

In this dynamic landscape, here are the top three suggestions for how to best take advantage of the AI and Copilot revolution as an app maker/developer:

<b>Leverage AI for Learning and Development</b>	<b>Prioritize User Experience (UX)</b>	<b>Master New Skills</b>
<p>With the advent of generative AI, it has never been easier to enhance your development skills. ChatGPT makes an excellent coach for learning how to write PowerFX or troubleshooting your code and formulas. Copilot, by generating apps and flows, provides valuable examples to incorporate into your own projects.</p>	<p>As more makers begin creating apps and using Copilot for generation, UX will become more important than ever. Understanding how to enhance an app to optimize usability will be critical to standing out in an increasingly crowded field.</p>	<p>Learning how to use Copilot's capabilities to the fullest will make you more effective as a developer. While it is possible to be a developer without these tools, those who figure out how to master these capabilities to enhance their core skills will be the most effective in their roles.</p>

Microsoft's Copilot is not an autopilot, but rather an AI-powered assistant designed to augment and support human creativity and problem-solving. The world of low-code development is changing rapidly, and Copilot is at the forefront of this evolution.

We feel that Power Platform is ideally situated to take advantage of the AI revolution, enabling makers to quickly build applications and design and deploy user interfaces for AI technology.



# Maximizing Copilot



## Prompt Engineering for Microsoft Power Platform Copilot

The quality of the results you will get from conversational AI greatly depend on the way prompts are crafted. These prompts form the backbone of human-computer interactions, dictating the scope and depth of responses generated by the AI. If your prompt is too vague, you may not get the results that you want, and this applies to Copilot as well as ChatGPT.

## What is Microsoft Power Platform Copilot?

Copilot is a preview feature in Power Platform that allows you to create automation and apps using natural language expressions. Some of the available Copilots include:

- Power Apps Copilot: Describe an app to build it
- Power Pages Copilot: Create a form in a webpage or generate text for a website
- Power Virtual Agents Create with Copilot: Create topics in your chatbot using a prompt
- Power Automate Describe It to Build It: Write a prompt and Power Automate will build a flow based on your description

## What Microsoft Power Platform Copilot Isn't

While Copilot is a powerful tool for creating solutions, it isn't the only tool you'll ever need. The current iteration of Copilot is not a replacement for:

- Solution Architects: Copilot can create single data tables and back-end logic, but it does not replace the need for an architect when a solution requires design work that moves beyond the basics, such as designing a relational data model.

- **Developers:** Putting Copilot in the hands of experienced developers is necessary because the tool can only be fully utilized by someone who understands its limitations and can fill in the gaps to create a complete solution. For example, Copilot is limited in its ability to interface with other applications or services, including those in the Power Platform. Developers are still needed to add Power Apps or Power Pages components to flows, work with AI Builder, or add comments or desktop actions to flows, among other items.
- **UX/UI Designers and Researchers:** Copilot is an assistant for low-code developers, not a fully-fledged design tool. Designers and researchers are still indispensable for creating compelling virtual experiences for users that promote ease of use, productivity, and alignment with company branding standards.

## Prompt Engineering: The CIDI Method

The concept of “prompt engineering” is strategically designing prompts to maximize the efficacy of AI models like Copilot. The CIDI (Context, Instruction, Details, Input) method is one approach used for this purpose. Let’s break it down:



**Context:** Sets the stage for the AI and explains the situation or environment in which the instruction is to be executed



**Instruction:** The task the AI needs to perform that must be clear and precise



**Details:** Additional information that aids in executing the instruction more effectively



**Input:** The data required for the task, including user input or any other data relevant to the task

Note: Not all parts of CIDI are necessary in all prompts. The point is to try to include enough information for the AI to complete the task successfully.

## CIDI Method in Power Platform Copilot

In the context of Power Platform Copilot, prompt engineering using the CIDI method is vital to optimize the tool's potential. Let's examine how:



**Context:** In Power Platform Copilot, context could refer to the problem the user is attempting to solve. For example, the user might be trying to build an app for task management. The context would inform Copilot about the user's goal and the type of application they are building.



**Instruction:** Here, the user would provide a clear directive of what they want Copilot to do. It could be something like "Create a task in the task management app."



**Details:** In this step, the user provides extra information that could help Copilot execute the task better. For instance, details for the instruction above could have specific fields to be included in the task such as "Task Name," "Due Date," "Priority Level," etc.



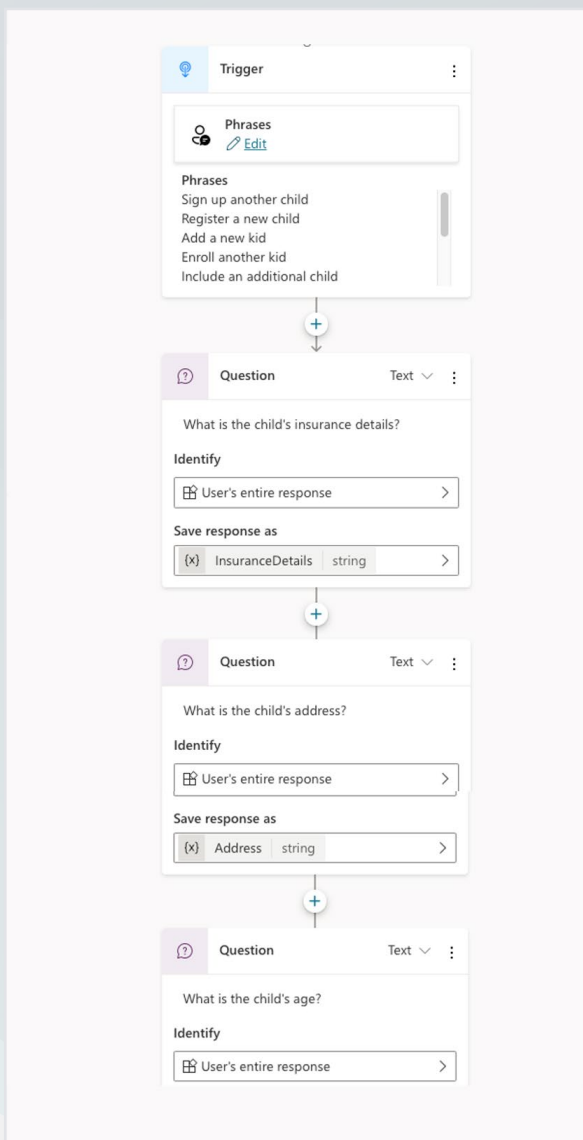
**Input:** The final piece of the puzzle is the input data. For Power Platform Copilot, this could come from the user or from existing data within the platform. In our example, input data could be the specific values the user wants to assign to the new task.

## Making the Most of Power Platform Copilot with the CIDI Method

Here are some examples of Copilot prompts that follow the CIDI approach:

In this example, we are building a chatbot in Power Virtual Agents (PVA) and want parents to be able to register their children for summer camp using the “Create with Copilot” feature. The following prompt includes complete information that will result in a complete topic:

In this scenario, the context is the summer camp registration system, the instruction is to capture child registration, and the details include the specific information required from each child. The input would come from the users interacting with the chatbot to provide the necessary data.



“

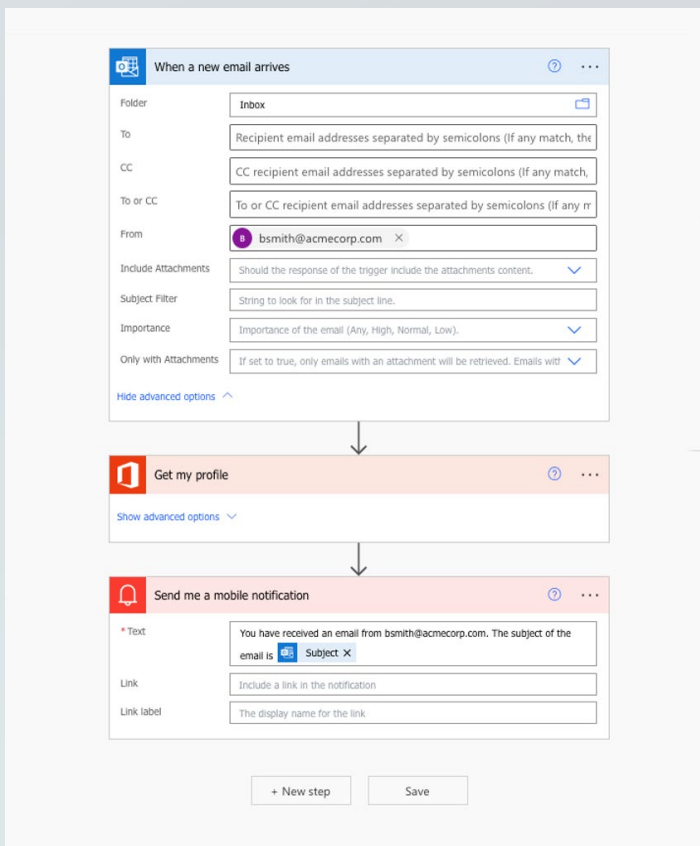
*In the context of our summer camp registration system, capture child registration for camp. Collect the following info about each child: insurance details, address, age, grade, and a selection from a list of common summer activities. After collecting these details, the chatbot should present a summary of the registration to the user and an option to continue with the registration.*

”



Another excellent example of the CIDI approach: "I am going on a trip but I want to be notified when I get an email from an important customer so I can quickly respond." Here's how we could write the prompt for Power Automate's Describe it to build it feature:

In this prompt, the context is the user's need to stay informed about specific emails. The instruction is to create a notification when a particular email arrives, with details about the source of the email and the desired notification format. The input data, in this case, would be the incoming email from **bsmith@acmecorp.com** and the folder in Outlook to monitor.



“

*I want to be notified when I receive a message. When an email arrives in my Outlook inbox folder from **bsmith@acmecorp.com**, notify me with a mobile notification with the subject of the received email.*

”

By using an approach like CIDI, you can dramatically improve the results that you get from generative AI and Microsoft Copilot. But you can also take this further for more complex prompts. [See a great example of how Joe Unwin \(Flow Joe\) used multiple paragraphs to generate a more complex PVA topic including multiple adaptive cards.](#)

Here are some links that might be helpful as you delve further into prompt engineering:

[Power Platform Prompt Samples Library](#)

[Prompt Engineering Techniques on Microsoft Learn](#)

## Do we still need AI Builder?

Microsoft has long been a leader in document automation, including capabilities like intelligent text extraction and invoice processing automation via Azure Form Recognizer and AI Builder. These solutions include general prebuilt models, as well as the ability to create custom models to process structured or unstructured documents.

Given the advancements in generative AI, do we still need to train models like AI Builder and Azure Form Recognizer?

When considering the approach that you should take with document automation, like invoice processing, consider the unique capabilities of each technology:

### AI Builder/Azure Form Recognizer:



**Specificity:** AI Builder has features for processing forms and documents, including invoices. It can read and extract structured information from these forms, making it a powerful tool for data extraction.



**Handwritten and Scanned Document Processing:** AI Builder can process handwritten text and scanned documents. This makes it a versatile solution for businesses that deal with a variety of document formats, including paper-based invoices or manually filled forms.



**Integration:** AI Builder is well-integrated into Power Platform, which includes Power Apps, Power Automate, and Power BI. If you're already using these platforms or need to build AI capabilities into your applications or workflows, AI Builder can be a suitable choice.



**Ease of Use:** With a user-friendly interface designed for business experts, AI Builder does not require extensive machine learning expertise. This makes it easy to build, train, and implement models for specific tasks.

### Azure OpenAI (ChatGPT or similar):



**Flexibility:** Azure OpenAI is a versatile tool that can be used for a wide range of tasks beyond form extraction. It is excellent for tasks that involve natural language understanding, such as information retrieval, text generation, translation, and more.



**Contextual Understanding:** While AI Builder excels at extracting specific fields, Azure OpenAI has superior capabilities when it comes to understanding the broader context of text. This can be beneficial when interpreting or analyzing invoice data, not just extracting it.



**Language Models:** Azure OpenAI leverages advanced language models like GPT-4, which can understand and generate human-like text. This can help with tasks that involve complex text interpretation, or generation of summaries, descriptions, etc.

### Key differences:

- With Azure OpenAI, you will need to produce the right prompt to indicate to GPT what to extract.
- GPT does not natively handle scanned and handwritten documents.
- GPT has a limit to the number of characters (tokens) you can send in one request, so if an invoice has a lot of text, it can become an issue
- A call to GPT today is more expensive than the invoice processing model

### Complementary use — it's not either/or:

These two tools can be used together to leverage the strengths of both. The good thing is that with AI Builder customers have access to the invoice processing model and [the GPT model](#). So, the customer can try both approaches and see which one works better.

You can also use these two models together. For instance, AI Builder could be used to extract structured data from printed or handwritten invoices or forms. Then, this structured data could be passed to Azure OpenAI for further processing, such as generating summaries, creating automated responses, or handling cases where the invoice processing model is not able to extract some data and try with GPT.

Azure OpenAI can be leveraged from Power Automate, AI Builder, and Power Automate Desktop, so you can easily extend your automations using generative AI.



## Why Hitachi Solutions?

Hitachi Solutions helps its customers successfully compete with the largest global enterprises using powerful, easy-to-use, and affordable industry solutions built on Microsoft cloud services.



[Contact](#) Hitachi Solutions today to get started!

Follow Us:



 **Hitachi Solutions**

[global.hitachi-solutions.com](http://global.hitachi-solutions.com)