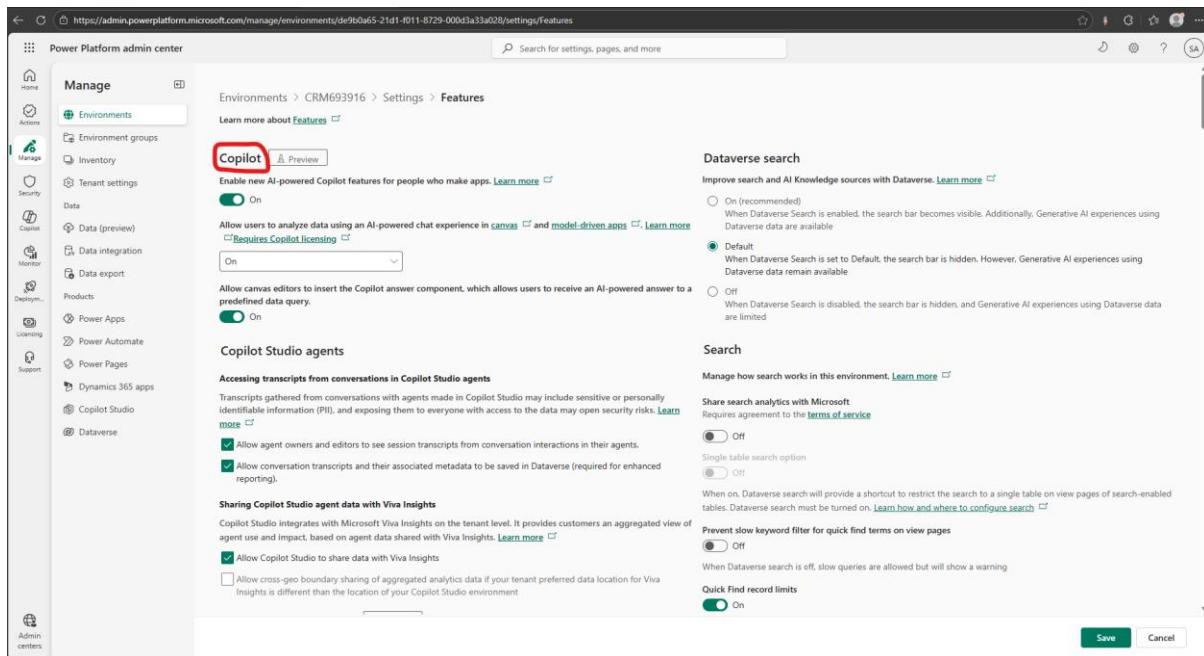


Hands-On Demo: Enhancing OOB agent for Model-Driven App in Dataverse

Prerequisites

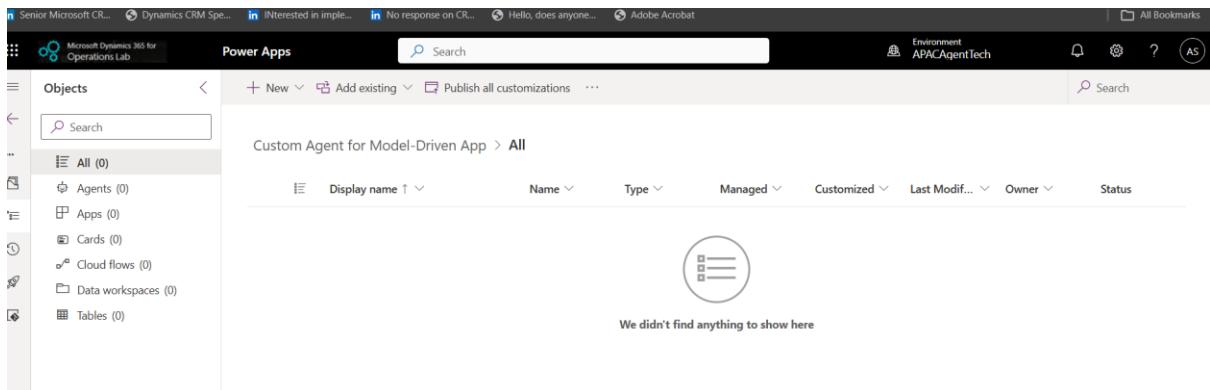
Please make sure that the following prerequisites are completed

1. Access to a Dataverse environment
2. Copilot studio license user license. This is required to publish the agent.
3. In Power platform admin center, go to **Settings > Product > Features** and confirm that the Copilot feature is enabled.



Step 1: Creating the Solution

Login to the Maker Portal. Click on **Solutions** and then select **New Solution**. Name your solution “Custom Agent for Model-Driven App” and add a meaningful publisher prefix if needed. Remember, a solution acts as a container for all components, such as apps, tables, and automation.

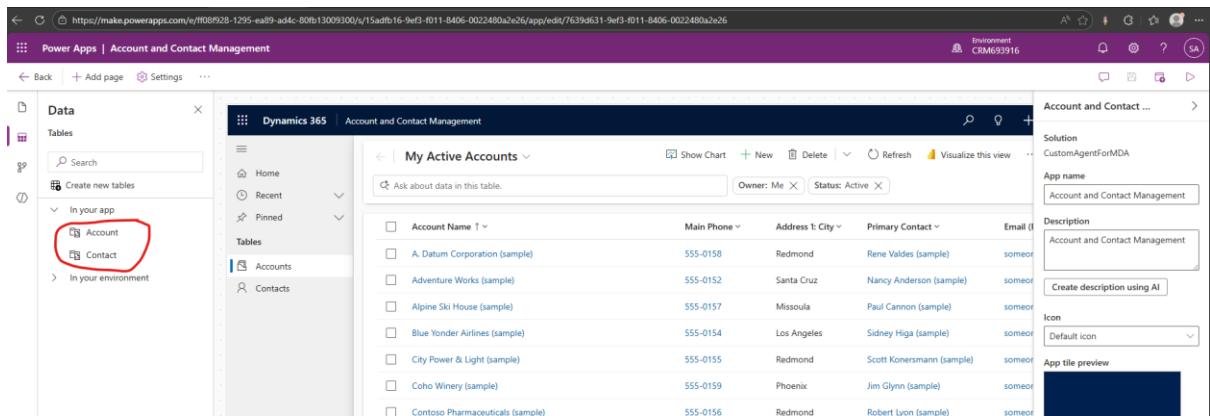


Step 2: Adding a Model-Driven App

Inside your new solution, click **New**, choose **App**, and then select **Model-Driven App**. Name your app “Account and Contact Management.” Using clear, descriptive names will help you and others manage and maintain the app more easily in the future.

Step 3: Adding Tables

On the left navigation panel, click **Data** and expand **In Your Environment**. Hover over **Account**, click the three dots, and select **Add to App**. Do the same for the **Contact** table. These tables serve as the core data entities for your app. Rename the area name to “Tables” to make the navigation panel more intuitive. If you wish to change the layout, remember that navigation can be customised to fit your needs.



Step 4: Configuring the Agent

Under **Agents**, open **Agent Assistance**. Find the **App Assistant Agent**, click the three dots, and select **Configure**. The configuration might take a few moments. This step connects the agent to your app’s data and functionality. Once done, click the three dots again and choose **Edit in Copilot Studio**.

The screenshot shows the Power Apps | Account and Contact Management interface. On the left, there's a sidebar with sections like Agents, Agent assistance, In your feed, and In your environment. The main area displays a Dynamics 365 view titled 'My Active Accounts' with a table of account data. A context menu is open over an account entry, with one option 'Edit in Copilot Studio' highlighted.

This opens the low-code editor where you can refine your agent's behaviour. In the new tab, publish the agent to make it available within your app.

The screenshot shows the Copilot Studio interface for the 'Copilot in Power Apps - Account and Con...' agent. The 'Overview' tab is selected. Key sections include 'Details' (Name: Copilot in Power Apps - Account and Con..., Description: None provided), 'Select your agent's model' (set to GPT-4.1 (Default)), 'Analytics' (showing conversation sessions, engagement, and satisfaction score), 'Triggers' (with an 'Add trigger' button), and 'Instructions' (describing the agent's purpose). On the right, there's a 'Test your agent' pane showing a sample conversation with the agent.

Step 5: Testing the Agent's OOB capabilities

Return to your app and open the Copilot side pane. Try interacting with the agent by running a sample query, such as "Show me all accounts in Redmond." If you are wondering about real-time updates, know that the agent retrieves data based on the most current information available in Dataverse at the time of your query.

The screenshot shows the Dynamics 365 interface with the 'Accounts' table selected. On the left, there's a navigation bar with links like Home, Recent, Pinned, Accounts, and Contacts. The main area displays a list titled 'My Active Accounts' with columns for Account Name, Main Phone, Address 1: City, Primary Contact, and Email (Primary Contact). A Copilot window is open on the right, showing an 'Analyze' card with the question 'What are the top 10 accounts with the highest...'. Below it, an 'Ask' card asks 'What accounts do not have a contact?'. The Copilot interface includes sections for 'Show me all the accounts which are in the city Redmond', 'You asked to see all accounts located in Redmond. The following accounts were found: City Power & Light (sample), Contoso Pharmaceuticals (sample), and A. Datum Corporation (sample), all with their main phone numbers listed. No other accounts in Redmond were found.', and a 'View Prompts' section.

Step 5: Customizing the agent to give it Create ability

Step One: Opening the Custom Agent and Adding a New Topic

- Access Copilot Studio:** Log in to Copilot Studio and locate your Model-Driven App's custom agent under **Agent Assistance**.
- Edit the Agent:** Click the three dots next to your agent, select **Edit in Copilot Studio**, and open the low-code editor.
- Add a New Topic:** In the topics section, click **+ New topic**. Name the topic (e.g., "Create Task Activity")

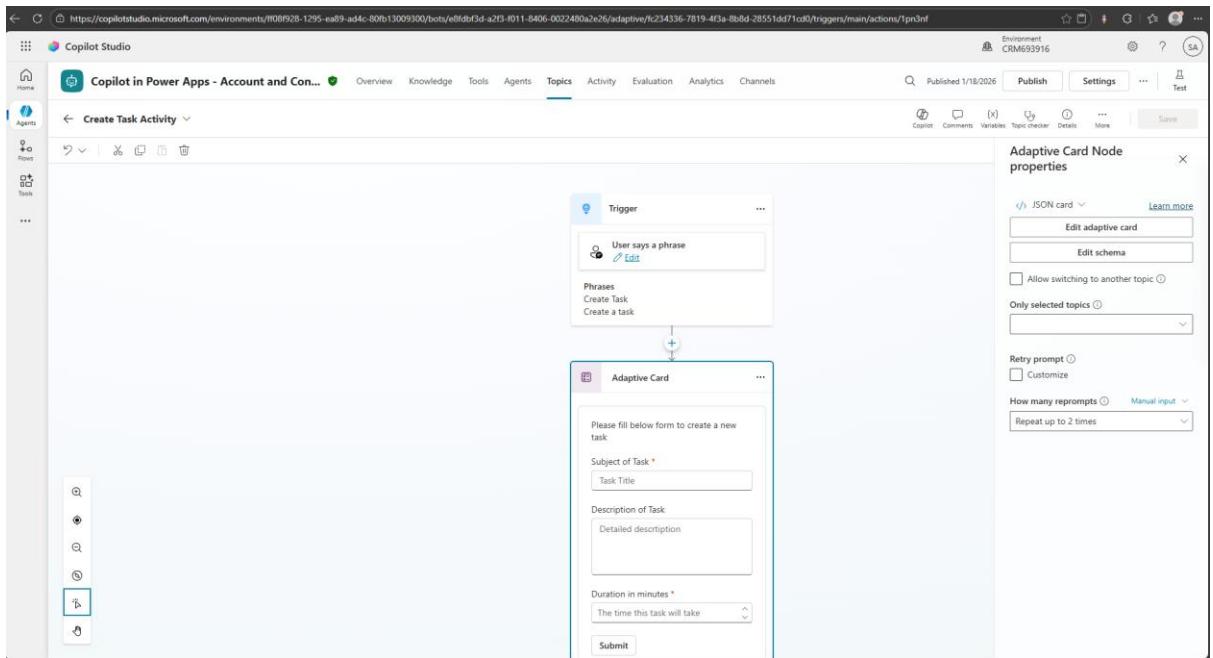
The screenshot shows the Copilot Studio interface. On the left, there's a sidebar with icons for Home, Agents, Tools, and more. The main area has a header "Copilot in Power Apps - Account and Con..." with tabs for Overview, Knowledge, Tools, Agents, Topics (which is selected), Activity, Evaluation, Analytics, and Channels. A search bar at the top right says "Published 1/18/2026". Below the header, a modal window titled "+ Add a topic" is open, showing "From blank" and "Add from description with Copilot" options. To the right of the modal is a table with columns: Name, Type, Trigger, Last modified, Editing, Errors, and Enabled. The table lists several topics, such as "AppCopilot DV Table Context", "AppCopilot.GetFeatureSetting", "BingSearchFallback", etc. Each row has a "Edit" button. On the far right, there's a "Test your agent" section with a message "Hello, I'm your app copilot.", a timestamp "45 minutes ago", and a text input field "Ask a question or describe what you need".

4. Add trigger phrases such as “create task” or “create a task.”

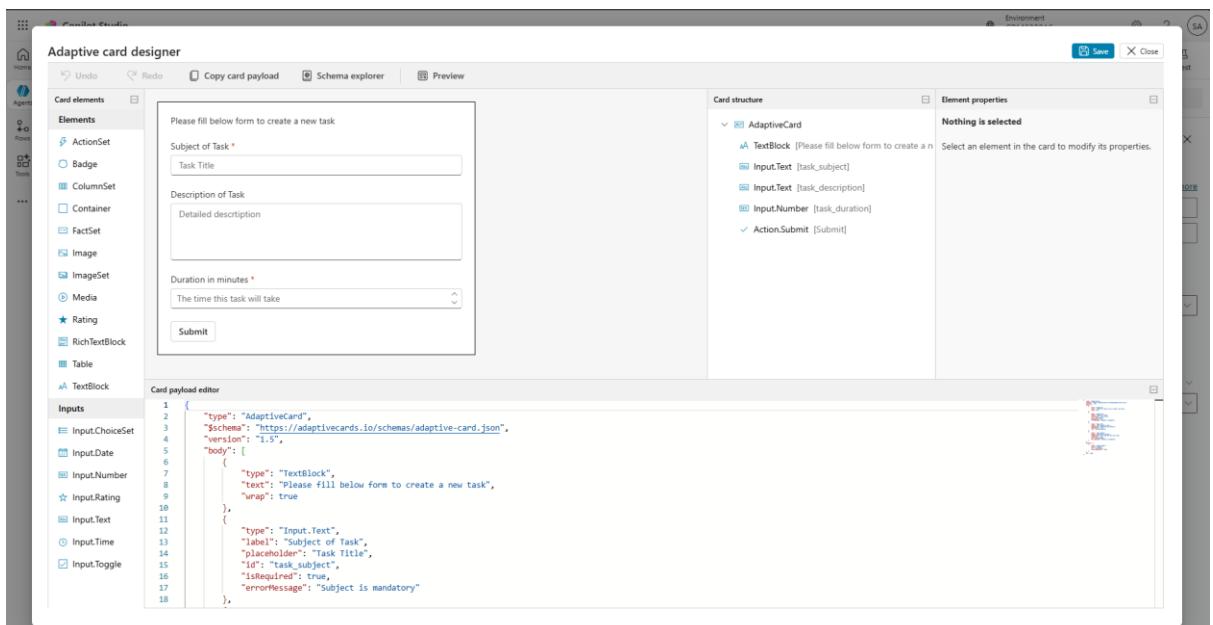
This screenshot shows the "Create Task Activity" configuration screen in Copilot Studio. The top navigation bar is identical to the previous screenshot. The main area shows a "Trigger" configuration with a card titled "User says a phrase" containing the phrase "Create Task". There are other sections visible below, such as "Phrases" and "Create Task". On the right side, there are buttons for Copilot, Comments, Variables, Topic checker, Details, More, Publish, Settings, and Test.

Step Two: Configuring the Adaptive Card with Required Fields and IDs

- 1. Insert an Adaptive Card:** Within your new topic, choose **Ask a question** and select **Adaptive Card**.



2. **Design the Card:** Add fields for task creation: **Subject** (single-line, required), **Description** (multi-line, optional), and **Duration in Minutes** (number, required). Assign IDs: **task_subject**, **task_description**, **task_duration**.



The Adaptive card JSON will be something like this:

{

```

"type": "AdaptiveCard",
"$schema": "https://adaptivecards.io/schemas/adaptive-card.json",
"version": "1.5",
"body": [
  {
    "type": "TextBlock",
    "text": "Please fill below form to create a new task",
    "wrap": true
  },
  {
    "type": "Input.Text",
    "label": "Subject of Task",
    "placeholder": "Task Title",
    "id": "task_subject",
    "isRequired": true,
    "errorMessage": "Subject is mandatory"
  }
]
  
```

```

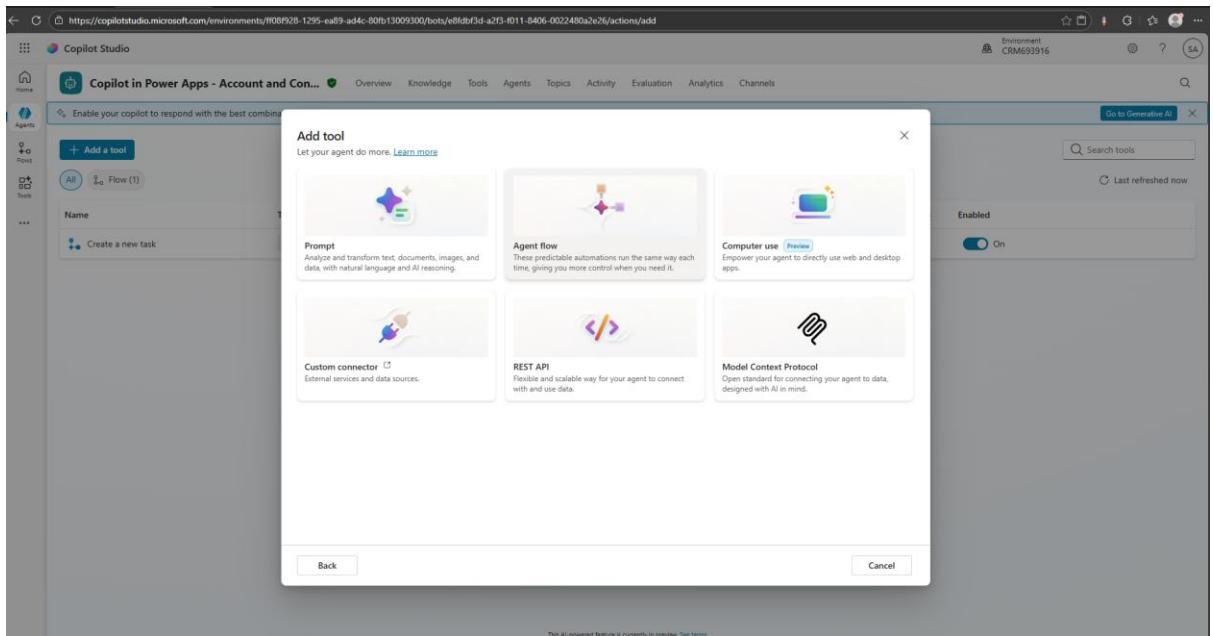
        "type": "TextBlock",
        "text": "Please fill below form to create a new task",
        "wrap": true
    },
    {
        "type": "Input.Text",
        "label": "Subject of Task",
        "placeholder": "Task Title",
        "id": "task_subject",
        "isRequired": true,
        "errorMessage": "Subject is mandatory"
    },
    {
        "type": "Input.Text",
        "label": "Description of Task",
        "placeholder": "Detailed description",
        "isMultiline": true,
        "id": "task_description"
    },
    {
        "type": "Input.Number",
        "label": "Duration in minutes",
        "placeholder": "The time this task will take",
        "isRequired": true,
        "id": "task_duration",
        "errorMessage": "Duration is mandatory"
    }
],
"actions": [
    {
        "type": "Action.Submit",
        "title": "Submit",
        "id": "task_submit",
        "associatedInputs": "auto"
    }
],
"id": "task"
}

```

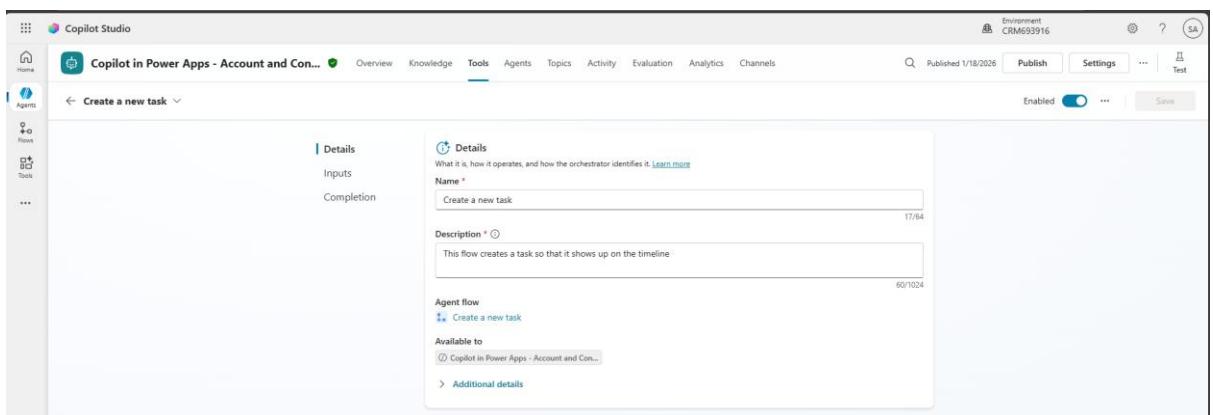
3. **Save the Card:** Review the card layout and save your changes.

Step Three: Creating the Power Automate Flow for Task Creation

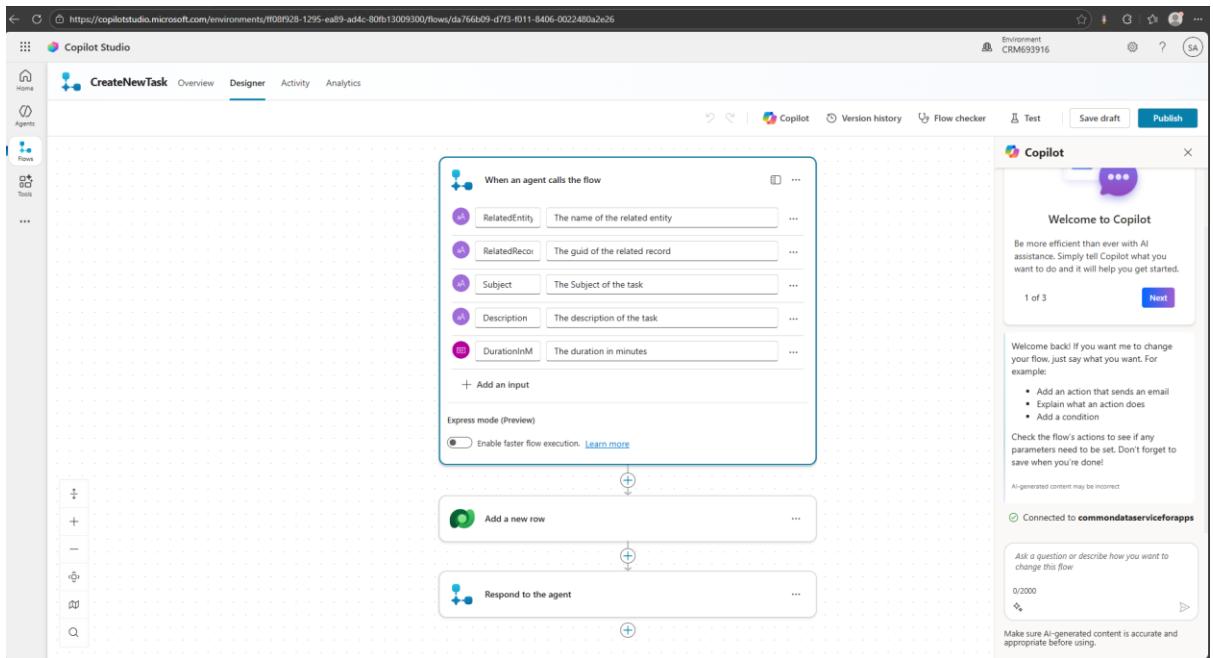
1. **Open Tools section:** In Copilot Studio, select **Add a tool button** and choose **Agent Flow**.



2. Configure the Flow: Name the flow “Create a New Task.”



3. Add variables: **relatedEntity** (text), **relatedRecord** (text), **subject** (text), **description** (text), (number). Add a Dataverse “Add a new row” step for the Tasks table, mapping subject, description, and duration, and use expressions for regarding fields.



Add a new row

Table name *

Tasks

Subject *

 Subject X

Description

 Description X

Due Date

Enter the expected due date and time.

Duration

 DurationInMinutes X

Owner (Owners)

Enter the user or team who is assigned to manage the record. This field is updated e...

Priority

Select the priority so that preferred customers or critical issues are handled quickly. ▾

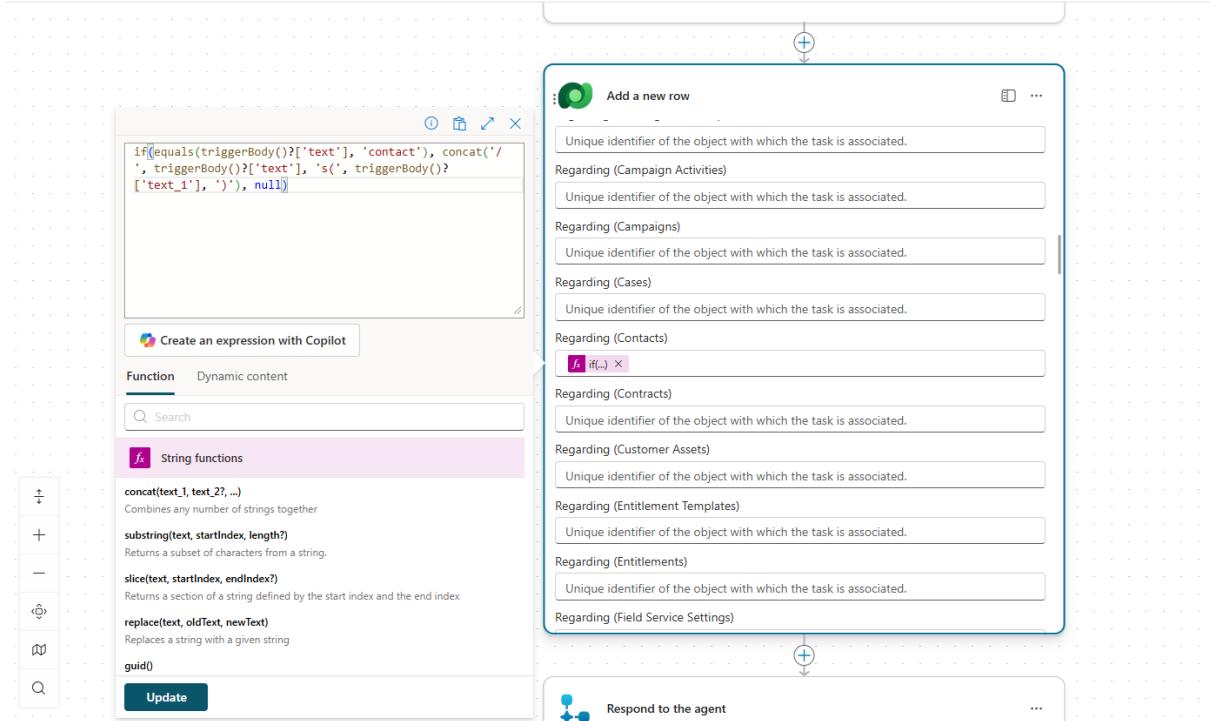
Regarding (Accounts)

 if(...) X

Regarding (Ad Placements)

Unique identifier of the object with which the task is associated.

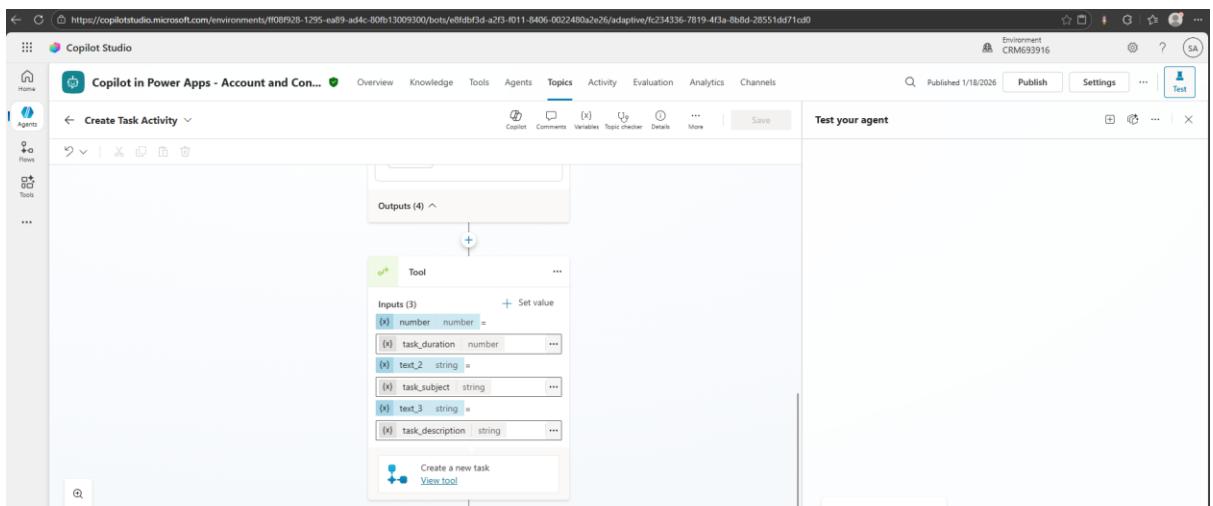
4. Provide this Expression for Account: `if>equals(triggerBody()?'text', 'account'), concat('/', triggerBody()?'text', 's(' triggerBody()?'text_1', ')'), null)`
5. Provide this Expression for Contact: `if>equals(triggerBody()?'text', 'contact'), concat('/', triggerBody()?'text', 's(' triggerBody()?'text_1', ')'), null)`



6. Save and Test the Flow: Test the flow to confirm it creates tasks as expected.

Step Four: Integrating the Flow with the Topic and Mapping Variables

1. Link the Flow: In your topic, add the Power Automate flow as an action step.



2. Map Variables: Map the adaptive card fields (subject, description, duration) to their respective flow parameters. For related entity and record, use custom values with out-of-the-box variables.

The screenshot shows the Copilot Studio interface for creating a new task. The task is titled "Copilot in Power Apps - Account and Con...". The "Inputs" section is active, displaying five input fields:

- RelatedEntityName ***: Set to "Custom value" with a value of "Global.PA__Copilot_Model..."
- RelatedRecordGuid ***: Set to "Custom value" with a value of "Global.PA__Copilot_Model..."
- Subject ***: Set to "Dynamically fill with AI" with a "Customize" option.
- Description ***: Set to "Dynamically fill with AI" with a "Customize" option.
- DurationInMinutes ***: Set to "Dynamically fill with AI" with a "Customize" option.

The "Completion" section shows "After running:" set to "Don't respond (default)".

Give this variable in `RelatedEntityName`:

`Global.PA__Copilot_Model_PageContext.pageContext.entityTypeName`

Give this variable in `RelatedRecordGuid`:

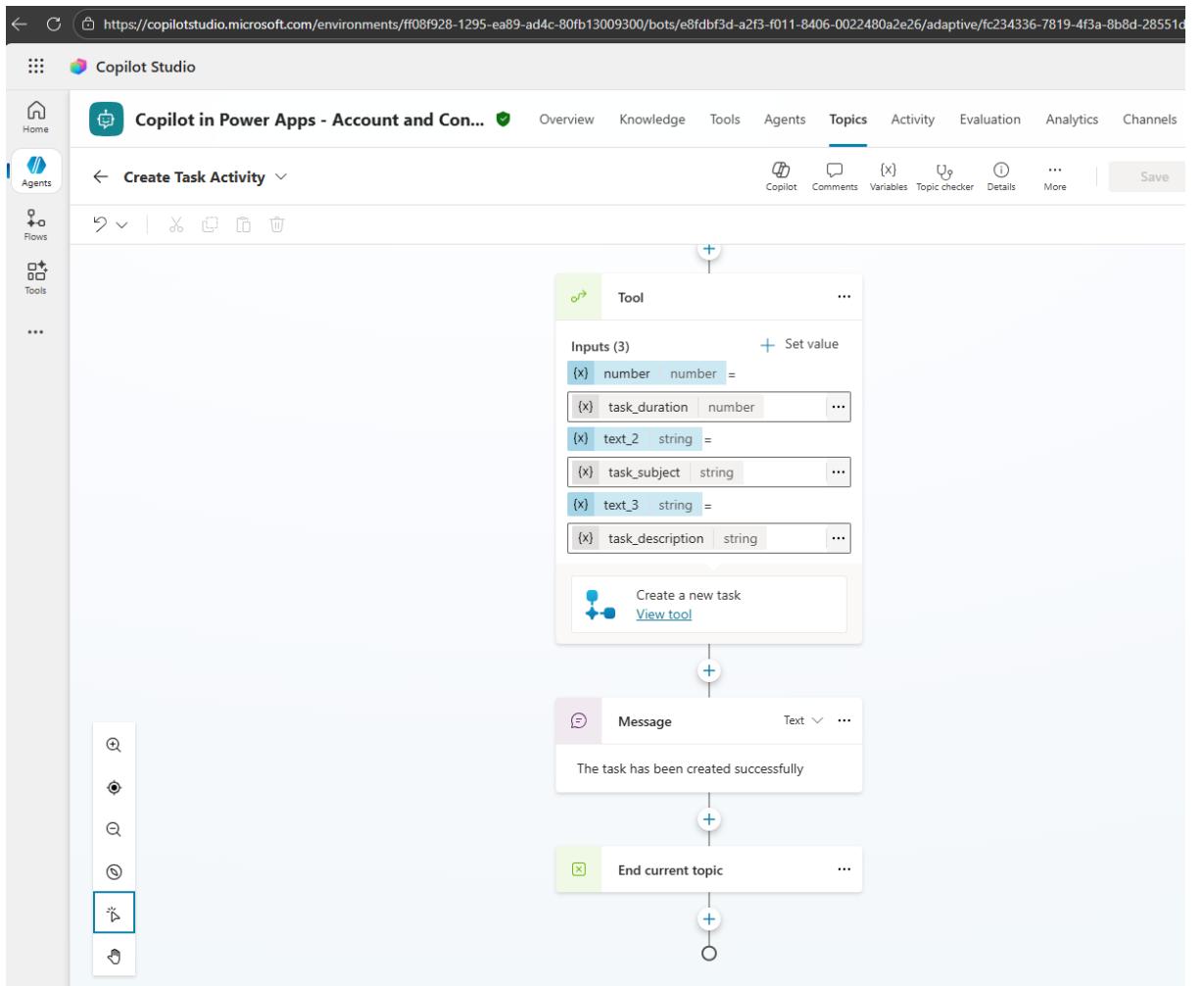
`Global.PA__Copilot_Model_PageContext.pageContext.id.guid`

These are context variables which automatically tell the agent which record is open which chatting with the agent.

3. **Handle Output:** Use the flow's output to confirm task creation or display an error if necessary.

Step Five: Adding a Confirmation Message and Ending the Topic

1. **Confirmation Message:** After the flow runs, add a **Send a message** step to confirm successful task creation (e.g., “Your task was created successfully and added to the account timeline.”).



2. **End the Topic:** Use the **End topic** action to close the conversation gracefully.

Testing: Verifying Task Creation in an Account Timeline

1. **Open the Account Record:** In your app, navigate to the relevant account (e.g., Adventure Works).
2. **Test the agent:** Type create task and fill the adaptive card and click submit.

The screenshot shows the Microsoft Dynamics 365 interface. On the left, there's a navigation bar with 'Home', 'Recent', 'Pinned', and 'Tables' sections. Under 'Tables', 'Accounts' is selected. The main area displays the 'Summary' tab for an account named 'Adventure Works (sample)'. Key details shown include an Annual Revenue of \$60,000.00, 4,300 Number of Employees, and System Administrator as Owner. To the right of the main screen is the Copilot Studio interface, which includes a 'Chat' section with a text input field and a 'Create a task' button. Below the chat is a form for creating a new task, with fields for 'Subject of Task' (set to 'Draft a contract for enhanced support'), 'Description of Task' (set to 'The renewal of the enhanced support is coming soon. Draft a legal document for its renewal'), and 'Duration in minutes' (set to 180). A 'Submit' button is present at the bottom of the task form.

3. You should see a message that the task is created successfully and the task is visible in the timeline.

This screenshot is similar to the one above, showing the Microsoft Dynamics 365 interface with the Copilot Studio extension. The main difference is that the timeline on the left side of the main screen now lists the newly created task, indicating that the task has been successfully created and is now visible in the timeline.

Conclusion

By following these steps, you can equip your Copilot Studio custom agent with the ability to create tasks, streamlining your workflow and enhancing user experience.