Microsoft

Autonomous Agents with Microsoft Copilot Studio

A Guide for Power Platform Makers

Microsoft worked with Shane Young, Power Platform MVP, to build out Copilot Studio training for Power Platform makers as part of a paid collaboration. This paper is intended to provide considerations for builders when building; it is not intended as a step-by-step.

Table of contents

1. Introduction: Autonomous Agents with Copilot Studio for Power Platform Makers 2

2. Leveraging Existing Power Platform Skills in Copilot Studio 2

2.1. Power Fx: Your Formulas Extend to Agent Logic 2

2.2. Power Automate Cloud Flows: Actions Your Agent Can Execute 3

2.3. Dataverse: Your Data Modeling Expertise is Key 4

2.4. More Power Platform Skills That Shine in Copilot Studio 4

3. Autonomous Contract Review Agent: A Guide for Power Platform Developers 5

3.1. Planning Your Contract Review Agent 5

3.2. Building the Agent 5

4. Deployment and Ongoing Management: Ensuring Agent Health, Security, and Scalability 12

4.1. Monitoring & Issue Detection: Keeping a Close Watch on Your Agent 12

4.2. Compliance & Security Monitoring: Ensuring Compliance and Security 14

4.3. Scaling & Optimization: Handling High Contract Volumes 15

5. The Power Platform Maker's Advantage: Leveraging Power Platform Expertise for Agents 17

1. Introduction: Autonomous Agents with Copilot Studio for Power Platform Makers

This paper introduces Microsoft Copilot Studio autonomous agents for experienced Power Platform developers. It outlines how existing skills can be applied to build these agents. Copilot Studio enables the creation of two types of Custom Agents: autonomous and conversational. This guide focuses on autonomous agents, which are designed to operate independently to perform tasks. A key advantage for Power Platform developers is that their existing skills are highly relevant for building these agents. Copilot Studio extends the capabilities of familiar Power Platform tools.

Why should a Power Platform developer care about Copilot Studio?

- **Shared Foundation:** Copilot Studio utilizes the Power Platform foundation, presenting familiar concepts and components.
- **Skill Applicability:** Expertise in Power Fx, Power Automate, and Dataverse is applicable to building agents, allowing developers to leverage their current knowledge.
- **Expanded Automation Capabilities:** Autonomous agents are suited for automating complex, multi-step, dynamic tasks that may require decision-making based on context.
- **Natural Language Interaction:** Give your instructions in user-friendly, natural language. Imagine building your solutions just by asking in plain English.
- **Generative AI Integration:** Agents can utilize generative AI for tasks like content generation, summarization, and orchestrating actions based on instructions.

This guide uses the scenario of an autonomous contract review agent to illustrate how Power Platform developers can apply their skills. The example involves processing contracts from email, extracting information, and generating a summary report automatically.

2. Leveraging Existing Power Platform Skills in Copilot Studio

Power Platform developers possess core competencies applicable to building agents in Copilot Studio. This section outlines how specific skills map to agent development:

2.1. Power Fx: Your Formulas Extend to Agent Logic

Remember writing those Power Fx formulas in your canvas apps? **Power Fx is natively supported in Copilot Studio.** You can use the same familiar formulas to define agent logic, manipulate data, and control the flow of your agent's actions.

- **Familiar Syntax:** It's the same Excel-like formula language you already know. No need to learn a new coding language.
- **Data Manipulation:** Use Power Fx to parse text, perform calculations, validate data, and transform information within your agent's topics and actions.
- **Conditional Logic:** Implement complex conditions and branching logic, just like you do in Power Apps. Fine-tune your agent's behavior with the power of formulas.

Example: Imagine your contract review agent needs to validate the contract date. You can use a Power Fx formula right in Copilot Studio to check if the date string is in the correct format, just like you would in a Power App.

Bottom Line: Your Power Fx skills are directly transferable. You can add logic to your agents without stepping outside the low-code environment. (Using Power Fx in Copilot Studio)

2.2. Power Automate Cloud Flows: Actions Your Agent Can Execute

If you're a Flow builder, get ready to directly apply those same skills in Copilot Studio. **Your agents can directly invoke Power Automate cloud flows as actions.** This means any workflow you've built (or can build) in Power Automate – updating Dynamics 365, calling APIs, sending emails – can be triggered by your AI agent.

- **Reuse Existing Flows:** With some slight modifications, you can bring your existing Power Automate workflows directly into Copilot Studio.
- **Create Flows On-the-Fly:** Build new flows right from the Copilot Studio authoring canvas when you need custom workflows.
- **Familiar Flow Designer:** It's the same intuitive visual designer found in Flow, with conditions, loops, connectors. All your flow-building skills apply.
- **Connectors:** Leverage the existing library of Power Platform connectors (SharePoint, Outlook, Teams, SQL, and thousands more). Your agent can tap into many of the same systems and services you already connect to in your flows.

How it works:

1. **"Run a flow from Copilot" Trigger:** Start your Power Automate flow with this special trigger.
2. **"Respond to Copilot" Action:** Use this action within your flow to send results back to your agent (within a speedy 100-second window) (Responding to an Agent).

Example:
For our contract review agent, you might create a Power Automate flow to...

- Extract key entities from the contract text (using AI Builder or other text processing actions).
- Save contract details and extracted entities to Dataverse.
- Generate a summary report in Word or PDF format.
- Send an email notification with the report attached.

Your agent can then trigger this flow whenever a new contract arrives, automating the entire review process.

Bottom Line: Your Power Automate skills are *directly* empowering your agents. Your ability to automate business processes with flows is the foundation for your agent's "skills." It's like giving your agents an automation engine, built with your expertise.

2.3. Dataverse: Your Data Modeling Expertise is Key

Microsoft Dataverse is the backbone of Power Platform solutions, and your data modeling skills are just as important for agents. **Agents can read and write to Dataverse tables** as part of their tasks.

- **Familiar Data Structure:** Your knowledge of Dataverse tables, columns, and relationships is crucial for designing agents that interact with data effectively.
- **Dataverse Actions:** Create actions that query Dataverse or other databases using Power Automate flows or built-in connector actions.
- **OData Filters:** Use your expertise in crafting OData filter queries to retrieve specific data from Dataverse, just like you do in Power Apps and flows.

Example:
Our contract review agent will likely use Dataverse to...

- Store contract records.
- Store extracted entities (contract value, dates, parties, etc.).
- Track the review status of each contract.

You can create actions for your agent to...

- Look up contract details in Dataverse based on a contract ID.
- Update the review status of a contract record.
- Create new records for extracted entities.

Bottom Line: Your Dataverse expertise is essential for building agents that are data-driven and integrated with your business data. It's about applying your data modeling skills to create agents that can work with your organization's information (Lookups/Filtering with Actions).

2.4. More Power Platform Skills That Shine in Copilot Studio

Beyond Power Fx, Power Automate, and Dataverse, even more of your Power Platform and Microsoft ecosystem knowledge comes into play:

- **Connector Ecosystem:** You know the hundreds of connectors inside and out. Agents use the same connectors for services like SharePoint, Outlook, and more, just like your flows. Knowing which connector to use and how to configure it is a huge advantage.
- **Custom Connector Expertise:** Built custom connectors to call REST APIs? You can integrate those custom connectors as agent actions too, extending your agent with bespoke operations and your company's APIs (Using Custom Connector Actions).
- **Custom Topics with Copilot Studio:** Built chatbots with Power Virtual Agents in the past? Your conversational design skills are directly relevant. Copilot Studio agents evolved from Power Virtual Agents. You still define topics (dialogue flows) and can call actions within them, but now it's simpler with the ability to have Copilot generate topics on your behalf using natural language (Natural Language Topic Authoring).
- **AI Builder & Cognitive Services Know-How:** Used AI tools in apps or flows? Agents can invoke AI Builder models (like form processing or text extraction) as actions, too. You can create agents that leverage AI for document processing and more. (How to use AI Builder inside an Agent)

- **Solutions & ALM Best Practices:** You understand Power Platform environments and solutions. Agents and their actions are managed in solutions, just like flows and apps. Your knowledge of solution management and application lifecycles is key for organizing and deploying Copilot Studio components. (ALM for Agents)

In short, Power Platform developers are uniquely positioned to excel with Copilot Studio. Your existing skills are directly applicable for building autonomous agents.

3. Autonomous Contract Review Agent: A Guide for Power Platform Developers

Now, let's put your Power Platform skills to work and build an autonomous agent for contract reviews.

3.1. Planning Your Contract Review Agent

Before we jump into building, let's define the agent's objective and scope:

- **Agent Goal:** Automatically review incoming contracts received via email, extract key information, and generate a summary report.
- **Trigger:** Incoming email containing a contract document (e.g., PDF, Word doc).
- **Actions:**
  - **Extract Contract from Email:** Monitor designated inbox and retrieve contracts.
  - **Extract Entities:** Use AI Builder in an agent flow or Power Automate cloud flow to extract key entities (contract value, dates, parties, key clauses, etc.) from the contract.
  - **Store Data:** Save contract details and entities.
  - **Generate Report:** Create a summary report of the extracted information.
  - **Send Notification:** Post the summary report to Microsoft Teams to notify to relevant stakeholders.
- **Output:**
  - Records containing contract details and extracted entities
  - Summary report
  - Email notification with the report

For Power Platform Developers: Think of this as designing a Power Automate flow but with an agent using generative AI to orchestrate actions based on your data.

3.2. Building the Agent

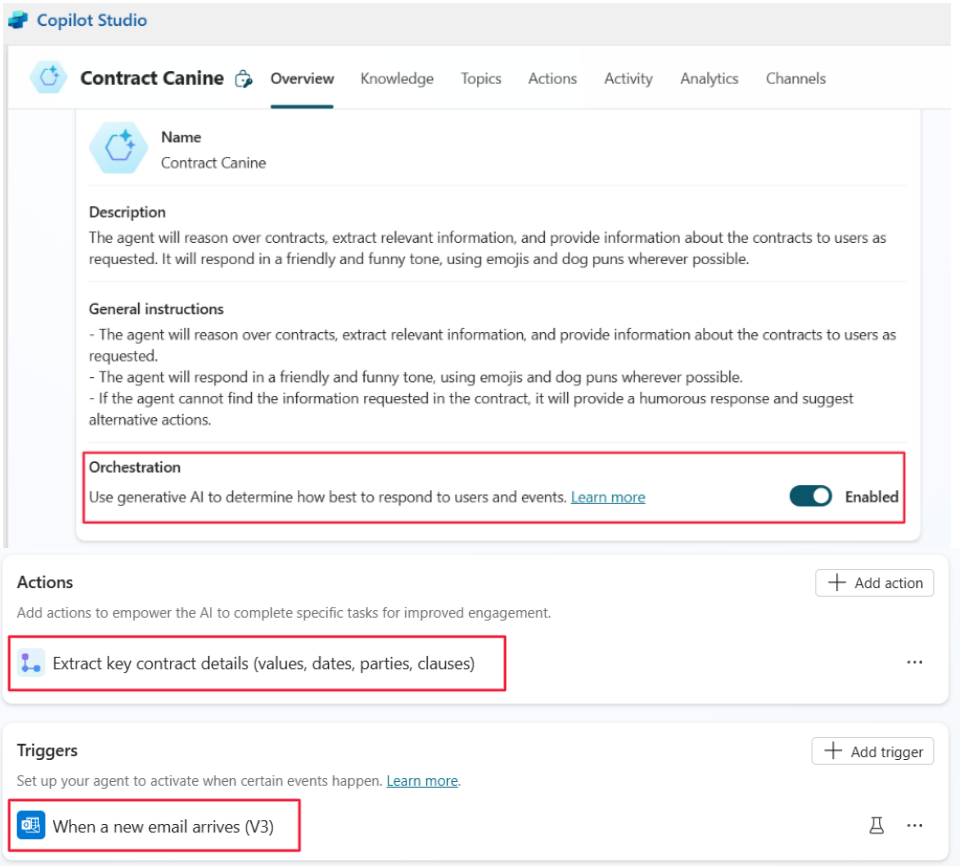
In this section, we outline in detail how to create the contract review agent. We'll use a variety of methods and products to ensure that our agent is capturing all relevant information from the contract as accurately as possible.

Step 1: Configure the Copilot Studio Agent and Actions

Create your autonomous agent and enable Generative AI orchestration. In Actions, add a new action for contract analysis. (Directly call the AI Builder model.) If using a flow, map the flow's outputs (like ContractValue, StartDate, etc.) to the flow's "Respond to Copilot" action to send information back to Copilot.

Add an event trigger so the agent runs on newly received contracts. For example, connect the Outlook / SharePoint event to the agent. Do this by calling the agent via Power Automate when the email arrives or by using built-in event support in Copilot Studio. You can remove the email trigger for your action's flow because your agent will now be triggered by receiving the email. (Event trigger overview)

Tools	Tips & Best Practices
Copilot Studio AI Builder Power Automate	**Generative orchestration** allows the agent to respond to event triggers (like an email arriving) and sequence actions to fulfill its task. Provide a clear, descriptive name and purpose for your **Action** (e.g., "Extract key contract details (value, dates, parties, clauses) from a document using AI Builder." This description is crucial; with generative orchestration, the agent will use this to decide when to invoke the action. (Orchestrate agent behavior with generative AI (preview)) Double-check the action's input and output definitions and test the action independently to ensure the agent correctly pulls data from the contract. Keep the action focused on only extraction and keep the output schema consistent while initially testing to make the agent's job easier.



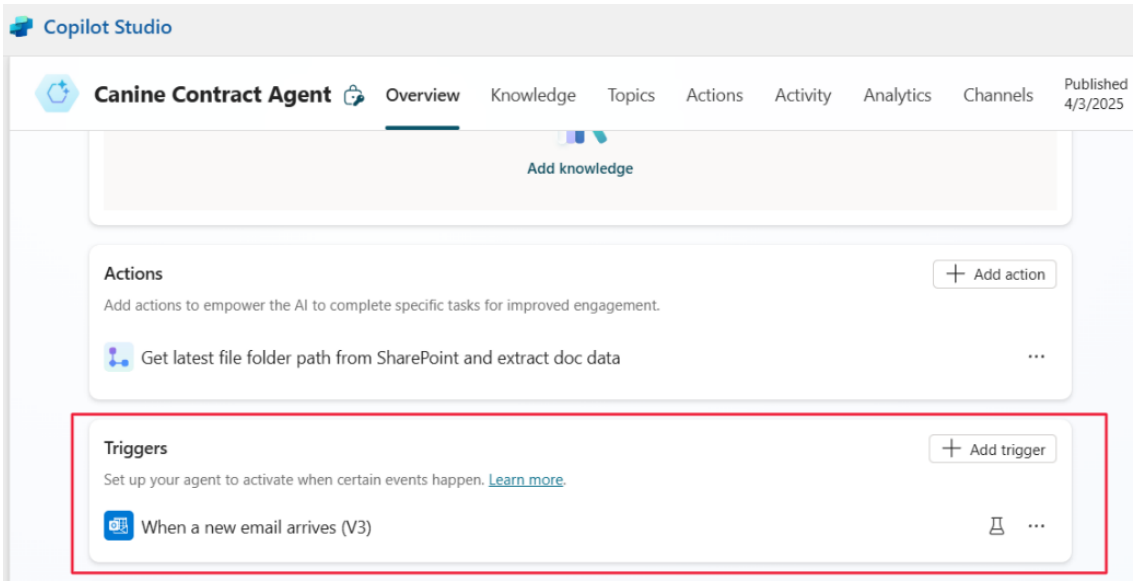
Pro Tip

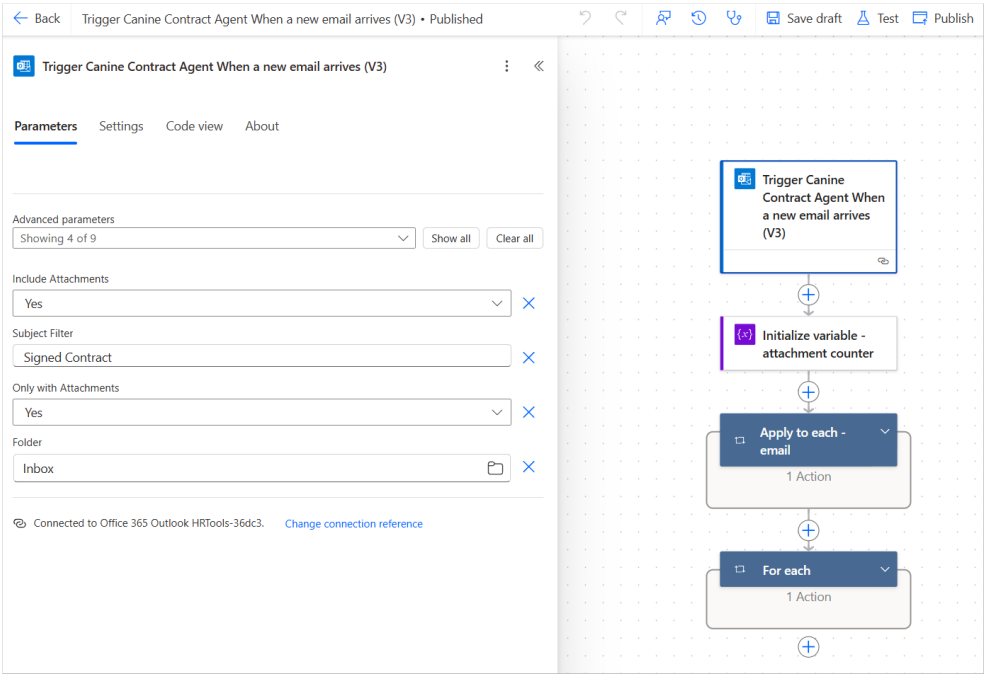
Consider Power Automate actions as functional components the agent can utilize. Your ability to create reusable, well-defined flows translates directly to creating actions that your agent can orchestrate based on data. Your understanding of event triggers in Power Automate is valuable to setting up autonomous triggers for your agent in Copilot Studio.

Step 2: Build an Agent with a Trigger That Uses Power Automate

Create a Power Automate cloud flow to handle incoming contract emails from your Office 365 Outlook inbox. Apply specific conditions or filters (e.g., sender, subject, or attachment type) so only relevant emails trigger the workflow. Save the email attachment – the contract – to a SharePoint library or Dataverse table column. Be sure to record the file's location or ID.

Tools	Tips & Best Practices
Power Automate cloud flow Outlook trigger "When a new email arrives (V3)"	Save your contract files with proper metadata (like email ID, sender, received date) to help tracking and to serve as an audit log. We'll use the file reference to trigger the agent's next workflow; this will ensure no contract email is missed and each contract file is available for AI processing.





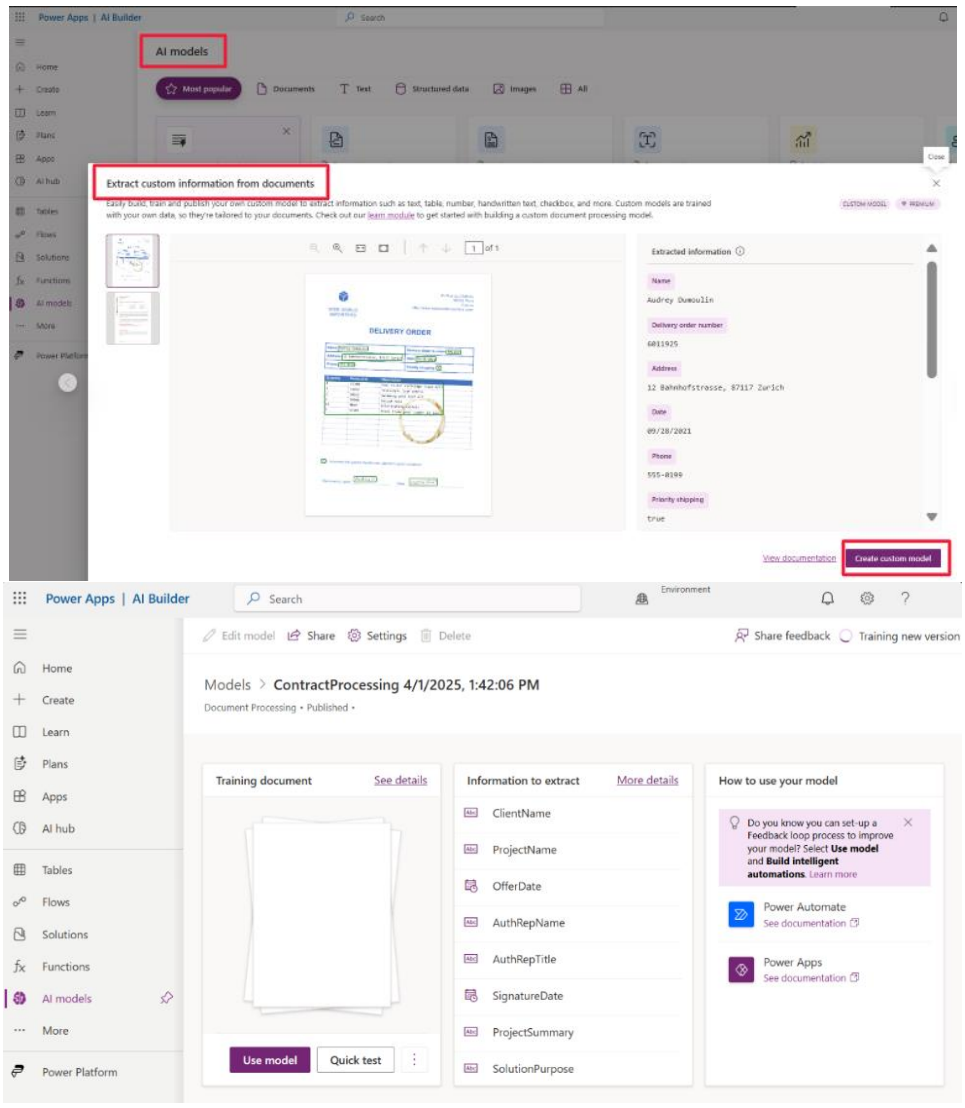
Pro Tip

Consider Power Automate actions as functional components the agent can utilize. Your ability to create reusable, well-defined flows translates directly to creating actions that your agent can orchestrate based on data. Your understanding of event triggers in Power Automate is valuable to setting up autonomous triggers for your agent in Copilot Studio.

Step 3: Prepare an AI Model for Contract Data Extraction

Set up an AI Builder document processing model to identify critical contract fields such as contract value, effective dates, involved parties, and key clauses. (Create a document processing custom model)

Tools	Tips & Best Practices
AI Builder Document Processing Model	• Ensure you have at least 5 contracts for custom model training. • Use general document type if contracts have varied formats; fixed document type if they have a standard layout. • Use the latest AI Builder model (v4.0 preview) for improved accuracy, better character recognition, and confidence scores for complex elements like tables; preview in a test environment. (What's new and planned for AI Builder, Try the new model version (preview), Model version 4.0) • Carefully label fields to help the model learn important terms (e.g., "Total Contract Value", "Start Date", "Termination Clause"). • Set a threshold confidence score (e.g., 80%) below which the agent will flag the entry for human review or re-training to minimize errors.



Pro Tip

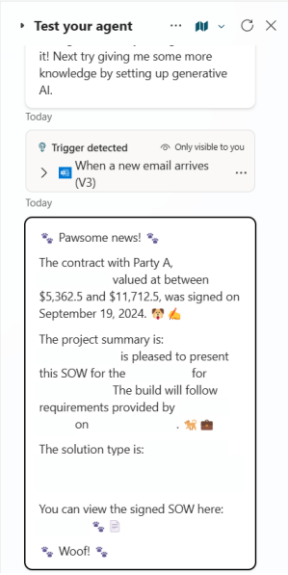
Consider Power Automate actions as functional components the agent can utilize. Your ability to create reusable, well-defined flows translates directly to creating actions that your agent can orchestrate based on data. Your understanding of event triggers in Power Automate is valuable to setting up autonomous triggers for your agent in Copilot Studio.

Step 4: Implement Generative AI for Summarization

Once the contract's key data is extracted, the agent needs to produce a useful summary report.

Agent-generated answer with generative orchestration: with generative orchestration enabled, let the agent's AI formulate the summary. For even better results, enable the new deep reasoning model. Ensure the extracted fields are passed into the agent's conversation context. (Copilot Studio does this by capturing action outputs into variables.) The agent's final response can then include a narrative like, "The contract between Party A and Party B, valued at $x, runs from [StartDate] to [EndDate]. Key provisions include [summarize key clauses]." Generative orchestration allows the agent to autonomously combine action outputs and any relevant knowledge to produce this summary Orchestrate agent behavior with generative AI (preview)).

Tools	Tips & Best Practices
Copilot Studio (Topics) Copilot Studio (Actions) Copilot Studio (Generative Orchestration) AI Builder (Structured Outputs) Azure OpenAI Connector	• Most of your agent's behavior will be driven by Instructions, but you still have topics available where you want the agent's response to be more structured or use a specific format. • For even more control, create a second action with either AI Builder or Azure OpenAI Connector using a generative AI prompt to summarize the contract text or extracted data. For example, a "Summarize Contract" action could take the full text or key points and return a formatted summary. Instruct it with a prompt like, "Draft a concise summary of the contract, including parties, duration, value, and any clauses related to termination or dispute" • Using the new Prompt Builder features, you can even enforce an outline or JSON structure in the summary (AI Builder JSON Structured Outputs). This action can be called by the agent right after extracting the data, giving the agent a polished summary to deliver. • Whatever your approach, guide the AI by providing some context or template. Include a brief system or user message in the conversation or action prompt listing the information that should be included in the agent's summary. This helps ensure the agent doesn't omit critical details.



Pro Tip

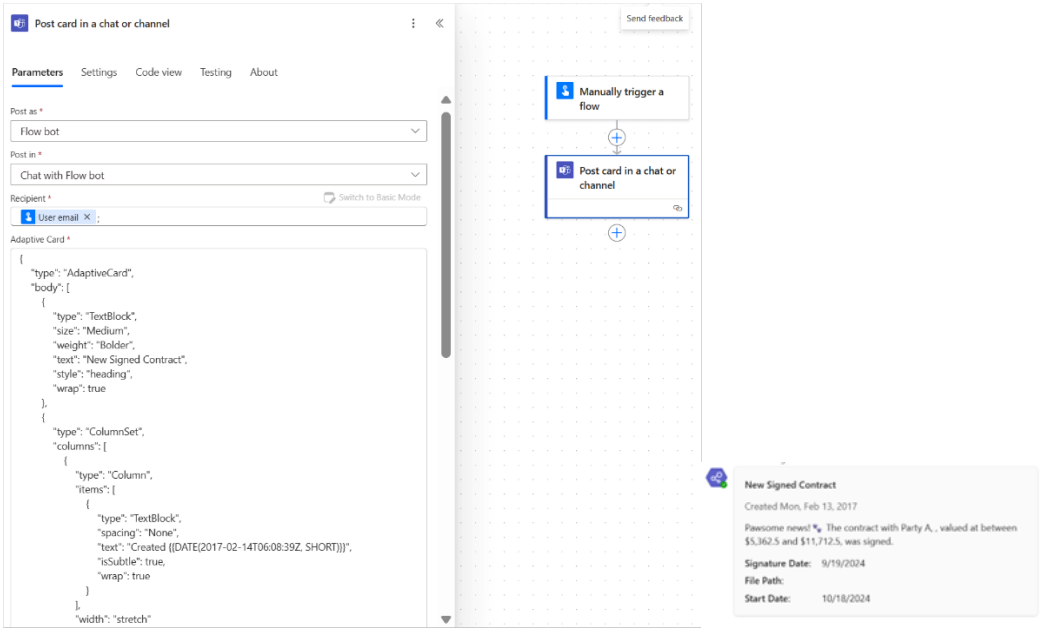
Your Power Fx skills can be used to further refine the summarization process. You can use Power Fx formulas or Flow expressions to manipulate the extracted data before it's used for summarization, or to format the summary output in specific ways.

Step 5: Store and Distribute the Summary Report

After the agent generates the contract summary, decide how you'll output and share it. One approach is to post an Adaptive Card in Microsoft Teams containing the key points and any relevant metadata. This provides timely visibility to your colleagues — right in Teams — without needing to sift through emails or portals.

You can also maintain a permanent record of each contract analysis by storing the summary in a SharePoint list or Dataverse table. For example, create a SharePoint list called "Contract Summaries," with columns for each key field plus a space for the full narrative. The agent (via flows or connectors) can automatically add a new item every time a contract is processed, giving stakeholders a centralized view of all contract summaries.

Tools	Tips & Best Practices
Power Automate (Adaptive Cards in Teams) SharePoint list or Dataverse table (with flows or connectors)	Design a flow that takes the agent's summarized output and posts an Adaptive Card to a Teams channel or chat, so your colleagues can quickly see the key highlights and take any follow-up actions. Use Dataverse if you plan to build further workflows or Power Apps with the data; it's easier due to Dataverse's seamless integration with Copilot Studio agents. Additionally, notify via Teams or email where a timely notification is required.



Pro Tip

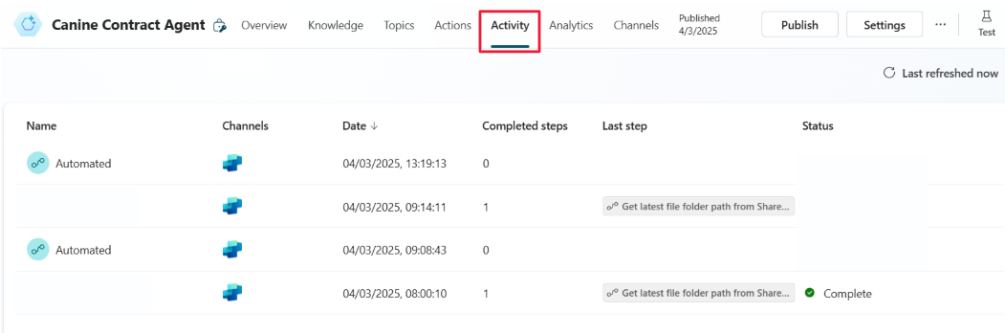
You can use your flow expertise to post Adaptive Cards, create SharePoint list items, or write to Dataverse tables — whatever fits best with your organization's workflow. Think of this as building the "delivery system" for your agent's insights, making sure every contract summary reaches the right people in the right place.

Step 6: Test and Refine the Agent Workflow for Accuracy

Conduct end-to-end testing using a variety of sample contracts. Send a few contract emails (or simulate the trigger) and observe the agent's behavior. Verify that the extracted values match the contract and the summary is accurate and well-structured.

Tools	Tips & Best Practices
Copilot Studio (test canvas) AI Builder	• In Copilot's Studio's test canvas, you can monitor the activity map to see which actions the agent is choosing and in what order (Review Agent Activity). Ensure that your agent is choosing the right actions in the right order.

- If you find errors in extraction, consider retraining the AI Builder model with additional samples or adjusting field definitions.
- If the summary missed certain elements, refine the prompt or add clarifications in the agent's instructions. Also, double check the descriptions for actions and topics. Generative orchestration is designed to handle multi-step queries and even multiple intents (Orchestrate agent behavior with generative AI), but providing high-quality descriptions and context is key to guiding the AI.
- Make use of AI Builder confidence scores; for example, if a crucial field like "Contract Value" has low confidence, review that or have the agent include a note in the summary indicating uncertainty.
- Incorporate user and stakeholder feedback – perhaps you'll identify new fields to extract or clauses to flag. Continuous improvement ensures the agent remains accurate and effective as formats and requirements evolve.



Pro Tip

Your Power Platform testing and debugging skills are important for validating your autonomous agent. Use the testing tools in both Copilot Studio and Power Automate to thoroughly validate your agent's behavior and identify any areas for improvement. Think of this as applying quality assurance practices to your AI agent creation.

4. Deployment and Ongoing Management: Ensuring Agent Health, Security, and Scalability

Deploying your autonomous contract review agent is a significant milestone, but it's just the beginning of its lifecycle. To realize the full potential of your agent and ensure its long-term success, you need a strategy for monitoring, compliance, security, and scalability. This section provides a detailed guide to these critical aspects, drawing on the latest Copilot Studio features and best practices. Keep these guidelines in mind regardless of your particular use case.

4.1. Monitoring & Issue Detection: Keeping a Close Watch on Your Agent

Effective monitoring is crucial for autonomous Copilot Studio agents to quickly detect failures and performance issues. A proactive monitoring strategy minimizes downtime and ensures your agent consistently delivers value.

- **Native Copilot Studio Monitoring:** Leverage Copilot Studio's **built-in monitoring features** for insights into agent behavior.

    - **Action History Tracking:** Utilize the **action history tracking** feature, which logs every step your agent takes. This detailed log includes system interactions, API calls, and topic traversals, providing a granular view for troubleshooting and analysis. This built-in history allows you to replay agent runs, step-by-step, to pinpoint exactly where issues occur. (Action usage)

- **Azure Application Insights Integration:** For more comprehensive monitoring, **integrate your agent with Azure Application Insights**, a component of Azure Monitor.
    - **Telemetry Capture:** Copilot Studio can send telemetry data to Application Insights, including incoming and outgoing messages, triggered topics, custom events, and performance metrics (Capture telemetry with Application Insights - Microsoft Copilot Studio | Microsoft Learn).
    - **Advanced Analytics:** Application Insights enables analytics. Use **Kusto queries** to delve into conversation logs, analyze error patterns, and identify trends (Capture telemetry with Application Insights - Microsoft Copilot Studio | Microsoft Learn).
    - **Custom Dashboards:** Build **custom dashboards** in Azure Monitor to visualize agent performance indicators, error rates, and user interaction patterns, tailored to your specific needs (Basics of visualizing data in Azure Monitor).
    - **Real-time Performance Tracking:** Gain **real-time visibility** into agent performance, allowing you to identify and react to performance degradation quickly (Integrating Azure Application Insights with Microsoft Copilot Studio for Enhanced Monitoring and Error Logging).
    - **Alerting:** Set up **Azure Monitor alert rules** on telemetry data. Configure alerts to notify administrators via email or other channels if critical events occur, such as exceptions, response time spikes, or error rate increases (Copilot Studio Security | Power Platform Developer Blog).

- **Power Platform Flow Analytics:** If your agent utilizes Power Automate flows for actions (as is common), leverage the **Power Platform's built-in monitoring for flows.**
    - **Flow Run History:** Access **real-time flow run logs and error details** directly within the Power Automate portal to troubleshoot individual flow executions.
    - **Power Platform Admin Center Analytics:** Utilize the **Power Platform Admin Center's Analytics reports** for a broader, environment-wide view of flow performance (View analytics for Power Automate cloud flows - Power Platform | Microsoft Learn).
    - **Error Reports:** Specifically, use the **Errors report** in the Admin Center to identify recurring flow error types, failure counts, and last failure times, helping pinpoint problematic flow actions (View analytics for Power Automate cloud flows - Power Platform | Microsoft Learn).
    - **Flow Error Handling & Notifications:** Implement **error handling within your flows** (retry policies, catch blocks) to improve robustness. Configure **failure notifications** in Power Automate to automatically alert flow owners when errors occur.

Pro Tip: Combine Copilot Studio's agent-level monitoring with Power Platform's flow-level monitoring for a complete picture of your autonomous agent's health. Think of it as monitoring both the "brain" (Copilot Studio agent) and the "muscles" (Power Automate flows) of your automated contract review process.

- **Automated Alerts and Diagnostics:** Proactively catch issues by setting up **automated alerts and diagnostics** across your monitoring layers.
    - **Application Insights Alerts:** Configure Azure Monitor alerts (based on Application Insights telemetry) for abnormal spikes in agent errors, slow response times, or other critical performance indicators.

- **Power Automate Admin Flows:** Create scheduled Power Automate administrative flows to periodically check for failed flow runs in the past hour or day and automatically notify support teams if failures are detected.

Pro Tip: Go beyond simply viewing logs – automate issue detection and notification. Proactive alerting ensures timely awareness of problems and can take swift action to resolve them, minimizing any impact on contract processing.

By implementing this multi-layered monitoring approach – combining Copilot Studio's native logs, Azure Monitor/Application Insights telemetry, and Power Platform flow analytics – you establish a robust system for proactively detecting and addressing any performance degradation, exceptions, or unexpected behavior in your autonomous contract review agent. This ensures high availability and consistent performance for your autonomous agents.

4.2. Compliance & Security Monitoring: Ensuring Compliance and Security

Maintaining strict compliance and robust security for sensitive contract data processed by your Copilot Studio agent is non-negotiable. Implement a comprehensive security framework encompassing access control, data protection, and auditability.

- **Role-Based Access Control (RBAC) and Authentication:** Enforce **strong authentication and role-based access control (RBAC)** at every level of your agent solution.
  - **Azure AD Authentication: Require Azure AD login** for any user interacting with your Copilot Studio agent, ensuring verified user identities and preventing unauthorized access (Configure user authentication in Copilot Studio).
  - **Agent Sharing and Deployment Control:** Share agents only with **authorized users or teams**. Deploy agents to controlled environments like dedicated Microsoft Teams channels or via Teams admin consent to restrict access to approved departments. (Agent sharing)
  - **Dataverse Security Roles:** Leverage Dataverse security roles to implement **fine-grained data access control**, restricting permissions at the environment, table, row, and even column level (Access controls for Dataverse and Power Platform - Microsoft Cloud for Sovereignty | Microsoft Learn). Ensure the agent's service account or managed identity has only the minimum necessary permissions to access contract data (Security role overview).

- **Data Protection and Encryption:** Leverage built-in Power Platform and Azure security features to protect contract data.
  - **Encryption at Rest and in Transit:** Dataverse and SharePoint, where contract data is likely stored, provide **default encryption at rest and in transit**, protecting data both while stored and during transmission (Access controls for Dataverse and Power Platform - Microsoft Cloud for Sovereignty | Microsoft Learn).
  - **Column-Level Security and Field Masking (Dataverse):** For highly sensitive contract fields (e.g., financial details, personal information), implement **column-level security or field masking** in Dataverse to further restrict access, even for authorized users or the agent's service account. Apply these concepts to your own unique use case (Field level security to control access).
  - **SharePoint Permissions and Sensitivity Labels:** Secure SharePoint sites used as knowledge sources with **strict permissions**, controlling who can access contract documents. Consider using **Microsoft Purview Information Protection sensitivity labels** to classify and encrypt contract files, adding an extra layer of protection and control. Your autonomous agent will respect all information protection labels so long as you implement them (Enable sensitivity labels for files in SharePoint).

- **Audit Logging and Security Monitoring:** Implement audit logging and integrate with security monitoring tools for proactive threat detection and compliance reporting.
    - **Microsoft Purview Audit Logs:** Enable **Microsoft Purview Audit logs** to capture a detailed record of agent usage and administrative actions, including agent creation, deletion, configuration changes, topic updates, and user interactions (Microsoft Purview setup guides).
    - **SIEM Integration (Microsoft Sentinel):** Integrate Purview audit logs with **Microsoft Sentinel** (or your organization's SIEM) for real-time security monitoring and threat detection (Integrate Sentinel and Purview). Set up custom alerts in Sentinel to detect anomalies like unusual agent access patterns or suspicious query volumes.
    - **Data Retention Policies (Microsoft Purview):** Utilize Microsoft Purview to define **data retention policies for Copilot interactions**, separate from standard Teams chat logs, to comply with data retention regulations for contract-related data. While this may not be as important for autonomous agents given that humans won't be interacting with them directly, you'll still want to familiarize yourself with this capability (Learn about retention for Copilot and AI apps).

- **Data Loss Prevention (DLP) Policies:** Enforce **strict Data Loss Prevention (DLP) policies** in Power Platform to govern data flow and connector usage within your Copilot Studio agent.
    - **Connector Classification:** Classify connectors used by your agent (Power Automate connectors, knowledge source connectors) as "Business" or "Non-Business" in DLP policies. Your autonomous agent will respect all connector classifications (Connector classification).
    - **Prevent Data Exfiltration:** Create DLP policies to **prevent sensitive contract data from being inadvertently shared with unauthorized services or external systems** via connectors (Configure data loss prevention policies for agents). For example, block the agent from sending contract details via unapproved email connectors or social media.
    - **Enforce DLP Compliance: Strictly enforce DLP policies** in the Power Platform environment hosting your agent. Use the Power Platform Admin Center or PowerShell to ensure DLP policies are active and effectively govern Copilot Studio connectors (Turn on DLP enforcement for agents).

Pro Tip: Implement a layered security approach, combining access controls, data protection, audit logging, and DLP policies to create a robust security posture for your autonomous contract review agent.

By diligently implementing these compliance and security monitoring measures – RBAC, data protection, audit logging, and DLP – you establish a framework to safeguard confidential contract data, maintain regulatory compliance, and proactively detect and respond to any security threats or anomalies.

4.3. Scaling & Optimization: Handling High Contract Volumes

As your organization increasingly relies on your autonomous agent, scalability and performance optimization become more important. Ensure your agent can handle high volumes efficiently and reliably. These practices can apply to lots of use cases, not just contract analysis.

- **Performance Tuning and Efficient Design:** Optimize your agent's design for performance from the outset.
    - **Streamline Agent Logic:** Design efficient conversation flows and action sequences within Copilot Studio. Minimize unnecessary steps or complex topic traversals to reduce processing time.

    - **Solution Management and ALM:** Utilize Copilot Studio's **Solution Management and Application Lifecycle Management (ALM) features** to promote well-tested, optimized agent versions from development to production (Create and manage solutions in Copilot Studio)
    - **Performance Profiling with Application Insights:** Use **Application Insights** to identify performance bottlenecks. Measure response times for topics and actions, pinpointing slow API calls or computationally intensive steps (Optimizing custom copilot (agent) performance with Azure Load Testing: A comprehensive guide - Forward Forever).
    - **Offload Heavy Processing:** For computationally intensive tasks (like analyzing large contract documents), **offload processing to external services** like Azure Functions or Logic Apps (invoked via Power Automate). Pre-process documents in parallel and have the agent retrieve pre-analyzed results.
    - **Optimize Power Automate Flows:** Design Power Automate flows triggered by the agent for efficiency. **Enable concurrency** in flows where safe to allow parallel processing of multiple contracts. Avoid unnecessary loops or resource-intensive operations within flows. (Implement parallel execution and concurrency)

- **Capacity Planning and Quota Management:** Plan for sufficient capacity and manage Copilot Studio quotas as your contract volume grows.
    - **Copilot Studio Message Quotas:** Be aware of Copilot Studio's **message rate quotas** (currently a default of 8,000 messages per minute per environment) (Quotas and limits - Microsoft Copilot Studio | Microsoft Learn). While generous, monitor usage and request quota increases from Microsoft support if you anticipate exceeding these limits in high-volume scenarios.
    - **Message Capacity Planning: Estimate message consumption** based on your contract processing volume and the complexity of your agent's actions. Factor in that autonomous actions and generative AI features consume more messages (Manage message capacity - Microsoft Copilot Studio | Microsoft Learn).
    - **Message Pack Acquisition and Pay-as-you-go: Plan for sufficient message capacity** by acquiring additional message packs or enabling **pay-as-you-go billing** (now available for Copilot Studio) to handle peak loads and growth without service interruptions (Setup pay-as-you-go).
    - **Power Platform Admin Center Monitoring: Regularly monitor Copilot Studio message consumption** in the Power Platform admin center to track usage against your capacity and set up alerts for approaching limits (Viewing message usage).

- **Continuous Optimization:** Embrace a mindset of **continuous optimization** for long-term scalability and efficiency.
    - **Telemetry-Driven Optimization:** Use monitoring telemetry data to **identify performance trends and bottlenecks**. Focus optimization efforts on areas where the agent is slowest or least efficient.
    - **Iterative Refinement:** Continuously refine your agent's logic, actions, and flows based on performance data and user feedback.
    - **Stay Updated with Platform Enhancements:** Keep abreast of new Copilot Studio features and performance improvements released by Microsoft. Leverage new capabilities like generative AI-driven topic/action selection to enhance agent efficiency (Copilot Release Plans).

Pro Tip: Scaling isn't just about throwing more resources at the problem. It's about smart design, efficient code, proactive monitoring, and continuous optimization. Treat scalability as an ongoing journey, not a one-time fix. By focusing on performance tuning, capacity planning, and leveraging the scalability of Azure and Power Platform, you can confidently scale your Copilot Studio agent to handle even the most demanding processing workloads without sacrificing performance or reliability.

Addressing monitoring, compliance, security, and scalability contributes to developing a reliable and manageable autonomous agent solution suitable for enterprise deployment.

5. The Power Platform Maker's Advantage: Leveraging Power Platform Expertise for Agents

For developers building autonomous agents with Copilot Studio, existing Power Platform skills provide a significant advantage. Knowledge of Power Fx, Power Automate, Dataverse, and the associated ecosystem is highly applicable.

Copilot Studio integrates with and extends Power Platform capabilities, enabling the development of agents that incorporate AI to handle more complex or dynamic automation tasks.

Embrace this opportunity to...

- **Expand your skillset:** Add agent development capabilities to your existing Power Platform skillset
- **Address complex process:** Apply agent technology to automate dynamic or multi-step business processes
- **Deliver advanced solutions:** Build solutions that leverage AI for enhanced automation and task handling.