

Power Automate Desktop Development Best Practices

Speaker Name

Microsoft – Power CAT



Scope of the document

- All features mentioned are available as of January 2024.
- This document will focus on Power Automate Desktop & Desktop Flows.
- All topics are general recommendations and should not be seen as mandatory.

Why best practices?

Development best practices are crucial for ensuring that automated processes are not only efficient and effective but also robust and reliable. These practices are fundamental to the success of RPA initiatives, providing a framework for creating solutions that deliver consistent value to your organization.

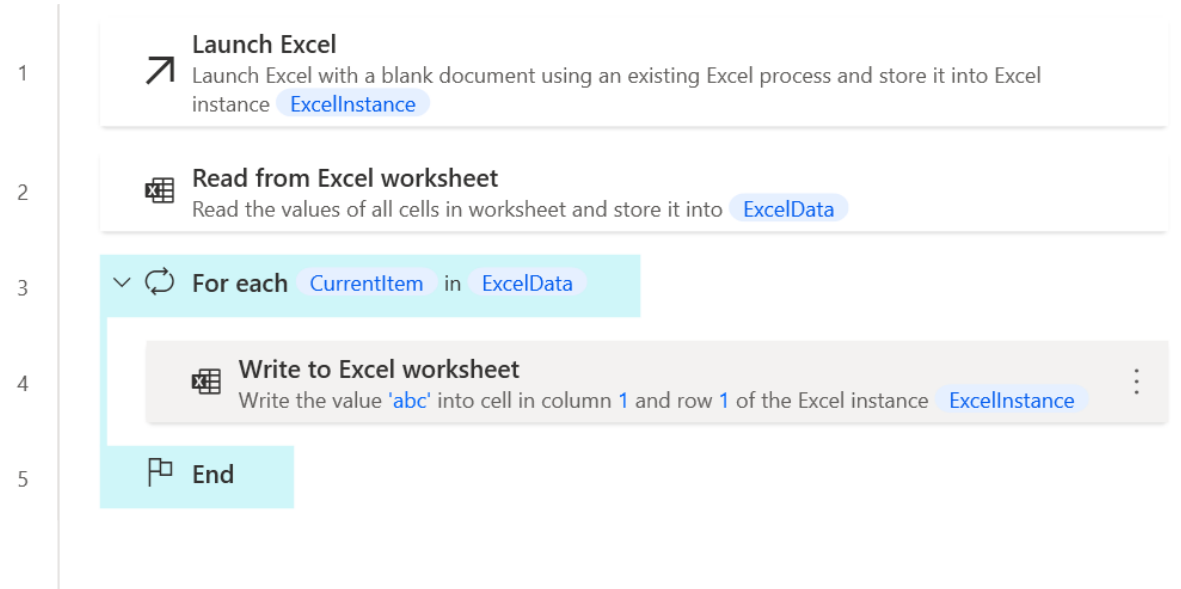
- ✓ **Performance:** Optimizes RPA bots to execute tasks swiftly and accurately, reducing processing time.
- ✓ **Maintainability:** Facilitates easy updates to RPA workflows, accommodating changes in processes.
- ✓ **Security:** Safeguards sensitive data handled by RPA bots against breaches.
- ✓ **Data:** Manages data flow within RPA processes, ensuring accuracy and integrity.
- ✓ **Coding Standards:** Adheres to best coding practices for readability and consistency in RPA scripts.
- ✓ **Supportability:** Ensures RPA solutions are easy to support and troubleshoot, with clear documentation.
- ✓ **Usage:** Monitors how RPA bots are utilized to ensure they meet the organization's needs.
- ✓ **Design:** Involves careful planning of RPA workflows for optimal performance and user experience.



Performance

Avoid loops across many records

- Avoid loops on large datasets, for example excel files with 1000's of records.
- Or loops with a high loop index.
- Try to convert these actions in a script, for example a python or .Net script.



Limit Data Retrieval

- Avoid retrieving more data than necessary.
- If an action only requires a subset of data, use filtering or pagination options provided by connectors to retrieve only the required data. This reduces the amount of data transfer and processing, resulting in improved performance.

Read from Excel worksheet



Reads the value of a cell, a named cell or a range of cells from the active worksheet of an Excel instance

[More info](#)

Excel instance: ⓘ

Retrieve: ⓘ

Start column: ⓘ

Start row: ⓘ

End column: ⓘ

End row: ⓘ

> Advanced

> Variables produced ExcelData

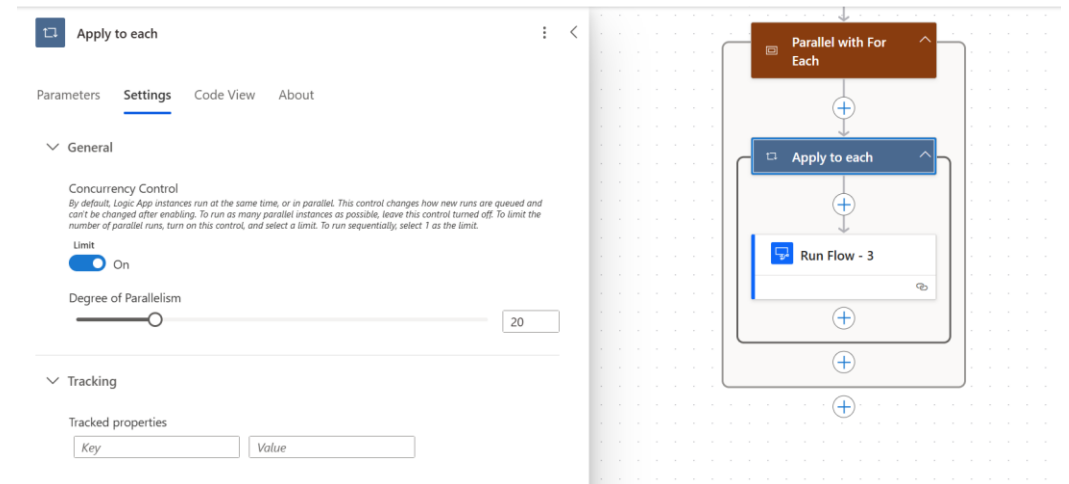
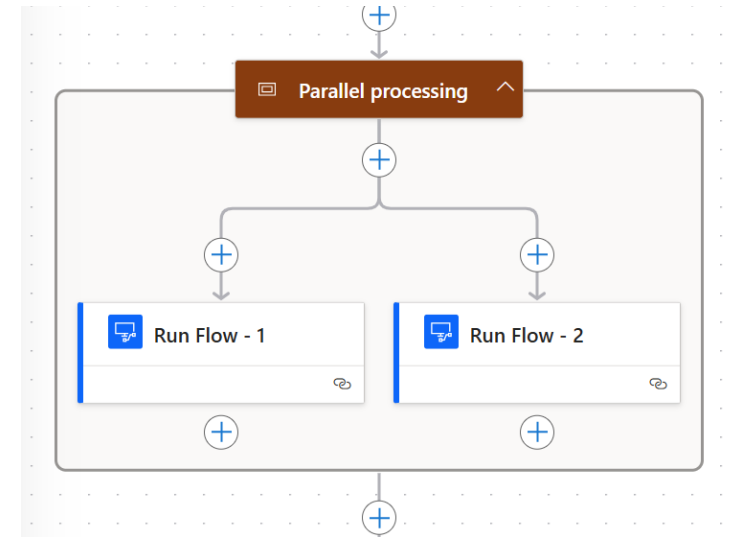
On error

Save

Cancel

Implement parallel processing

- Identify opportunities to parallelize actions or process multiple tasks simultaneously.
- Power Automate provides parallel branches or "Apply to Each" loops to execute actions in parallel.



Work Queues

- Use Work Queues and (Hosted) Machine Groups to scale large workloads of data and enable parallel processing of data.

Work queue items

Name	Enqueue time	Status	Expires	Value	Processing notes
005	Jul 24, 04:47 PM	Queued	Jul 29, 04:47 PM	{ "InvoiceId": "I-1010", "InvoiceDate": ...	—
004	Jul 24, 04:47 PM	Queued	Jul 29, 04:47 PM	{ "InvoiceId": "I-1009", "InvoiceDate": ...	—
003	Jul 24, 04:24 PM	Queued	Jul 29, 04:24 PM	{ "InvoiceId": "I-1008", "InvoiceDate": ...	—
002	Jul 24, 04:24 PM	Queued	Jul 29, 04:24 PM	{ "InvoiceId": "I-1007", "InvoiceDate": ...	—
001	Jul 24, 04:23 PM	Processing	Jul 29, 04:23 PM	{ "InvoiceId": "I-1006", "InvoiceDate": ...	—

Create new hosted machine group

Step 3 of 6

Select bots usage

Bots in a group are created when needed. Whenever a desktop flow waits in the queue and no bot is available, a bot is created automatically.

A bot is created as long as the maximum number of bots for this group isn't reached and you have enough hosted RPA add-ons assigned to your environment.

[Read more information about the licensing requirements](#)

Select the committed and maximum number of bots

Define the committed and maximum number of bots you want to assign to this group.

Note

- **Max Bots** allows your hosted machine group to scale beyond the number of Committed Bots, based on the available capacity in your environment. The Max Bots setting offers flexibility to handle increases in workload or unexpected demand surges
- **Committed Bots** (preview) guarantees the number of available bots during auto-scaling. This ensures that the necessary resources will always be available for processing your flows, providing consistent performance for your critical flows.
- Go to the **Capacity Utilization** page to view and manage your hosted RPA capacity utilization within your environment.

Maximum number of bots

5 bots

Committed number of bots

2 bots

Back Next

Process work queue items

Indicates to the orchestrator that the machine is ready to process one or more work queue items, if available [More info](#)

Select parameters

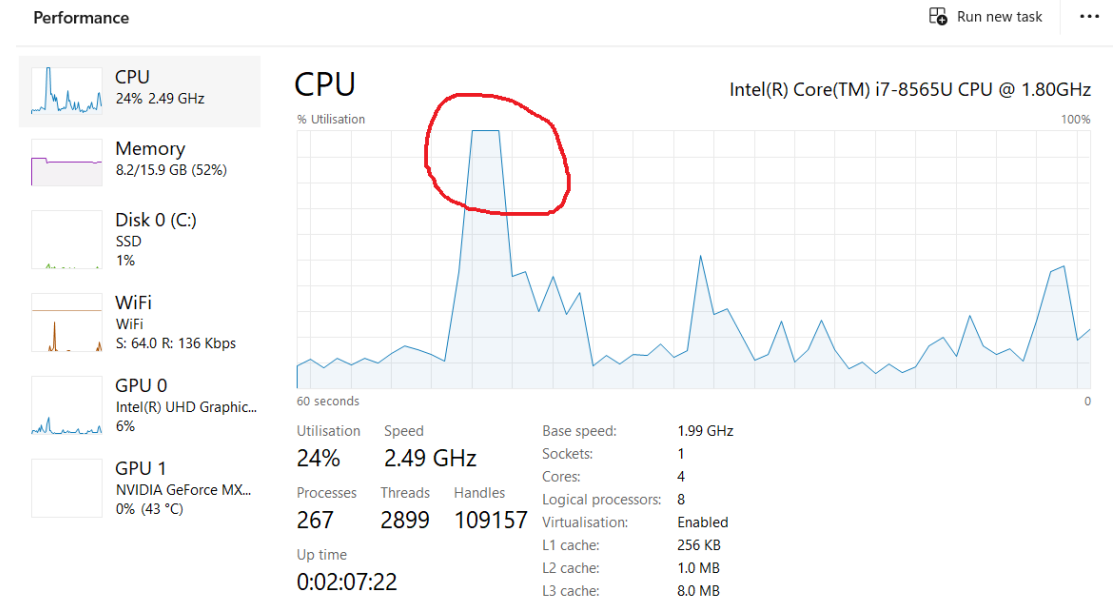
Work queue: Demo PAD Queue {x} ⓘ

> Variables produced: WorkQueueItem

On error Save Cancel

Monitor and optimize resource usage

- Keep track of the resource usage of your flow, such as memory, CPU, or network bandwidth.
- If your flow is resource-intensive, consider optimizing resource allocation or adjust to ensure efficient resource utilization.
- Especially for long running processes



Test and iterate

- Test your flow with large datasets, high transaction volumes, or under stress conditions to identify performance bottlenecks.
- Use the results of performance testing to iterate and refine your flow, implementing optimizations where necessary.

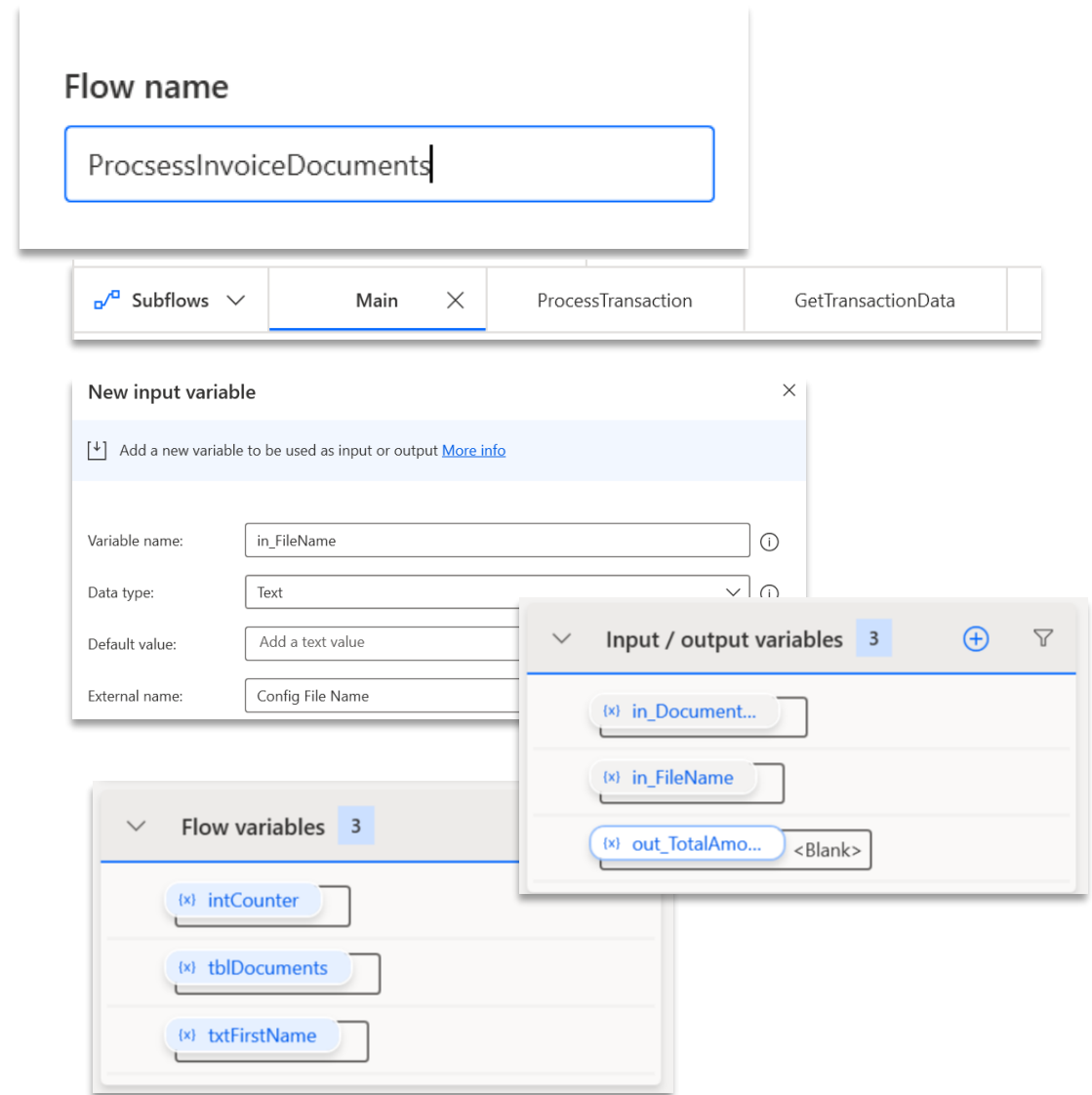




Coding Standards

Naming Conventions

- To develop clear and well-defined flows.
- Use [Camel Case](#) and/or [Pascal Case](#).
- Use naming conventions for:
 - Flows
 - Sub flows
 - Variables (Input, output and flow variables)
 - Ui Elements
 - Connection Reference Names
- Add prefix in or out to Input and Output variables.
- Add a datatype prefix to variable names.
- Use underscore if it makes your variables more readable.
- Be as concise as possible.
- Add clear and understandable external names for the input and output variables.
- Sub Flows should contain the verb describing what the flow does (besides Main)
- A connection reference name starts with a noun to describe the connection account, followed by the connection type, the solution name and a unique identifier.



Sub Flows

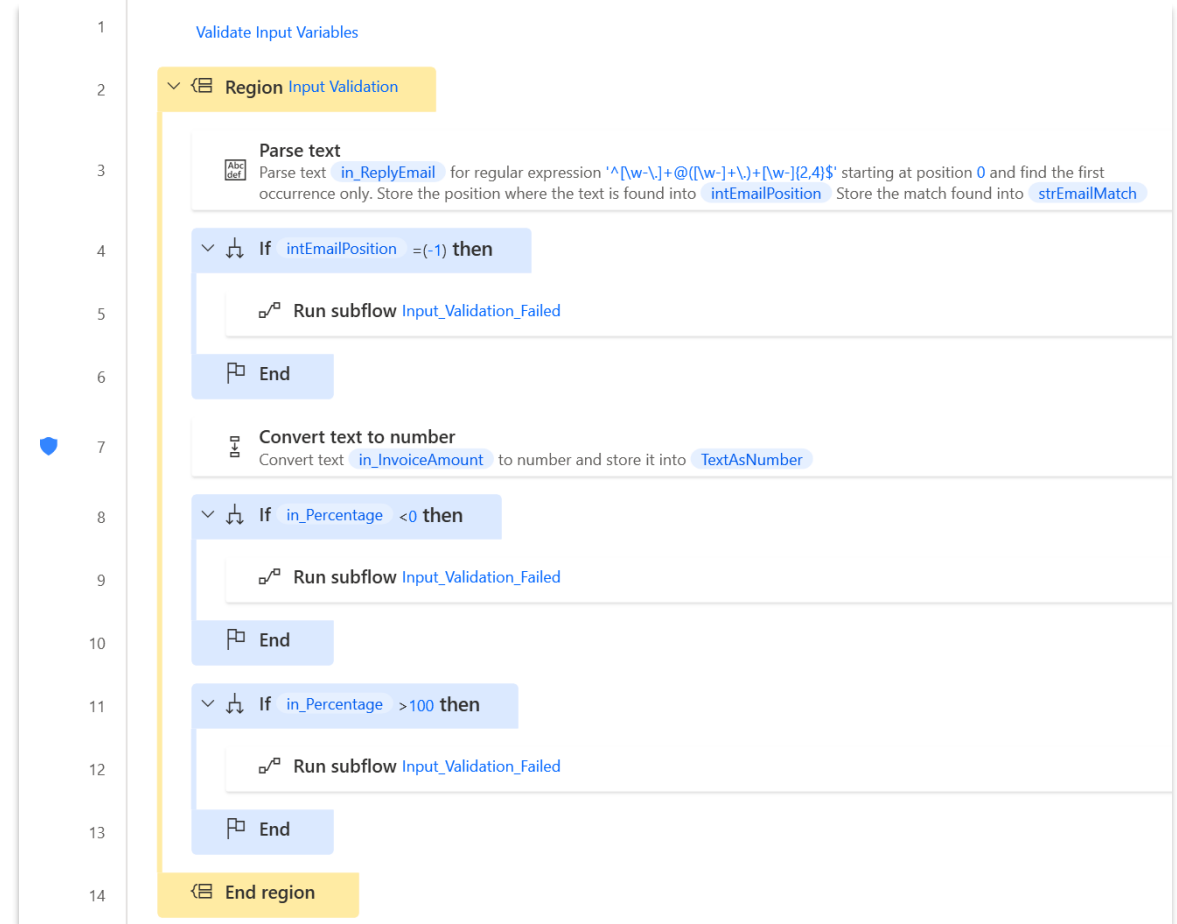
- Sub Flows are groups of actions, which may be referenced as a group in a desktop flow.
- They are a reusable set of actions within a desktop flow.
- Every Flow contains one Main sub flow, which is the start of the desktop flow.
- Identify repeatable actions and put them in a sub flow.
- If a sub flow is built in a customized way, then it can be reused across other Desktop Flows too by copy pasting into them.

The image shows a software interface for managing subflows. At the top, a horizontal pane titled 'Subflows' contains four tabs: 'Main', 'ProcessTransaction', and 'GetTransactionData'. The 'Main' tab is currently selected. Below this pane are two dialog boxes. The first dialog, titled 'Add a subflow', includes a tip about segmenting flows into subflows, a text input field for 'Subflow name' containing 'Subflow_1', and 'Save' and 'Cancel' buttons. The second dialog, titled 'Run subflow', includes a tip about running subflows with arguments, a section titled 'Select parameters' with a dropdown menu for 'Run subflow:', and 'Save' and 'Cancel' buttons.

Subflows	Main	ProcessTransaction	GetTransactionData
<div><h3>Add a subflow</h3><p>Segmenting a flow into subflows makes it easier to manage—especially if it has a lot of actions or complexities. More info</p><p>Subflow name: <input type="text" value="Subflow_1"/></p><p><input type="button" value="Save"/> <input type="button" value="Cancel"/></p></div> <div><h3>Run subflow</h3><p>Run a subflow specifying any required arguments More info</p><p>Select parameters</p><p>Run subflow: <input type="text" value=""/> ⓘ</p><p><input type="button" value="Save"/> <input type="button" value="Cancel"/></p></div>			

Input Validation

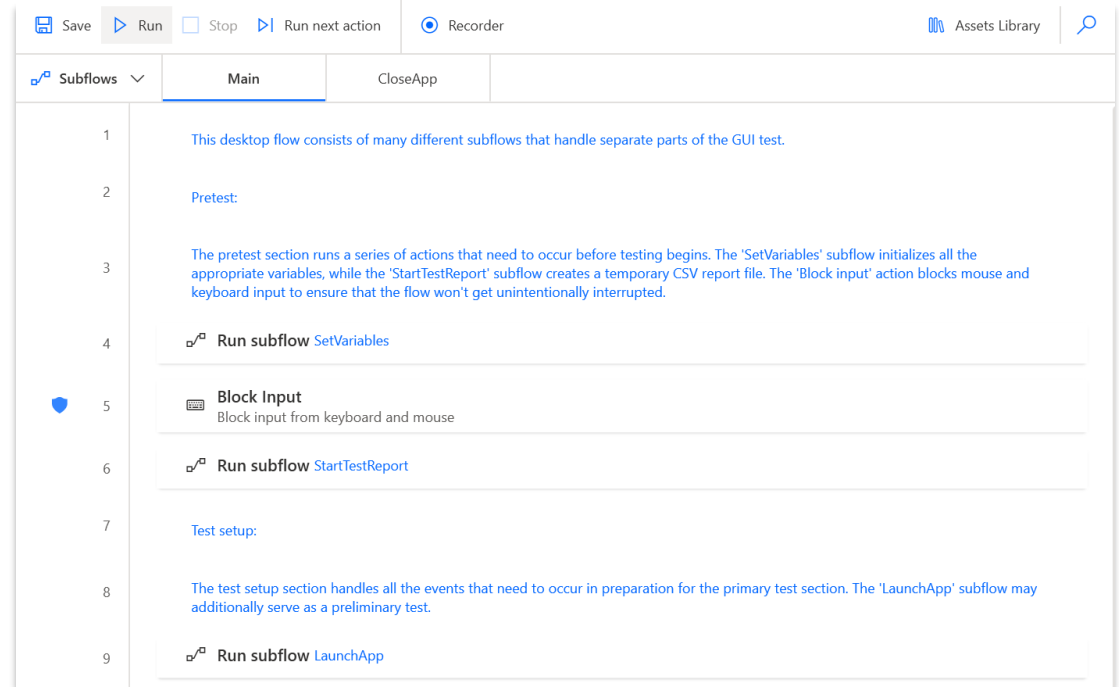
- Validate data format; like email, phone number, specific text patterns, ...
- Validate data type; integer, text, dates, ...
- Validate required fields, check if they are not empty, blank or null.
- Validate ranges, for example a number must fall into a specific range.
- Proper error handling, send error notifications, user friendly error messages.



Supportability

Add Comments in Main and Sub Flows

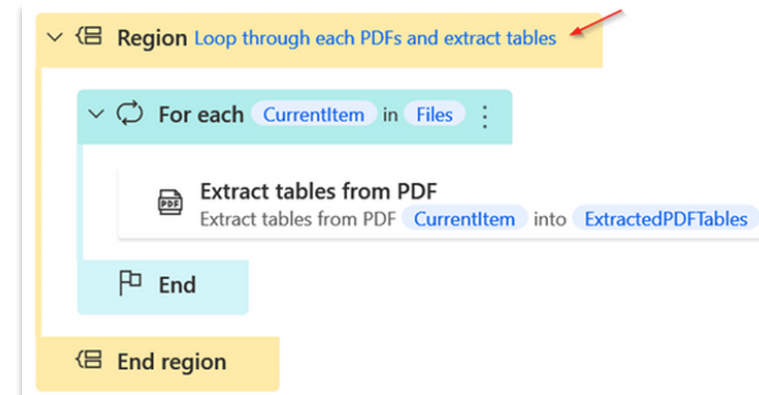
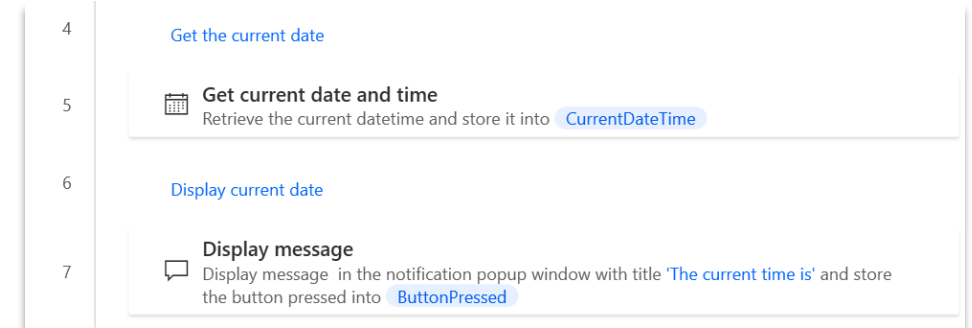
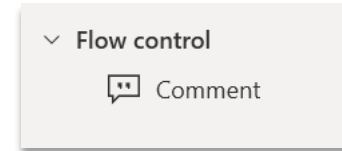
- Start your Main flow with a comment to give a short introduction of the bot: a short description, what is the audience, what is the process, are there any other bots involved (for example a producer, consumer or triggered by a cloud flow, ...).
- Add a comment in each sub flow to briefly introduce the intention of the sub flow.



Comments & Regions

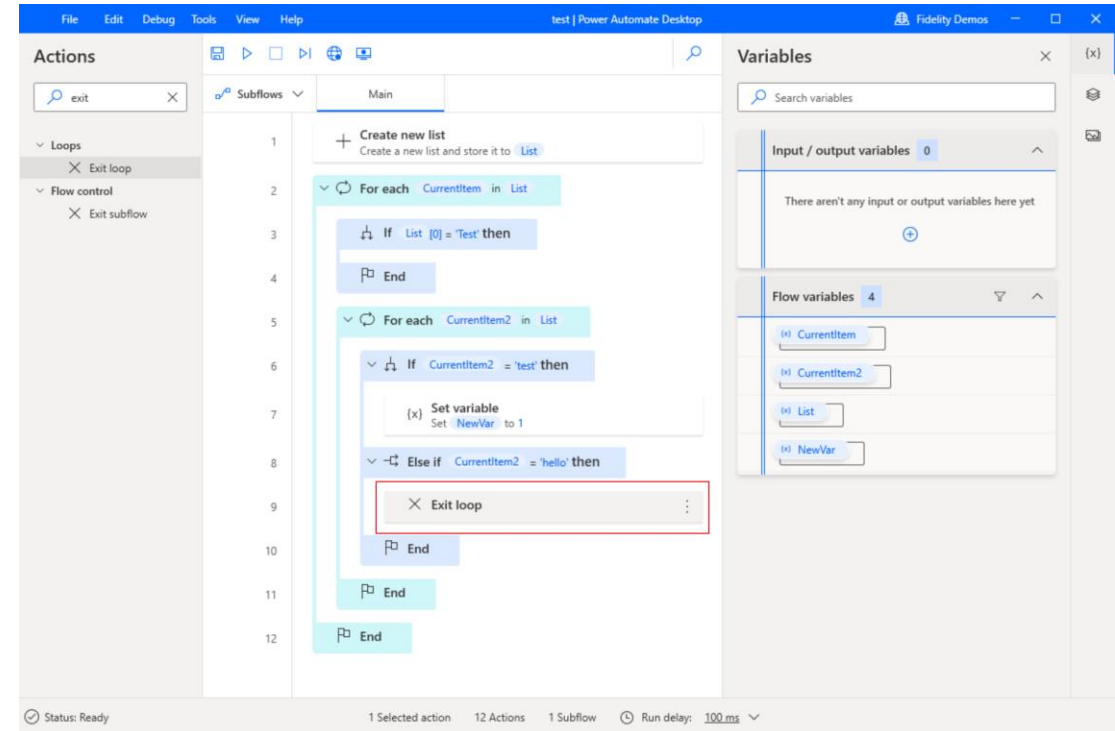
- Add comments to increase code readability.
- Comments will help you to understand the code when you need to review the code after a couple of months or when another developer needs to review the code.
- Add comments when fixing bugs.
- Add a comment at the start of each sub flow to explain the purpose of the flow.
- Regions organize your actions into logical groups that can be collapsed or expanded in the designer.

[Flow control actions reference - Power Automate | Microsoft Learn](#)



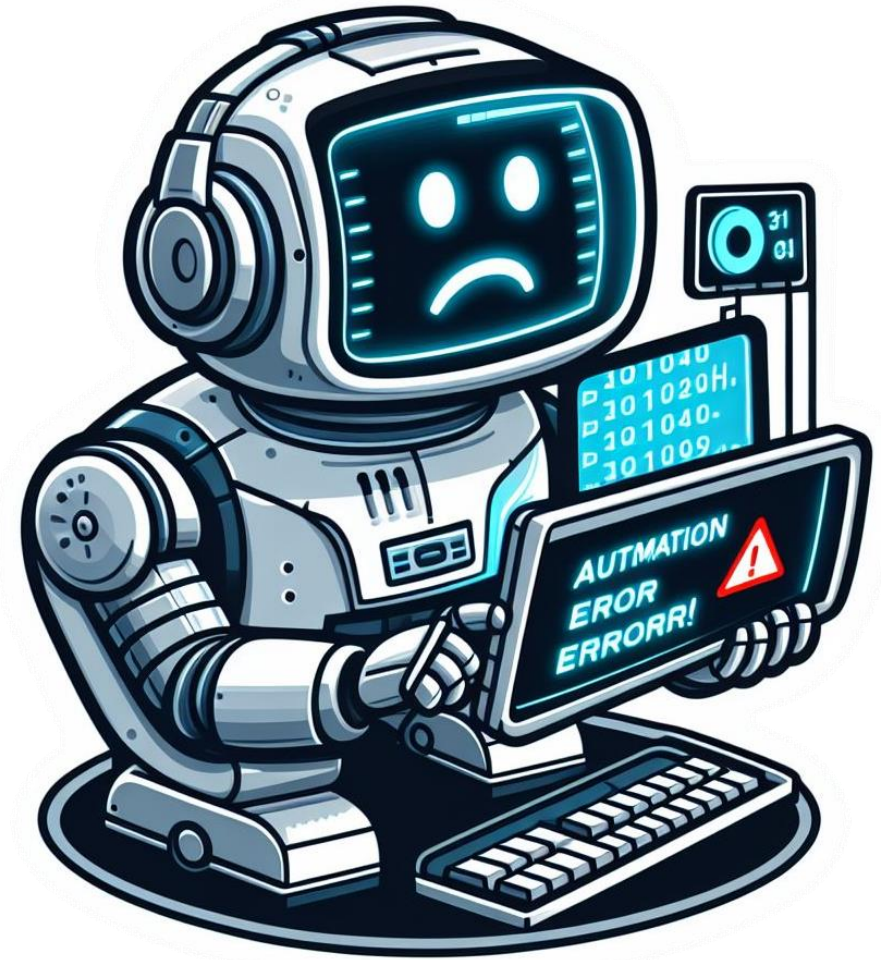
Avoid nesting loops

- Multiple nested loops often make your code hard to read.
- Apply a rule to have a max depth across your flows (for example 3).



Error Handling

- Identify potential points of failure within your flows.
- Error handling will make your flows more resilient and reliable.
- Log error messages (Log Message, File, ...)
- 2 types of error handling:
 - Single action
 - Multiple actions



[Handle errors in desktop flows - Power Automate | Microsoft Learn](#)

Single Action

- In-action error handling.
- On Error button per action.
- With rules, variables can be set, and sub flows can be executed.

The screenshot shows a dialog box titled "Launch new Microsoft Edge" with a close button in the top right corner. Below the title bar, a light blue banner contains a shield icon and the text "The following rules will apply if the action fails [More info](#)".

Below the banner, there is a section for "Retry policy" with a dropdown menu set to "None" and an information icon. Below this is a section for "All errors" with a checkmark icon, a "New rule" button, and a "Clear all" button. The text "No rules added" is displayed in the center. Below this, there are two buttons: "Continue flow run" and "Throw error".

Below the "All errors" section, there is a section for "Advanced" with a chevron icon. At the bottom of the dialog, there are three buttons: "Return to parameters", "Save", and "Cancel".

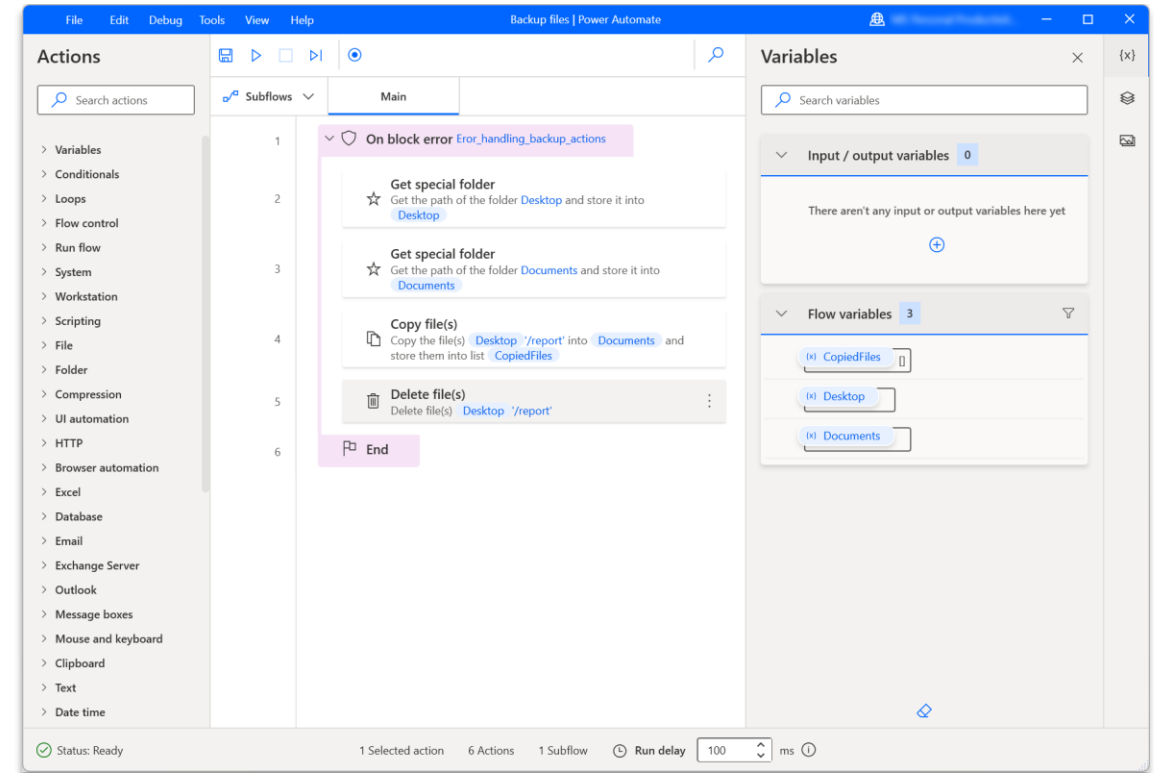
Retries

- Retry the action when it fails.
- Retry policy can be fixed or exponential.
- Retry System errors, for example: cannot launch Edge browser.
- Do not retry Business errors, for example: File path was not found.
- When retry limit is reached, navigate the flow to properly handle the error.

The screenshot shows a 'Copy file(s)' dialog box with a close button (X) in the top right corner. Below the title bar, a light blue banner states: 'The following rules will apply if the action fails [More info](#)'. Below this, the 'Retry action if an error occurs' toggle switch is turned on and is highlighted with a red rectangular box. Underneath the toggle, the 'Times' is set to 1 and the 'Interval' is set to 2 seconds. Below these settings, there is a section for error rules. It shows a dropdown menu set to 'All errors' with a shield icon, and buttons for 'New rule' and 'Clear all'. Below this, it says 'No rules added'. At the bottom of this section are two buttons: 'Continue flow run' and 'Throw error'. Below the error rules section is an 'Advanced' section with a chevron icon. At the very bottom of the dialog are three buttons: 'Return to parameters', 'Save', and 'Cancel'.

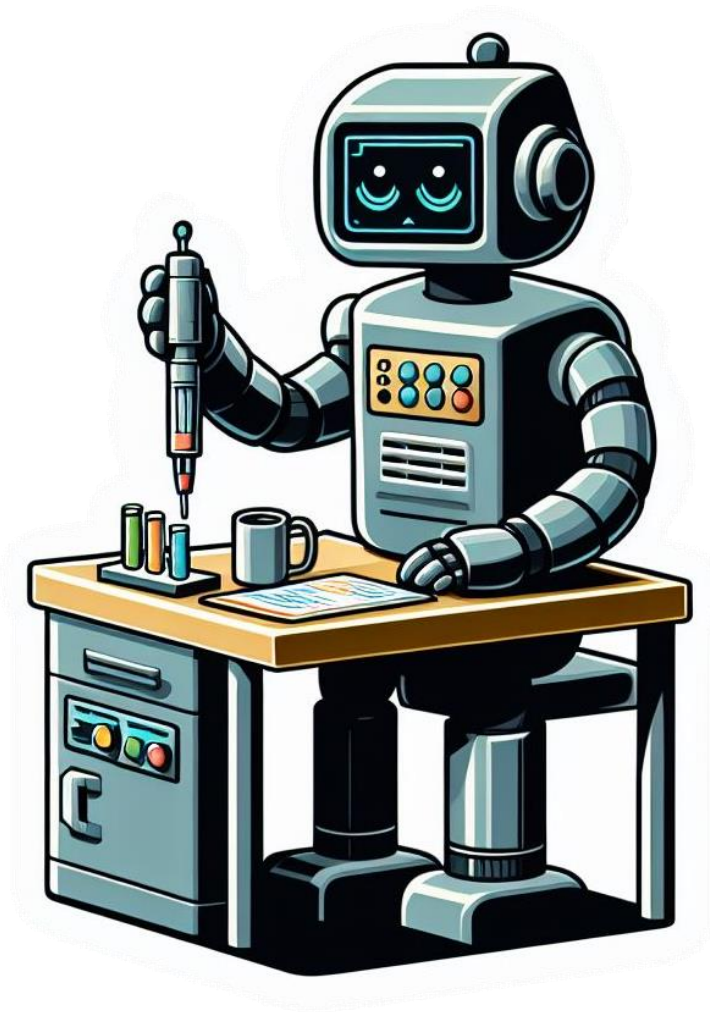
Block of actions

- This action offers almost the same options as the “On error” settings of single actions but also allows you to capture unexpected logic errors, such as trying to access a list item from an out-of-bounds position.
- To retrieve the latest occurred error in a desktop flow and use it in later actions, use the **Get last error** action



Testing

- Test different scenarios.
- Validate test results.
- Test error and exception handling.
- Perform end-to-end testing.
- Test flows performance, for example with large datasets,...
- Do regression testing after upgrading to a new Power Automate Desktop version.

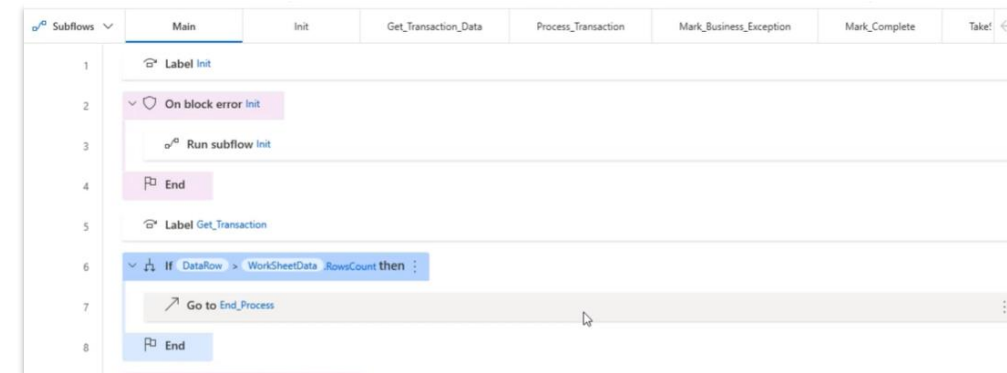
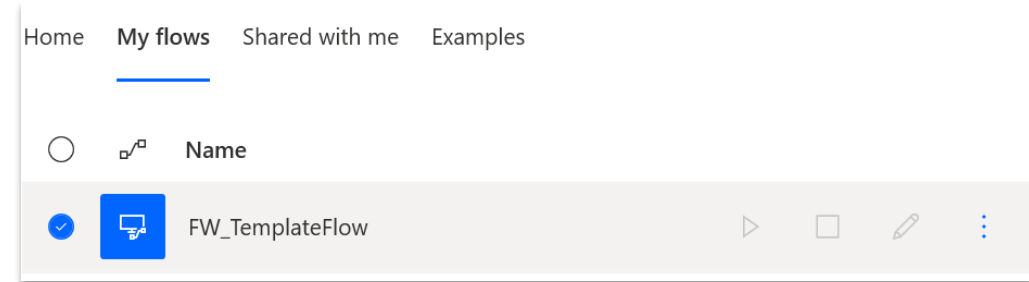




Design

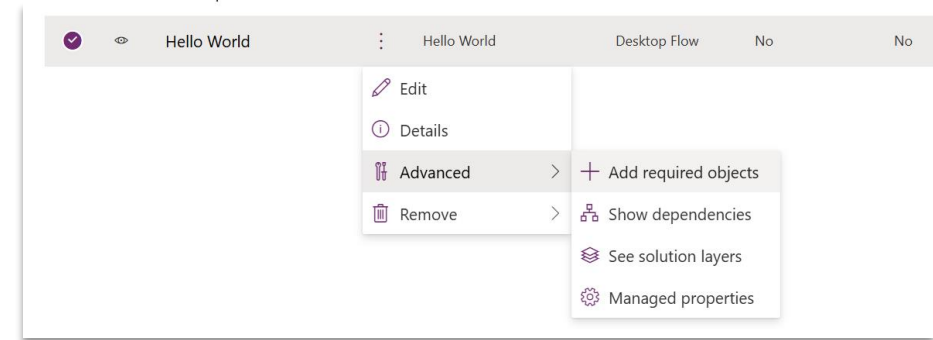
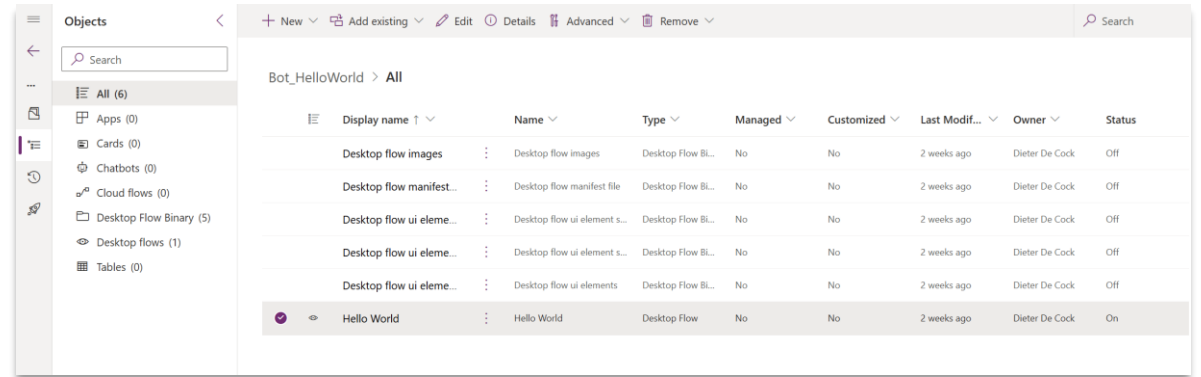
Framework

- Templates will help to standardize desktop flow development.
- Prefix the template desktop flows and make them read only.
- Create a templates that contain:
 - Initialization of the bot
 - Get work queue item
 - Processing work queue item
 - Mark complete
 - Mark Business Exception
 - Mark System Exception
 - Logging & Error handling
 - Take Screenshot



Solutions

- Add Power Automate Desktop flows in a solution.
- Use preferred solution feature ([Set a preferred solution](#))
- Use one solution for one automation process / bot.
- When your flows are in a solution you will automatically use connection references instead of connections.
- Have all your solution artifacts in one place: cloud flow(s), desktop flow(s), connection references, queues, tables, etc.
- Do not forget to include Desktop Flow Binaries in your solution.
- Before exporting your solution, make sure you have all solution artifacts in the solution, use "Add Required Objects"



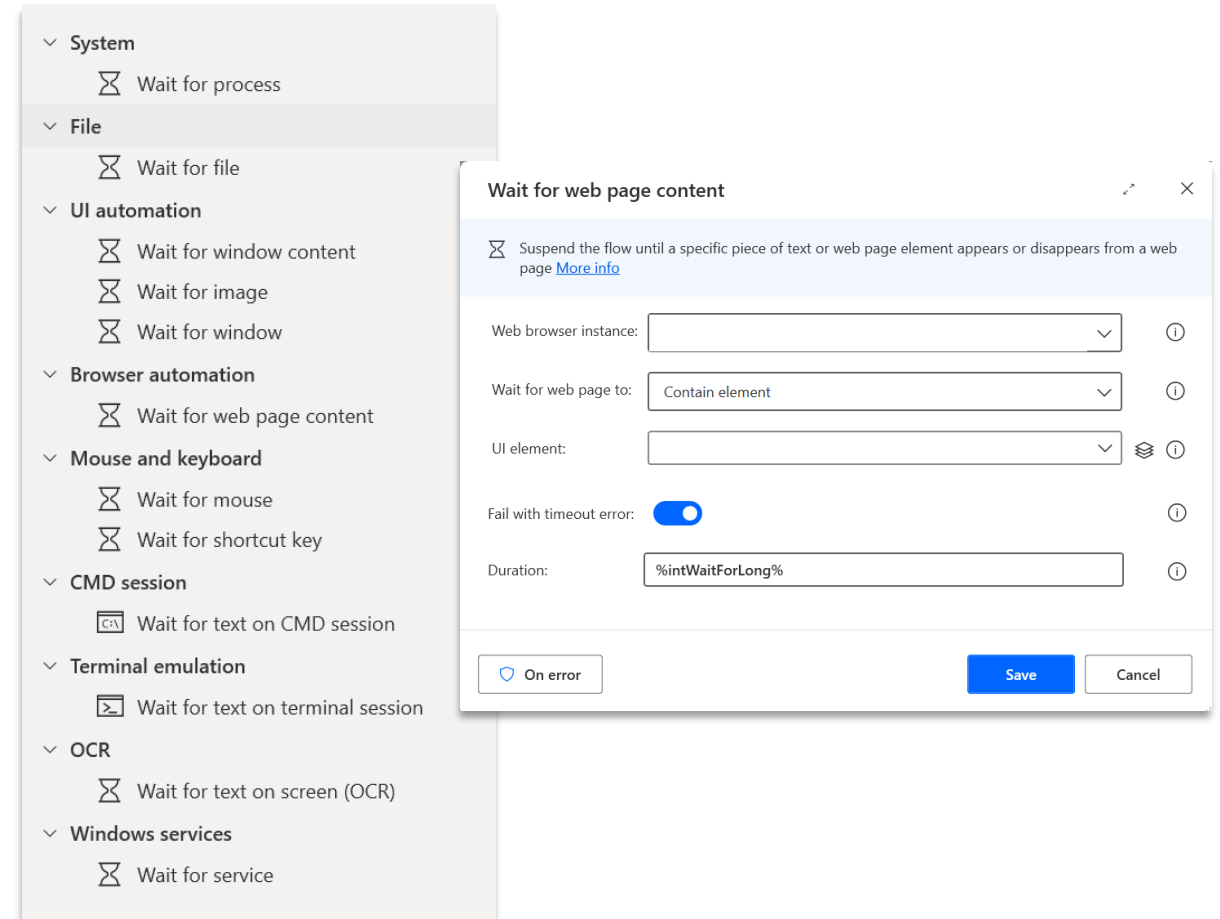
Bot Config

- When a bot is initialized, it usually needs certain parameters to be able to run properly. These parameters can come from input parameters, config files, (Dataverse) tables, etc....
- Popular formats for configuration are text files, JSON, XML, Excel files, ...
- Configuration should only be accessible to the bot.
- Files can be stored on a shared drive, OneDrive, SharePoint.
- In some cases, Windows Environment Variables can be used (has dedicated PAD actions)
- Key Value pairs are easy to read.

	A	B
1	Key	Value
2	SourceFilePath	C:\Test\InputFile.xlsx
3	GenerateReport	TRUE
4	SendEmailTo	abc@email.com; pqr@email.com
5	CompressFileBeforeSending	FALSE

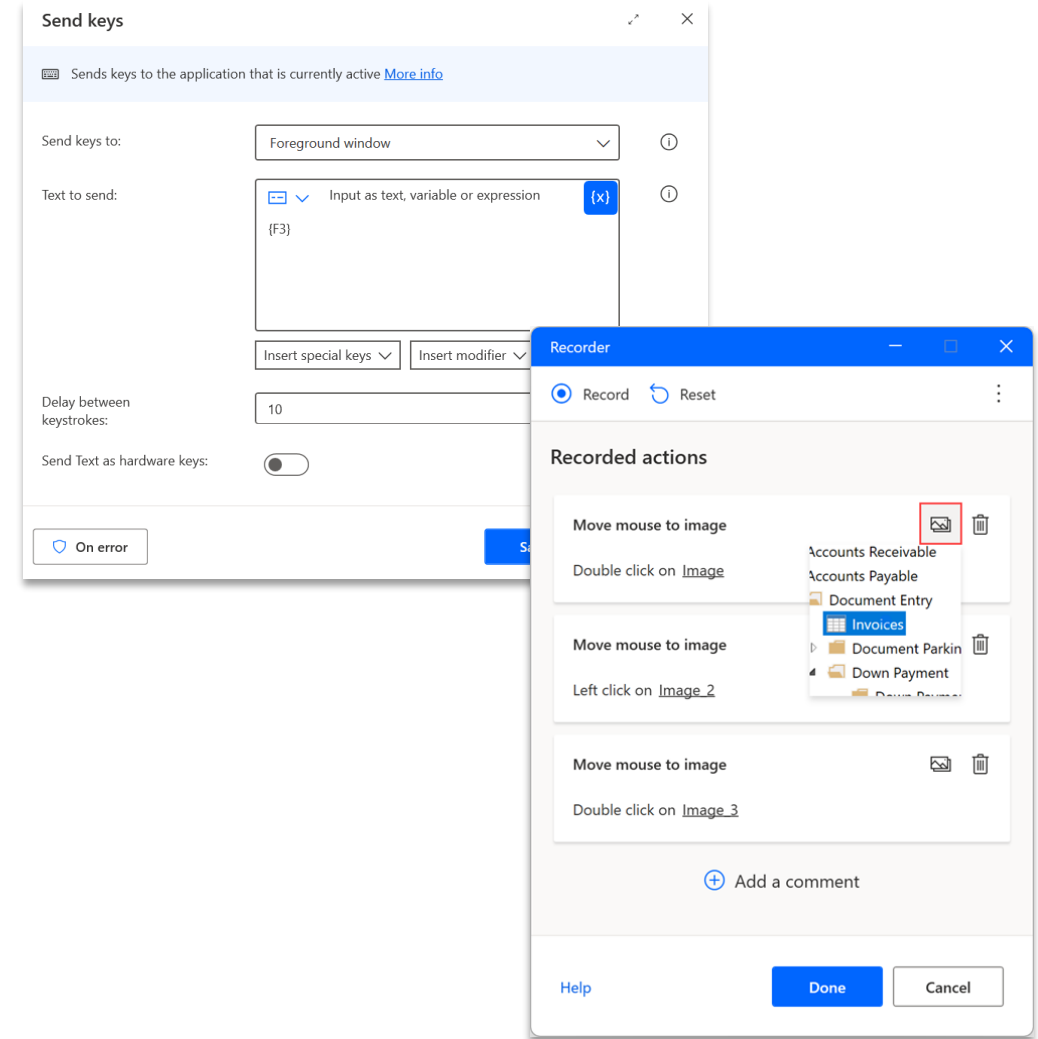
Wait actions

- Use “Wait for ...” actions instead of delays.
- Add a timeout
- And handle the timeout error
- Use global Wait variables to configure the time to wait (might be different per application)



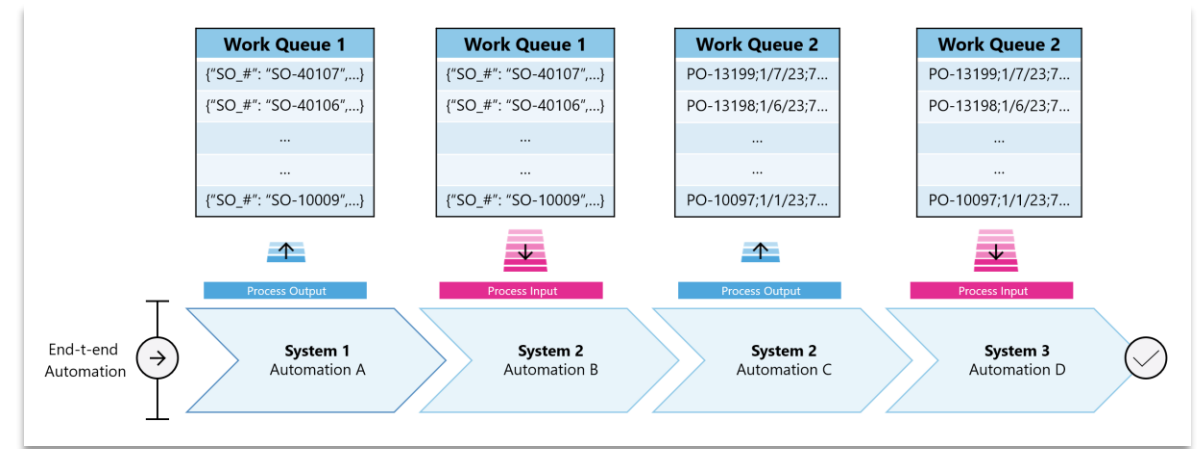
Send Keys

- Send Keys action should be used as a last resort, when nothing else is available.
- Look for alternatives, for example scripting actions (PowerShell, Python, VBScript)
- In some cases, [Image Recording](#) might help.
- But in some cases, **Send Keys** action can be a valid (and only) option.



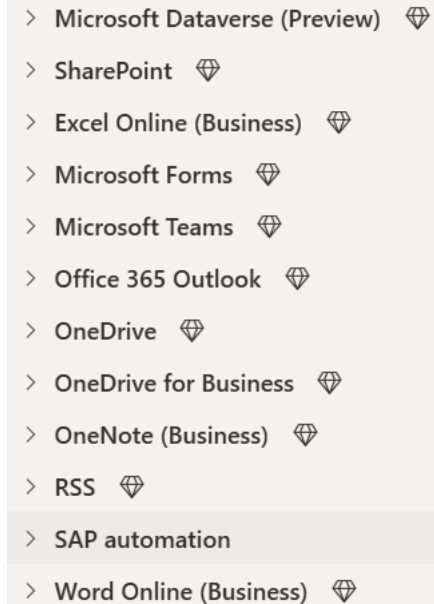
Work Queues

- Queues store process-relevant data.
- And enable asynchronous communication between complex processes and automations.
- Work queues improve efficiency, scalability, and resiliency of automations, helping prioritize high-priority tasks.
- using work queues in combination with other digital workforce management tools, such as hosted machine groups, advanced analytics with Power BI, and process mining, organizations can gain deeper insights into the performance of their automation.
- Add queues to the solution.



Cloud Connectors

- Several cloud connectors are available in Power Automate Desktop. (and more will follow)
- Whenever possible use the cloud flow actions instead of the native Power Automate Desktop actions. a
- Cloud Flow connectors will use API endpoints, which are more stable, secure than the native Power Automate Desktop actions.




A screenshot of the Power Automate Desktop connectors list. The list is displayed in a light gray background with a vertical scrollbar on the right. Each connector is preceded by a right-pointing chevron (>). Most connectors have a small diamond icon to their right, indicating they are available in the desktop version. The connectors listed are: Microsoft Dataverse (Preview), SharePoint, Excel Online (Business), Microsoft Forms, Microsoft Teams, Office 365 Outlook, OneDrive, OneDrive for Business, OneNote (Business), RSS, SAP automation, and Word Online (Business). The 'SAP automation' connector is highlighted with a darker gray background.

- > Microsoft Dataverse (Preview)
- > SharePoint
- > Excel Online (Business)
- > Microsoft Forms
- > Microsoft Teams
- > Office 365 Outlook
- > OneDrive
- > OneDrive for Business
- > OneNote (Business)
- > RSS
- > SAP automation
- > Word Online (Business)

Reusable components

- There are 2 options to create reusable components:
 - Sub Flows
 - Child Flows
- **Sub flows** are reusable groups of actions that can be invoked within the same flow.
- A **child flow** is another desktop flow which is triggered from another flow
- Child flows are triggered as separate flows, so there is an overhead in initializing the flow, setting up the logging, etc..

9

 Run subflow [LaunchApp](#)



10

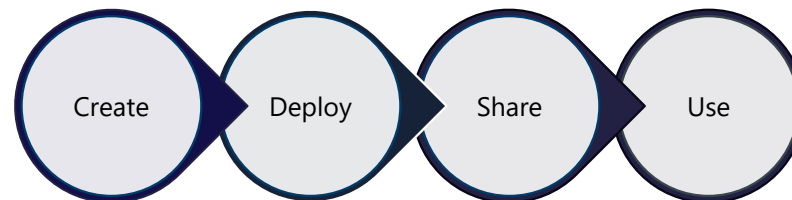
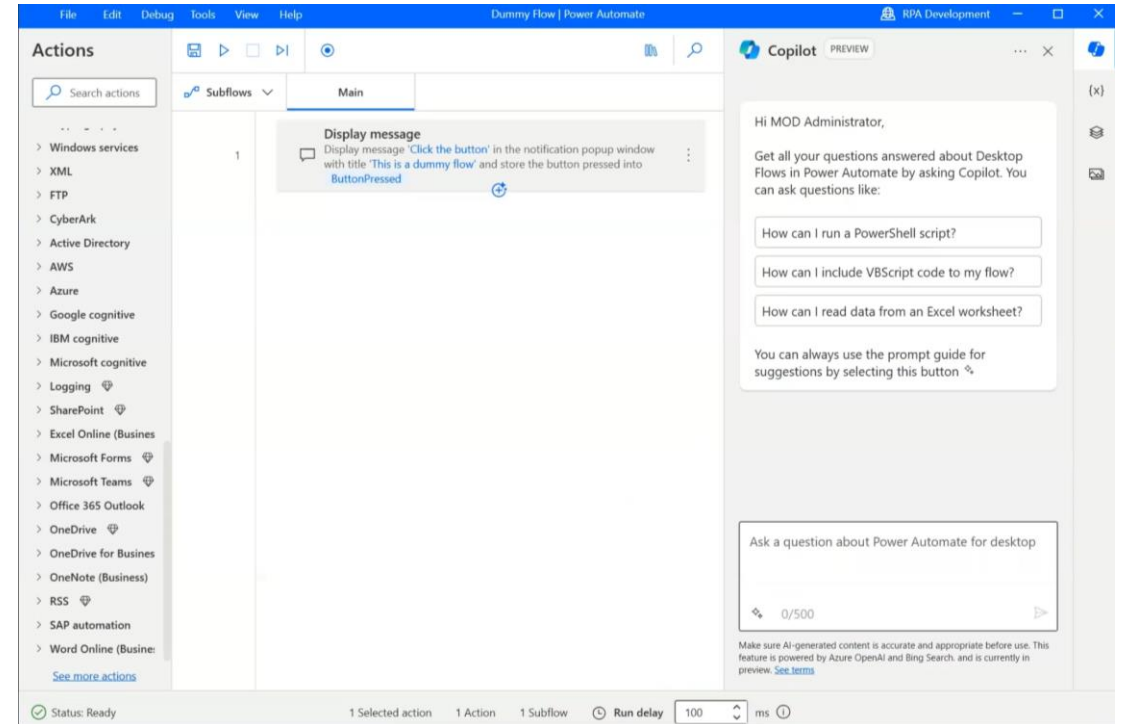
 Run desktop flow
Run flow with name: [Hello World](#)



Custom Actions

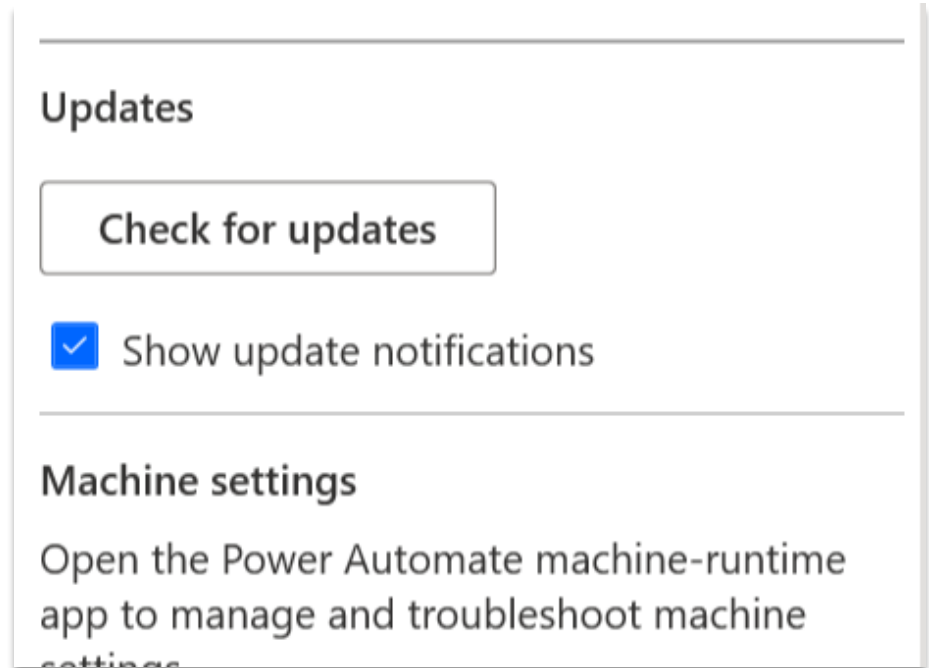
Custom actions allows you to create **your own reusable actions** that can be used across multiple desktop flows.

Custom actions are created using the Power Automate for desktop **actions SDK**, which provides a set of APIs that allow makers to create custom actions using **.NET language C#**. Custom actions can also be shared with other users through the custom actions section in Power Automate.



Keep up with updates

- Stay informed about new features, enhancements, and bug fixes in Power Automate.
- Review release notes.
- Regularly update your flows.
- Utilize the latest capabilities to ensure optimal reliability, resiliency, accuracy, and performance.



[Released versions for Power Automate for desktop - Release Notes | Microsoft Learn](#)



Security

Credentials

- The Credentials page in Power Automate allows you to create, edit, and share sign in credentials using Azure Key Vault or CyberArk and use them in desktop flow connections.
- Use Credentials to configure Desktop Flow connections for unattended runs.
- Credentials also support password rotation.

The image shows two overlapping windows from the Power Automate interface. The top window is titled 'Create a credential' and has a close button (X) in the top right corner. It features a vertical sidebar on the left with three radio buttons: 'Define credential name' (selected), 'Select credential store', and 'Select credential values'. The main area of this window is divided into two sections: 'Define credential name' and 'Description'. The 'Define credential name' section has a text input field labeled 'Credential name *' with the placeholder text 'Type a name for your credential'. The 'Description' section has a text input field labeled 'Enter a description'. The bottom window is titled 'Desktop flows' and has a close button (X) in the top right corner. It features a blue icon of a computer monitor with a plug symbol. Below the icon, it says 'Desktop flows' and 'Microsoft Premium'. A message states: 'If you don't see a machine or group, you can register a new machine now. To see recently registered machines or groups, refresh the list. Please note that machine in pending deletion state will not appear here'. There is a 'Learn more' link. Below this is a section titled 'Windows Credential (preview) *' which contains a dropdown menu labeled 'Select credential', a search bar with the placeholder 'Search', and a '+ New credential' button. Below the search bar is a list of credentials: 'SAP automation credential 01', 'RPA User 01 credential', and 'RPA credential 1'. A 'Create' button is located at the bottom right of this window. The word 'Standard' is visible at the very bottom of the image.

Create a credential

Define credential name

Credential name *

Type a name for your credential

Description

Enter a description

Desktop flows

Microsoft Premium

If you don't see a machine or group, you can register a new machine now. To see recently registered machines or groups, refresh the list. Please note that machine in pending deletion state will not appear here

[Learn more](#)

Windows Credential (preview) *

Select credential

Search

+ New credential

SAP automation credential 01

RPA User 01 credential

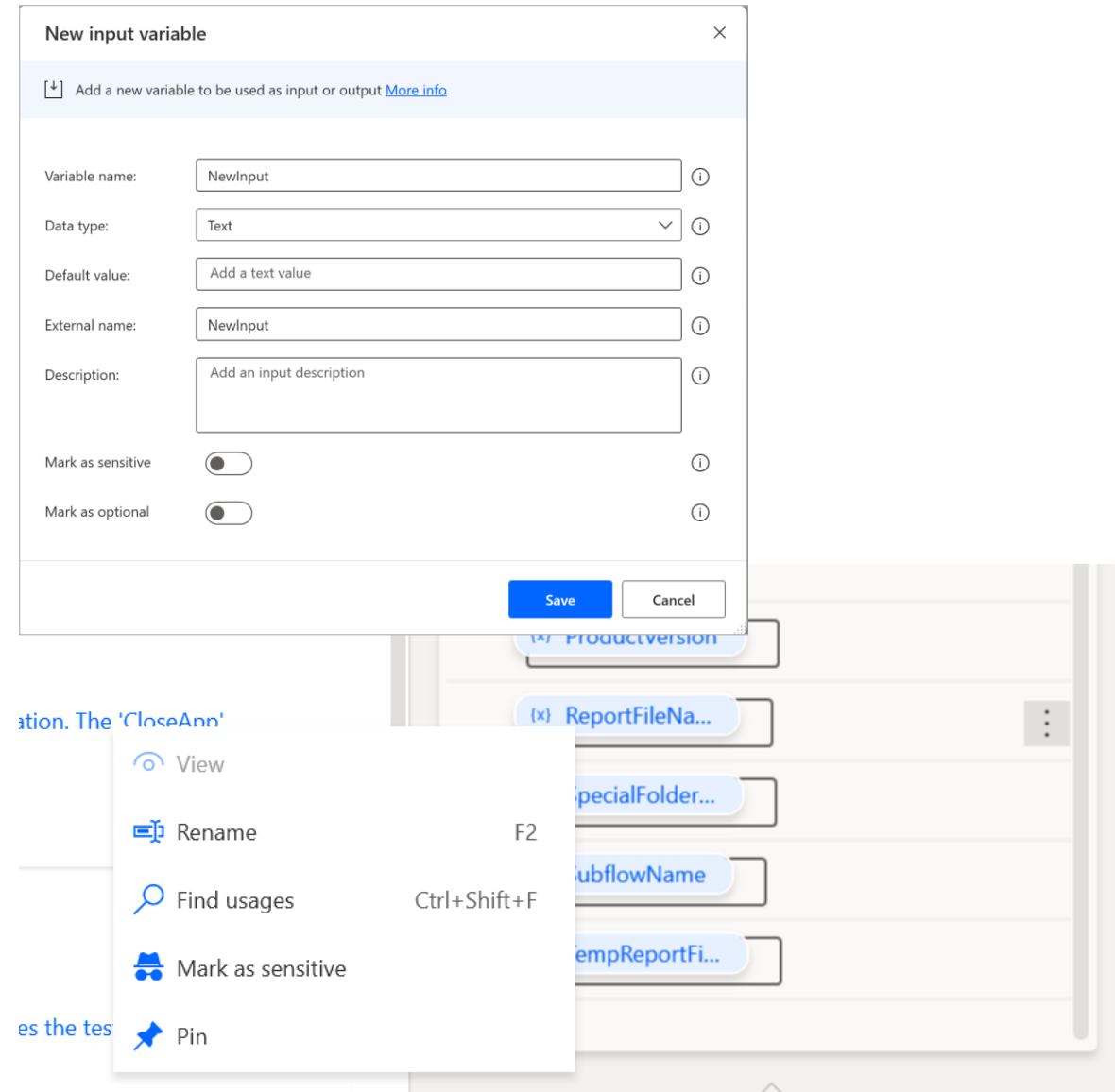
RPA credential 1

Create

Standard

Sensitive values

- Sensitive variables are not visible in the Power Automate Desktop designer during runtime, and they are also not logged in the desktop flow logs.
- Input, Output and Flow variables can be set to sensitive.
- Keep the scope of sensitive variables as small as possible. For example, keep the sensitive variable within a specific sub flow.
- Do not store a sensitive value into plain text variable. For example, using a sensitive password variable in a non-sensitive connection string. This is a potential security risk.
- Do not pass sensitive variables from one desktop flow to another flow. Instead retrieve them from a key vault or such.





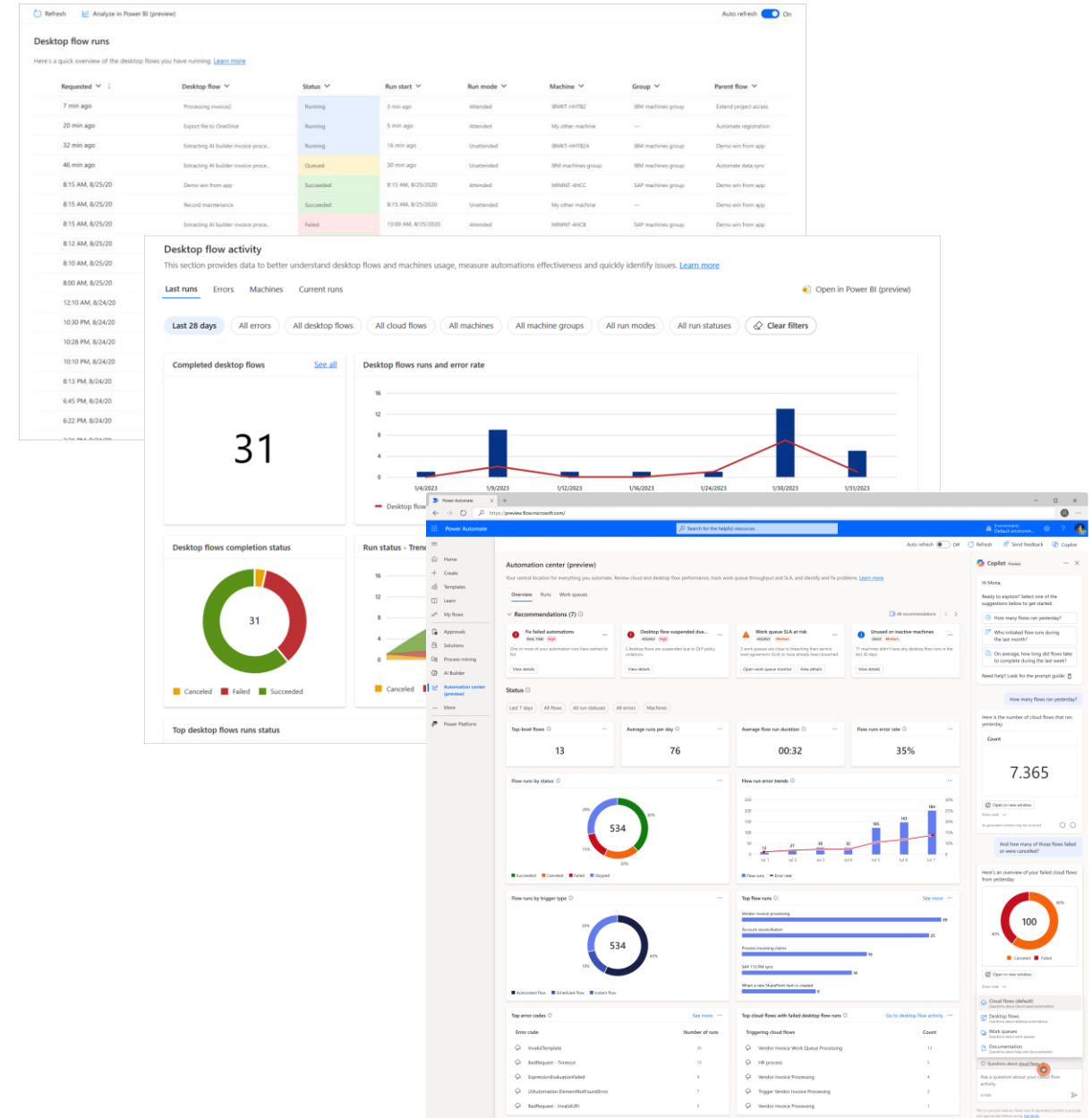
Usage

Logging

- Use **Log Message** action to send custom log messages to desktop flow action logs.
- Log message is a premium action.
- Log level can be Info, Warning, Error.
- Message has a limit of 250 chars.
- Alternative: use your own custom logging mechanism.
 - Write to text file (with append content setting)
 - Write to application insights
 - Custom Action for logging
- Use Get Last Error action to get line number and error description.
- Use logs for keeping track of progress and not only for logging errors.
- Record important details such as input values, output values, intermediate results, or error messages

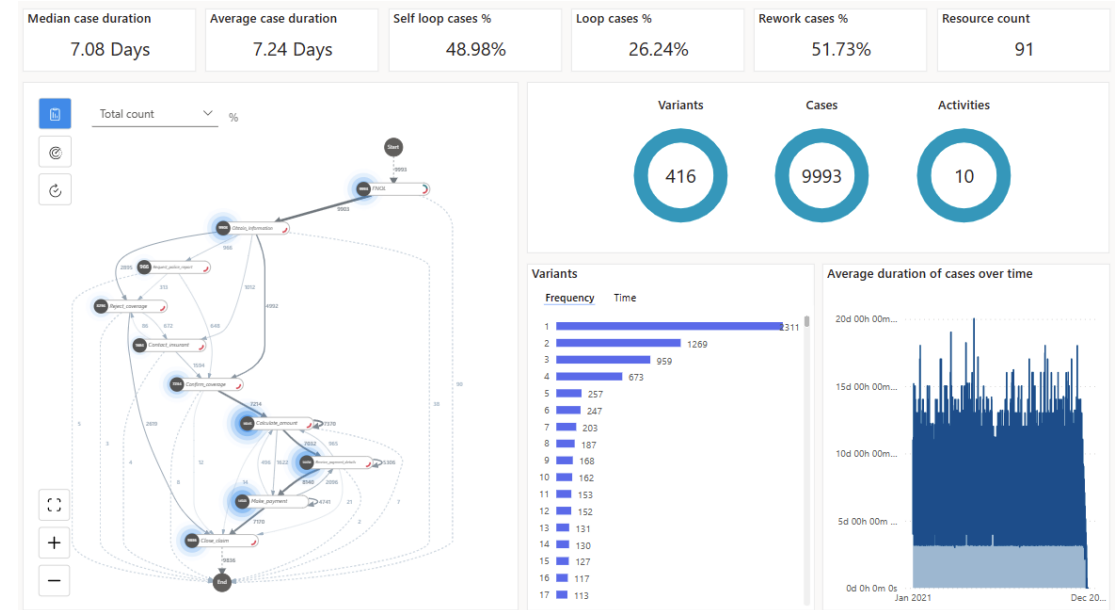
Monitoring

- **Desktop Flow Runs:** Monitor all your desktop flow runs in just one location to keep your automations running smoothly.
- **Desktop Flow Activity:** Provides dashboards, tables, and graphs to monitor desktop flows, measure effectiveness, and quickly identify issues.
- **Monitor run details:** View all the details of one of your desktop flow runs for the current environment.
- **Automation center:** enables you to gain a holistic view of all automation related data, including recommendations, execution logs, performance metrics, and an integrated copilot.



Process Mining

- Use [Process Mining](#) to analyze the run history of RPA flows and optimize the performance of the desktop flows.
- The Desktop Flows template can identify potential issues:
 - Errors
 - Inefficiencies
- The template provide recommendations for improvement.



[Power Platform templates for process mining - Power Automate | Microsoft Learn](#)



Questions?

Thank you!