 Microsoft Power Automate Desktop

Lab 01: Automate using UI elements

Hands-on lab step-by-step

July 2024

Microsoft Power Automate Desktop – Advanced Workshop

Contents

[Microsoft Power Automate Desktop 1](#_Toc171456731)

[Goals for this lab 1](#_Toc171456732)

[Prerequisites 1](#_Toc171456733)

[Exercise 1: Automate a desktop application with PAD 2](#_Toc171456734)

[Exercise 2: Challenge – Perform a math operation with a dynamic selector 13](#_Toc171456735)

[Terms of Use 14](#_Toc171456736)

# Microsoft Power Automate Desktop

This lab is subject to the Terms of Use found at the end of this document.

## Goals for this lab

|  |  |
| --- | --- |
| After this lab you will be able to:   * Use UI elements to interact with applications without resorting to image recognition and absolute coordinates. * Edit UI element properties and avoid common errors when interacting with dynamic elements | The time to complete  this lab is [40] minutes. |

## Prerequisites

Please note that some labs, especially later labs, do reference previous labs in reference to capabilities and previous tasks. The labs have been designed so if you have access to a Microsoft Power Automate Desktop trial, you can get started from most lab without having to complete the previous module to be able to move forward. However, for the best experience that shows the features and functionality that is possible within the product, it is recommended you have completed specific modules before starting some of the labs.

For Lab 01: Automate using UI elements, you need:

* A computer with internet access.
* The application Power Automate Desktop installed in your computer. If you don’t have the application installed, please download it here: <https://go.microsoft.com/fwlink/?linkid=2102613>
* Be able to log into your corporate tenant.

## Exercise 1: Automate a desktop application with PAD

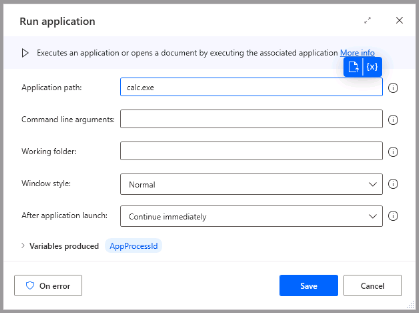
### Task 1: Log into Power Automate Desktop

1. Open the **Power Automate Desktop** app on your computer
2. Log into the application using your corporate account
3. If you don’t have a Power Automate Premium license, start a trial by clicking on the **Go Premium** button at the top right corner of the application



### Task 2: Create a desktop flow and open a desktop application

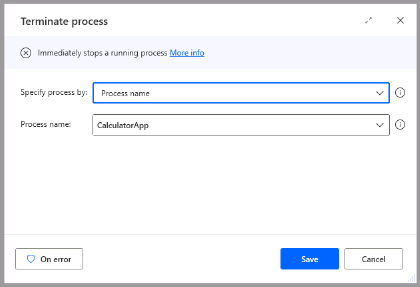
1. Make sure you create your automations in your own **Personal Development environment**. If you don’t have one, follow the steps here: [Get your developer environment - Power Apps | Microsoft Learn](https://learn.microsoft.com/en-us/power-apps/maker/maker-create-environment#create-your-own-developer-type-environment)
2. Click on **+ New Flow** and create a flow named Lab 01: Automate using UI elements
3. Once on the flow designer, look for action **Run application** from the **System** actions group
4. Open the Windows Calculator with this action setting the **Application path** to calc.exe



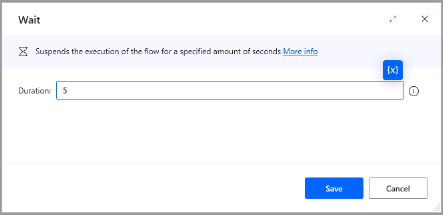
1. Click the **Save** button of the **Run application** action
2. Click the **Run** button of the flow and ensure that the Windows Calculator app is opened correctly. Close the Windows Calculator for now
3. Add the action **Terminate process** to your flow

|  |  |
| --- | --- |
| Lights On outline | **Pro tip**: It’s always a good idea to close all applications (using the **Terminate process** action) at the end of your flows to avoid conflicting application sessions between process runs. |

1. Configure the **Terminate process** action setting **Specify process by:** to Process name and **Process name:** to CalculatorApp



1. Click the **Save** button of the **Terminate process** action
2. Add an action to **Wait** between **Run application** and **Terminate process** and set the delay to 5 seconds. This will allow you to see how the automation opens the Calculator before closing it.

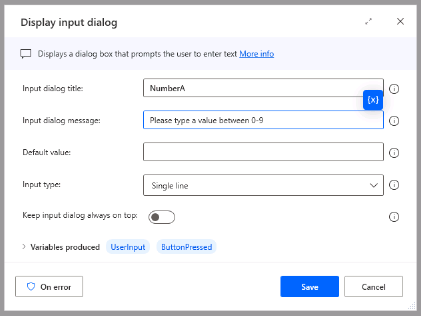


1. Click the **Run** button of the flow and test that the Calculator app opens and closes correctly.

### Task 3: Interact with UI elements of the application

1. Add a **Display input dialog** action after the **Run application action**
2. Configure this action as follows:

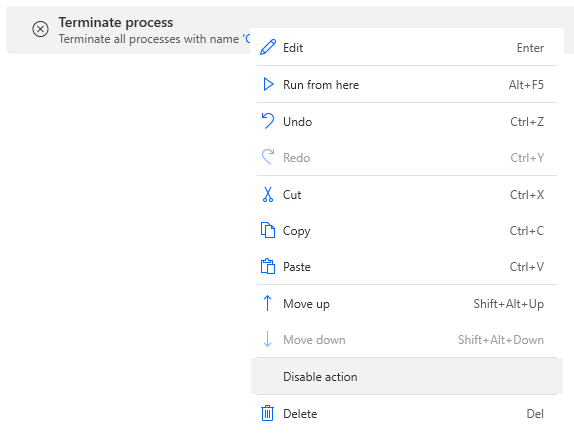
* Input dialog title: NumberA
* Input dialog message: Please type a number between 0-9



1. Rename the UserInput variable produced by this action to NumberA
2. **Save** this action
3. **Copy** the action by right-clicking it and modify it as follows:

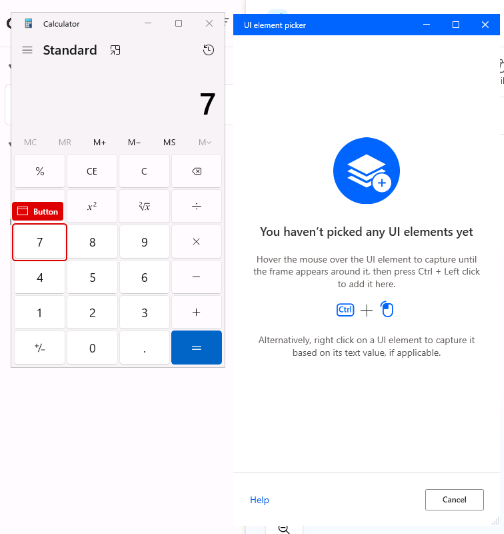
* Input dialog title: NumberB
* Input dialog message: Please type a number between 0-9

1. Rename the UserInput variable produced by this action to NumberB
2. Right-click on the **Terminate process** action and **Disable this action**.

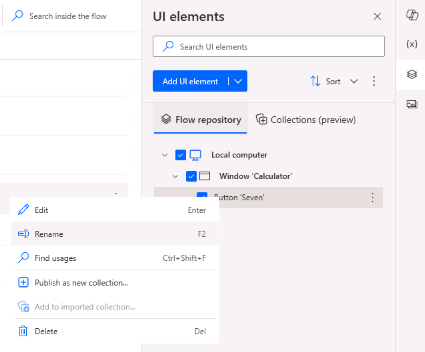


1. Search for the **Click UI element in window** action and drag It to your flow, right after the second **Display input dialog** step.
2. Open the **UI element** selector and click on the **Add UI element** button
3. While pressing the **Ctrl** key, click on any of the numbers in the Windows Calculator app to identify the UI element

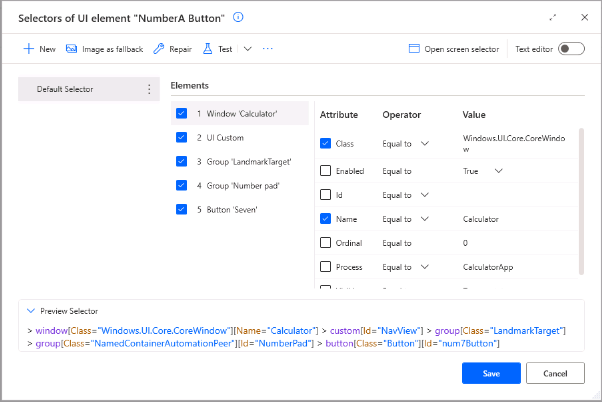
|  |  |
| --- | --- |
| Warning outline | We will dynamically change which number is selected on the Windows Calculator app later on. |



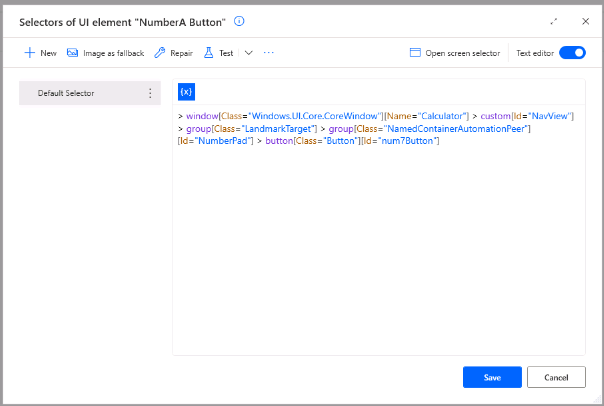
1. On the Flow Designer, open the UI elements panel to the right of the screen, and locate the UI element representing the Windows Calculator button you just selected. Rename it to NumberA Button



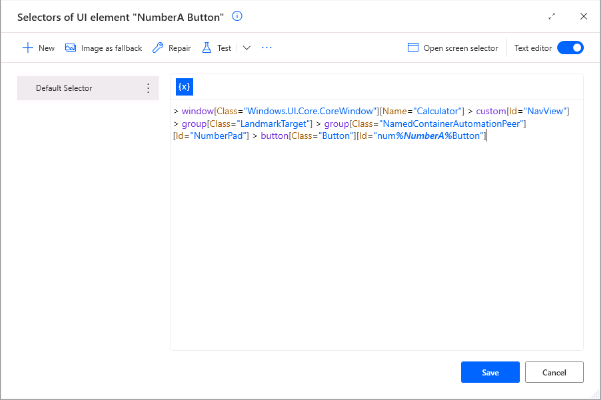
1. Double-click on the UI element for **NumberA Button** to discover its properties



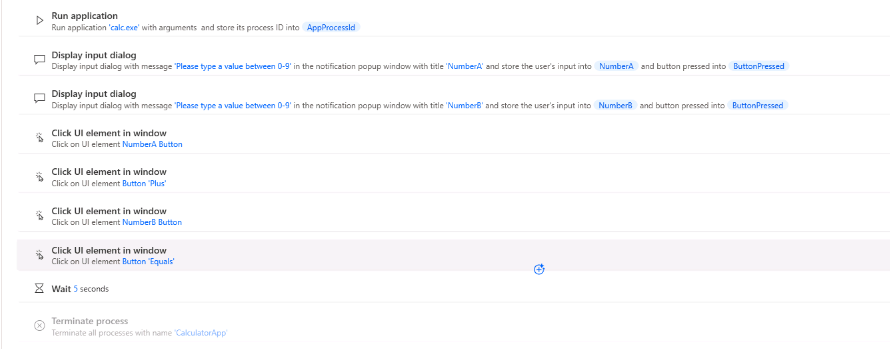
1. On the top-right corner of this window, switch the **Text editor** on



1. Locate the section of the selector [Id=”num7Button”] and replace it by [Id=”num%NumberA%Button”]. This will dynamically select the button on the Calculator that matches the input value. Click on **Save**



1. Repeat steps 8 through 14 to select the button for **NumberB**
2. Add one more **Click UI element in window** button to select the operand button of the Windows Calculator app. Add the UI element for the **SUM** button of the calculator. **Remember to place this step between the NumberA and NumberB UI selectors**
3. Repeat the previous step to select the **EQUAL** button of the calculator. **Remember to place this step after the NumberB UI selector**



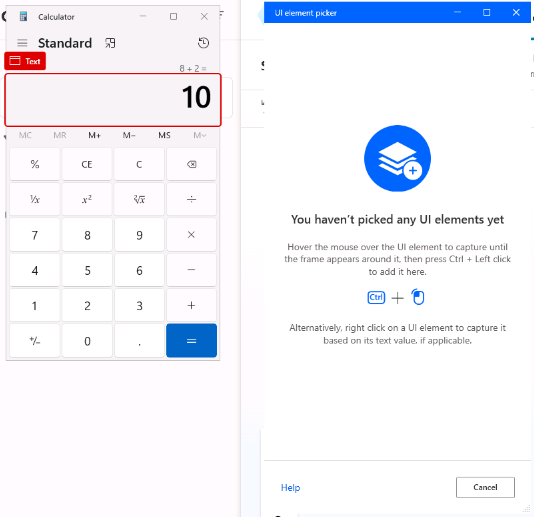
1. **Save** the flow and click on the **Run** button to test your progress so far

By the end of this section, your process should be able to perform a simple sum on the Windows Calculator and return the result.

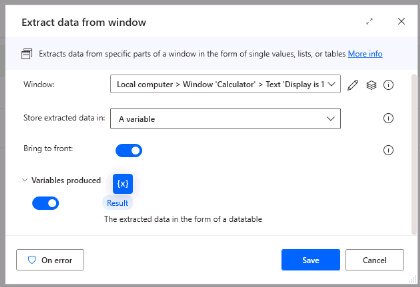


### Task 4: Extract data from a window

1. Search for the action **Extract data from window** and drag it to your process, right after the **Click on UI element ‘Equal Button’**
2. Expand the **Window** parameter of this action, and click on **Add UI element**
3. While pressing the **Ctrl** key, click on the result of the operation on the Windows Calculator, as shown on the image below



1. Expand the **Show extracted data** parameter of this action and select **A variable**.
2. Rename the variable produced by this step to Result. **Save** the action



1. On the Flow Designer, open the UI elements panel to the right of the screen, and locate the UI element representing the Windows Calculator Result you just selected. Rename it to Result UI Field

A screenshot of a computer

Description automatically generated

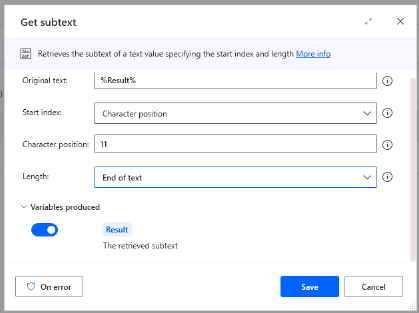
1. Click on the **Run** button of the flow and check the value of the **Result** variable after finishing the process. The value should be similar to “Display is 5”

A screenshot of a computer

Description automatically generated

1. Search for the action **Get subtext** and drag it to your process, right after the **Extract data from window** step
2. Configure the action parameters as indicated:

* **Original text:** %Result%
* **Start index:** Character position
* **Character position:** 11
* **Length:** End of text
* **Variables produced:** Select the existing Result variable



1. **Save** this action
2. Search for the action **Display message** and drag it to your process after **Get subtext**. Configure its parameters as shown:

* **Message box title:** Result is:
* **Message to display:** %Result%



1. Enable the **Terminate process** action at the end of your flow. Save it and execute it for a final test.

## Exercise 2: Challenge – Perform a math operation with a dynamic selector

On the last exercise, you created a flow to sum two digits. Now, try to modify your flow to include a selector for the math operation you want to perform, a sum, a subtraction, a multiplication or a division.

**Hint:** Consider using a **list variable** to create a list of available math operations (sum, minus, divide, multiply). Also consider using the **Display select from list dialog** action to show a dialog with options.

Take some time to analyze this challenge and share your ideas with the group. Good luck!

## Terms of Use

By using this document, in whole or in part, you agree to the following terms:

### **Notice**

Information and views expressed in this document, including (without limitation) URL and other Internet Web site references, may change without notice. Examples depicted herein, if any, are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

### **Use Limitations**

Copying or reproduction, in whole or in part, of this document to any other server or location for further reproduction or redistribution is expressly prohibited. Microsoft provides you with this document for purposes of obtaining your suggestions, comments, input, ideas, or know-how, in any form, ("Feedback") and to provide you with a learning experience. You may use this document only to evaluate its content and provide feedback to Microsoft. You may not use this document for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this document or any portion thereof. You may copy and use this document for your internal, reference purposes only.

### **Feedback**

If you give Microsoft any Feedback about this document or the subject matter herein (including, without limitation, any technology, features, functionality, and/or concepts), you give to Microsoft, without charge, the right to use, share, and freely commercialize Feedback in any way and for any purpose. You also give third parties, without charge, the right to use, or interface with, any Microsoft products or services that include the Feedback. You represent and warrant that you own or otherwise control all rights to such Feedback and that no such Feedback is subject to any third-party rights.

### **DISCLAIMERS**

CERTAIN SOFTWARE, TECHNOLOGY, PRODUCTS, FEATURES, AND FUNCTIONALITY (COLLECTIVELY "CONCEPTS"),

INCLUDING POTENTIAL NEW CONCEPTS, REFERENCED IN THIS DOCUMENT ARE IN A SIMULATED ENVIRONMENT

WITHOUT COMPLEX SET-UP OR INSTALLATION AND ARE INTENDED FOR FEEDBACK AND TRAINING PURPOSES

ONLY. THE CONCEPTS REPRESENTED IN THIS DOCUMENT MAY NOT REPRESENT FULL FEATURE CONCEPTS AND MAY

NOT WORK THE WAY A FINAL VERSION MAY WORK. MICROSOFT ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH

CONCEPTS. YOUR EXPERIENCE WITH USING SUCH CONCEPTS IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

THIS DOCUMENT, AND THE CONCEPTS AND TRAINING PROVIDED HEREIN, IS PROVIDED “AS IS”, WITHOUT WARRANTY

OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING (WITHOUT LIMITATION) THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. MICROSOFT DOES NOT

MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, THE OUTPUT THAT DERIVES FROM USE OF THIS DOCUMENT OR THE CONCEPTS, OR THE SUITABILITY OF THE CONCEPTS OR INFORMATION CONTAINED IN THIS DOCUMENT FOR ANY PURPOSE.