

# Nested Flow OCR and Image Based Testing

Creation Date: 2/9/2023

## Table of Contents

Introduction .....	1
Image Snip And Examine.....	2
Snipping.....	2
Capture Window .....	4
OCR Read From Image .....	8
Windows Automation Element OCR Options .....	9
OCR Read .....	9
OCR Click By Text .....	12

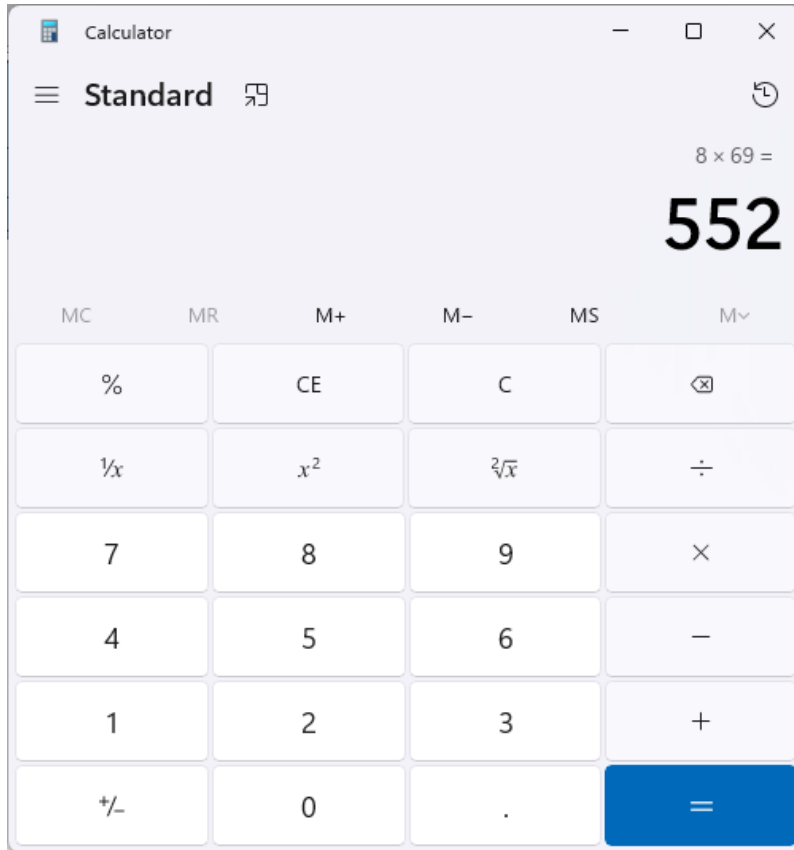
## Introduction

NestedFlow tool provides Tesseract based OCR options which can be used for testing applications where traditional testing techniques do not work.

## Image Snip And Examine

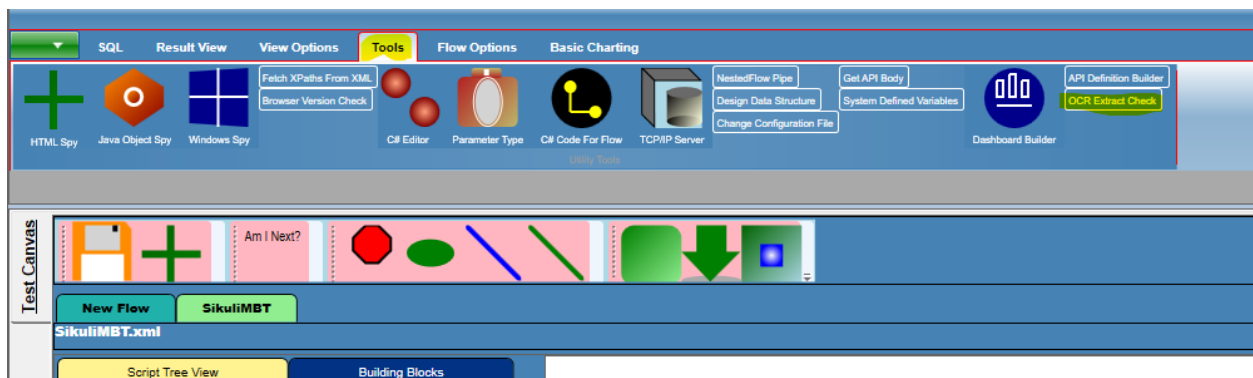
NestedFlow comes with inbuilt image snip and capture tool which can be used to measure effectivity of the OCR functionality. Below are two scenarios which can be performed for testing.

### Snipping

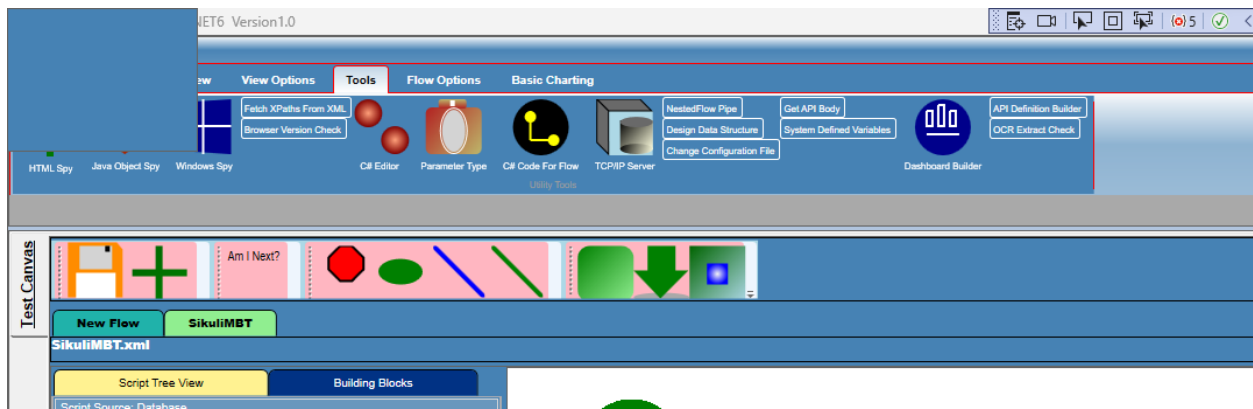


Suppose we did some calculations on the windows calculator and we need to see if the result can be read using OCR.

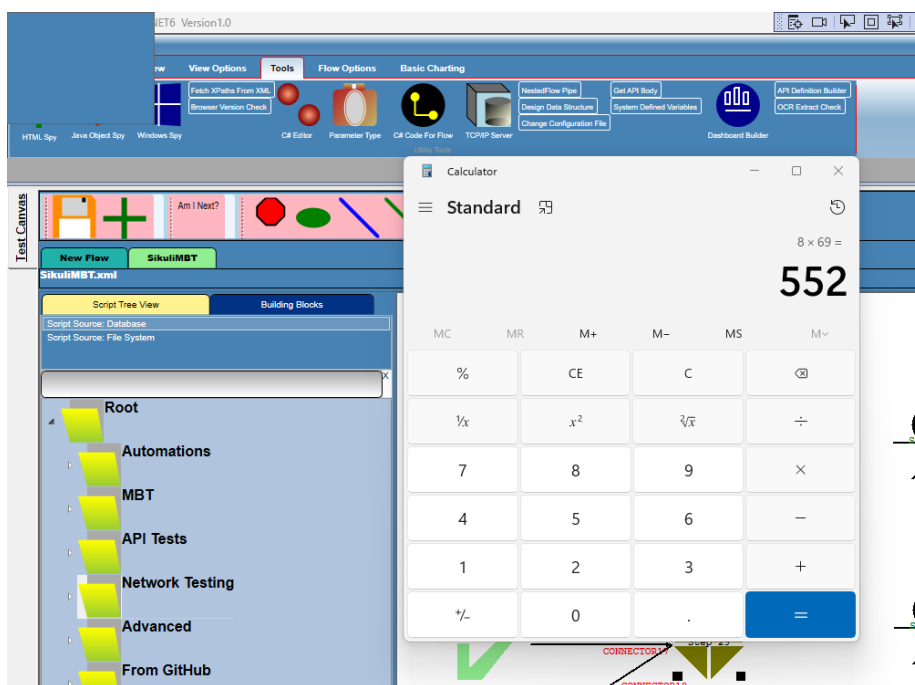
Click on Tools Tab on main ribbon on the tool and click on **OCR Extract Check** button



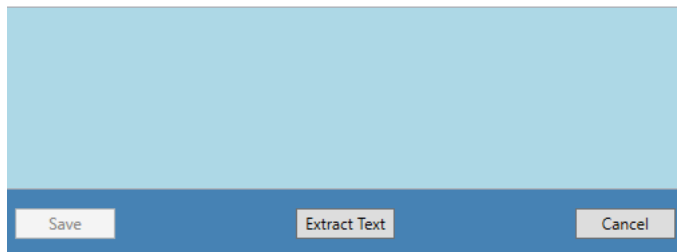
A blue rectangle will appear on the left side of the ribbon



Bring Calculator on the front and click on the blue rectangle

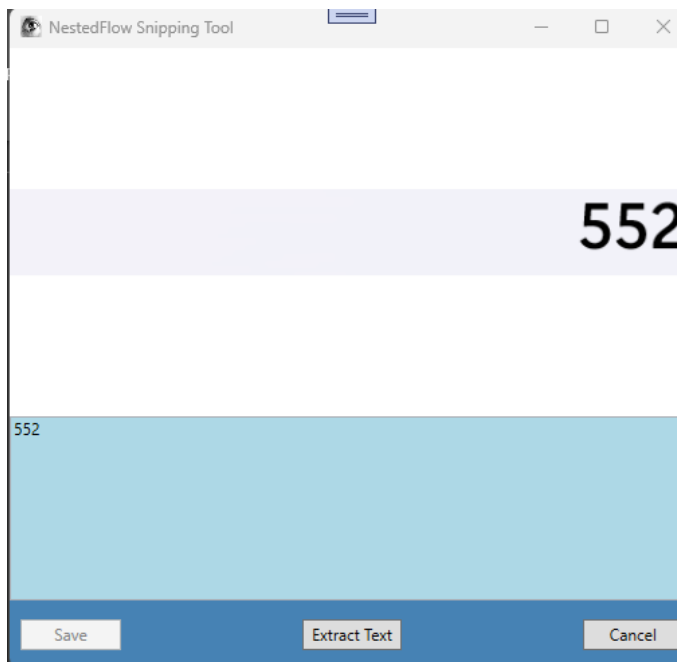


Mouse pointer converts to **+** shape and snip the calculator display part using mouse (just like you do in windows snipping tool)



Click on the **Extract Text** button

This action will try to read the text from the image

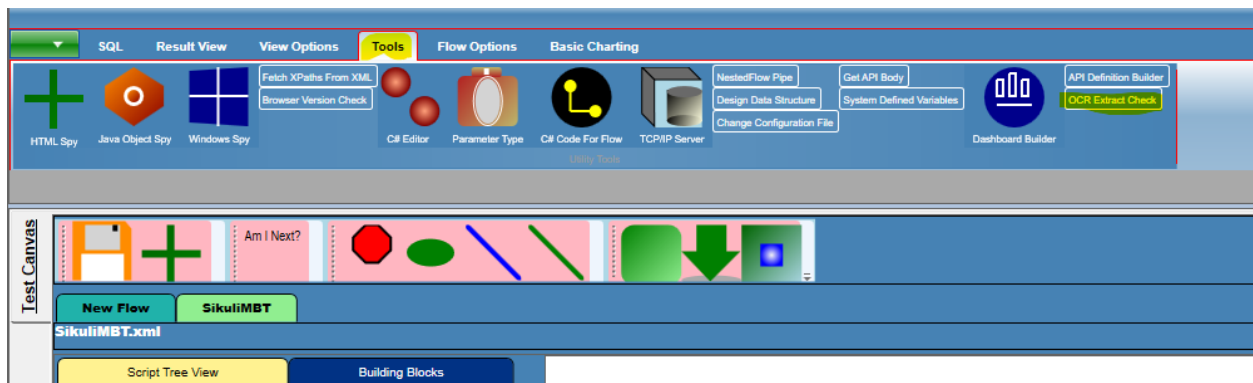


In this case, OCR accurately reads the value means OCR can be effective in the testing of windows calculator

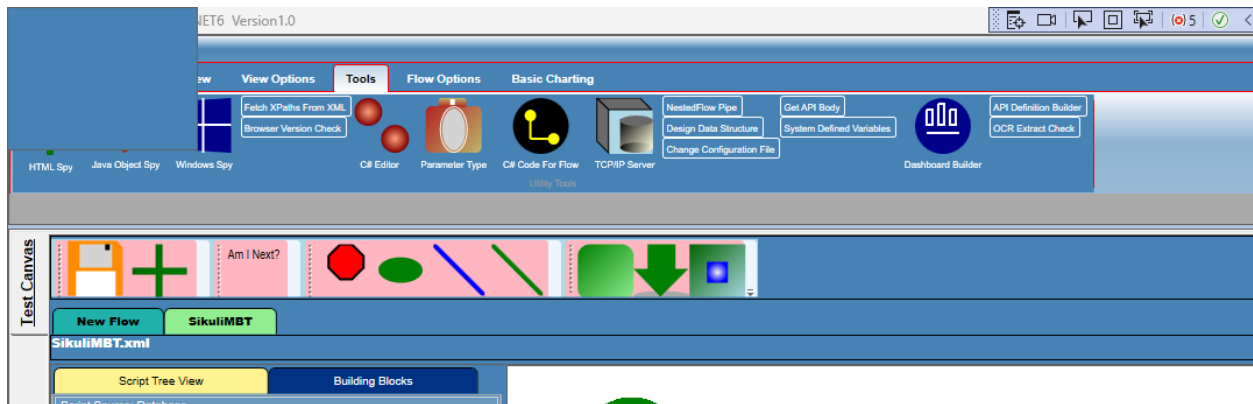
### Capture Window

Second approach is capturing entire application window image using the snipping tool

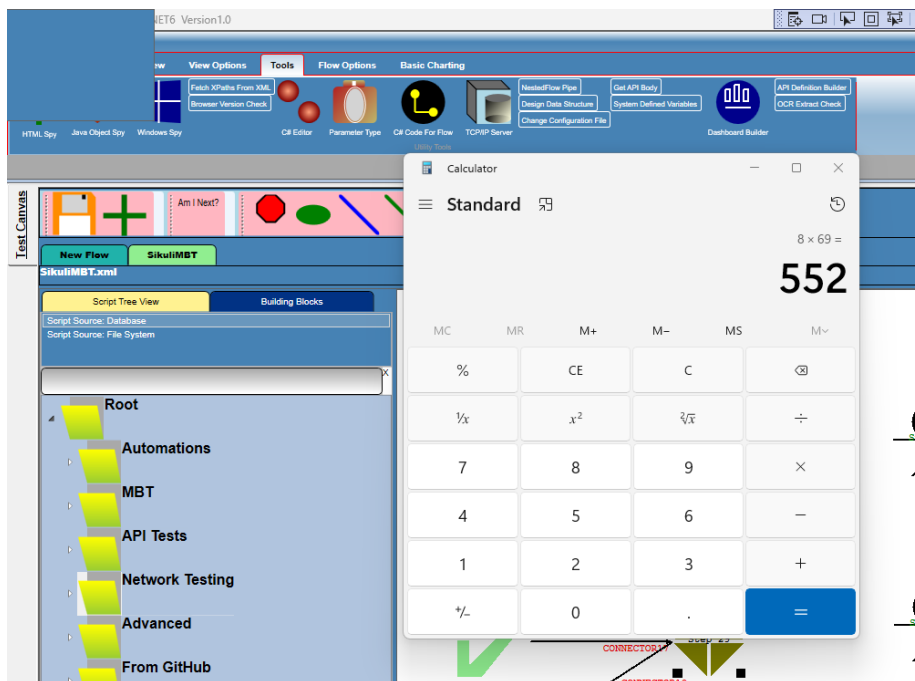
Click on Tools Tab on main ribbon on the tool and click on **OCR Extract Check** button



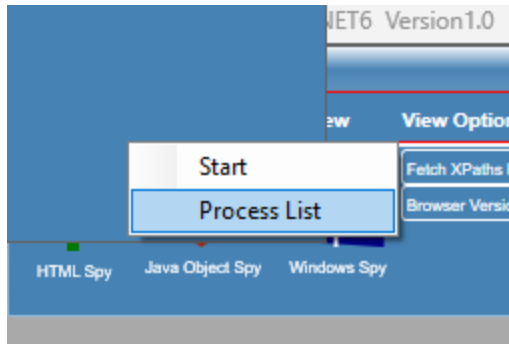
A blue rectangle will appear on the left side of the ribbon



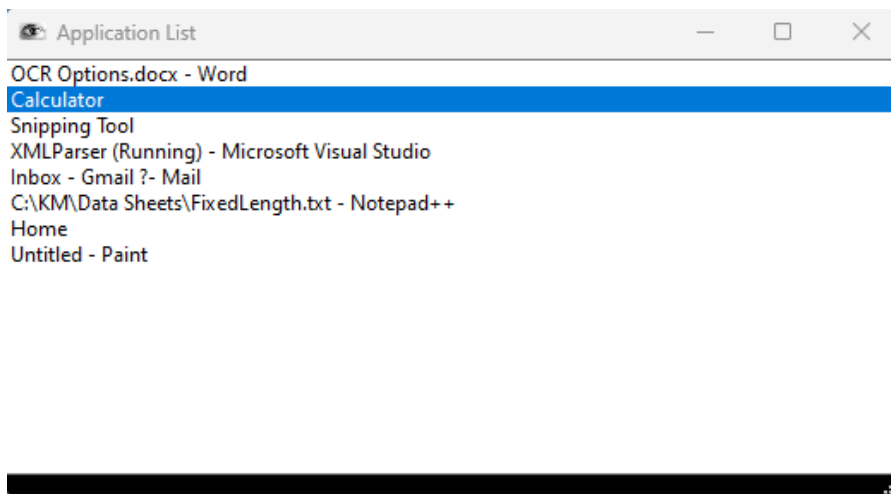
Right click on the blue rectangle



Click on **Process List** menu option

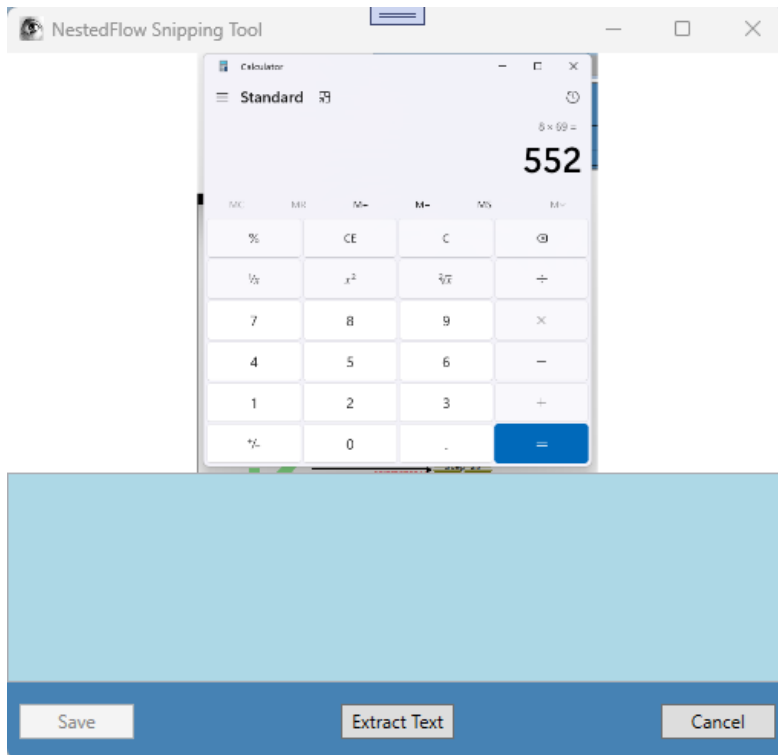


Choose the application to be captured by double clicking on it

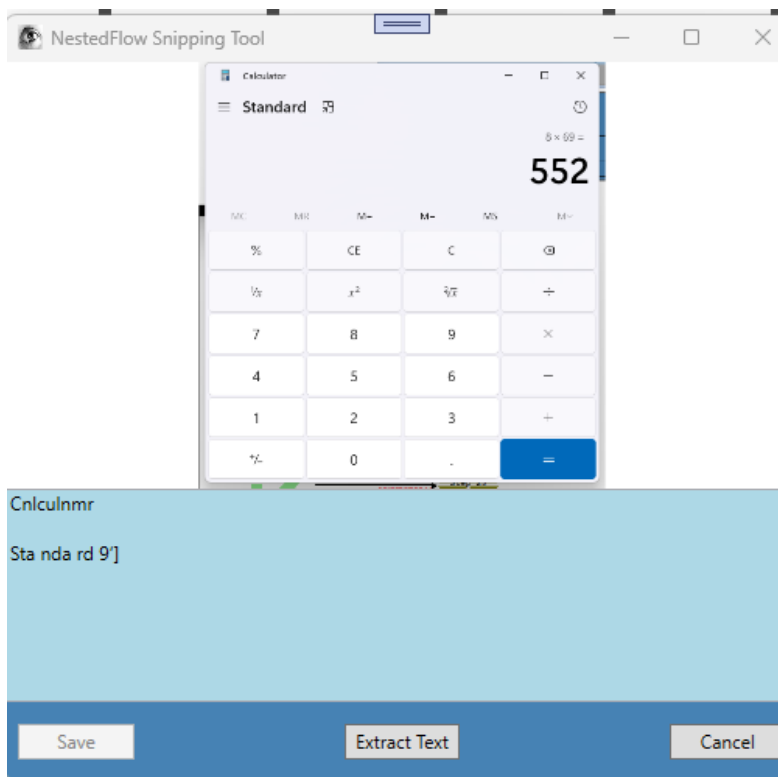


Entire application image will be captured





Upon clicking on **Extract Text** button and the result is completely improper



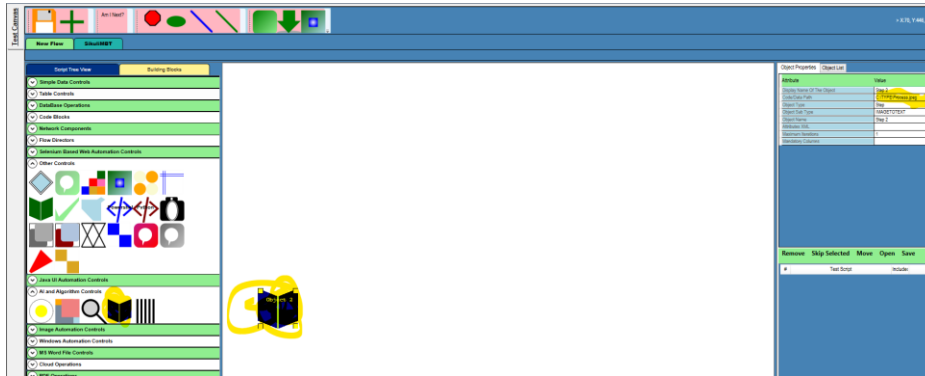
Means OCR has to be used in precise manner where it is applicable

## OCR Read From Image

Consider an image

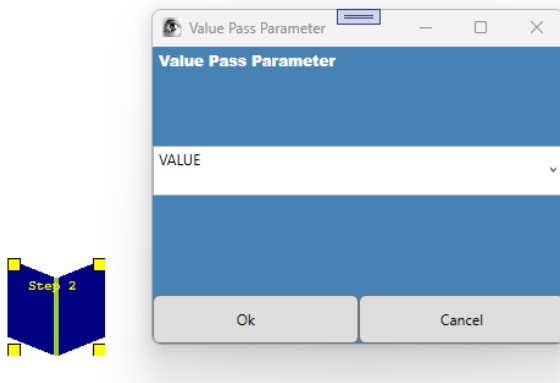
Shop for Holiday Gifts

Using Text from image from AI controls



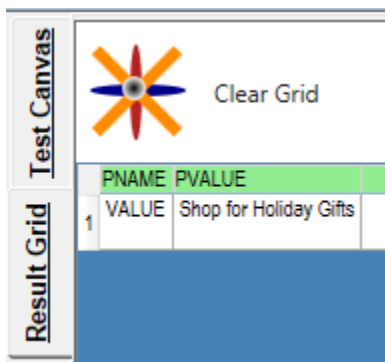
Right click on the dragged control and choose **Add Pass Attribute Name**

Enter parameter name and click on **OK**



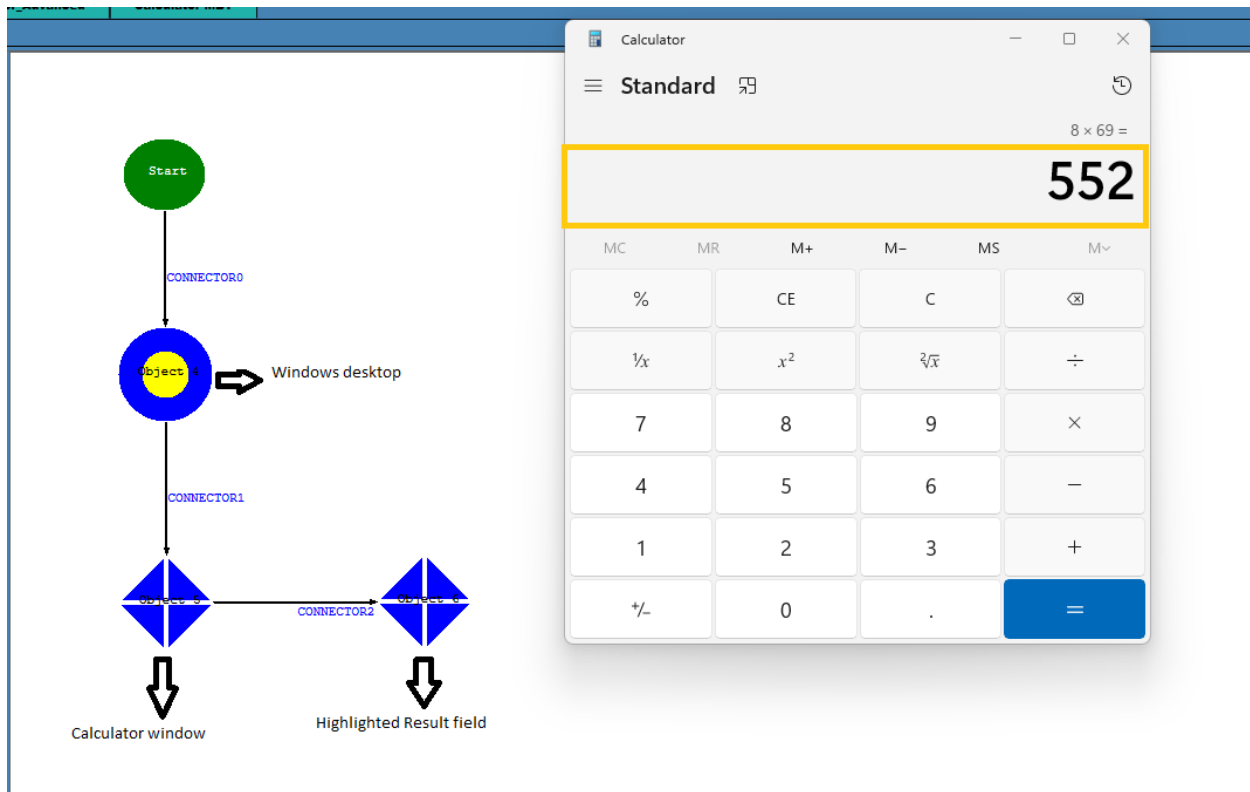
Right click on the dragged control and choose **Execute From Here** option

OCR reads the data accurately



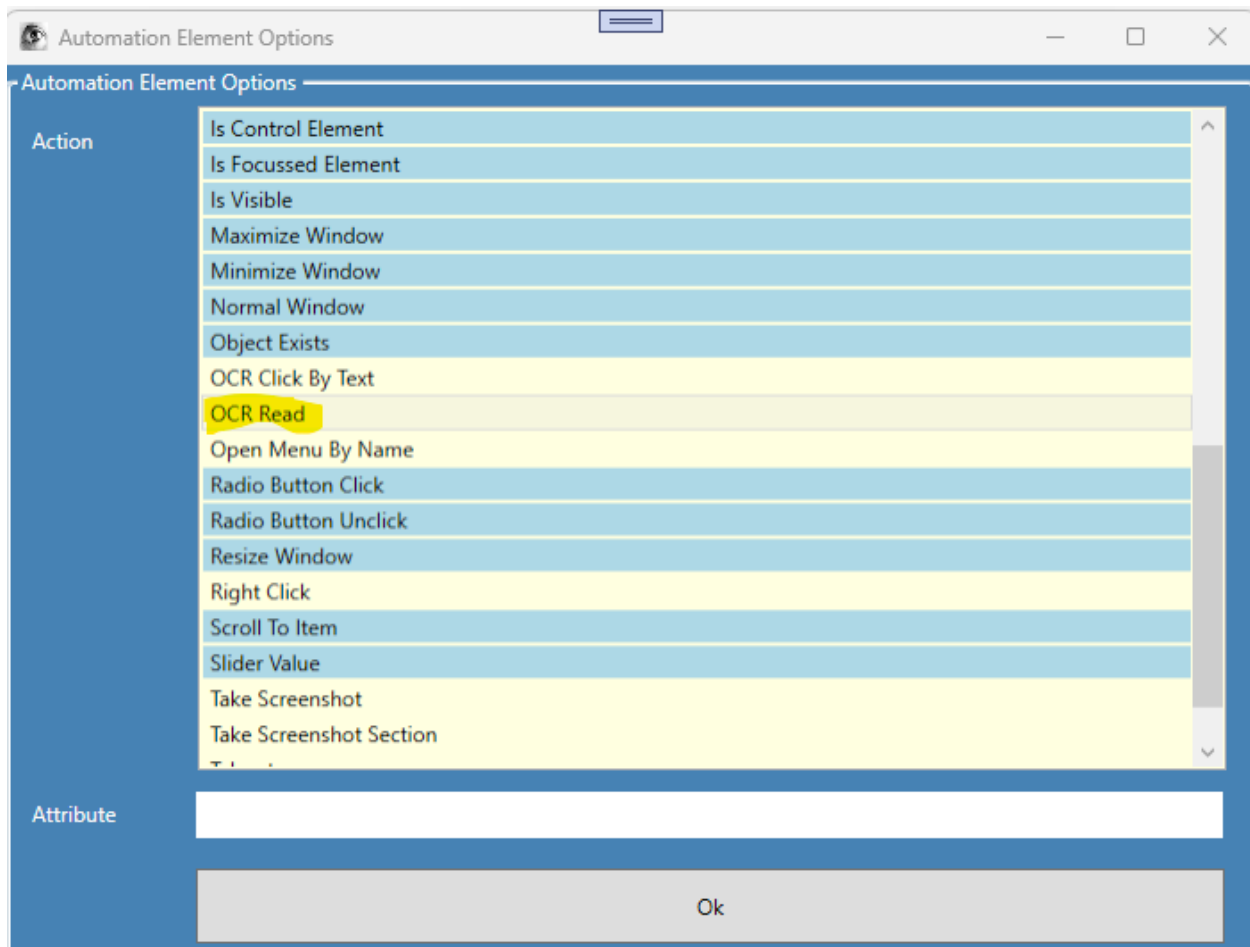
## Windows Automation Element OCR Options

### OCR Read



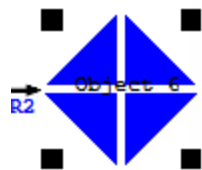
Consider a simple script which reads result field from windows calculator

Choose action -> OCR Read



Right click on the Windows element control and choose **Add Pass Attribute Name**

Enter parameter name and click on **OK**




Value Pass Parameter

**Value Pass Parameter**

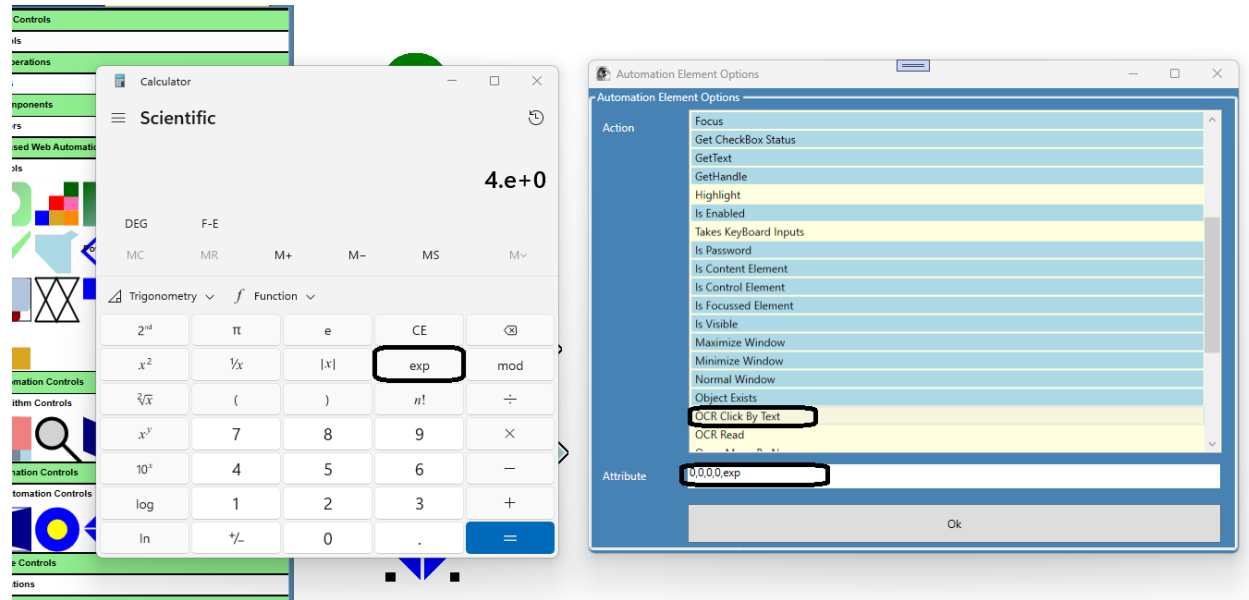
VALUE

Ok Cancel

Upon execution OCR reads the control text

Test Canvas	
 Clear Grid   Export To Excel	
Result Grid	PNAME PVALUE
	1 VALUE 552

## OCR Click By Text



OCR Click By Text → It can be used to search a text inside a windows element and clicks on the text

Function Name: **OCR Click By Text**

Attribute: x1,y1,width,height,string to find

X1 → start from windows element left corner + X1

Y1 → start from windows element top corner + Y1

Width → Width of the box to find string from left corner (if 0 then takes entire control width)

Height --> Height of the box to find the string from top corner (if 0 then takes entire control height)

String to find → which string to find and click on it in the search rectangle