

# NestedFlow Automation – Automating Chrome framework Desktop Applications

Creation Date: 9/28/2023

## Table Of Contents

Introduction .....	1
Case 1.: Microsoft ClipChamp .....	2
Logic: .....	9

## Introduction

In This document We will look into automation of Chrome framework based desktop applications. We will try to automate Microsoft ClipChamp and record a simple screen video in this example

## Case 1.: Microsoft ClipChamp

- Start off by a blank canvas in the NestedFlow Automation tool (Click on **New Test** button on the **Flow Options** ribbon tab)

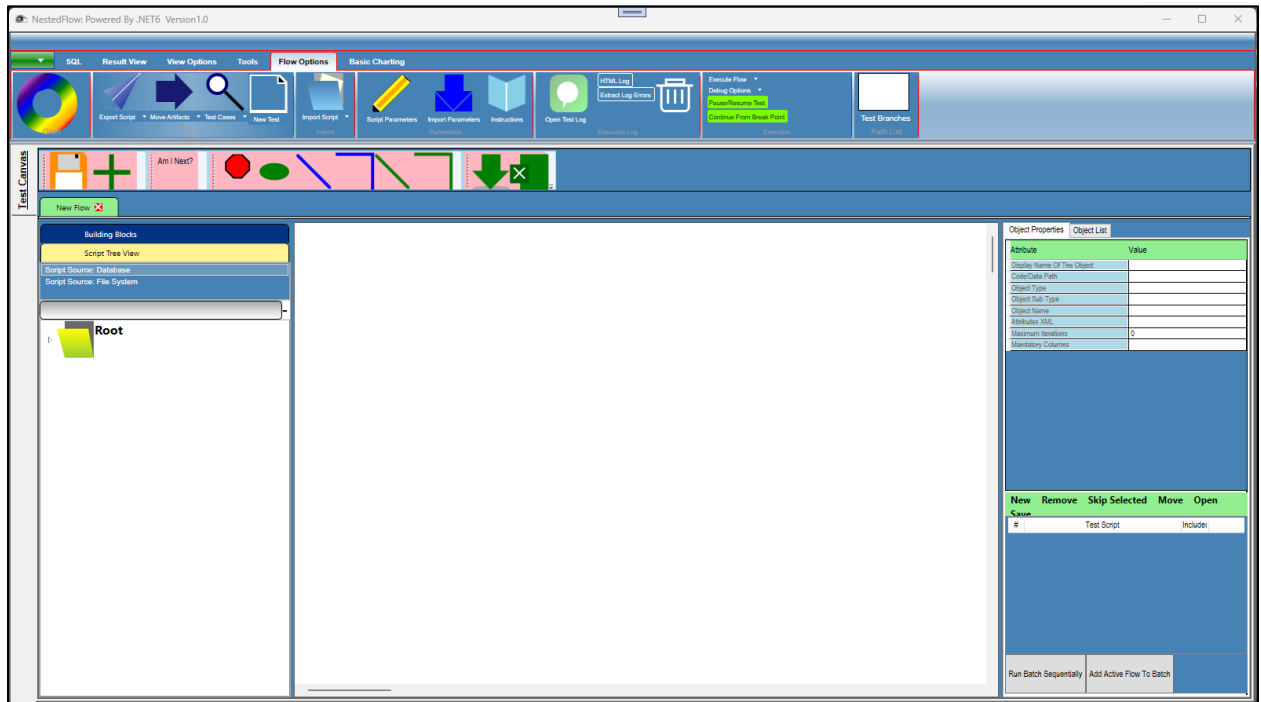


Figure 1: Blank Screen on the NestedFlow Automation Tool

- Drag and Drop Start and End components from shortcut menu

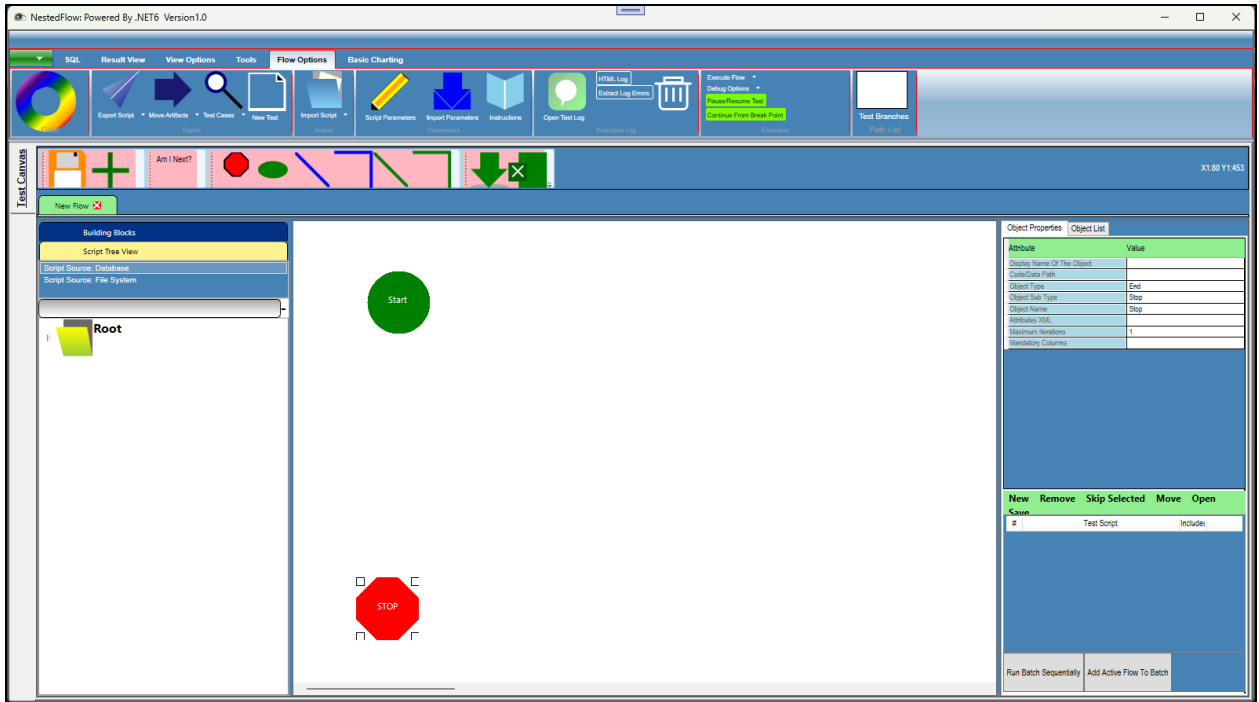


Figure 2: Start and End Component

- Drag and drop Invoke external Exe

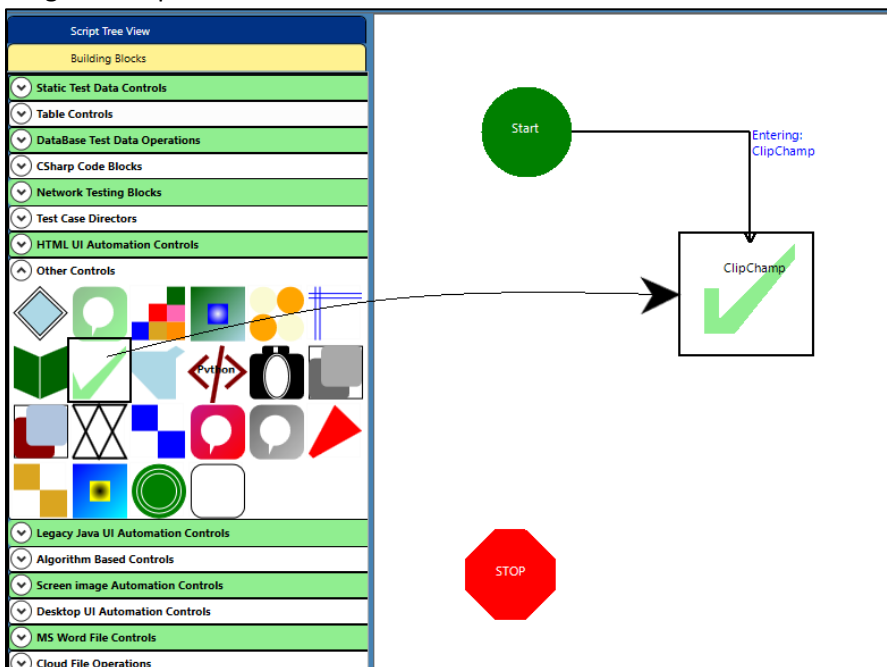


Figure 3: Invoke External Exe

- Right click on the component and choose **Add Exe or executable path with Arguments**

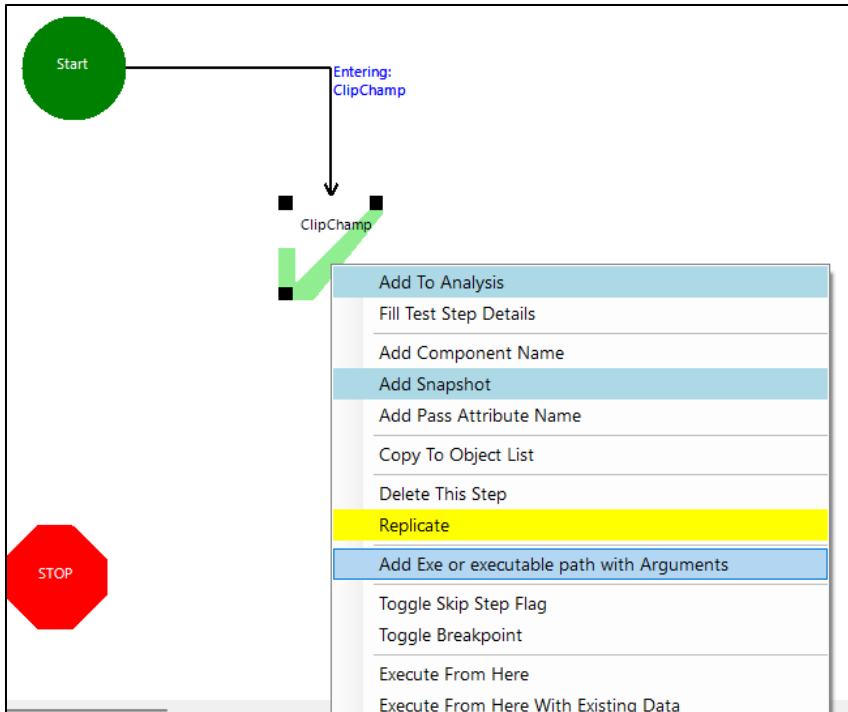


Figure 4: Add Exe or executable path with Arguments

- Add clipchamp.exe path and click on **Ok** button

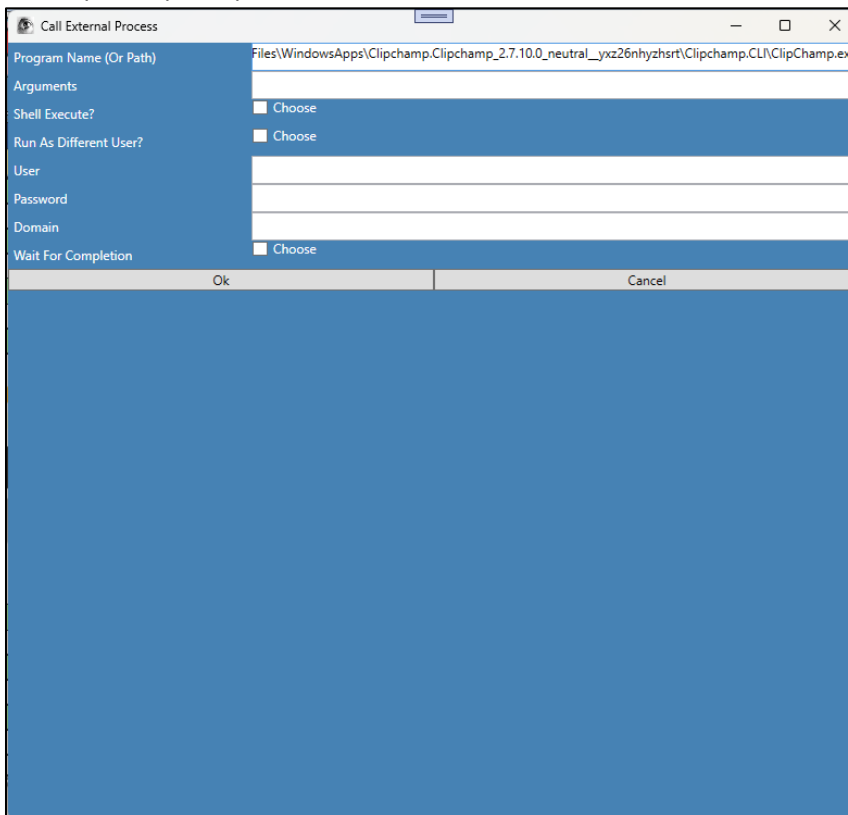


Figure 5: Exe Path

- Current state

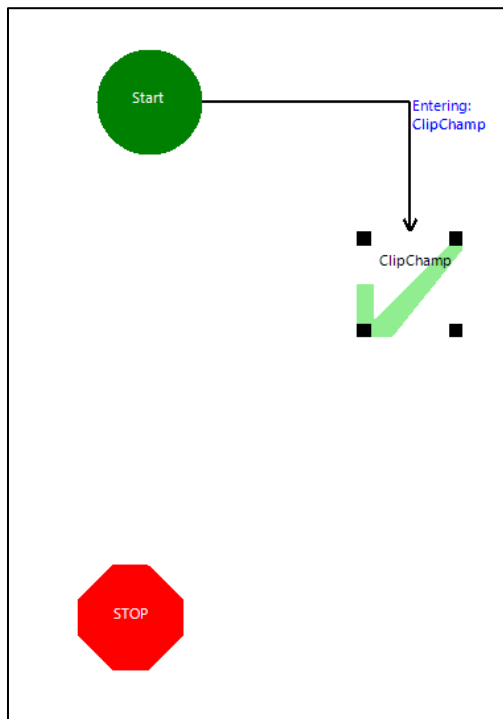


Figure 6: Current State

- Click on **Windows Object Spy** in the **Tools** ribbon menu

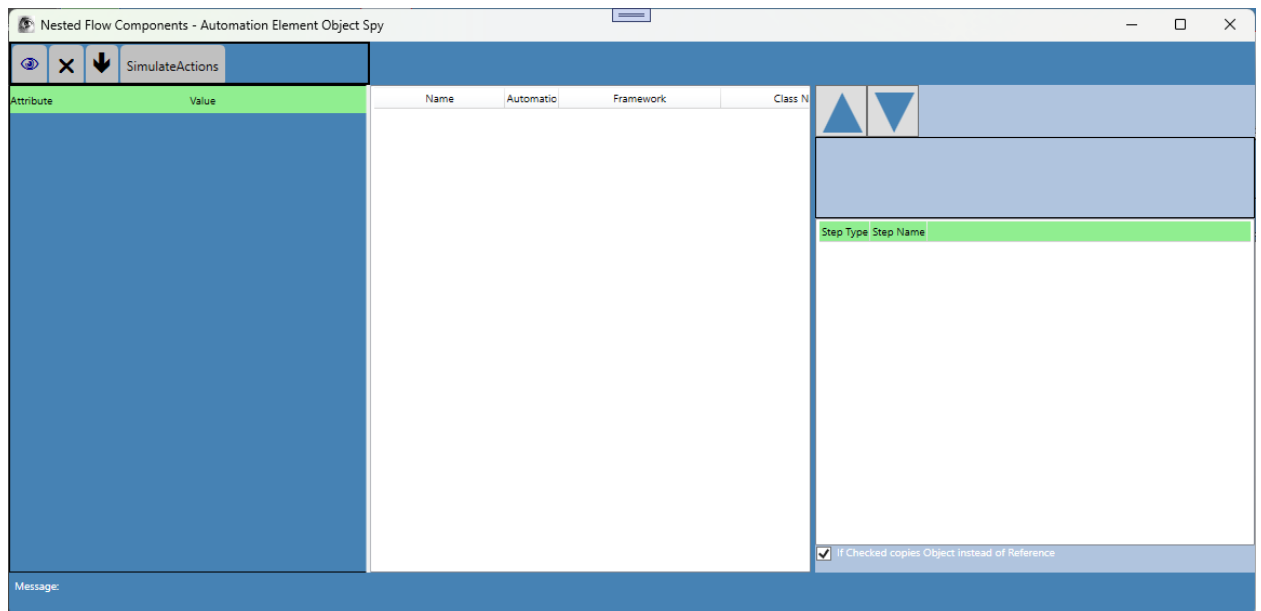


Figure 7: Windows Object Spy

- Open **ClipChamp**

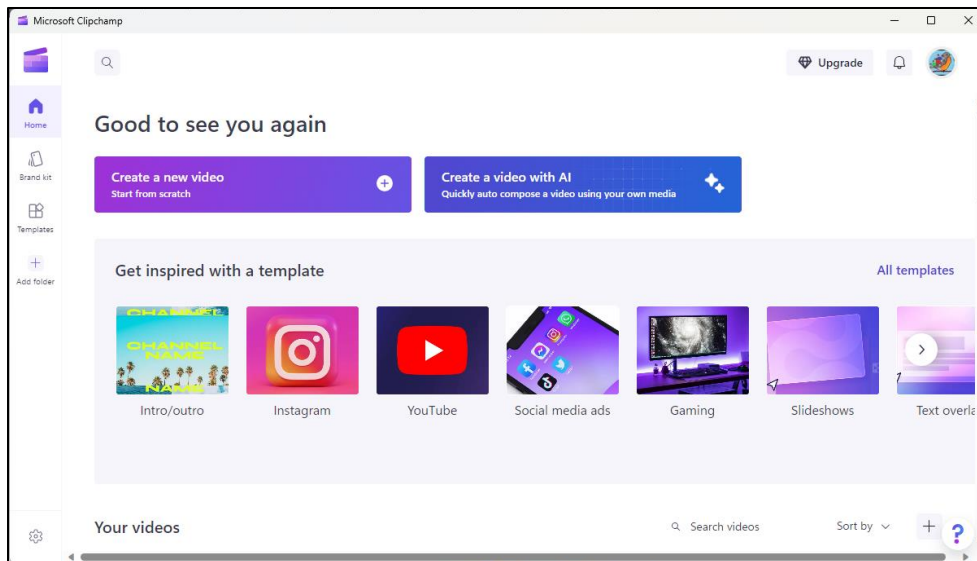


Figure 8: ClipChamp

Object Spy starts identifying the objects

- Capture **Create a new video** button

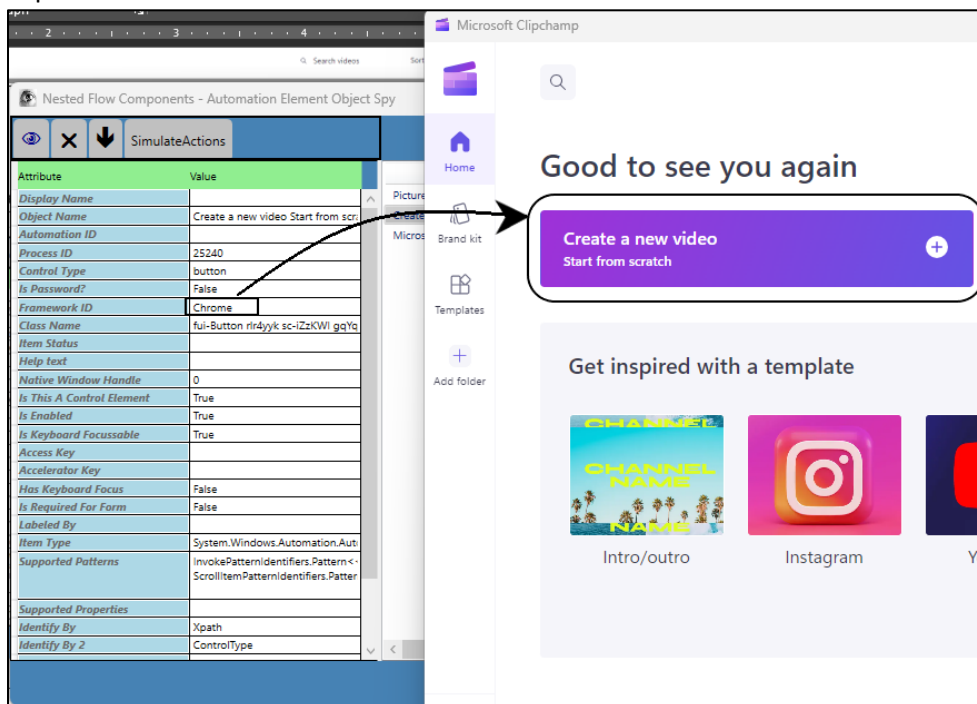


Figure 9: Capture Object Properties

Observe that the Identify By is Xpath

- Click on **SimulateActions** button and choose **Click**



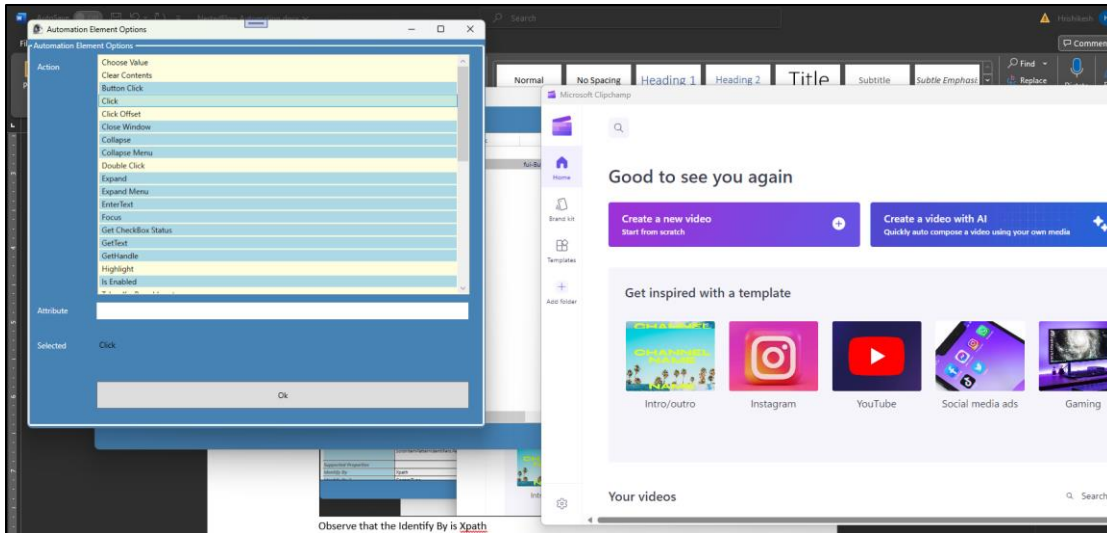
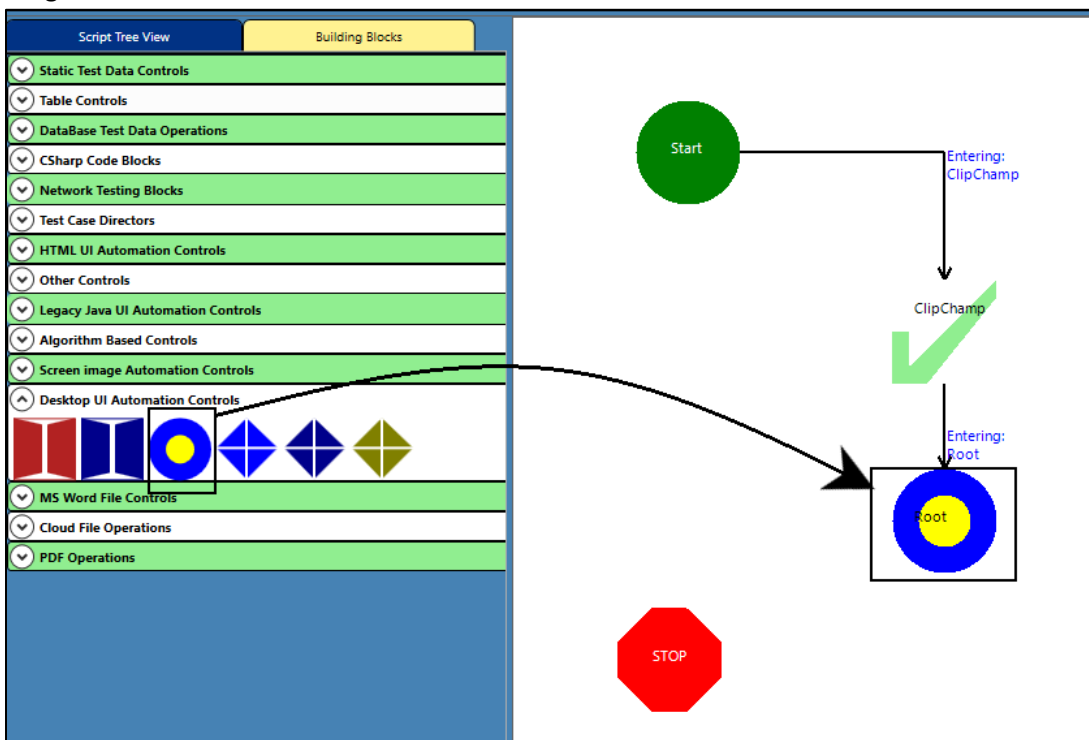


Figure 10: Simulate actions

You will see that the simulate action doesn't work. Which makes **Identify By Xpath** is not a right option for the Chrome framework elements.

- So, Windows Automation Framework parent child relation tree approach is the better approach here
- Drag **Windows Root** element to canvas



- First Let us capture the entire window

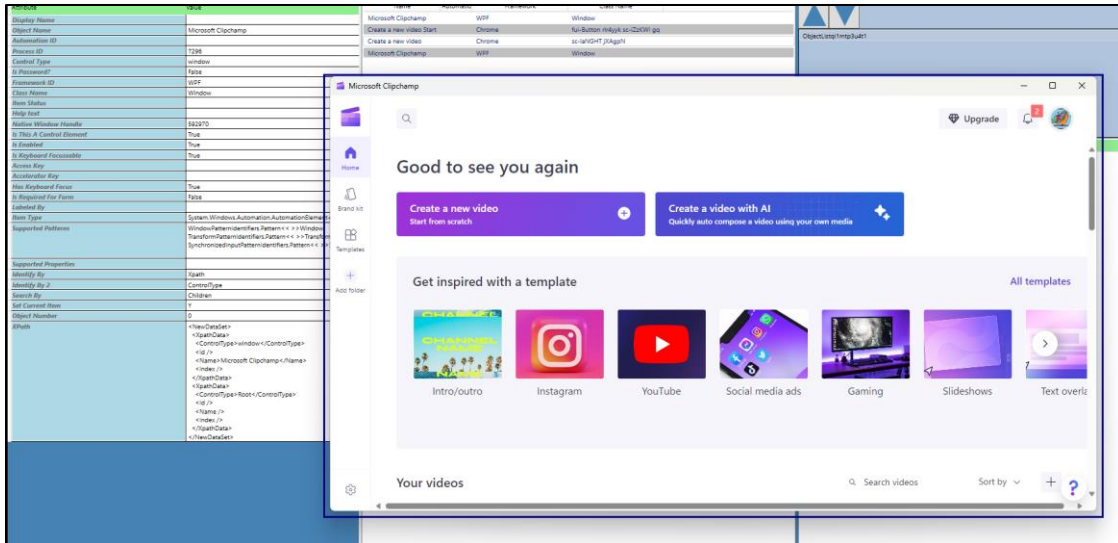


Figure 11: Capture Window

This window has the Name. So make the **Identified by** as Name (double clicking on the field value will allow you to change) Also make sure that the **Set Current Item** is set to Y

- Capture **Create a new video** button properties:

SimulateActions	
Attribute	Value
Display Name	
Object Name	Create a new video Start from scratch
Automation ID	
Process ID	23480
Control Type	button
Is Password?	False
Framework ID	Chrome
Class Name	fui-Button rlr4yyk sc-iZzKWi gqYqsG create __11od2dv ffp
Item Status	
Help text	
Native Window Handle	0
Is This A Control Element	True
Is Enabled	True
Is Keyboard Focussable	True
Access Key	
Accelerator Key	
Has Keyboard Focus	False
Is Required For Form	False
Labeled By	
Item Type	System.Windows.Automation.AutomationElement+Automa
Supported Patterns	InvokePatternIdentifiers.Pattern<< >> Invoke ScrollItemPatternIdentifiers.Pattern<< >> ScrollItem
Supported Properties	
Identify By	Name
Identify By 2	ControlType
Search By	Descendants
Set Current Item	N
Object Number	0
XPath	<NewDataSet>

Figure 12: Descendants

Logic:

1. First the control will be sent to Windows Root (Desktop)
2. Next, Window by Name "Microsoft Clipchamp" which is a **child** of Desktop is chosen and control is set to this item and all future searches will be done from this window

Attribute	Value
Display Name	
Object Name	Microsoft Clipchamp
Automation ID	
Process ID	7296
Control Type	window
Is Password?	False
Framework ID	WPF
Class Name	Window
Item Status	
Help text	
Native Window Handle	592970
Is This A Control Element	True
Is Enabled	True
Is Keyboard Focussable	True
Access Key	
Accelerator Key	
Has Keyboard Focus	True
Is Required For Form	False
Labeled By	
Item Type	System.Windows.Automation.AutomationElement+Automati
Supported Patterns	WindowPatternIdentifiers.Pattern<< >>Window TransformPatternIdentifiers.Pattern<< >>Transform SynchronizedInputPatternIdentifiers.Pattern<< >>Synchroniz
Supported Properties	
Identify By	Name
Identify By 2	ControlType
Search By	Children
Set Current Item	Y
Object Number	0
XPath	<NewDataSet> <XPathData> <ControlType>window</ControlType> <Id /> <Name>Microsoft Clipchamp</Name> <Index /> </XPathData> <XPathData> <ControlType>Root</ControlType> <Id /> <Name /> <Index /> </XPathData> </NewDataSet>

Figure 13:Window Capture

3. We will now capture "**Create a new video Start from scratch**" button with the same name. note that control will still be left at Window level by **Set Current Item = N** for this button

Display Name	
Object Name	Create a new video Start from scratch
Automation ID	
Process ID	23480
Control Type	button
Is Password?	False
Framework ID	Chrome
Class Name	fui-Button rlr4yyk sc-lZzKWl gqYqsG create __11od2dv ffp
Item Status	
Help text	
Native Window Handle	0
Is This A Control Element	True
Is Enabled	True
Is Keyboard Focussable	True
Access Key	
Accelerator Key	
Has Keyboard Focus	False
Is Required For Form	False
Labeled By	
Item Type	System.Windows.Automation.AutomationElement+Automa
Supported Patterns	InvokePatternIdentifiers.Pattern<< >>Invoke ScrollItemPatternIdentifiers.Pattern<< >>ScrollItem
Supported Properties	
Identify By	Name
Identify By 2	ControlType
Search By	Descendants
Set Current Item	N
Object Number	0

Figure 14: Button 1

Note: See the Search By option it is not Children but Descendants.

Children should be used when you have free moving window

For all other objects we should use Descendants

The button is not a direct child but nested Descendant

- Click manually on the button and on the next window capture “Record & Create” button

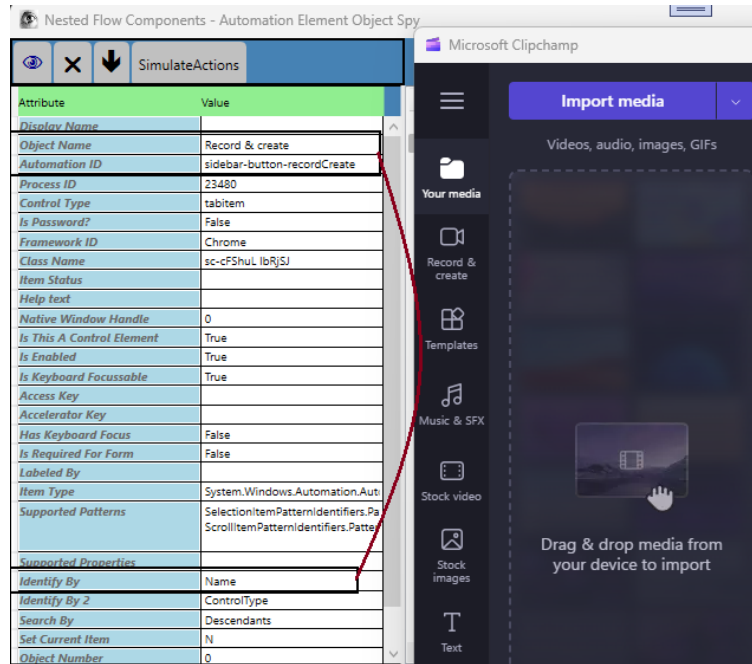


Figure 15: Button 2

Select item by Name or AutomationID (both will work for this item)

Click Manually on the button

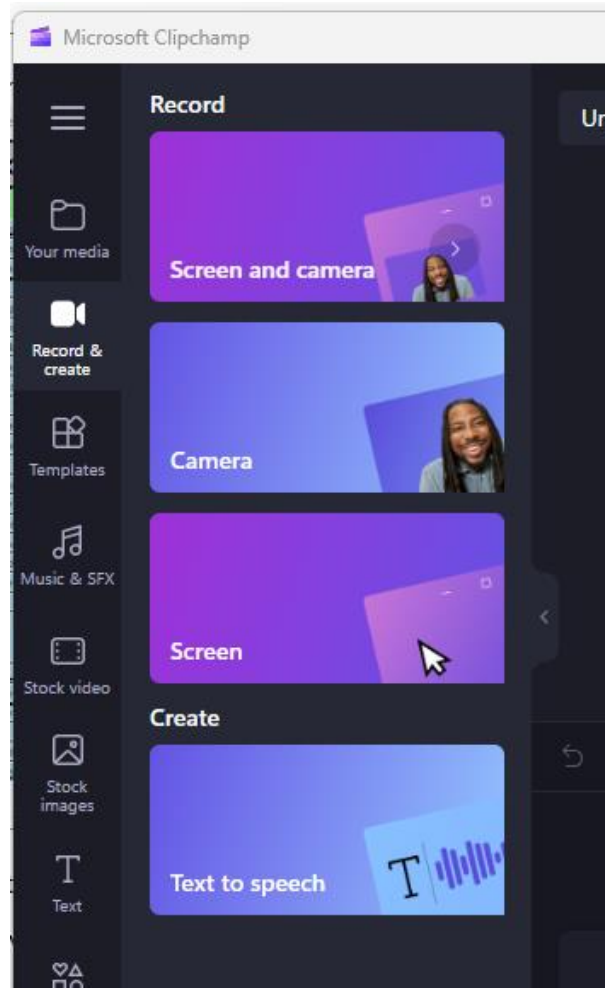


Figure 16: New Objects

5. Capture **Screen** button

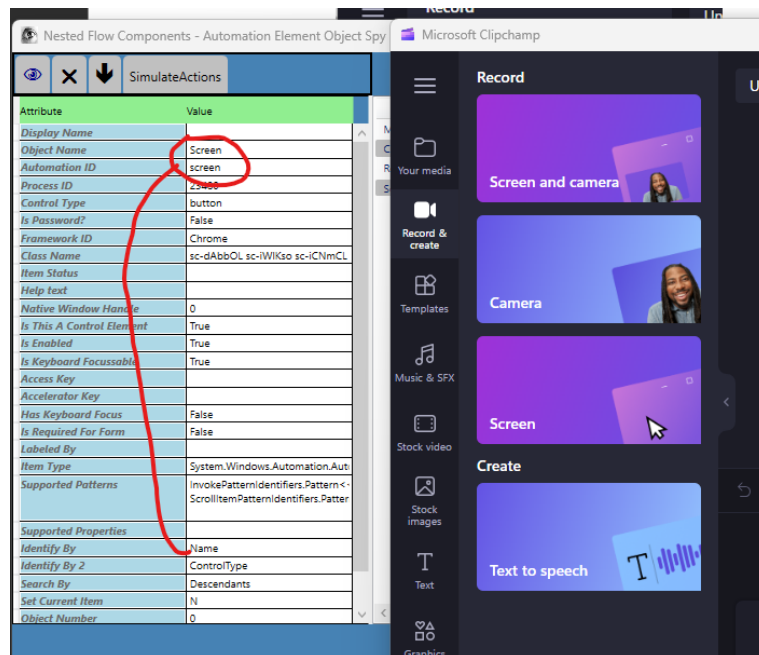


Figure 17: Button 3

Click Manually on the button

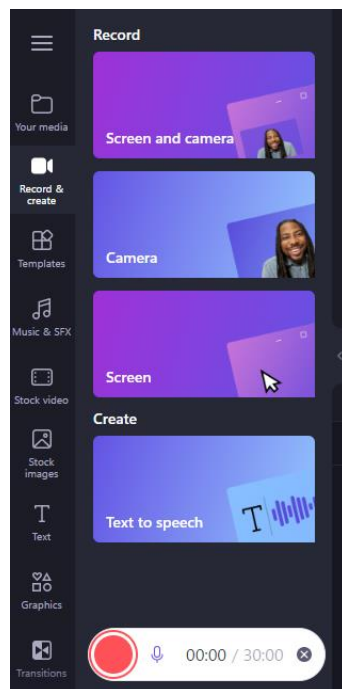


Figure 18: Opens Record Button

6. Capture Red Record button

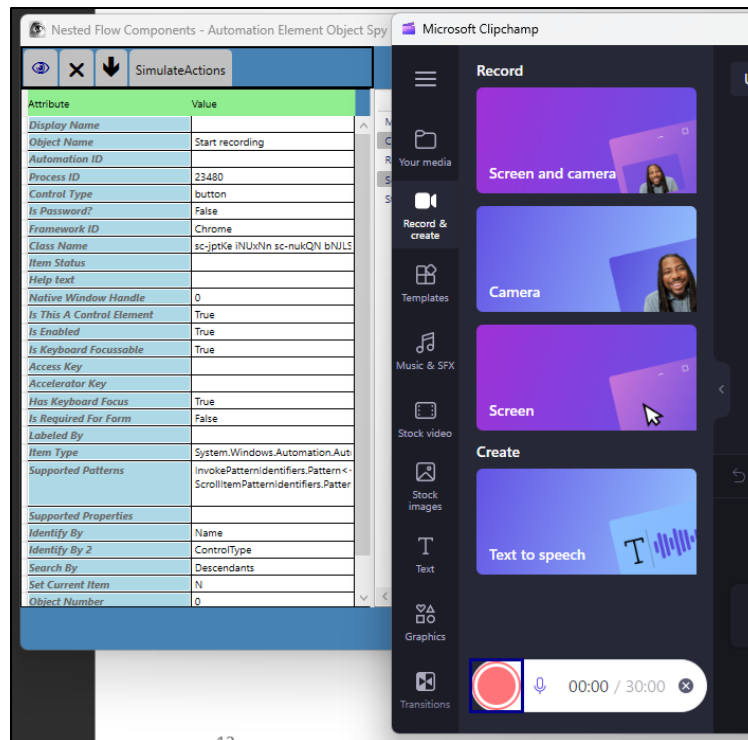


Figure 19: Record Button

Capture by Name and it is a Descendant of the Clipchamp Window  
Click the button manually

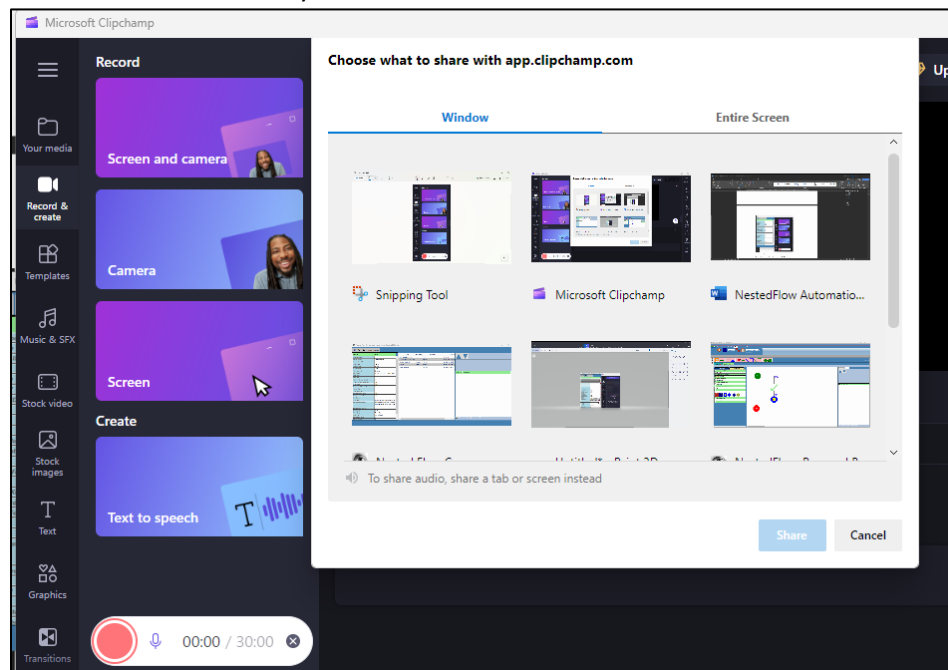


Figure 20: Choices



The choice page opens. We want to choose “Entire Screen” choose record sound and click on Share button but, this Entire Choices window is not recognized. Hence we have to use SendKeys option which is in **HTML UI Automation Controls** section

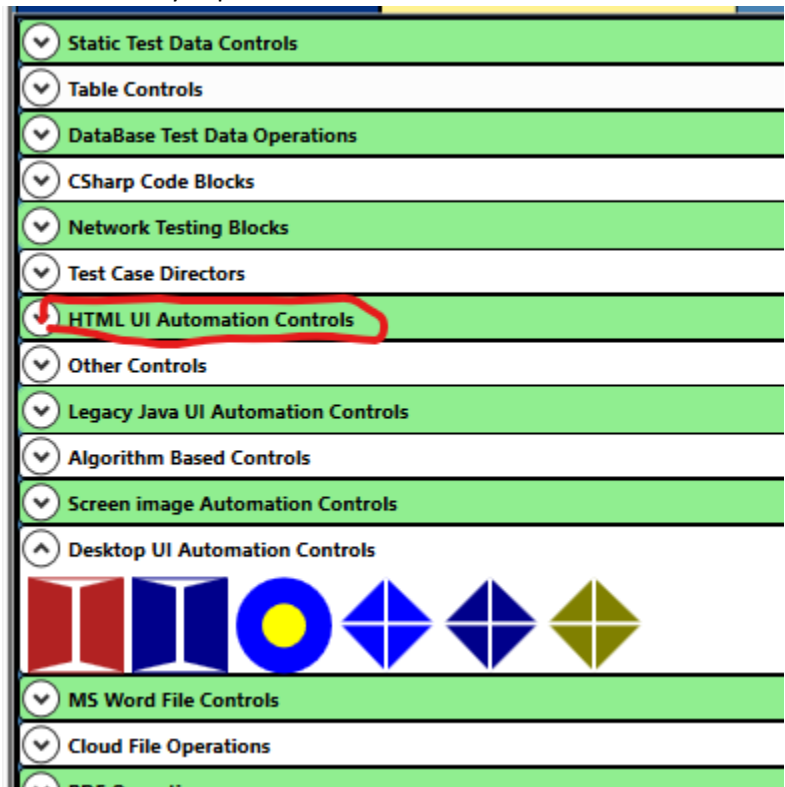


Figure 21: Tool box

- All the objects we need for this automation are captured. Now click on double click on each item so that the items land in Object list

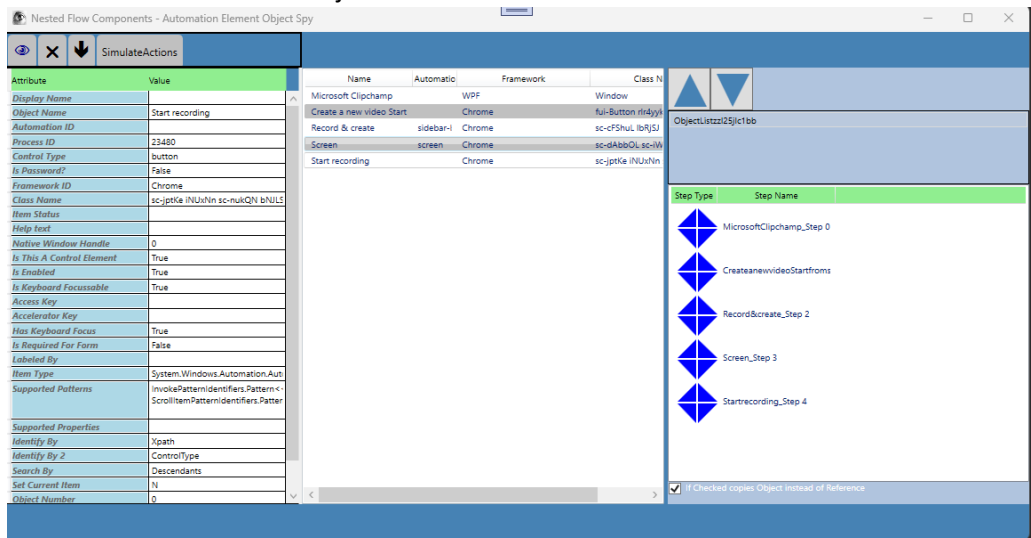



Figure 22: Object List

- Click on Return button 
- All the items will be available in the Object List to the right of Canvas

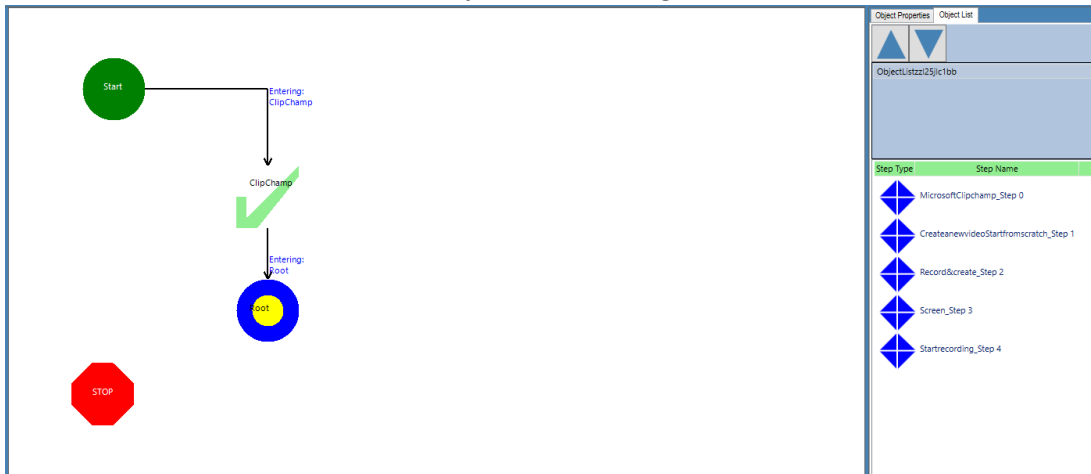


Figure 23: Object List on Canvas

- After the Root, create a Wait for window object to wait for clip champ to completely load

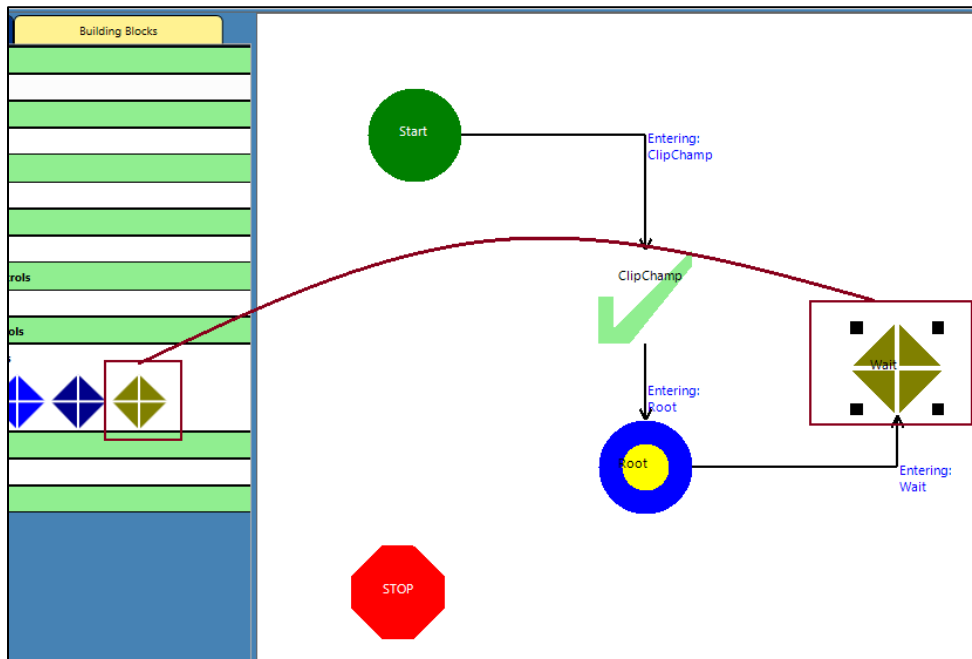
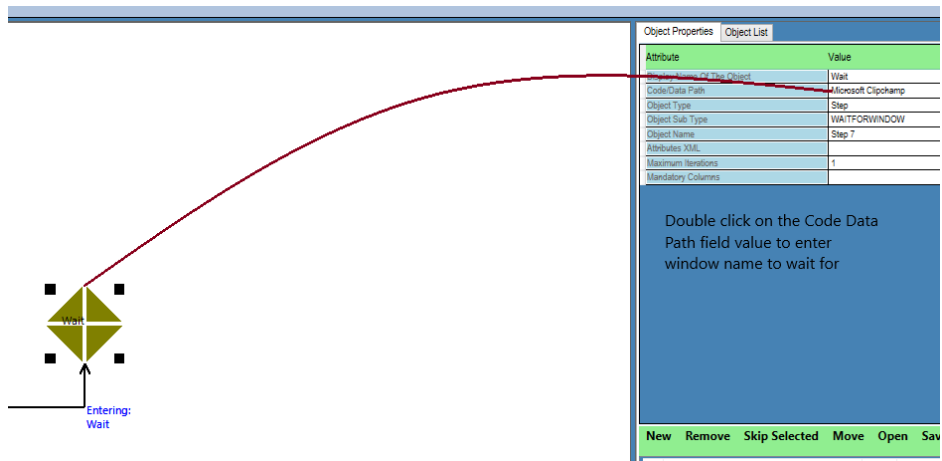
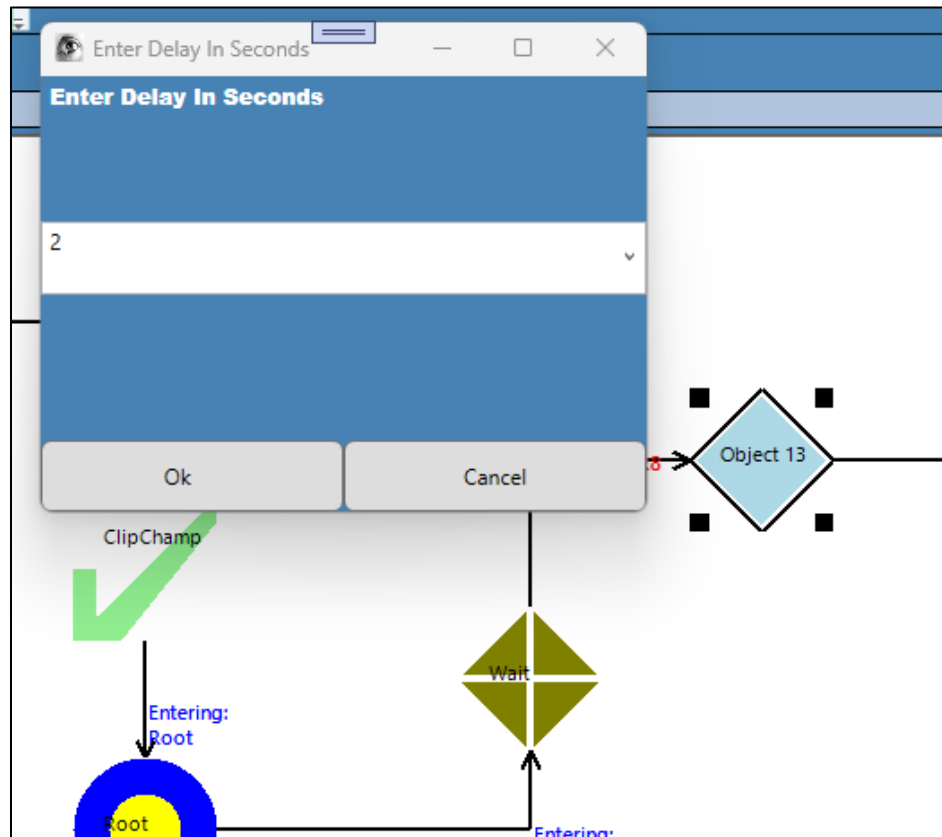


Figure 24: Wait for windows

- Enter window to wait as **Microsoft Clipchamp**



- Some times it is suggested to add Delay object to create a delay for the Clipchamp components to load fully



- Drag the entire Object page to Canvas and join the wait object to the first Automation Element object on the page

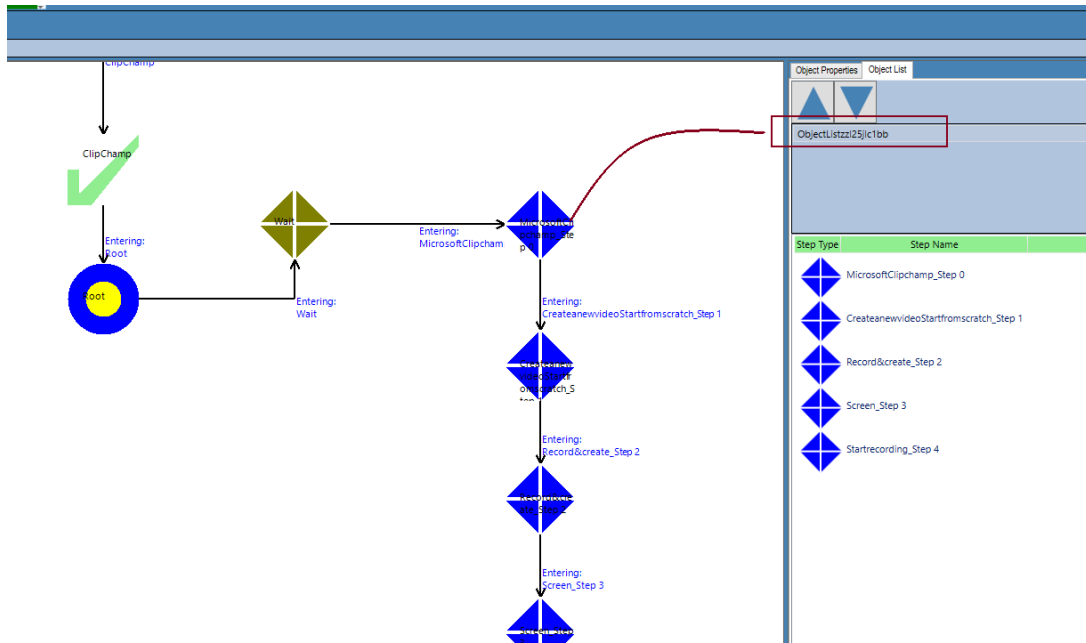


Figure 25: Drag Page

- Right click on the first Automation Element object (which is the main window of Clipchamp) and click on **Add Windows Object Action** menu
- Choose **Focus** action for this object. For All the other objects choose **Click** Action

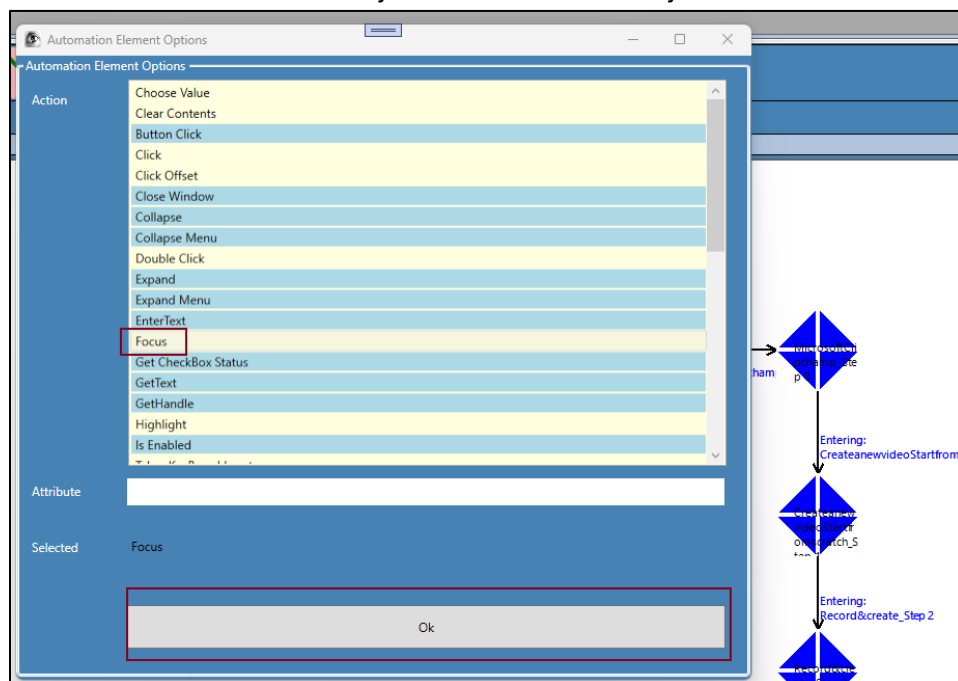


Figure 26: Action

- Add a control + Tab sendkeys simulation

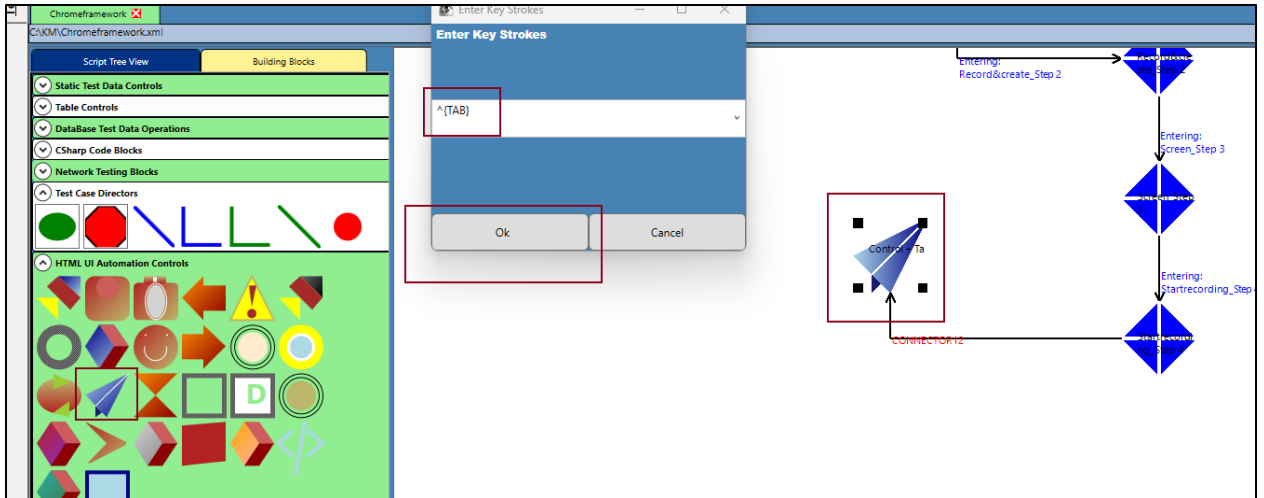


Figure 27: Send Keys

- Add another sendkeys to simulate TAB + TAB + INSERT + TAB + ENTER

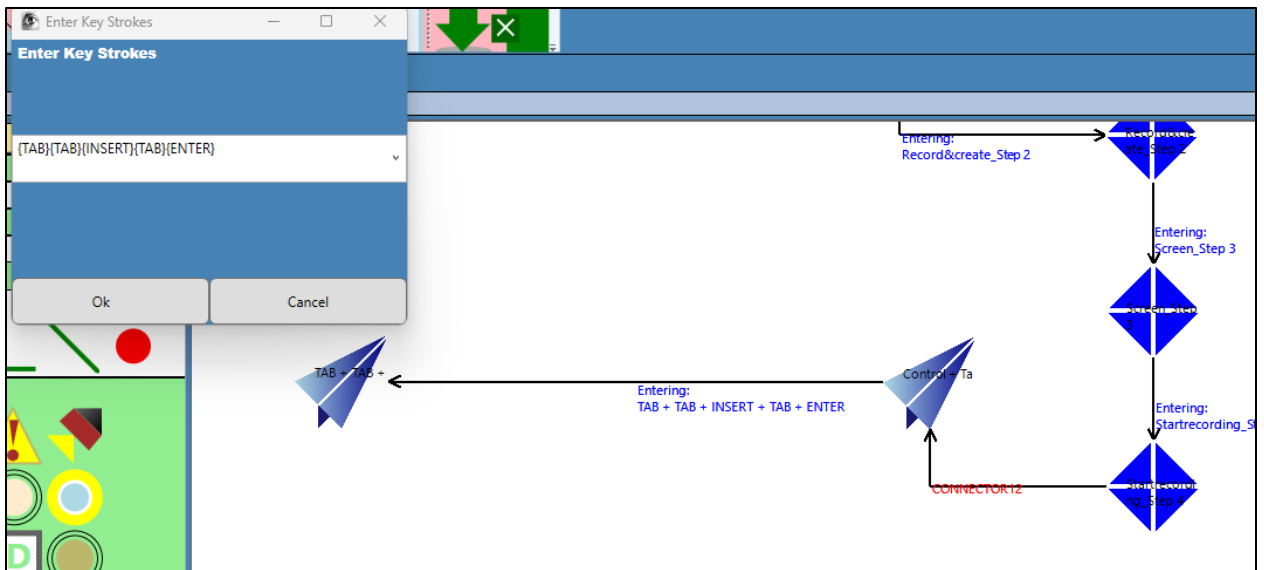


Figure 28: Send Keys 2

Note: Sendkeys reference <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.sendkeys.send?view=windowsdesktop-7.0>

- Add delay to create a 5 second screen video

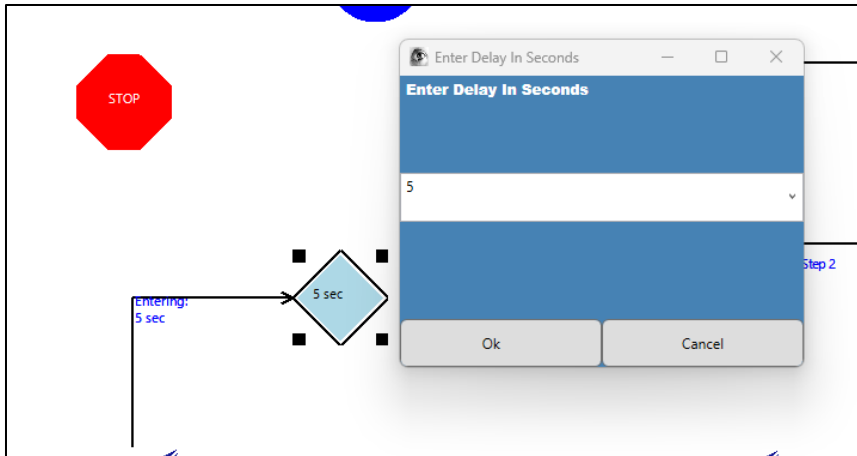


Figure 29: Delay

- Capture properties of stop recording and add that also to the flow

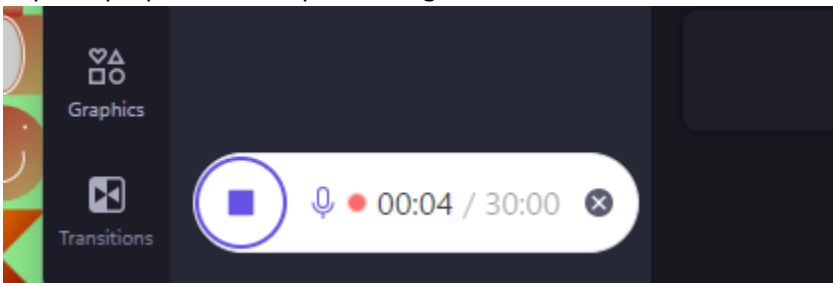


Figure 30: Stop Recording

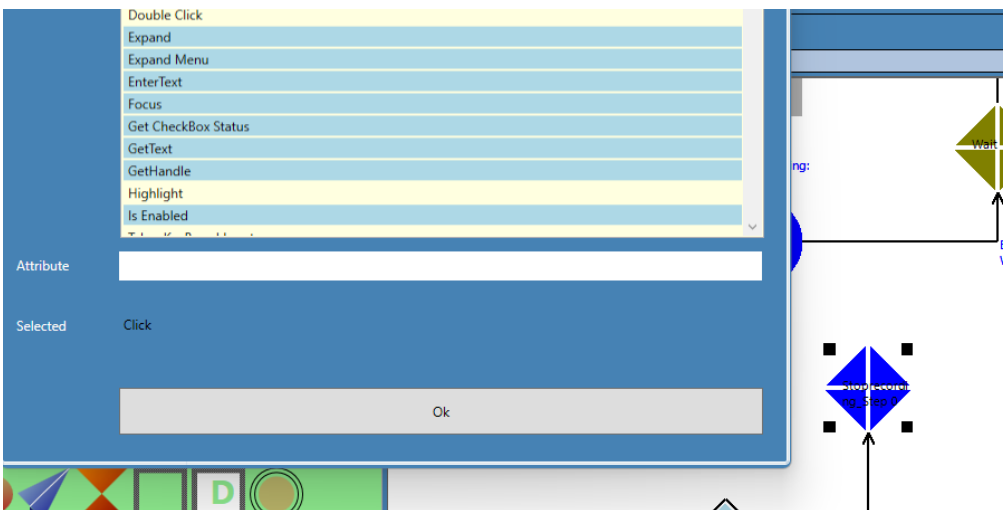
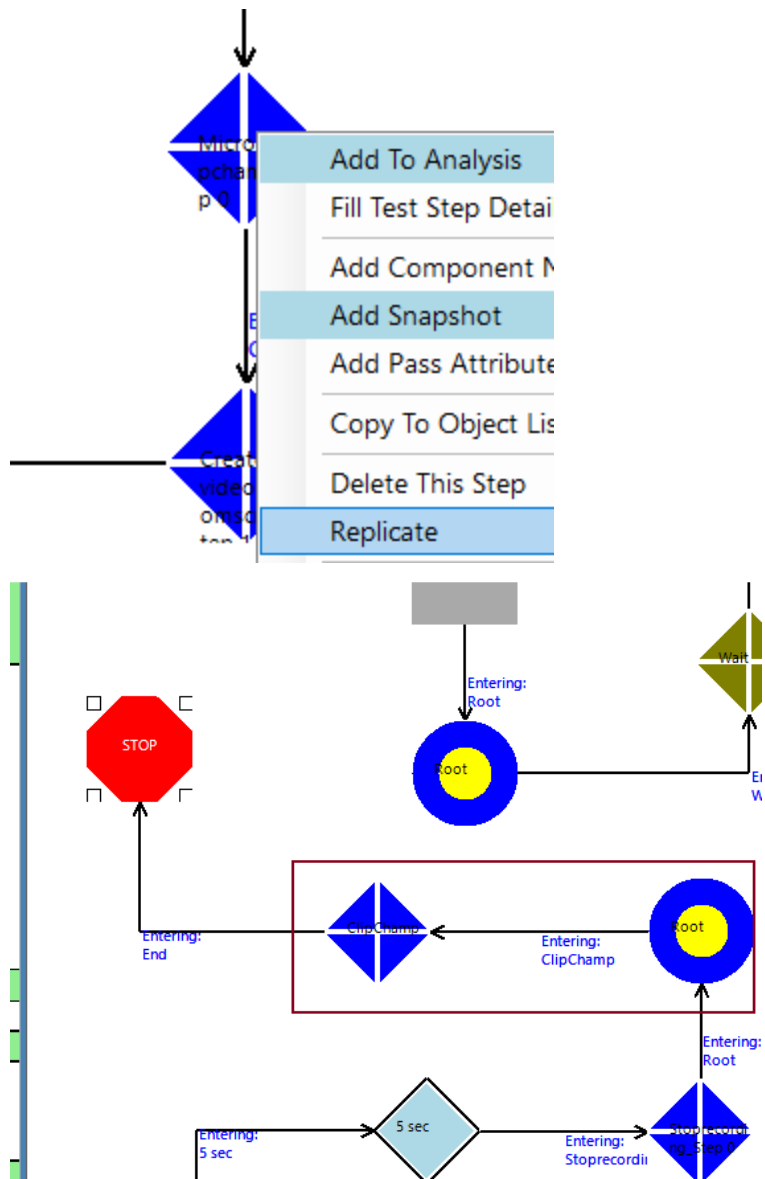


Figure 31: Click Stop Recording

- Since the control is not with root, create another Root element so that the object can be recreated for closing the form and then replicate the clipchamp window object to create copy



- Choose "Close Form" action for the Clipchamp window and join it to the End Block.
- The completed flow with descriptions is as follows

