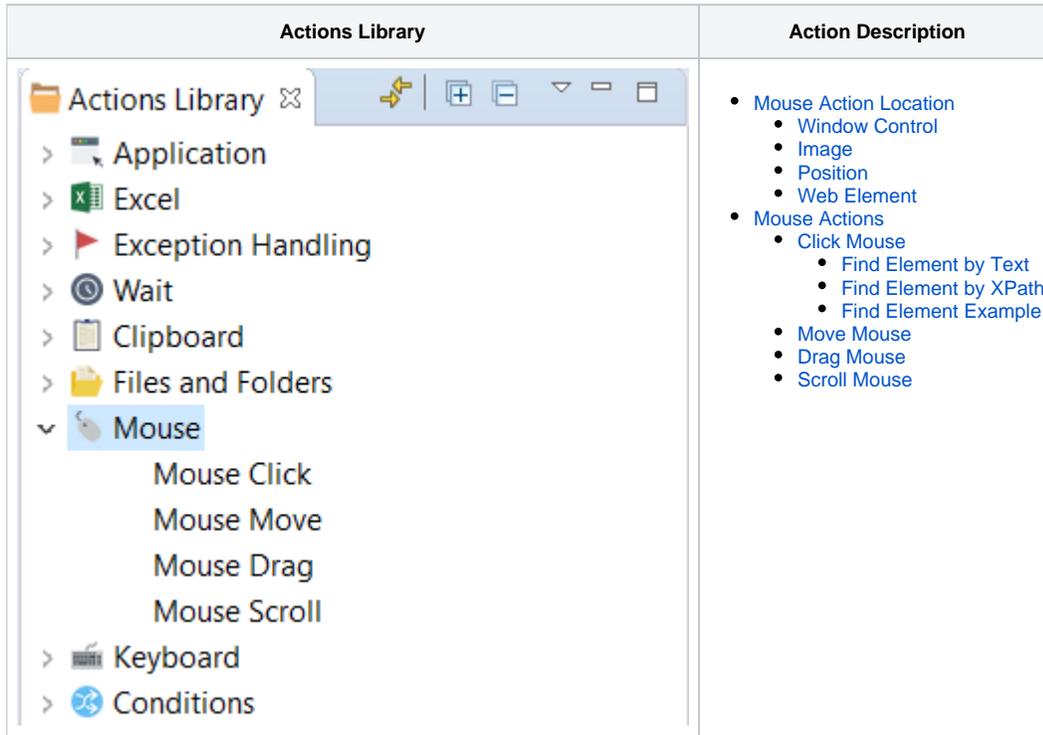


# Mouse

 We have moved to a new documentation platform. This section is no longer supported. For the up-to-date information, see [Mouse](#).

The Mouse action group is intended to interact with user interface using mouse. All the **Mouse** actions are automatically recorded when you manipulate your mouse during the [recording](#) process.

You can add the Mouse [Click](#), [Move](#), [Drag](#), or [Scroll](#) actions manually by dragging from [Actions Library](#) to Actions Flow.



## Mouse Action Location

Each mouse action needs a specific location (target), that is **a place on the computer screen or application where the action is applied to**. Currently, WorkFusion Studio supports four types of targets.

- [Window control](#) – a Windows Desktop application component selected by its [property](#), e.g., `Class`, `Name`, etc. Refer to [Inspector](#) for more information.
- [Image](#) – a graphical image of a UI control (button, input field, link, etc.) which Bot tries to find and click (hover, drag, scroll) while script execution. See [Editing Images](#) for more details.
- [Position](#) – X and Y coordinates of a point from the top left screen corner.
- [Web element](#) – an XML or HTML document element selected using a [Text](#) or [XPath](#) expression. That is the most reliable way to specify the location, as it depends on the web document (page) structure, not on the color scheme or location on the screen that may vary from one machine to another.

## Window Control

In this mode, WorkFusion Studio [captures UI controls of Windows applications](#) selected for Bot to click during script execution.

 The mode is used for Windows applications only, i.e. **you cannot select an object on a web page**, or inside certain documents.

By default, RPA Recorder uses Window controls for capturing user actions in the recording mode and tries to find the same elements and click on them during script execution. If you record your actions in a browser, Window controls are recorded by default for mouse actions on the web page, but will fail during execution. **You need to edit mouse actions in the browser by switching them to the [Image](#) mode** (images are recorded as a fallback option).

Actions Flow	Active	#
Launch Application (calc; wait for 0ms; poll every 0ms)	<input checked="" type="checkbox"/>	1
Window (Calculator)	<input checked="" type="checkbox"/>	2
Click Mouse (click left button)	<input checked="" type="checkbox"/>	3
Click Mouse (click left button)	<input checked="" type="checkbox"/>	4
Click Mouse (click left button)	<input checked="" type="checkbox"/>	5
Click Mouse (click left button)	<input checked="" type="checkbox"/>	6
Enter Keystrokes (Ctrl+c)	<input checked="" type="checkbox"/>	7
Launch Application (notepad; wait for 0ms; poll every 0ms)	<input checked="" type="checkbox"/>	8
Enter Keystrokes (Ctrl+v)	<input checked="" type="checkbox"/>	9

**Click mouse**

**Target**

Window control  Image  Position  Web element

Selector ?

[CLASS:Button; NAME:7]

[Capture from screen](#)

**Options**

Button: ?

Left button

Type: ?

Single click

**Advanced**

**Comments**

To set up the **Click Mouse** action for a **UI control**, follow a short instruction below.

**i** Before using **Click Mouse** for a control, make sure to switch to the application window with the help of the **Window** action.

1. Select **Window control** in **Target**.
2. Enter a **selector** for your application control.

- a. Use the **Capture from screen** option (1) and click the desired window control in the application (2). The selector is automatically entered into the **Selector** field (3).

**i** There are cases when you need to click a control without the selector being assigned to the element, e.g., to explore other application options or switch to another application. Use the black menu as a prompt.

- i. To click the control without its selector being selected, use the left mouse click + **Ctrl**.
- ii. To cancel capturing, press **Esc**.

- b. Use **Inspector** to view and select properties of you application objects. For more details, refer to the **Inspector** section.

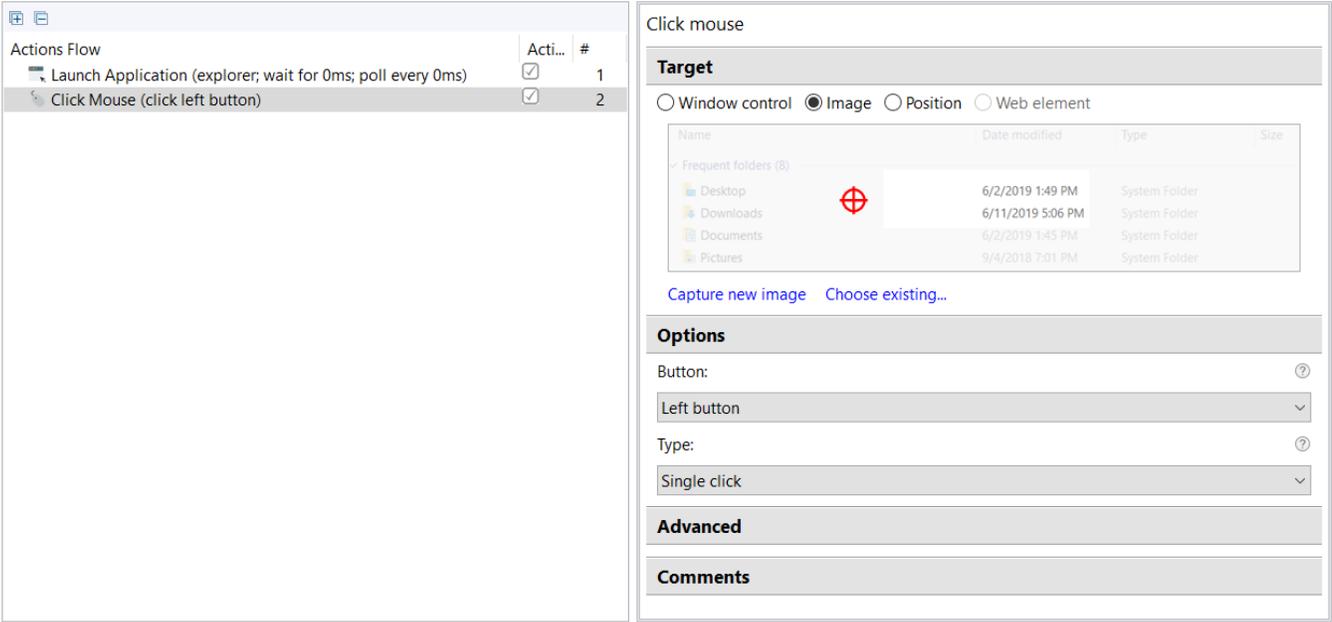
### Use Case

In the example above, the button "7" is selected in Calculator.

1. Make sure Calculator is open and click **Capture from screen**.
2. Switch to Calculator and click **7**.
3. The element selector consisting of the `Class` and `Name` properties is automatically inserted into the **Selector** field.

## Image

By default, Bot clicks (hovers, scrolls, drags) directly into the center of the captured image.



Name	Date modified	Type	Size
Frequent folders (8)			
Desktop	6/2/2019 1:49 PM	System Folder	
Downloads	6/11/2019 5:06 PM	System Folder	
Documents	6/2/2019 1:45 PM	System Folder	
Pictures	9/4/2018 7:01 PM	System Folder	

[Capture new image](#) [Choose existing..](#)

**Options**

Button: 

Left button 

Type: 

Single click 

**Advanced**

**Comments**

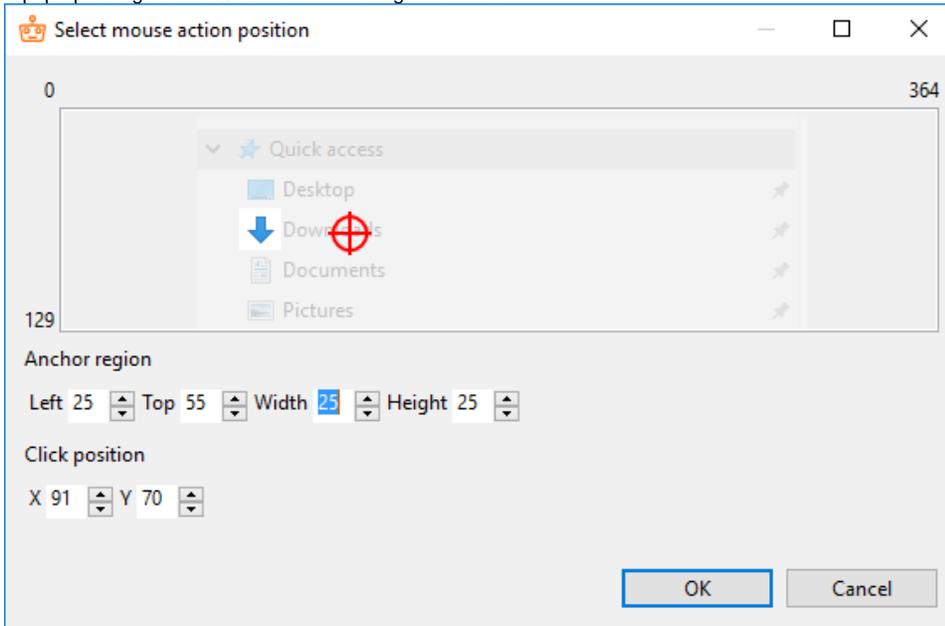
To set the crosshair to a custom place, do as follows.

1. Click the image.



[Capture new image](#) [Choose existing...](#)

A pop-up dialog with 100% zoom on the image is shown.



2. Set up the location for Bot to click.

- **Anchor region** – the selected area that defines what Bot should find on the screen (an anchor) as a reference point to location where it should perform the action.
  - Edit values in the **Left**, **Top**, **Width**, **Height** fields,
  - Or use **Arrow Up/Arrow Down** to increase/decrease the value to 1 respectively, or **PgUp/PgDn** keys to increase/decrease the value to 5 respectively.
- **Click position** – the place for Bot to click.
  - Click on the image to move the crosshair to this position,
  - Or set the values in the **X** and **Y** fields.

### Purpose of Anchor and Crosshair

When executing a script, Bot:

1. in the currently open window, searches for an area that looks practically the same as the **Anchor** area.
2. if the area is found, performs action at the **Crosshair** position (determined by the **Offset** parameter).
3. if the area is not found, informs about an error occurred, and the execution stops.

#### Useful Tips:

- The image in the **Anchor** area should contain static info (input label, icon, heading, table column name) that will not be changed if new data is displayed.
- Bot will NOT search for image that is outside the **Anchor** area (grayed-out part).
- The crosshair can be placed inside or outside the **Anchor** area, depending on where the action is applied.

## Position

 Clicking on position is not a stable solution, because it requires the same window size, screen resolution, and content layout for each Bot launch.





Mouse actions by position in web are NOT supported.

Do not select this option if the Mouse action is nested in **Open Website** or **Switch to Browser** action.

To make Bot click on screen coordinates, do as follows.

1. Select the **Position** option.
2. Enter the **X** and **Y** coordinates of the action point (from the top-left screen corner).

Click mouse

**Target**

Window control  Image  Position  Web element

---

X:  Y:  Measured from the top left corner of the screen

**Options**

Button:

Type:

**Advanced**

**Comments**

## Web Element



### Attention

You can use **Click Mouse** with **Web element** with the [Open Website](#) action only.

The action is performed on web page elements, e.g., buttons, links, or input fields.

## Mouse Actions

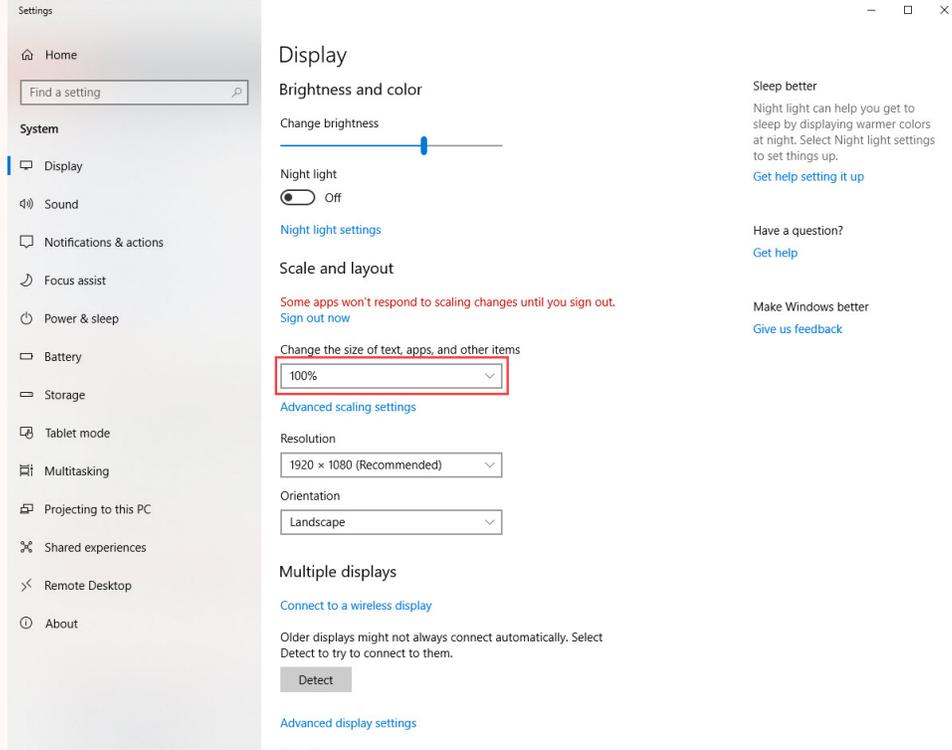


### Warning!

If you use Internet Explorer as a browser for automating, make sure your Windows display scaling is set up to 100%, otherwise a Bot will not be able to click on the needed elements.

1. Go to **Start > Settings > System > Display**.

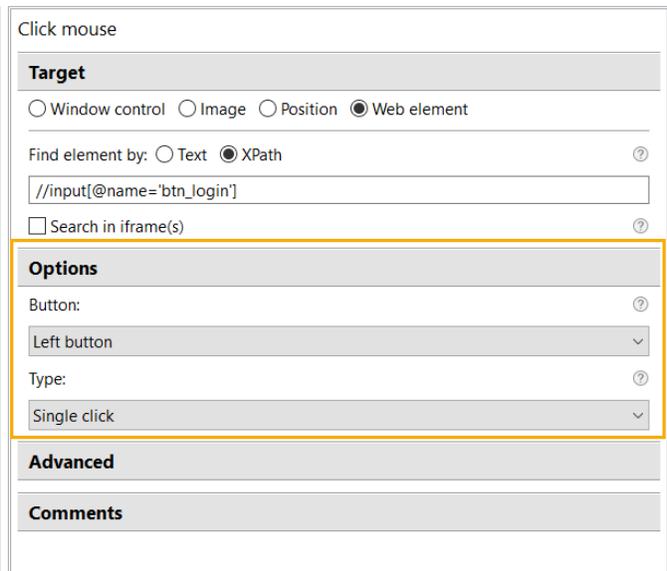
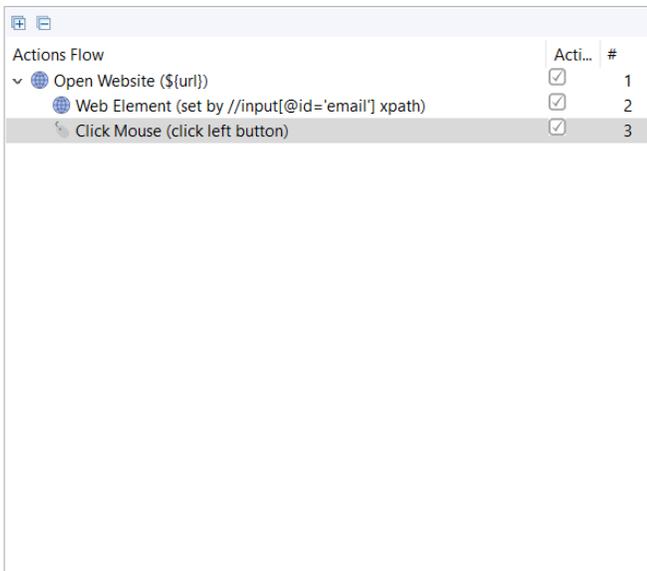
2. In **Scale and Layout**, select 100% for **Change the size of text, apps, and other items**. The changes apply automatically.



## Click Mouse

The Click Mouse action represents a mouse click on a given image, XPath, or position, and supports different **types of clicks** and **buttons**.

- Mouse button:
  - left
  - right
  - middle (wheel)
- Types of click:
  - single
  - double
  - triple
  - hold down
  - release



## Find Element by Text



### Attention

You can use **Click Mouse** with **Web element** with the [Open Website](#) action only.

1. Add **Click Mouse** to **Open Website**, choose **Web element** as **Target**, and select the **Find element by Text** option.

The screenshot shows the UiPath Actions Flow on the left and the Click mouse configuration panel on the right. In the Actions Flow, the 'Click Mouse (click left button)' action is selected. The configuration panel shows the following settings:

- Target:** Web element (selected)
- Find element by:** Text (selected)
- Text:** Robotic Process Automation
- Exact match:** checked
- Search for links only:** checked
- Search in iframe(s):** unchecked
- Options:** Button: Left button, Type: Single click
- Advanced:** (empty)
- Comments:** (empty)

2. Enter **Text** of your link in the respective field.
3. Select **Exact match** to enable the strict search.
4. Select **Search for links only** to search for the text in the links only, e.g. in the HTML elements enclosed in the `<a>` tag.



### Example

When you use **Find element by Text** with selected **Search for links only**, e.g.,

#### Text

My Site

Bot clicks the following link.

```
<a href='http://mysite.com'>My Link</a>
```



#### Attention!

If the text in your link is enclosed to some other tags, e.g., `<bold>`, Bot is not able to find the text.

```
<a href='http://mysite.com'><bold>My Link</bold></a>
```

5. Select **Search in iframe(s)**, if the link you need to click on is located in an iframe (or iframes) as shown on the image below.



### Note

The iframe containing the link can be enclosed in another iframe. In this case, you should define each iframe separately.

XPath of parent iframe(s) ?

<code>{target_iframe}</code>	✘
<code>{parent_iframe}</code>	✘
+ Add XPath of iframe	

For this purpose, click **Add XPath frame** and define the iframe that holds the iframe with the link.

When defining the iframes, you should keep to the following order.

- The first iframe – the iframe containing the link.
- The second iframe – the parent iframe that the previous iframe is enclosed in.
- More enclosed iframes – all of them are defined one by one so as the root iframe takes the last place.

### i Frame vs. iframe

Frames and iframes perform similar functions of embedding a resource into the web page, still they are different. The frameset splits a browser window into multiple frames, with each frame having its own contents not intercepting the other frame. An iframe, on the other hand, embeds a frame directly in line with other elements of a web page. In spite of this difference, **frames and iframes are treated in the same way in WorkFusion Studio.**

- Choose the button to be used and the click type.

## Find Element by XPath

### ! Attention

You can use **Click Mouse** with **Web element** with the **Open Website** action only.

- Add **Click Mouse** to **Open Website**, choose **Web element** as **Target**, and select the **Find element by XPath** option.

Actions Flow	Acti...	#
Open Website (\$url)	<input checked="" type="checkbox"/>	1
Click Mouse (click left button)	<input checked="" type="checkbox"/>	2
Wait for 3000 ms	<input checked="" type="checkbox"/>	3

Click mouse

**Target**

Window control  Image  Position  Web element

Find element by:  Text  XPath

`{web_element}`

Search in iframe(s)

XPath of parent iframe(s)

`{target_iframe}` ✘

+ Add XPath of iframe

< >

**Options**

Button:

Type:

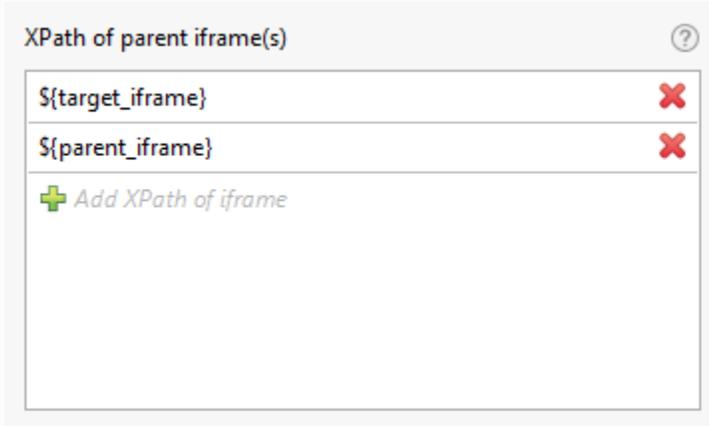
**Advanced**

**Comments**

- Enter your XPath in the respective field. For more details see the [How to get XPath](#) section.
- Search in iframe(s)** – use the option, if the element you need to click on is located in an iframe (or iframes).

### i Note

The iframe containing the web element can be enclosed in another iframe. In this case, you should define each iframe separately.



For this purpose, click **Add XPath of iframe** and define the iframe which holds the iframe with the web element.

When defining the iframes you should keep the following order.

- a. The first iframe – the iframe containing the web element.
- b. The second iframe – the parent iframe that the previous iframe is enclosed in.
- c. More enclosed iframes – all of them are defined one by one so as the root iframe takes the last place.

4. Choose the button to be used and the click type.

## Find Element Example

In this sample we will use Express Edition to open **Wikipedia**, search for **RPA**, and open the page referred to **Robotic Process Automation**.

#	Step	Recorder Action	Settings
1	Open the website.	Open Website	<b>Site URL</b> Value: <a href="https://www.wikipedia.org/">https://www.wikipedia.org/</a>
3	Enter text to search for.	Web Element	<ol style="list-style-type: none"> <li>1. <b>Mode:</b> Set value</li> <li>2. <b>Input:</b> <code>\${text_to_search} = RPA</code></li> <li>3. <b>Options XPath of the element:</b> <code>//input[@id="searchInput"]</code></li> </ol>
4	Start search.	Enter keystrokes	<ol style="list-style-type: none"> <li>1. <b>Mode:</b> Key combination</li> <li>2. <b>Input:</b> ENTER</li> </ol>
5	Click the link in the search results to go to the page.	Click Mouse	<ul style="list-style-type: none"> <li>• <b>Target</b> <ul style="list-style-type: none"> <li>• <b>Web element</b></li> <li>• <b>Find element by Text</b></li> <li>• Text to find: <code>Robotic Process Automation</code></li> <li>• Select <b>Exact match</b></li> <li>• Select <b>Search for links only</b></li> </ul> </li> <li>• <b>Options</b> <ul style="list-style-type: none"> <li>• <b>Button:</b> Left button</li> <li>• <b>Type:</b> Single click</li> </ul> </li> </ul>

The screenshot displays the RPA Express interface. On the left, the 'Actions Flow' pane shows a sequence of five actions: 'Open Website (\$url)', 'Web Element (set by //input[@id=searchInput] xpath)', 'Enter Keystrokes ((ENTER))', 'Click Mouse (click left button)', and 'Wait for 1200 ms'. The 'Click Mouse' action is selected. Below it, a 'Recorder Variables' window shows a table with the following data:

Name	Type	Default Value
url	String	https://www.wikipedia.org/
text_to_search	String	RPA
+ Add variable...		

On the right, the configuration panel for the 'Click mouse' action is shown. It includes sections for 'Target', 'Options', and 'Advanced'. The 'Target' section has radio buttons for 'Window control', 'Image', 'Position', and 'Web element' (selected). Below this, there are options for 'Find element by:' (Text selected), a search text field containing 'Robotic Process Automation', and checkboxes for 'Exact match' and 'Search for links only'. The 'Options' section includes a 'Button:' dropdown set to 'Left button' and a 'Type:' dropdown set to 'Single click'.

Play the recording to make sure, that all data are entered correctly and the Robotic Process Automation page opens.

You can [download sample recording](#) for further tests.

**Note**

The recording was made with RPA Express 1.4.0.

## Move Mouse

The action represents a **mouse move (hover)** on a given image or coordinates and is recorded when user stops moving cursor for more than for 1 second.

Recording of the **Move Mouse** action is disabled by default. You can enable it in WorkFusion Studio Preferences.

recording-1499685604751.rpae

Replace with new recording
  Insert recording

Actions Flow	Active	#
Window (Desktop)	<input checked="" type="checkbox"/>	1
Click Mouse (click left button)	<input checked="" type="checkbox"/>	2
Click Mouse (click left button)	<input checked="" type="checkbox"/>	3
Move Mouse	<input checked="" type="checkbox"/>	4
Click Mouse (release left button)	<input checked="" type="checkbox"/>	5

**Move mouse**

Move to image
  Move to coordinates

Target location (click on image to edit)

access.log

Dropbox (WorkFusion)

eclipse

Offset X: 125 Offset Y: 35

[Capture new image](#)
[Choose new image](#)

---

**Advanced**

Comments:

**Use case**

The action can be used for automating dynamic menus navigation where you need to hover over an item to select its child.

The screenshot shows a menu structure where the 'Other...' option is selected under the 'Open Perspective' sub-menu, which is under the 'Perspective' menu item.

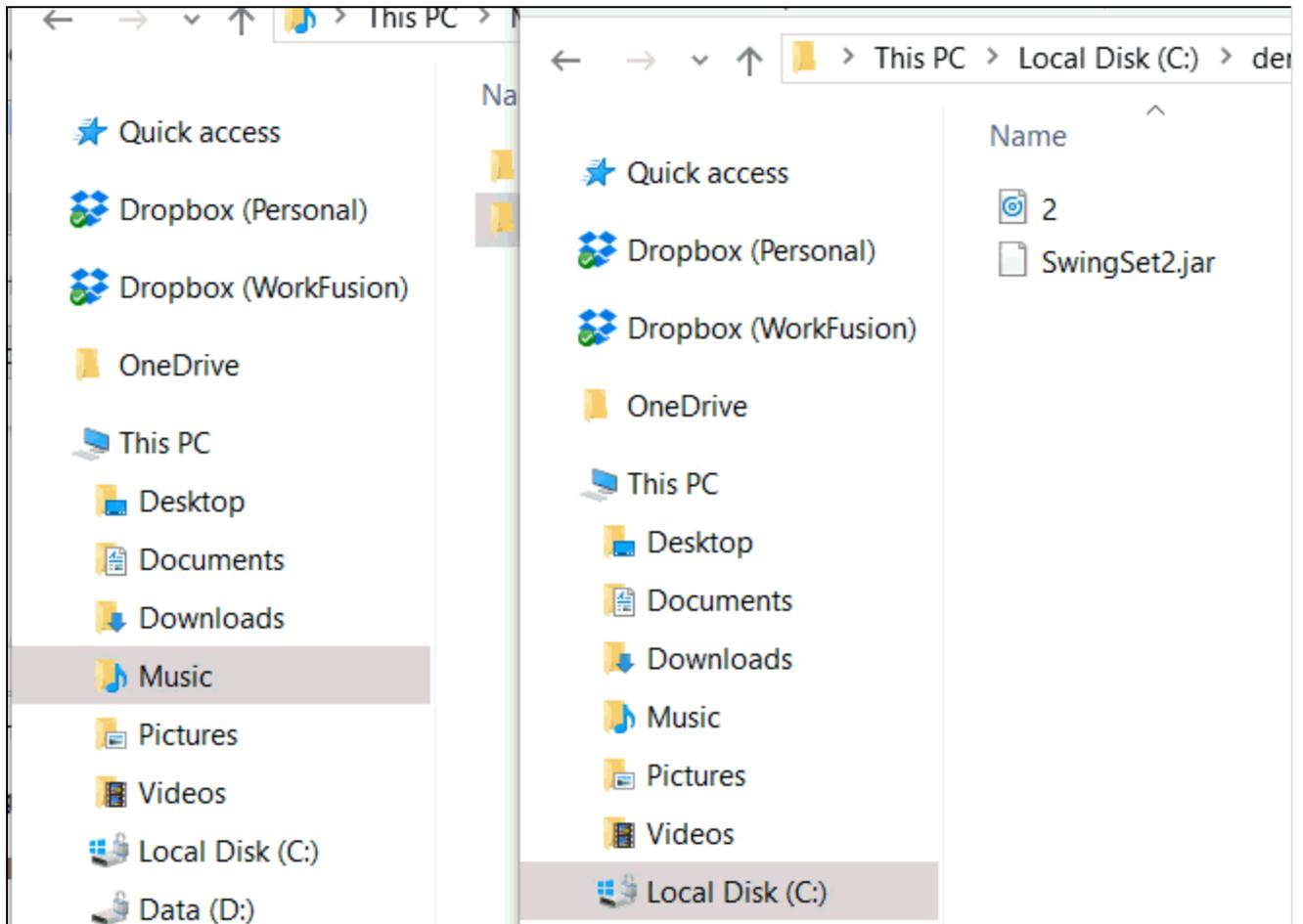
## Drag Mouse

The action represents a **drag and drop** operation from one point (**start**) to another point (**finish**) on the screen and is recorded when user performs a mouse click and moves cursor while holding the mouse button.

### Use case

The action can be used for dragging an element from one panel and dropping it to another, using slider controls.





Actions Flow	Acti...	#
☞ Mouse Click (Left-click 1 times)	<input checked="" type="checkbox"/>	1
> 📁 Window (Playlists)	<input checked="" type="checkbox"/>	2
> 📁 Window (Progress)	<input checked="" type="checkbox"/>	6
∨ 📁 Window (demo)	<input checked="" type="checkbox"/>	8
∨ ☞ <b>Mouse Drag</b>	<input checked="" type="checkbox"/>	9
☞ Start Point	<input checked="" type="checkbox"/>	10
∨ ☞ Intermediate Points	<input checked="" type="checkbox"/>	
☞ Mouse Move (to image)	<input checked="" type="checkbox"/>	11
☞ Finish Point	<input checked="" type="checkbox"/>	12
📁 Window (Playlists)	<input checked="" type="checkbox"/>	13

**Mouse Drag**

[Advanced...](#)

---

Comments:

The **Drag Mouse** action has no specific parameters and contains the following subactions.

- **Start Point** – the place where you start to hold the mouse button.

Actions Flow	Active	#
Click Mouse (click left button)	<input checked="" type="checkbox"/>	1
Window (demo)	<input checked="" type="checkbox"/>	2
Drag Mouse	<input checked="" type="checkbox"/>	3
Start Point	<input checked="" type="checkbox"/>	4
Intermediate Points	<input checked="" type="checkbox"/>	
Move Mouse	<input checked="" type="checkbox"/>	5
Finish Point	<input checked="" type="checkbox"/>	6
Drag Mouse	<input checked="" type="checkbox"/>	7
Start Point	<input checked="" type="checkbox"/>	8

Mouse press

Drag from image  Drag from coordinates

Drag with Left button ?

Target location (click on image to edit)

Access	Name
x (Personal)	SwingSet2

Offset X: 125 Offset Y: 35

[Capture new image](#) [Choose new image](#)

[Advanced...](#)

Comments:

- **Intermediate Points** – the place(s) where you stop moving cursor for more than 1 second while still holding the mouse button, e.g., when needed to drag an element to a nested list.

The block can contain the **Move Mouse** actions only.

Actions Flow	Active	#
Click Mouse (click left button)	<input checked="" type="checkbox"/>	1
Window (demo)	<input checked="" type="checkbox"/>	2
Drag Mouse	<input checked="" type="checkbox"/>	3
Start Point	<input checked="" type="checkbox"/>	4
Intermediate Points	<input checked="" type="checkbox"/>	
Move Mouse	<input checked="" type="checkbox"/>	5
Finish Point	<input checked="" type="checkbox"/>	6
Drag Mouse	<input checked="" type="checkbox"/>	7
Start Point	<input checked="" type="checkbox"/>	8

Move mouse

Move to image  Move to coordinates

Target location (click on image to edit)

Offset X: 125 Offset Y: 35

[Capture new image](#) [Choose new image](#)

[Advanced...](#)

Comments:

- **Finish Point** – the place where you release the mouse button (drop).

Actions Flow	Active	#
Click Mouse (click left button)	<input checked="" type="checkbox"/>	1
Window (demo)	<input checked="" type="checkbox"/>	2
Drag Mouse	<input checked="" type="checkbox"/>	3
Start Point	<input checked="" type="checkbox"/>	4
Intermediate Points	<input checked="" type="checkbox"/>	
Move Mouse	<input checked="" type="checkbox"/>	5
Finish Point	<input checked="" type="checkbox"/>	6
Drag Mouse	<input checked="" type="checkbox"/>	7
Start Point	<input checked="" type="checkbox"/>	8

Mouse Release

Drag to image  Drag to coordinates

Drag with Left button ?

Target location (click on image to edit)

Offset X: 125 Offset Y: 35

[Capture new image](#) [Choose new image](#)

[Advanced...](#)

Comments:

## Scroll Mouse

The action represents a **vertical scroll** operation from a defined point on the screen measured in lines (upwards or downwards) and is recorded when you scroll a page using mouse wheel.

The **Scroll Mouse** action has three properties. These are as follows.

- **Image** – Bot searches for the specified image and, if found, puts the cursor at the crosshair position and scrolls a defined number of lines.
- **Number of lines to scroll** – how many screen lines are to be scrolled.
- **Scroll direction** – either up or down.

Actions Flow	Active	#
Click Mouse (click left button)	<input checked="" type="checkbox"/>	1
Window (demo)	<input checked="" type="checkbox"/>	2
Drag Mouse	<input checked="" type="checkbox"/>	3
Drag Mouse	<input checked="" type="checkbox"/>	7
Scroll Mouse (17 lines down)	<input checked="" type="checkbox"/>	9

Scroll mouse

Scroll on image  Scroll on coordinates

Scroll  lines

Target location (click on image to edit)

[Capture new image](#) [Choose new image](#)

[Advanced...](#)

---

Comments:

---