**✅ What is SikuliX?**

**SikuliX** is an automation tool that lets you automate anything you see on your screen using **image recognition**.

* It uses **screenshots/images** (e.g., of buttons, fields) to identify and interact with GUI elements.
* Works on **any application**: desktop apps, browsers, games, etc.
* Built using **Java + OpenCV** for visual matching.

**🤔 Why use SikuliX?**

SikuliX is used when:

* The app under test **doesn't expose elements to standard automation tools** (like Selenium or Appium).
* You want to **automate desktop GUIs** (e.g., Calculator, Notepad, custom apps).
* You deal with **canvas, images, or Flash** content where HTML locators won’t work.
* You need to **click on icons/buttons/images** that cannot be selected via code.

**📅 When to use SikuliX?**

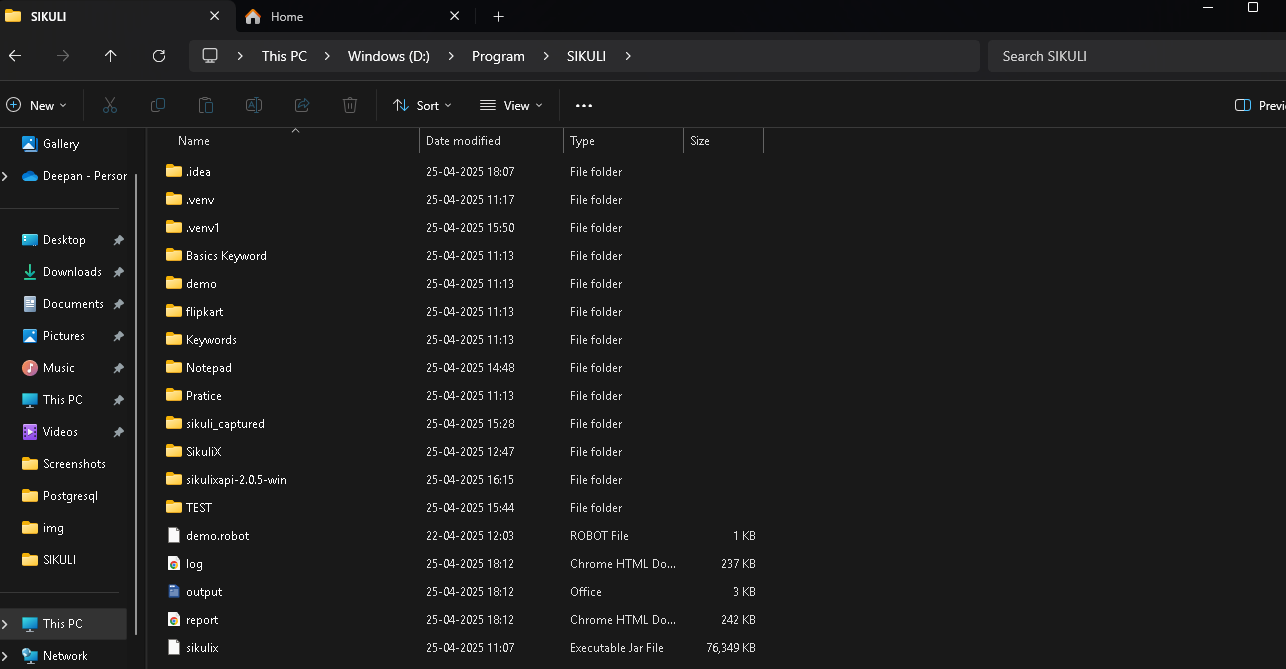
Use SikuliX in these situations:

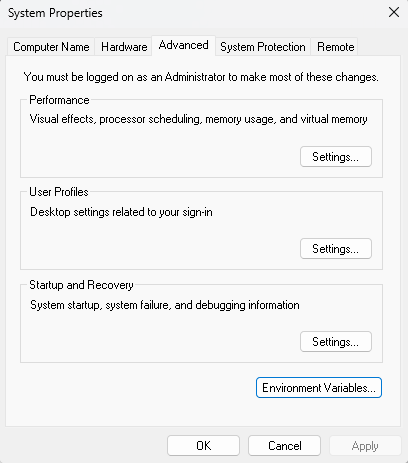
| **Scenario** | **Should Use SikuliX?** |
| --- | --- |
| Testing **Desktop Applications** | ✅ Yes |
| Automating **games or image-based software** | ✅ Yes |
| Testing **legacy software without APIs or DOM access** | ✅ Yes |
| Testing **web applications** (if Selenium works) | ❌ Better use Selenium |

**💡 Benefits of Using SikuliX**

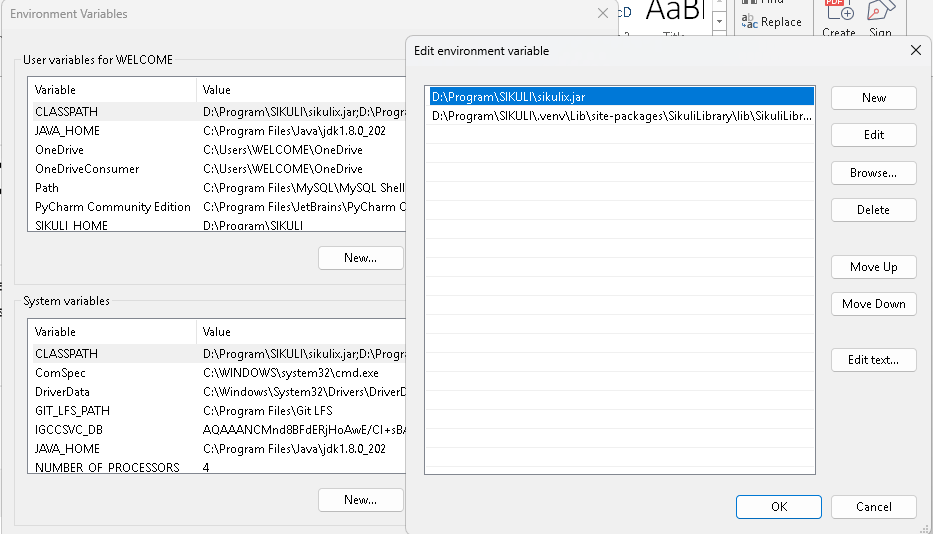
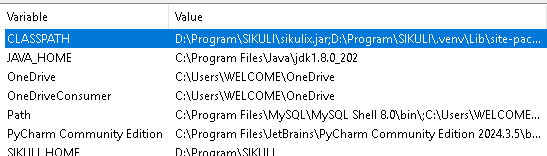
1. **Image-based automation** – Automate anything visible on the screen.
2. **Cross-platform** – Works on Windows, Linux, Mac.
3. **Simple** – Just use screenshots, no need to know internals.
4. **Can be integrated with**:
   * **Robot Framework**
   * **Java/Python** projects
   * **CI tools** like Jenkins

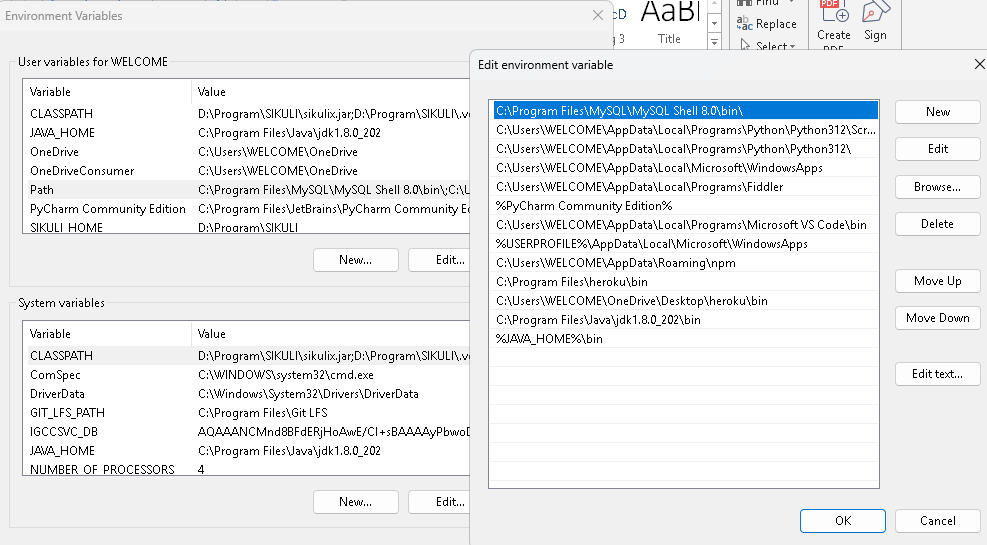
# pip install robotframework-sikulixlibrary





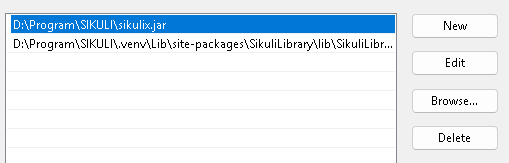
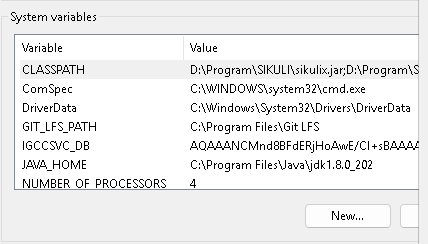
User Variable for welcome

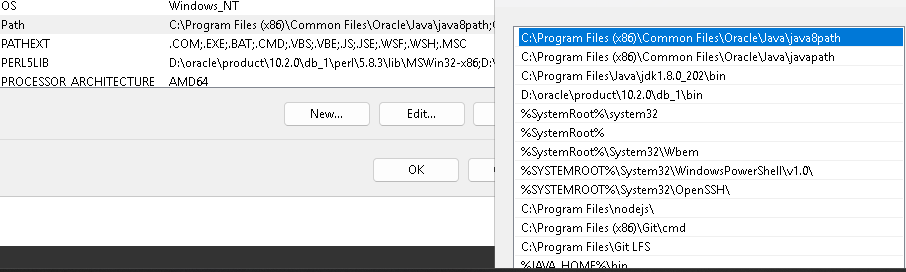
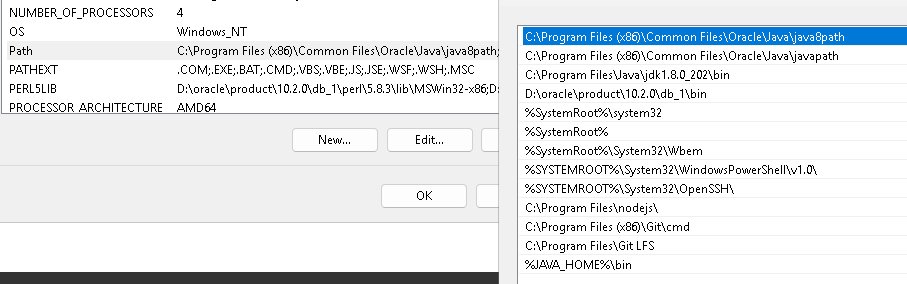


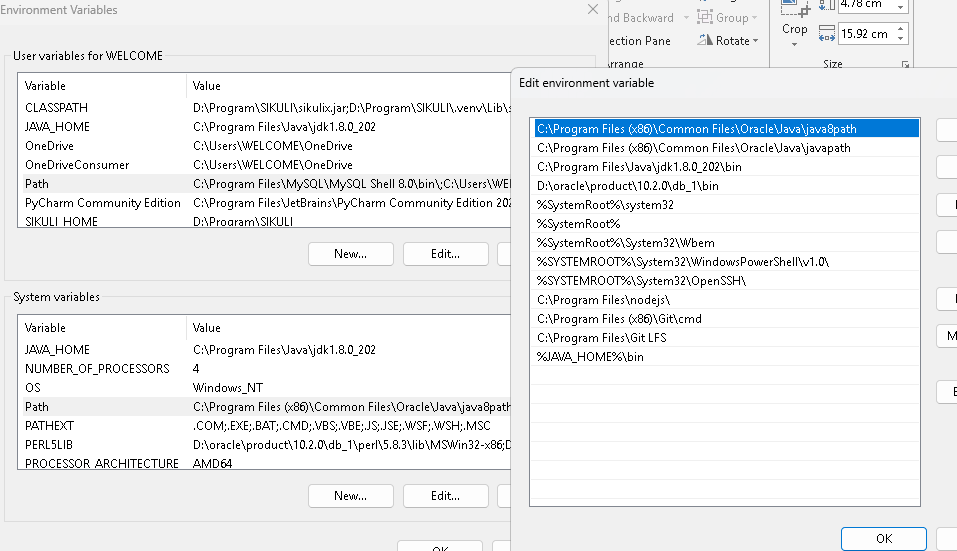
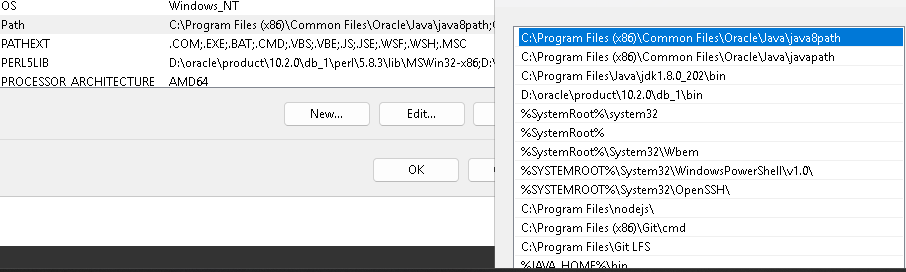




System Variable







SikuliX Keyword

1. Application Keyword

2. Image Keyword

3. Result Keyword

4. Region Keyword

1. Application Keyword

1. App Open

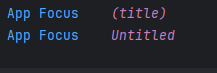
This keyword opens an application by giving the executable file path or application name



2. App Focus

This keyword brings an already opened application window to the front(focus)

Use the application name



3. App Close

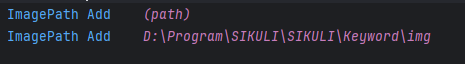
This keyword closes the application



2. Image Keyword

1. Image Path Add

Add a folder path where images are stored



2. Image Path Dump

List all current image path



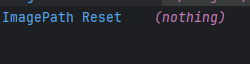
3. Image Path Remove

Remove a specific image path



4. Image Path Reset

Remove all the paths and reset



3. Result Keyword

1. Set Sikuli Result Dir

Set the folder where sikuli screenshot (logs, pass/fail images)



2. Set Passed Log Images

Decide if screen should be saved for passed action.



3. Set FailedLog Images

Decide if screenshots should be saved for failed actions.

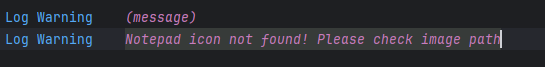


4. Set NotFoundLogImages

Save screenshots when image not found.



5. Log Warning

Log a arning message in the robot framework log

4. Region Keyword

Set of region keyword that look like they’re from Robot Framework or a visual automation tool (like sikuliLibrary or RPA Framework screen automation tools)

1. Region Configuration / setup

2. Finding / Waiting for elements

3. Mouse Actions

4. Keyword Actions

5. Drag and Drop

6. Highlighting Region

7. Text-Based Operations

8. ScreenShots

1. Region Configuration / setup

1. set offsetCenterMode (boolean)

2. region setDefaultSelectMode

3. regionsetAutoWait

4. region getAutoWait

5. region setFindFailedResponse

6. region getFIndFailedResponse

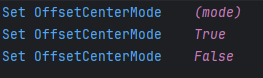
7. region SetRect

1. Set OffsetCenterMode

Sets whether the offset (click position) is calculated from the corner of the found match or from the top-left.

True – offset from center default

False – offset from top-left corner

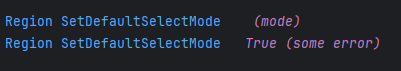


2. region setDefultSelectMode

Changes how a region is selected by default when using visual actions like find,wait etc

In sikuliX directly, this is usally handled by setting selection behaviors via API Native “region setDefaultSelectMode” is more robot framework style. In sikuliX you manually define selectin by choosing rectangular regions or with screen captures.

Region Set Default Select Mode RECT or POINT etcc



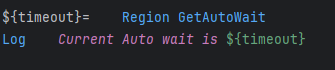
3. region setAutoWait

Sets the default wait time that SikuliX uses when trying to find something on the screen.



4. Region getAutoWait

Gets the current auto-wait timeout value



5. Region SetFindFailedResponse

Sets what hppens when find() fails

ABORT – Stop the script immediately

SKIP – Skip to the next action

PROMPT – Ask the user what to do



6. Region getFindFailedResponse

Gets the current find failed response settings

7. Region setRect(x, y, weight, height)

Manually sets the region boundaries by specifying x, y, height, width