

LECTURE 09.

IMAGE AND TEXT AUTOMATION

Robotic Process Automation

[28 November 2023]

Elective Course, 2023-2024, Fall Semester

Camelia Chisăliță-Crețu, Lecturer PhD

Babeș-Bolyai University

Acknowledgements

This course is presented to our Faculty with the support of UiPath Romania.



Contents

- **Image and Text Automation**
 - Motivation. Overview. Details
- **Image Automation**
 - Overview. Example
 - **Click Image Activity**
 - Details. Properties
 - **Click Text Activity**
 - Details. Properties
 - **Demo 1.** Click Image and Click Text
 - **Image Activities**
 - Details. Types
 - **Find Activity**
 - Overview
- **Keyboard Automation**
 - Details. Overview. Related Activities
 - **Demo 2.** Keyboard Automation
- **Information Retrieval**
 - Details. Functionalities
 - **Select&Copy Activity**
 - Details
- **Relative Scraping**
 - Details. Example
- **Demo 3.** Information Retrieval
- **References**

Image and Text Automation. Motivation

- **Image and Text automation** are used when
 - **selectors cannot be found by using regular methods;**
- E.g.: various applications like:
 - ones hosted on Citrix, Citrix Desktop,
 - scanned PDF, SAP;
- these are environments where selectors are not exposed by the application;
- **the scripting that allows selector generation is disabled.**

Image and Text Automation. Overview

Image based automation



- it relies on **user interface element recognition**.

Text based automation



- it relies on **identifying the text from certain user interface elements**.

Image and Text Automation. Details

- UiPath Studio provides activities for:
 - simulating **mouse input**: such as clicking, hovering or typing, text recognition and OCR (optical character recognition);
- these activities use:
 - **image recognition that work directly with images to identify UI elements**:
 - it can simulate human behavior by using images as a means of identifying UI elements;
 - **screen scraping to identify UI elements**:
 - it is able to extract text from running applications even when they are hidden or covered by another applications.

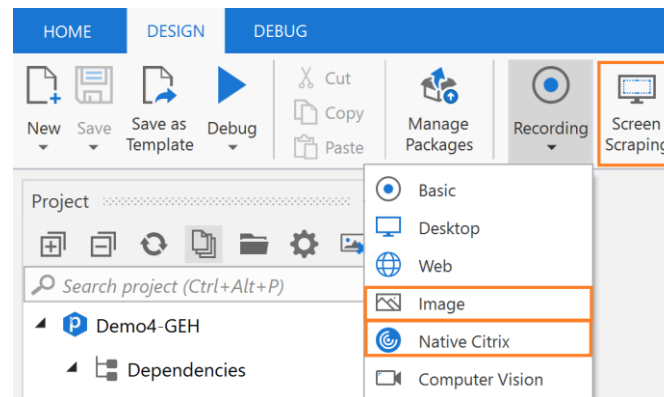


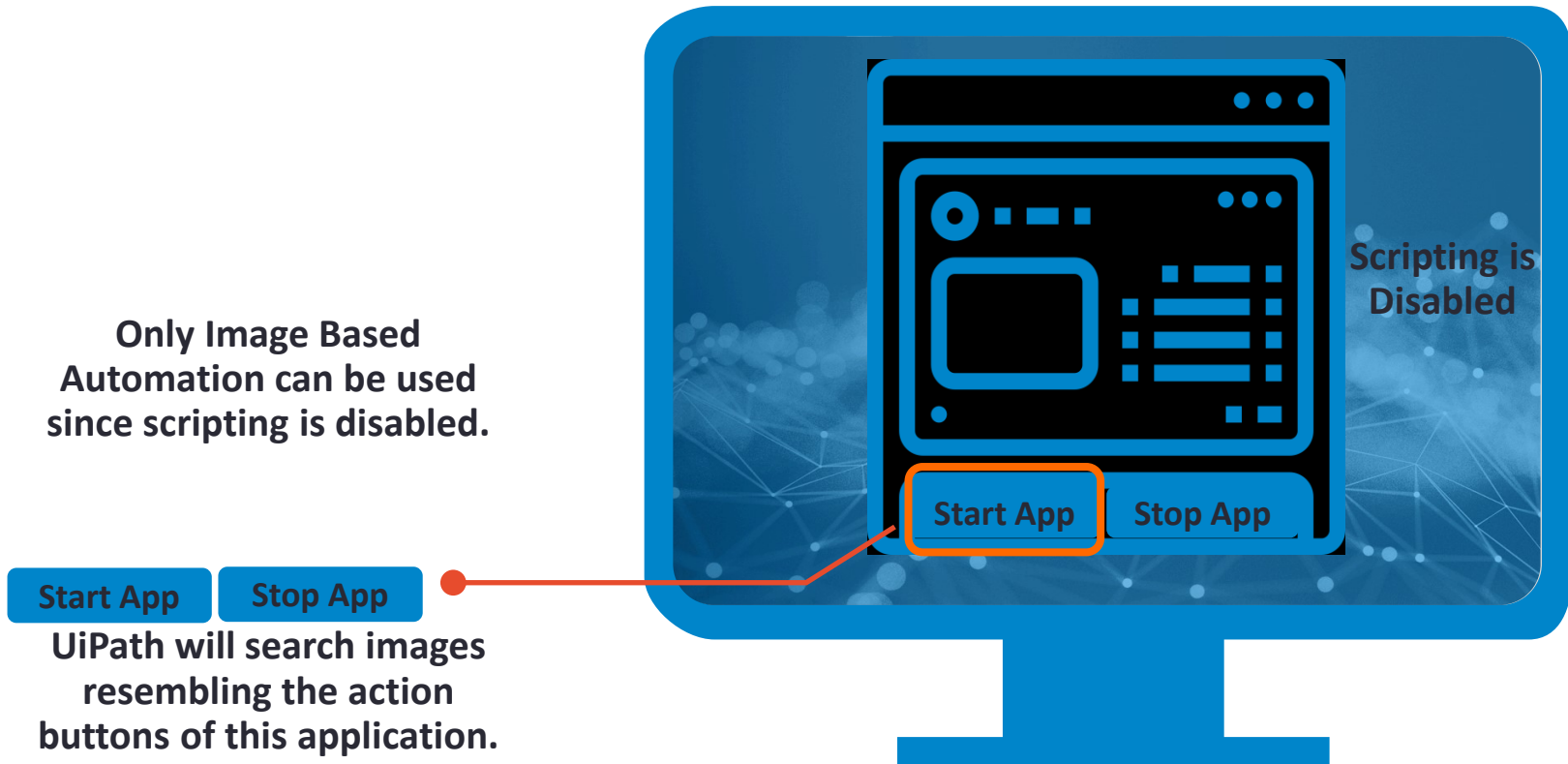
Image Automation. Overview

- **Image recognition** activities
 - can simulate human behavior by using images as a means of identifying UI elements;
 - are very useful in situations where **human behavior must be mimicked**.



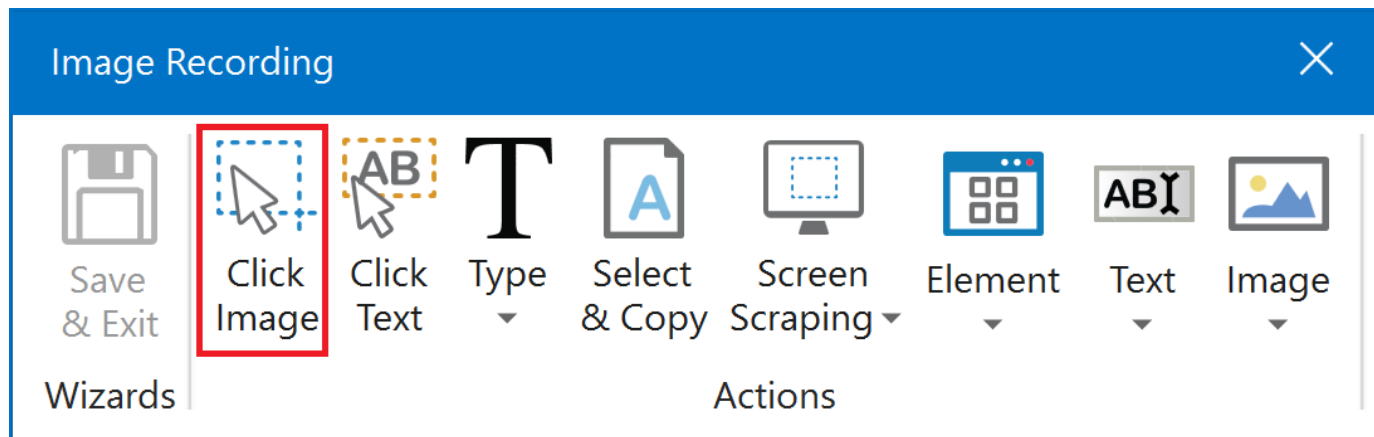
Image Automation. Example

- Image-based automation can be deployed in cases where scripting is not enabled.



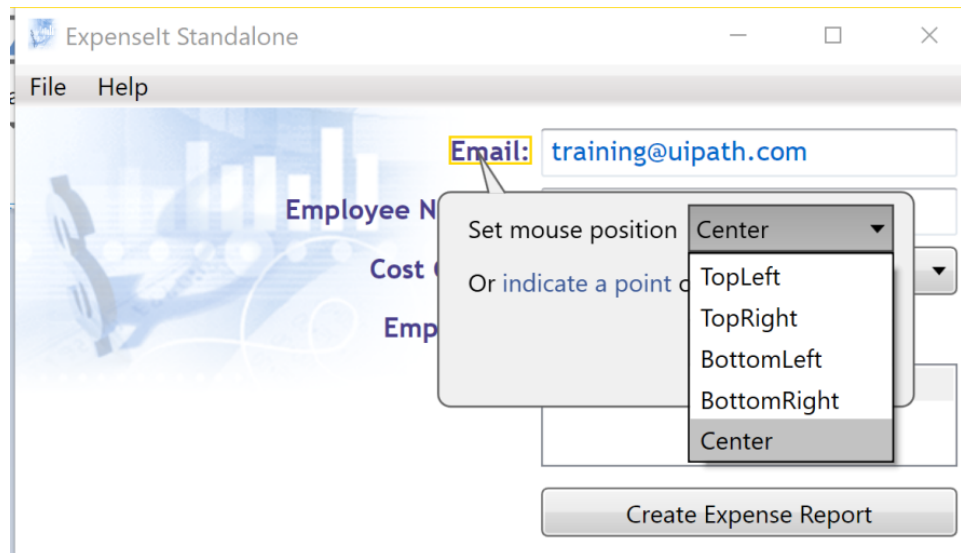
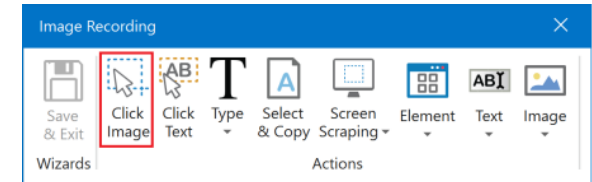
Click Image Activity. Details

- **Click Image** activity
 - allows to click on any selected area as long as this is unique on the screen/ virtual environment;
 - it scans the screen of the machine for UI elements which appear at random positions and return UI Element variables that have the clipping region set to the found element;
- **disadvantages:** such activities are fast and reliable, but sensitive to graphical variations, as they can fail if colors or background details change;
- different properties allow to customize the image recognition during execution.



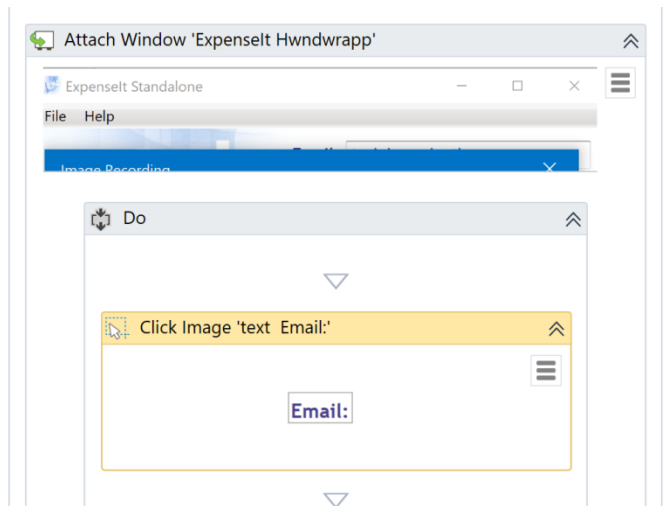
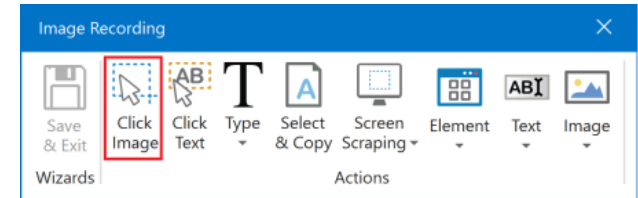
Click Image Activity. Properties (1)

- **Click Image** activity properties permit to
 - **set the position of the mouse where to perform click:**
 - *inside* the selected area: **Center, Top Left, Top Right, etc.;**
 - *outside* the selected area: **indicating an area or a point** where to perform click;
 - **this is useful when there is a single element to match to on the image;**



Click Image Activity. Properties (2)

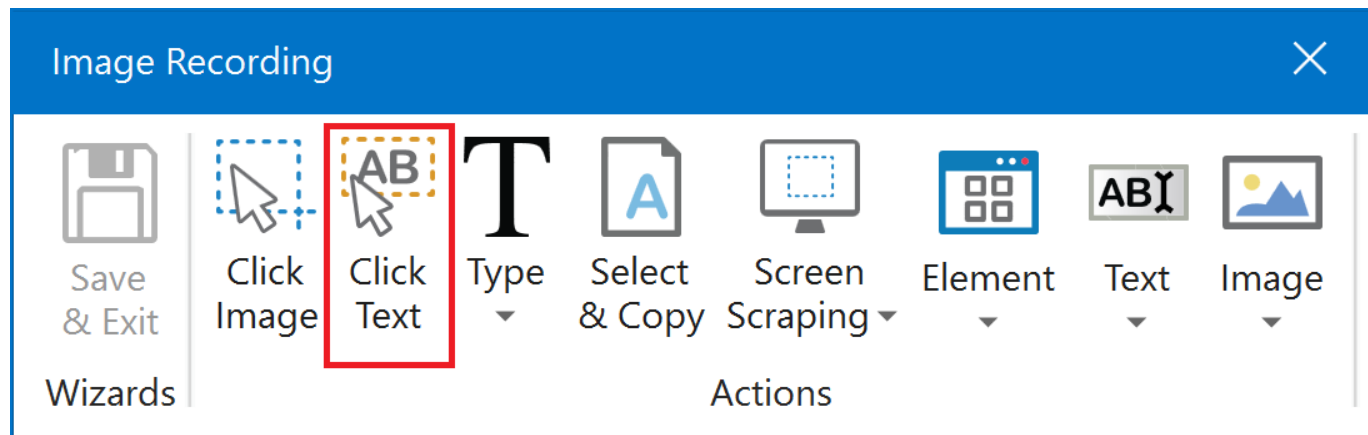
- **Click Image** activity properties permit to
 - **configure the accuracy of the matching action:**
 - **0.80** = good value, it allows a compromise between accuracy and reliability;
 - **1.00** = the image must match 100% to be registered as found;
 - **this is useful if graphical elements searched for may be slightly different;**



Display Name	Click Image 'text
Input	
ClickType	CLICK_SINGLE
Image	ImageTarget
Accuracy	0.8
Image	Enter a VB e...
MouseButton	BTN_LEFT
Target	Target
ClippingRegion	
Element	Enter a VB e...
Selector	"<ctrl name: ...
Timeout (milliseconds)	Enter a VB e...
WaitForReady	INTERACTIVE

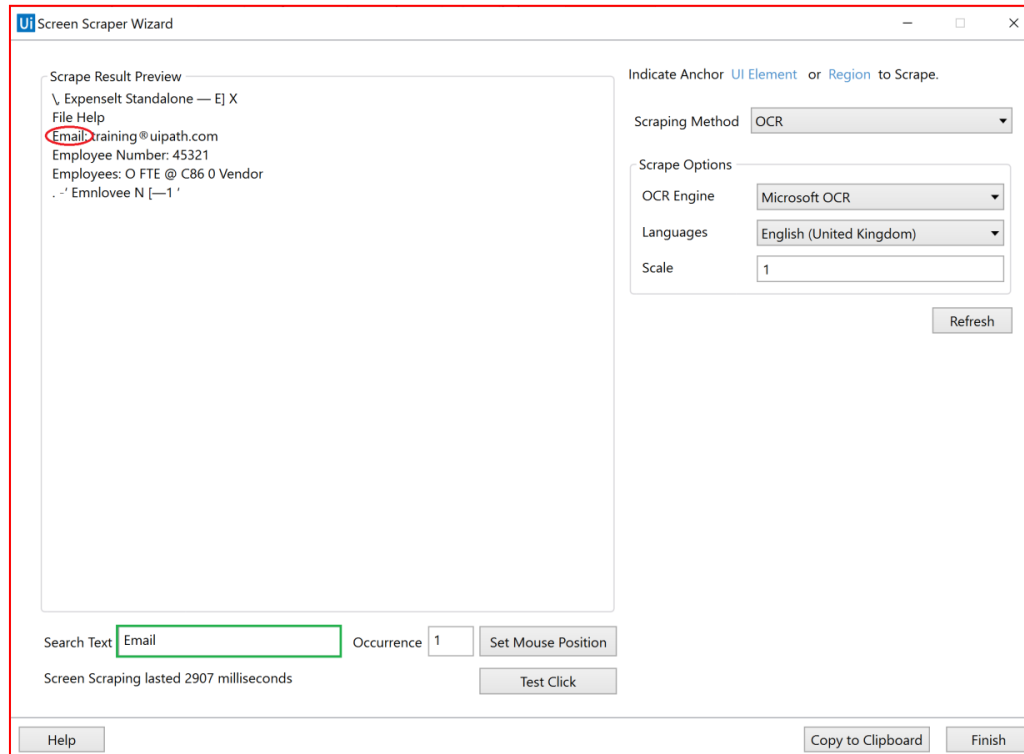
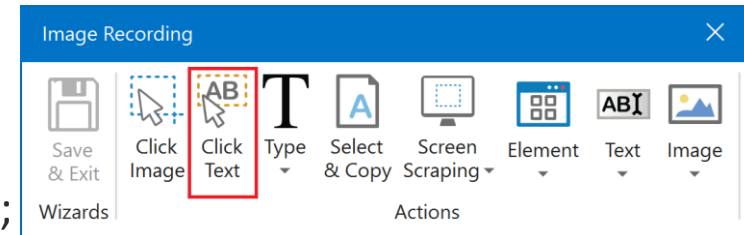
Click Text Activity. Details

- **Click Text** activity
 - allows to use an OCR engine to scan for a specific text available on the screen/virtual environment;
 - the window is similar to the one of Screen Scraping with some particularities for the mouse actions;
 - useful when window theme is different or the text is different, but is the same;
- **disadvantages: speed, OCR accuracy and its settings;**
- the properties refer mainly to the OCR engine used to recognize the text during execution.



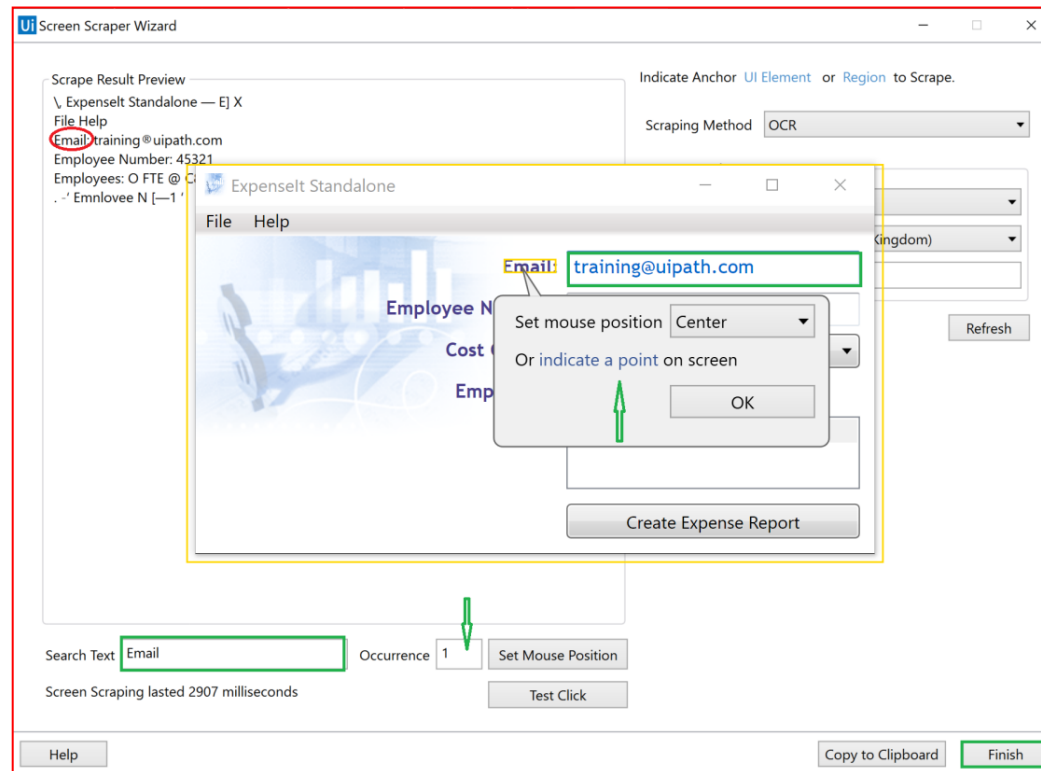
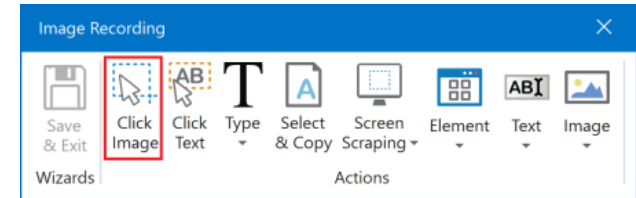
Click Text Activity. Properties (1)

- **Click Text** activity properties permit to
 - **set the text searched for:**
 - the user specifies the text to perform click on;



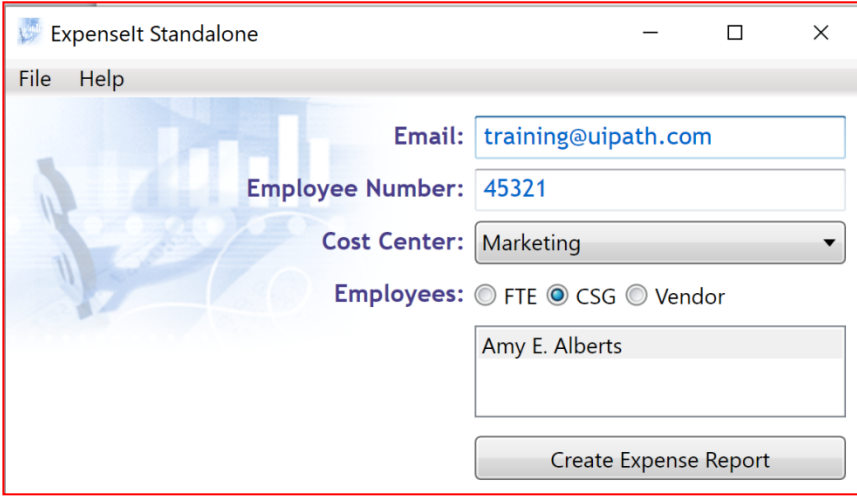
Click Text Activity. Properties (2)

- **Click Text** activity properties permit to
 - **configure the occurrences of the text searched for:**
 - the user specifies the number of occurrences and the position of the mouse (*inside, outside* the found text);



Demo 1. Click Image and Click Text

- Use the **Click Image** and **Click Text** activities to fill in the form required to generate a report;
 - application **Expenselt**;
 - Email: myEmail@company.com
 - Employee Number: 12345;
 - Cost Center: Marketing;
 - Employees: CSG;
 - select the employee Amy E. Alberts;



The screenshot shows a window titled "Expenselt Standalone" with a menu bar containing "File" and "Help". The main area contains a form with the following fields:

- Email:** A text box containing "training@uipath.com".
- Employee Number:** A text box containing "45321".
- Cost Center:** A dropdown menu with "Marketing" selected.
- Employees:** A group of radio buttons with "FTE", "CSG" (selected), and "Vendor".
- Employee Selection:** A list box containing "Amy E. Alberts".
- Create Expense Report:** A button at the bottom of the form.

Image Activities. Details

- **Find**-based activities for images
 - wait for specific UI components to appear;
- steps:
 - 1. it presents the model of UI element image indicated by the user to be searched for;
 - 2. once the UI component appears, the image activity with the UI Element variable and clipping image set region;
- the **Find**-based activities is a useful tool to identify the UI element, variable and component in the virtual machine;
- it provides the best decision choosing option that image is displayed or not; apart from this it can manage different actions;
- it enables to take and make decisions in image activities;
- it uses **Retry Scope** activity to handle the condition of image activity.

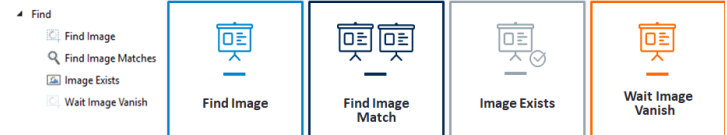
Find

- Find Image
- Find Image Matches
- Image Exists
- Wait Image Vanish







Image Activities. Types

- **Find Image** activity
 - waits for the images to appear in UI elements; when this process is completed, it changes the UI element image with the clipping region set; then, it provides an element along with UI elements;
 - source code: `UiPath.Core.Activities.WaitImageAppear;`
- **Find Image Match** activity
 - searches and matches a particular image in a UI element and delivers the collection of UI element;
 - in the collection, it keeps the clipping region set of matching screens.
 - source code: `UiPath.Core.Activities.FindImageMatches;`
- **Image Exists** activity
 - it specifies any images found with individual UI element;
 - source code: `UiPath.Core.Activities.ImageFound;`
- **Wait Image Vanish** activity
 - it waits for the image when it disappears from UI element;
 - source code: `UiPath.Core.Activities.WaitImageVanish.`



Find Activity. Overview

Find

-  Find Image
-  Find Image Matches
-  Image Exists
-  Wait Image Vanish



Find Image



Find Image Match





Image Exists



Wait Image Vanish



Event

-  On Image Appear
-  On Image Vanish



Monitoring Events

File

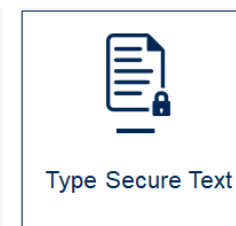
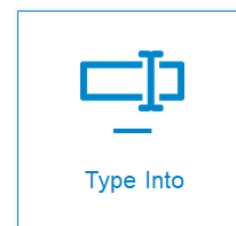
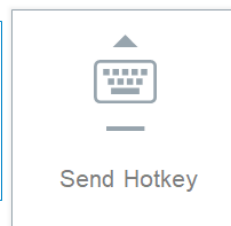
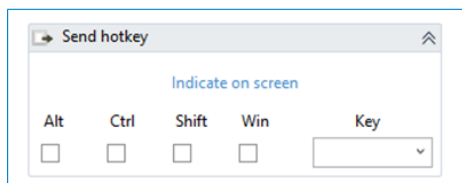
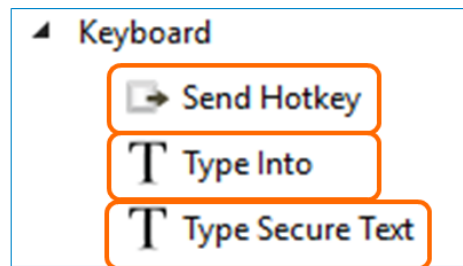
-  Load Image
-  Save Image



File Activities

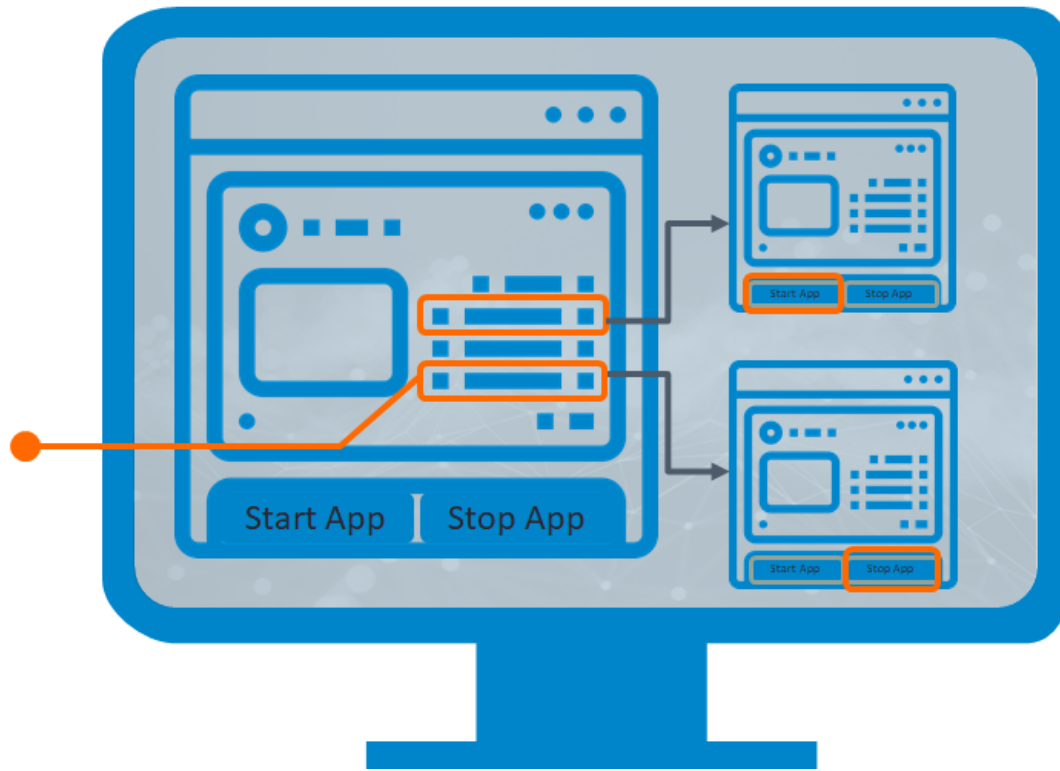
Keyboard Automation. Details

- UiPath Studio provides activities for:
 - simulate any type of **keyboard** input that a regular human user would use.
- the activity passes the UI element variable by input text;
- it identifies the action that is automated; during this process the target can automatically create screen functionality that identifies the UI elements in a proper region and produces selectors for them;



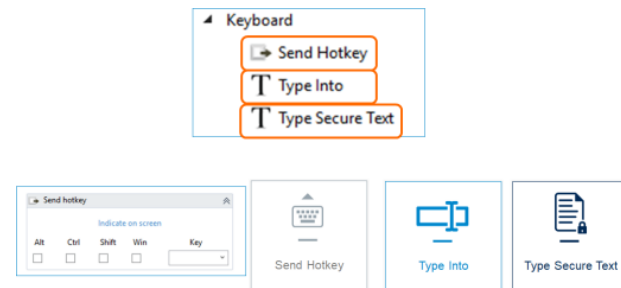
Keyboard Automation. Overview

With the use of hotkeys and shortcuts, specific actions can be set to reduce the number of steps performed.



Keyboard Automation. Related Activities

- activities that make the keyboard-based automation easier:
 - **Send HotKey** activity
 - is used to access the UI element application shortcut and to simplify the automation project;
 - **Type Into** activity
 - sends the keystrokes to an UI element;
 - supports the select and drop-down list components.
 - **Type Secure Text** activity
 - sends a secure string to a UI element which is useful for secure automation;
 - uses passwords that are stored in **SecureString**.



Demo 2. Keyboard Automation

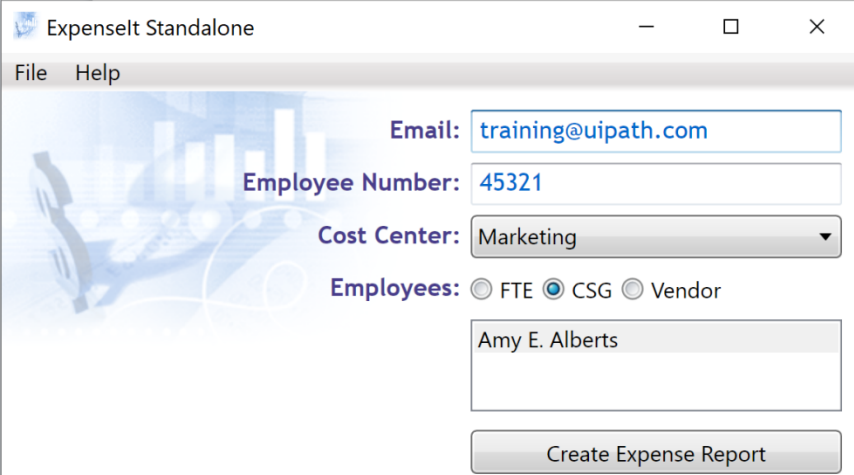
- Use the **keyboard automation** activities to fill in the form required to generate a report;

- application **Expenselt**;
 - Email: myEmail@company.com
 - Employee Number: 12345;
 - Cost Center: Marketing;
 - Employees: CSG;
 - select the employee Amy E. Alberts;

- possible sequence of keys:

- "[k(end)d(shift)d(home)u(shift)u(home)]myEmail@comany.com[k(tab)]"
+"[k(end)d(shift)d(home)u(shift)u(home)]12345[k(tab)][k(down)][k(tab)]
[k(right)][k(tab)][k(down)][k(tab)][k(enter)]"

- "[k(end)d(shift)d(home)u(shift)u(home)]" allows to select the entire content of the text field.

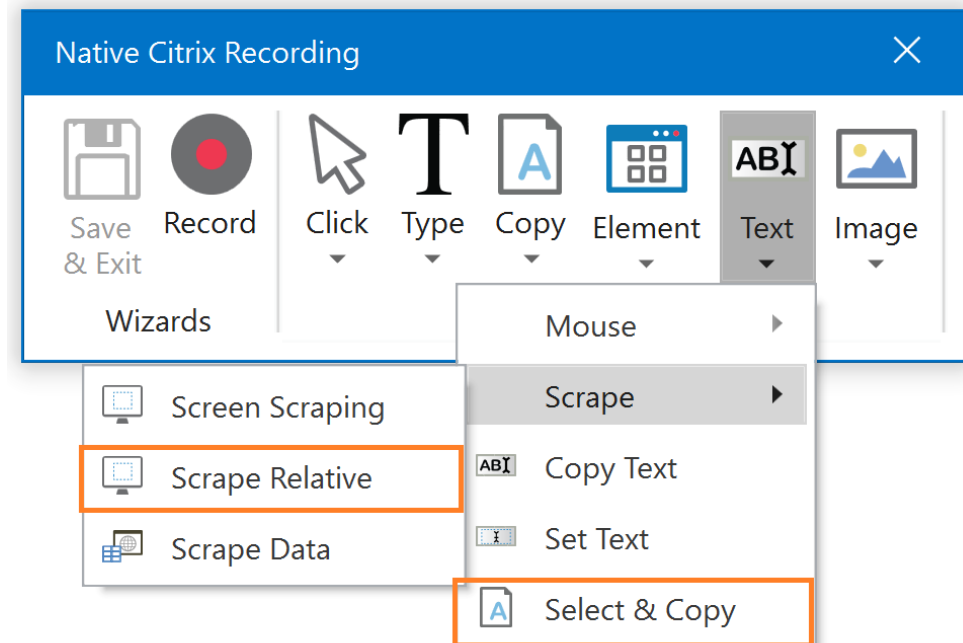


The screenshot shows a window titled "Expenselt Standalone" with a menu bar containing "File" and "Help". The main area features a light blue background with a bar chart and a hand holding a coin. On the right side, there is a form with the following fields and values:

- Email:
- Employee Number:
- Cost Center:
- Employees: ☐ FTE ☒ CSG ☐ Vendor
- A text box containing "Amy E. Alberts"
- A button labeled "Create Expense Report"

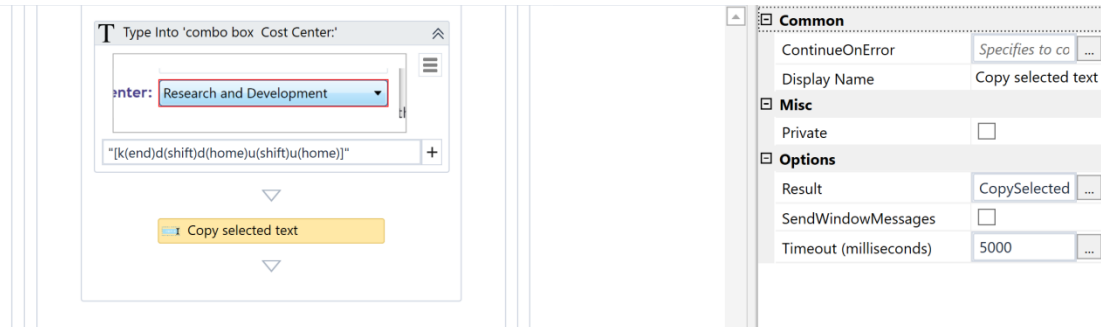
Information Retrieval. Details

- **Information retrieval** from
 - an application accessed by a Virtual Machine (VM) can be performed by:
 - **copy the data from an editable text** or
 - use Optical Character Recognition (**OCR**) to **extract the relevant information**;
- actions involved:
 - **Select&Copy**;
 - **Relative Scraping**;
- related actions:
 - **Copy Text**;
 - **Screen Scraping**.



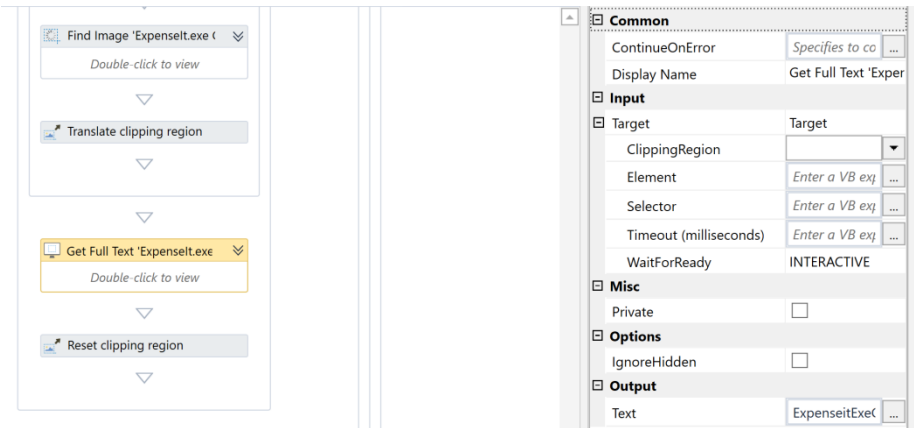
Select&Copy Activity. Details

- **Select&Copy** wizard
 - selects editable texts and copies it to the UiPath environment through the clipboard;
 - **all actions are performed on the active text field;**
 - two activities are generated by the wizard:
 - **Type Into** activity:
 - it is the command used to select the text:
`"[k(end)d(shift)d(home)u(shift)u(home)]";`
 - **Copy Selected Text** activity:
 - it involves copying the text activity from the target UI to the UiPath environment;
 - the copied text appear as **GenericValue** variable;
- this can be combined with any of the three input methods (*default, send messages, simulate type/click*).



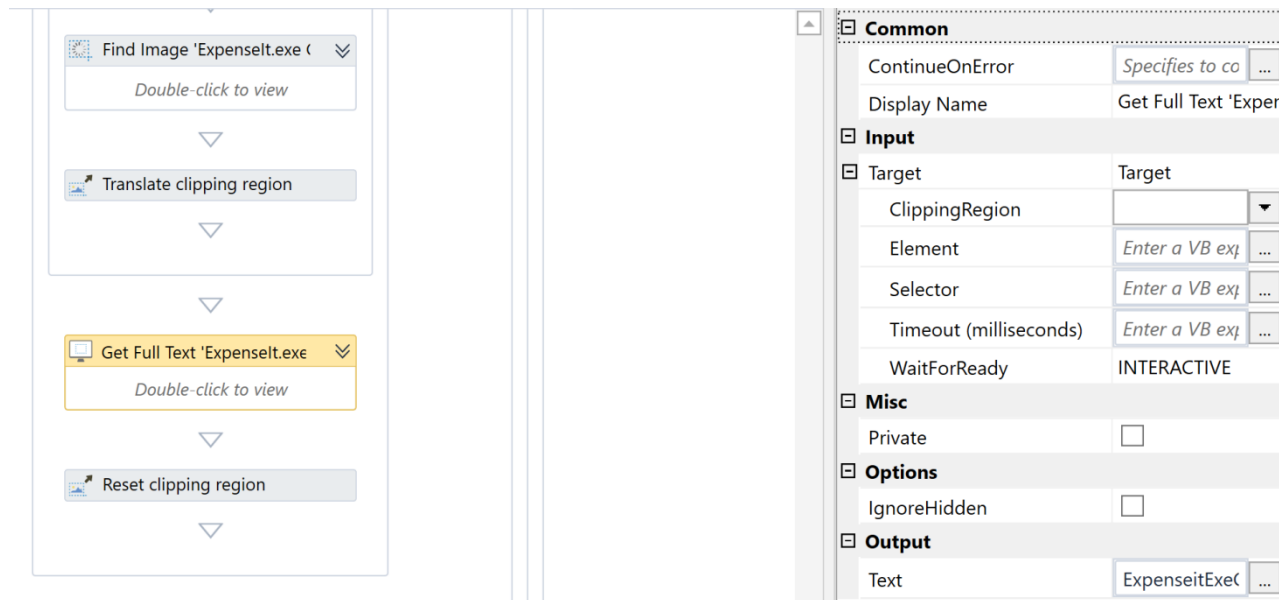
Relative Scraping. Details

- **Relative Scraping** wizard
 - scrapes just a portion of an image relative to an anchor;
 - it locates a fixed element on the screen and then using OCR extracts the information;
 - four activities are generated by the wizard:
 - **Find Image** activity:
 - used to find the anchor image;
 - **Translate Clipping Region** activity:
 - used to find the clipping region, offset to the anchor image;
 - **Get OCR Text** activity:
 - used to extract the data from the clipping region;
 - other **Get Text**-based activities may be used according to the output methods used, **FullText** or **Native**;
 - **Reset Clipping Region** activity:
 - used to avoid interference with other operations.



Relative Scraping. Example

- E.g.:
 - using OCR, it can be deployed information retrieval for customers that handle a large flow of documents such as Purchase Orders (PO);
 - or, someone can manually scan them and upload them on SAP;
 - each PO can be scanned out many documents and details can be extracted like **Invoice Number, Date, Tax, Total**, and so on;
 - the information can be organized in specific fields for further storage and handling.



Demo 3. Information Retrieval

- The use of **information retrieval** wizards to get information out of the created expense report;
 - application **Expenselt**:
 - **Select&Copy**;
 - **Relative Scraping**.

Create Expense Report

Email Alias: Someone@example.com

Employee Number: 57304

Cost Center: 4032

Expense Type	Description	Amount
Meal	Mexican Lunch	12
Meal	Italian Dinner	45
Education	Developer Conference	90
Travel	Taxi	70
Travel	Hotel	60

Add Expense

View Chart

Total Expenses (\$): 277

OK Cancel

References

- UiPath Docs - <https://docs.uipath.com/>
- UiPath Studio Docs - <https://docs.uipath.com/studio/standalone/2023.4>
- UiPath Forum - <https://forum.uipath.com/>
- UiPath Academy - <https://academy.uipath.com/>