**Input Methods**

Input actions require you or the robot to directly interact with an opened application or web page. There are three types of input methods for click and type actions, that differ in terms of compatibility and capability.

We generally recommend the **Simulate Type/Click** method as it is the fastest of the three and works in the background, but only if you do not need to send special keyboard shortcuts. If this does not work for you, try the **Windows Messages** method and then the **Default** one, as it is the slowest.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Capability Method** | **Compatibility** | **Background  Execution** | **Speed** | **Hotkey Support** | **Auto Empty Field** |
| 1 | **Default** | 100% | No | 50% | Yes | No |
| 2 | **SendWindowMessages** | 80% | Yes | 50% | Yes | No |
| 3 | **Simulate Type/Click** | 99% - Web Apps  66% - Desktop Apps | Yes | 100% | No | Yes |

The **Default** application simulates a click or type with the help of the hardware driver, while the **Simulate Type/Click** method uses the technology of the target application. Lastly, the **SendWindowMessages** works by sending a specific message directly to the target application.

**Output Or Screen Scraping Methods**

Output or screen scraping methods refer to those activities that enable you to extract data from a specified UI element or document, such as a .pdf file.

To understand which one is better for automating your business process, let’s see the differences between them.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Capability Method** | **Speed** | **Accuracy** | **Background  Execution** | **Extract  Text Position** | **Extract  HiddenText** | **Support  For Citrix** |
| 1 | FullText | 100% | 100% | Yes | No | Yes | No |
| 2 | Native | 80% | 100% | No | Yes | No | No |
| 3 | OCR | 30% | 98% | No | Yes | No | Yes |

**FullText** is the default method, it is fast and accurate, yet unlike the **Native** method, it cannot extract the screen coordinates of the text.

Both these methods work only with desktop applications, but the **Native** method only works with apps that are built to render text with the Graphics Device Interface (GDI).

**OCR** is not 100% accurate, but can be useful to extract text that the other two methods could not, as it works with all applications including Citrix. Studio uses two OCR engines, by default: Google Tesseract and Microsoft Modi.

1. **FullText:**

**Ignore Hidden** – when this check box is selected, the hidden text from the selected UI element is not copied.

1. **Native:**

**No Formatting** – when this check box is selected, the copied text does not extract formatting information from the text. Otherwise, the extracted text’s relative position is retained.

**Get Words Info** – when this check box is selected, Studio also extracts the screen coordinates of each word. Additionally, the **Custom Separators** field is displayed, that enables you to specify the characters used as separators. If the field is empty, all known text separators are used.

1. **Google OCR:**

**Languages** – only English is available by default.

**Characters** – enables you to select which types of characters to be extracted. The following options are available: **Any character**, **Numbers only**, **Letters**, **Uppercase**, **Lowercase**, **Phone numbers**, **Currency**, **Date** and **Custom**. If you select **Custom**, two additional fields, **Allowed** and **Denied**, are displayed that enable you to create custom rules on which types of characters to scrape and which to avoid.

**Invert** – when this check box is selected, the colors of the UI element are inverted before scraping. This is useful when the background is darker than the text color.

**Scale** – the scaling factor of the selected UI element or image. The higher the number is, the more you enlarge the image. This can provide a better OCR read and it is recommended with small images.

**Get Words Info** – gets the on-screen position of each scraped word.

1. **Microsoft OCR:**

**Languages** – enables you to change the language of the scraped text. By default, English is selected.

**Scale** – the scaling factor of the selected UI element or image. The higher the number is, the more you enlarge the image. This can provide a better OCR read and it is recommended with small images.

**Get Words Info** - gets the on-screen position of each scraped word.

Besides getting text out of an indicated UI element, you can also extract the value of multiple types of attributes, its exact screen position, and its ancestor.

This type of information can be extracted through dedicated activities that are found in the **Activities** panel, under **UI Automation > Element > Find** and **UI Automation > Element > Attribute**.

These activities are:

**Get Ancestor** – enables you to retrieve an ancestor from a specified UI element. You can indicate at which level of the UI hierarchy to find the ancestor, and store the results in a UiElement variable.

**Get Attribute** – retrieves the value of a specified UI element attribute. Once you indicate the UI element on screen, a drop-down list with all available attributes is displayed.

**Get Position** – retrieves the bounding rectangle of the specified UiElement, and supports only Rectangle variables.

UiPath Studio also features **Relative Scraping**, a scraping method that identifies the location of the text to be retrieved relative to an anchor.

You can also generate tables from unstructured data and store the information in DataTable variables, by using the **Screen Scraping Wizard**.

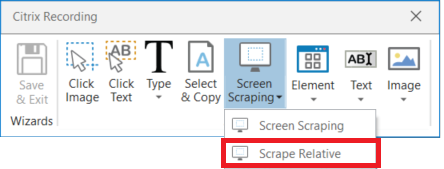
**Relative Scraping**

**Relative Scraping** is a technique that enables you to retrieve text from UI elements by using OCR technology. In situations where selectors cannot be found, the target UI objects are identified by using image recognition activities to look for adjacent labels or other elements.

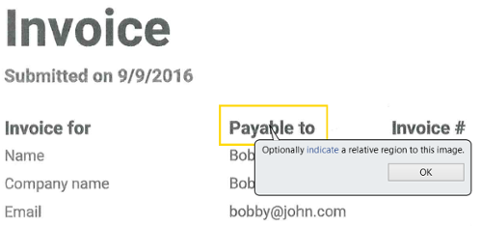
This technique is useful in retrieving text from certain UI elements that are difficult to access by using normal means, such as applications in virtual environments. Using visual labels of UI elements makes up for the inability to find selectors.

To use the **Scrape Relative** functionality, do the following:

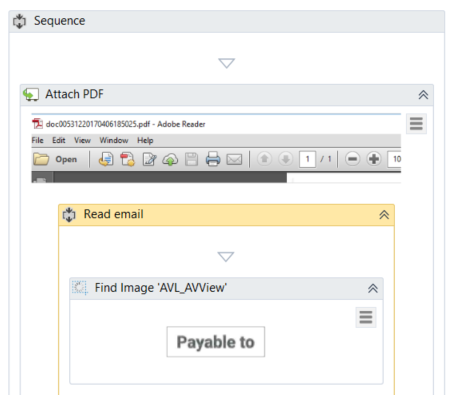
1. Start the **Citrix Recording Wizard**.
2. Click **Screen Scraping > Scrape Relative**.

[](https://files.readme.io/1ab665e-image_168.png)

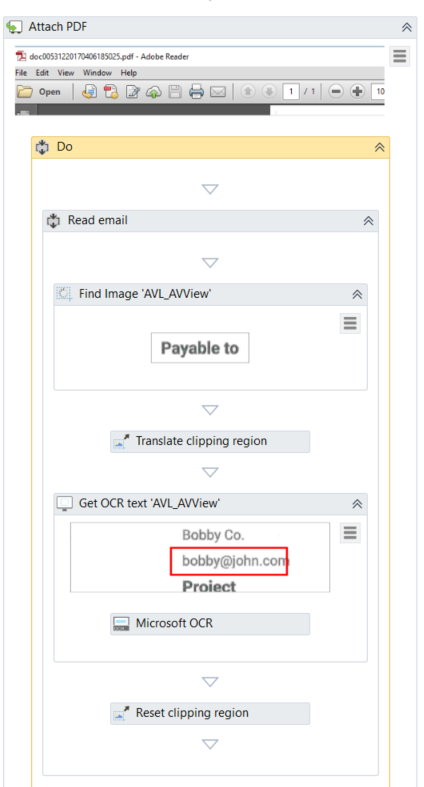
1. Select an **anchor**, which is the relative element used to identify the location of the target, such as the label of a text field.

[](https://files.readme.io/3cf08b6-image_169.png)

An **Attach Window** container is generated that sets focus to the app window and contains a **Find Image** activity that locates the position of the anchor on the screen.

[](https://files.readme.io/f2fb658-image_170.png)

1. Indicate the area where the target element is. A **Set Clipping Region** activity is generated, which translates the clipping region to where the target element can be found, relative to the anchor. Additionally, a **Get OCR Text** activity is generated that scrapes the target element. Since the clipping region is a shared resource, the recorder generates another **Set Clipping Region** activity which resets the clipping region, thus avoiding interference with other operations.

[](https://files.readme.io/4245272-image_171.png)