Kofax RPA

Getting Started with Desktop Automation
Version: 10.6.0

Date: 2019-06-19

KOFAX

Table of Contents

Preface

This guide provides a tutorial that walks you through the process of using Kofax RPA Desktop Automation to build a robot.

The instructions provided in this document assume that you have downloaded and installed Kofax RPA on your computer. See the chapter "Quick Start Guide" in the *Kofax RPAInstallation Guide* to start using the product.

Related Documentation

The documentation set for Kofax RPA is available here:[1]

https://docshield.kofax.com/Portal/Products/RPA/10.6.0_p2wddr4n2j/RPA.htm

In addition to this guide, the documentation set includes the following items:

Kofax RPA Release Notes
Contains late-breaking details and other information that is not available in your other Kofax RPA documentation.

Kofax RPA Installation Guide
Contains instructions on installing Kofax RPA and its components in a development environment.

Kofax RPA Administrator's Guide
Describes administrative and management tasks in Kofax RPA.

Help for Kofax RPA
Describes how to use Kofax RPA. The Help is also available in PDF format and known as *Kofax RPA User's Guide*.

Kofax RPA Getting Started with Document Transformation Guide
Provides a tutorial that explains how to use Document Transformation functionality in a Kofax RPA environment, including OCR, extraction, field formatting, and validation.

Kofax RPA Desktop Automation Service Configuration Guide
Describes how to configure the Desktop Automation Service required to use Desktop Automation on a remote computer.

---

[1] You must be connected to the Internet to access the full documentation set online. For access without an Internet connection, see the *Installation Guide*.

Kofax RPA Developer's Guide
Contains information on the API that is used to execute robots on RoboServer.

Kofax Analytics for RPA Release Notes
Contains late-breaking details and other information on Kofax Analytics for RPA.

Kofax Analytics for RPA Administrator's Guide
Contains instructions for installing and using Kofax Analytics for RPA.

Kofax RPA Integration API documentation

Contains information about the Kofax RPA Java API and the Kofax RPA .NET API, which provide programmatic access to the Kofax RPA product. The Java API documentation is available from both the online and offline Kofax RPA documentation, while the .NET API documentation is available only offline.

Note The Kofax RPA APIs include extensive references to RoboSuite, the original product name. The RoboSuite name is preserved in the APIs to ensure backward compatibility. In the context of the API documentation, the term RoboSuite has the same meaning as Kofax RPA.

Training

Kofax offers both classroom and computer-based training to help you make the most of your Kofax RPA solution. Visit the Kofax website at www.kofax.com for details about the available training options and schedules.

Getting help with Kofax products

The Kofax Knowledge Base repository contains articles that are updated on a regular basis to keep you informed about Kofax products. We encourage you to use the Knowledge Base to obtain answers to your product questions.

To access the Kofax Knowledge Base, go to the Kofax website and select **Support** on the home page.

Note The Kofax Knowledge Base is optimized for use with Google Chrome, Mozilla Firefox or Microsoft Edge.

The Kofax Knowledge Base provides:

• Powerful search capabilities to help you quickly locate the information you need.

Type your search terms or phrase into the **Search** box, and then click the search icon.

• Product information, configuration details and documentation, including release news.

Scroll through the Kofax Knowledge Base home page to locate a product family. Then click a product family name to view a list of related articles. Please note that some product families require a valid Kofax Portal login to view related articles.

- Access to the Kofax Customer Portal (for eligible customers).

  Click the **Customer Support** link at the top of the page, and then click **Log in to the Customer Portal**.

- Access to the Kofax Partner Portal (for eligible partners).

  Click the **Partner Support** link at the top of the page, and then click **Log in to the Partner Portal**.

- Access to Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.

  Scroll to the **General Support** section, click **Support Details**, and then select the appropriate tab.

Build a robot with Desktop Automation

Desktop Automation overview

Desktop Automation gives you the ability to build robots that can automate work processes involving Windows and Java applications on your networked computers. The main purpose of Desktop Automation is automated control of these applications. For Desktop Automation, Design Studio has a dedicated workflow language, editor, and steps.

The **Desktop Automation workflow** is a sequence of steps executed one after the other. The steps model how a user would interact with the application that is being automated.

Steps are the basic building blocks of the Desktop Automation workflow. Some steps are simple and perform one action such as moving a mouse or pressing a key. Others, called composite steps, may contain additional steps.

Use the **Desktop Automation Editor** to edit the Desktop Automation workflow. The editor presents a view of the robot and the applications being automated along with details on the robot state and buttons to control the robot manually.

For more information on Desktop Automation, see the Kofax RPA online Help.

How to configure Desktop Automation Service

To use Desktop Automation to automate applications on a remote computer, you need to install the Desktop Automation Service and connect the service to Design Studio. For details, see the *Kofax RPA Installation Guide* and "Configure Desktop Automation" in the Kofax RPA online Help.

As the network environment and applications may vary with each computer, the tutorial "How to build a robot with Desktop Automation" does not involve the use of applications on a remote computer. Therefore, you can build the tutorial robot in Design Studio without installing the Desktop Automation Service.

How to build a robot with Desktop Automation

This step-by-step tutorial shows how to create and edit the Desktop Automation workflow for a robot. The tutorial covers some of the most commonly used Desktop Automation functions and consists of five main parts:

- Using the built-in browser, extract information from the Training page on the Kofax website (https://education.kofax.com/training/schedule.php) about the first five training courses available in the schedule
- Add expressions to extract information based on a condition
- Using built-in Excel, write the extracted information to a spreadsheet
- Save the Excel file to a local folder
- Close the browser and Excel

Preliminary steps

Before proceeding to the main sections of the tutorial, complete the following preliminary steps. First, you need to create a robot in Design Studio, add a Desktop Automation step to it, and then create a type to store the extracted data.

1. **Create a robot**

   a. Start **Design Studio**.

   b. Click **File** > **New Robot**.

   c. Name the robot **training_schedule**, select a project, and then click **Next** > **Next**.
   By default, the Smart Re-execution (Full) execution mode is selected.

   d. Click **Finish**.
   By default, the End step is selected in the created robot.

   e. Insert an **Action Step** in the new robot.
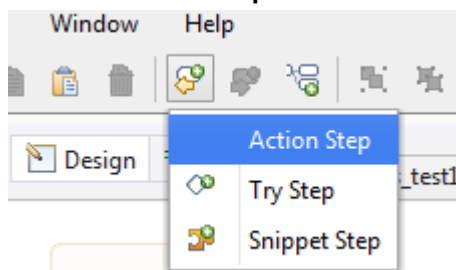
   f. To make it a Desktop Automation step, on the **Action** tab on the right, click **Select an Action** and then select **Desktop Automation**.
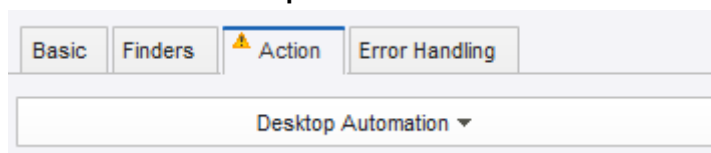
2. **Create a type**

   a. Click **File** > **New Type**.

   b. Name the type **training_schedule**, select a project, and then click **Next**.

   c. Click the plus sign to add new attributes to the type. Add the following attributes and specify their types:

Name	Attribute Type
date	Short Text
course	Short Text
location	Short Text
language	Short Text
can_register	Short Text
link	Short Text

   Short Text is a simple type that can contain text, not exceeding one line.

When finished, in the Desktop Automation step of the robot, in the **Workflow** property, click **Edit** to start designing the automation workflow. When the Desktop Automation Editor appears, proceed to the next section.

Open website and Excel

1. **Open the Kofax website with the built-in browser**

   a. To open the Training page of the Kofax website in the built-in browser, right-click the first program point (small circle) and click **Open step**.

   b. Copy and paste the following URL to the **URI** property:

   `https://education.kofax.com/training/schedule.php`

   c. Click **Step Over** 🔁 to execute this step. In the **Recorder View**, the website is opened in a new tab.
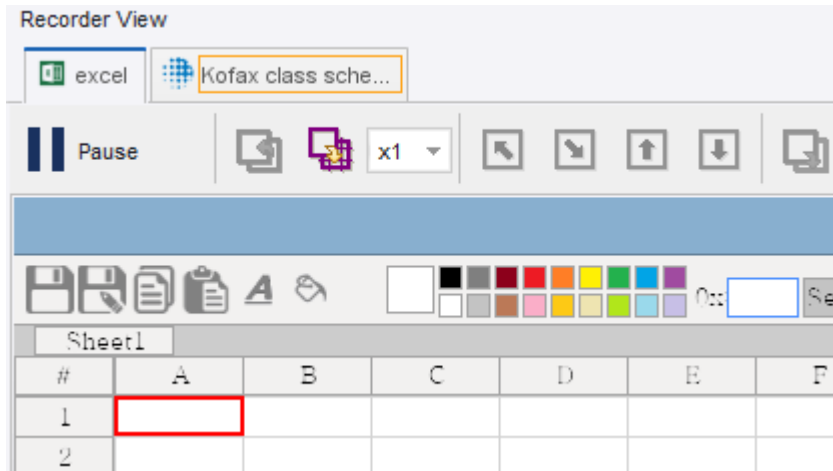
2. **Open the built-in Excel**

   a. To open the built-in Excel, right-click the next program point in the workflow and click **Open step**.

   b. In the **URI** property, type:

   ```
   excel://new
   ```

   c. Click **Step Over** ⤷ to execute this step. In the **Recorder View**, Excel is opened in a new tab.

When finished, proceed to the next section.

Extract information from website

1. **Locate the website area to extract information and add a loop**

   a. In the built-in browser, scroll down to the table with training courses. Right-click the first row of the table and click **Loop** > **Each Table Row** > **Exclude first row**.

   This action inserts in your robot a For Each Loop step that iterates over all table rows, except for the header row.

   b. In the workflow, expand the "For Exclude first row" step and do the following:

   - Expand the **Component** box and verify that the properties match the following screen.
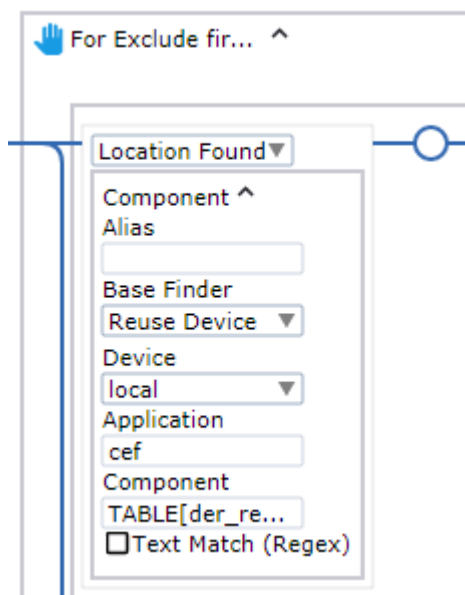
   - Expand the **For Each Loop** box, select **Iteration Variable** to store the iteration value, and enter **i** as the variable name. Also, verify that **Exclude First** is selected.
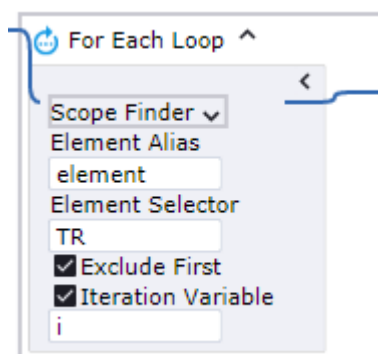
   - Add a condition to the loop to extract information only on the first five training courses (from the first five rows). At the bottom of the box, right-click the program point and click **Conditional step**. In the step, click the plus sign, click the text field, type the condition **i**

> **4**. Click the gray bar on the left so the equal sign appears, and the expression can be evaluated.
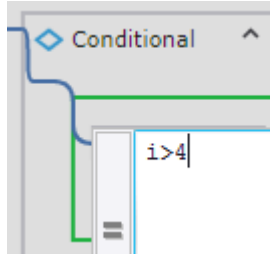
Then, right-click the program point next to the text field and click **Break step**. Double-click the program point next to the Conditional step to execute to this point.


The loop now iterates over the first five rows and then stops.

2. **Add steps to extract the information**

    a. Before adding steps that extract information from the table rows and write it to an Excel spreadsheet, for convenience, add a Group step to the For Each Loop step. In the **For Each Loop** box, right-click the program point to the right of the Conditional step and click **Group step**. For example, you can name it "Handle Schedule Data."

    b. In the Group step, expand the **Variables** box, specify a name for the variable to store extracted information, such as "schedule," and from the drop-down list, select the type

**training_schedule**. Double-click the program point inside the created Group step to execute to this point.
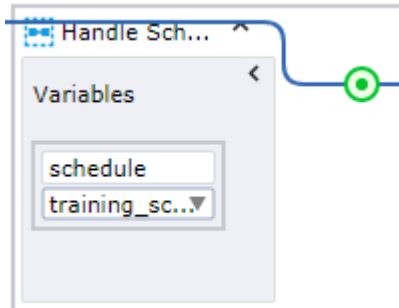
c.   In the built-in browser, in the table with training courses, right-click the date range in the first cell of the second row and then click **Extract Value From** > **Text Into** > **schedule: training_schedule** > **date: Text**.
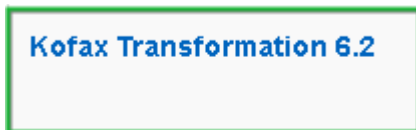
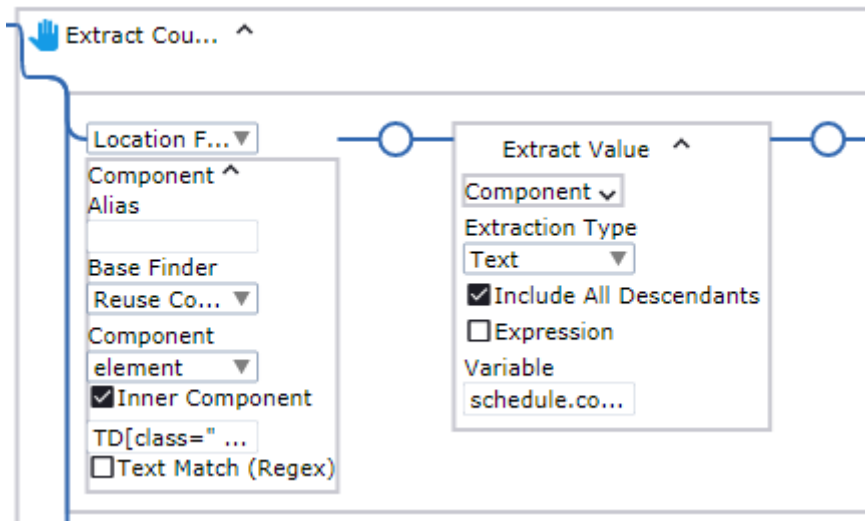Ensure that the date range is selected, but not the entire cell.
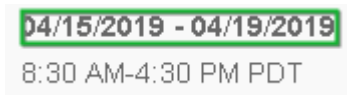
The Extract Value step is inserted in the workflow. For example, you can name it "Extract Date."
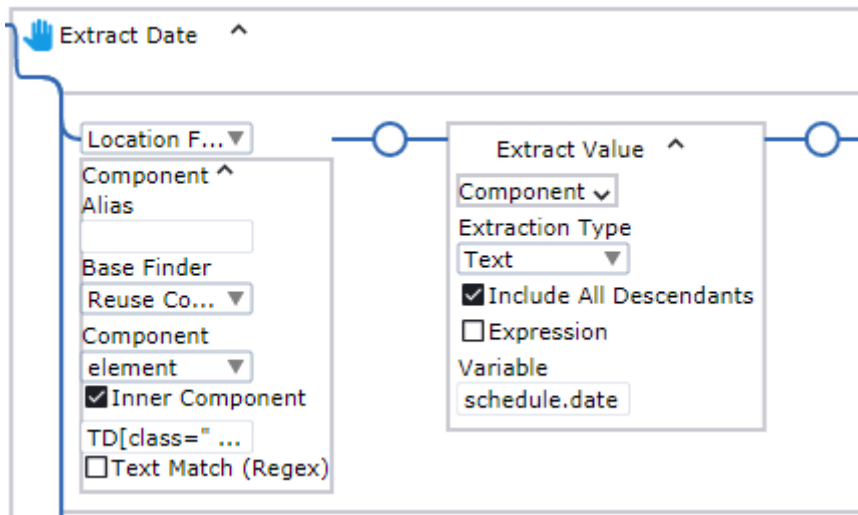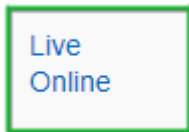

d.   Right-click the entire cell with the course name and click **Extract Value From** > **Text Into** > **schedule: training_schedule** > **course: Text**.

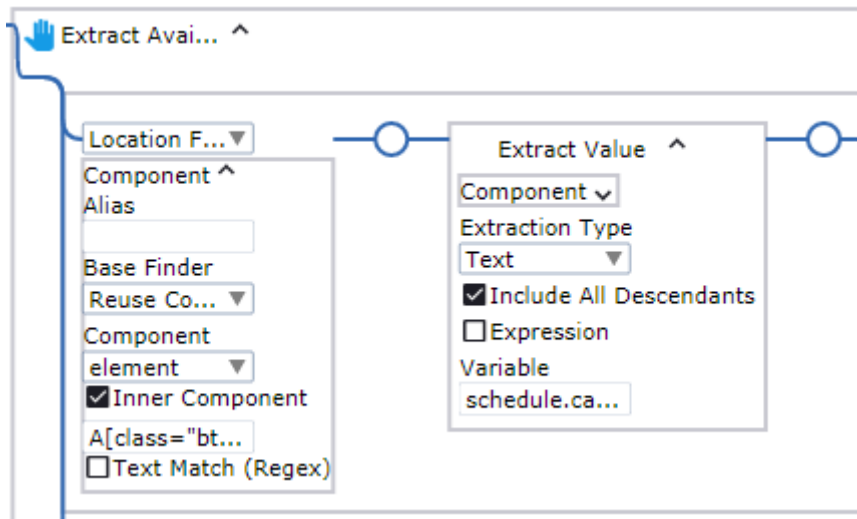The Extract Value step is inserted in the workflow. For example, you can name it "Extract Course Name."

e. Repeat the same action for the cells with the course location and language, but select the respective variable each time: **location** and **language**.
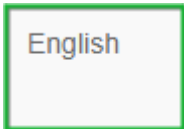

f. Right-click the Register button in the last cell of this row and click **Extract Value From** > **Text Into** > **schedule: training_schedule** > **can_register: Text**.

Ensure that the Register button is selected, but not the entire cell.

The Extract Value step is inserted in the workflow. For example, you can name it "Extract Availability."

g. Right-click the Register button once again and click **Extract Value From** > **Attribute** > **href into** > **schedule: training_schedule** > **link: Text**.

The Extract Value step is inserted in the workflow. For example, you can name it "Extract Link."
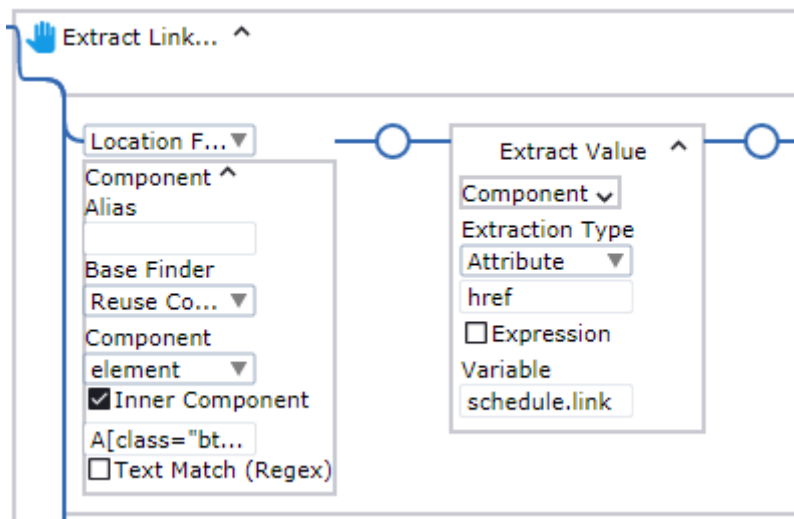
In the end, you have six Extract Value steps. Group the steps. For convenience, you can name the group "Extract Schedule."


When finished, proceed to the next section.

Add conditions to the extract steps

In the previous section, you extracted the value of the Register button into the variables of type Text. You can convert the extracted value of "can_register," which can be either *Register* or *Closed*, to a more detailed phrase, depending on whether the course is available for registration or closed. The "link" variable can be used together with the Guarded Choice step to create a condition to only extract the link to the registration form if the registration is open.

1. **Use a conditional expression to convert the value of the can_register variable**

   a. To add a conditional expression to convert the value of "can_register," expand the Extract Value step with this variable, expand the **Extract Value** box, and select **Expression**.

   b. Type the following in the **Expression** field: **$extract == "Register" ? "Registration is open at" : "Course is currently not available"**. Click the gray bar on the left so the equal sign appears, and the expression can be evaluated.
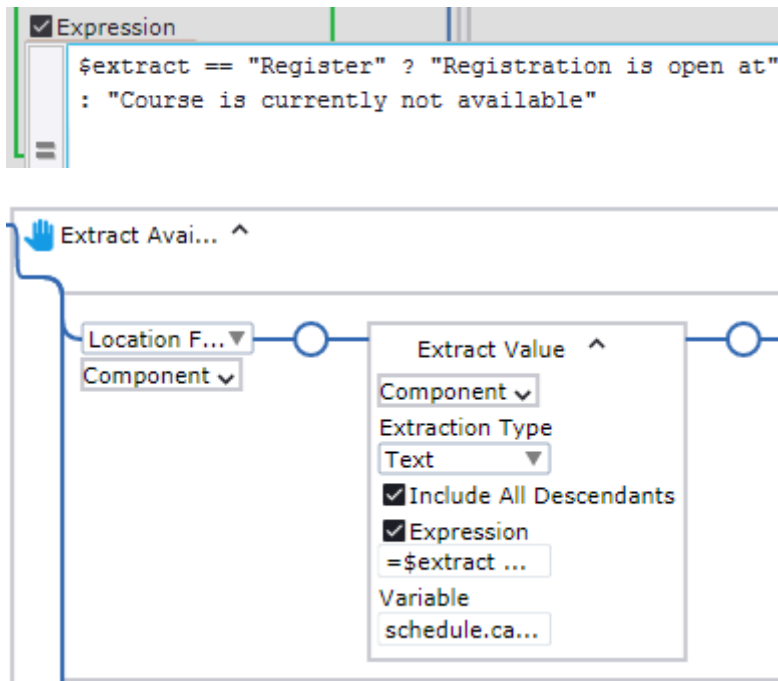

With this expression, when the extracted value is "Register," the robot converts it to "Registration is open at." When the value is "Closed," the robot converts it to "Course is currently not available."

2. **Use the Guarded Choice step to extract the link**

   The Guarded Choice step is used to set a number of conditions, each associated with some actions. Steps are executed for whichever guard is satisfied first. You can use this functionality to create a condition to only extract the link to the registration form if the registration is open.

   a. In the Extract Value step with the "link" variable, expand the **Component** box under **Guard Location Found** and select **Text Match (Regex)** to search for the matching text. In the text field, type **Register**.
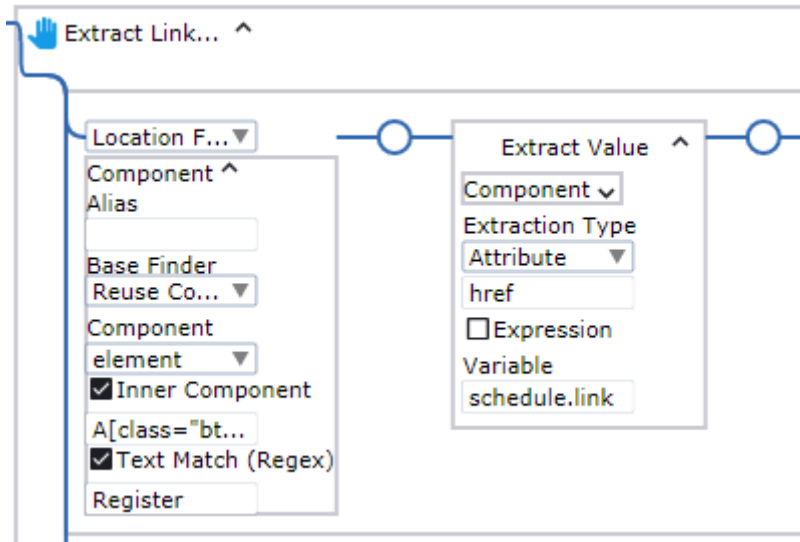
   b. In the **Guard** section below, set the guard to **Location Not Found**.

   c. Copy the entire Component box from the Location Found guard and paste it in the Location Not Found guard Component box.
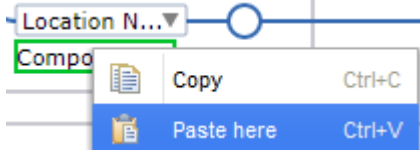
   Check that the settings of the two components now match.
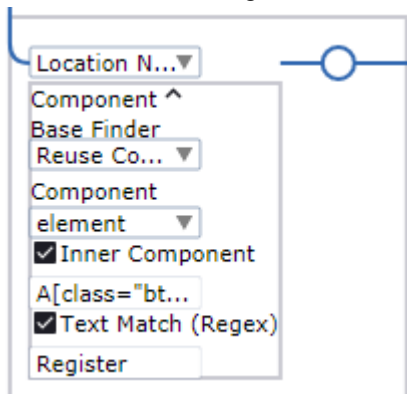
    d. Remove the **Throw TimeOutError** step as it is no longer necessary.

        With the created condition, the link to a training course registration form is only extracted if the button says **Register**, so the registration is open.

    e. Double-click the program point next to the group of Extract Value steps to execute to this point.

Tip After the group is executed, you can check the extracted values. In the **Workflow State** pane on the right, expand the **Variables** branch.

When finished, proceed to the next section.

Write extracted information to Excel

1. **Add steps that write the extracted information to an Excel spreadsheet**

   a. In Excel, right-click the A1 cell, and then click **Enter Text In Field** > **From variable** > **schedule: training_schedule** > **date: Text**.

   The "Input text from" step is inserted in the workflow. For example, you can name it "Input Date."
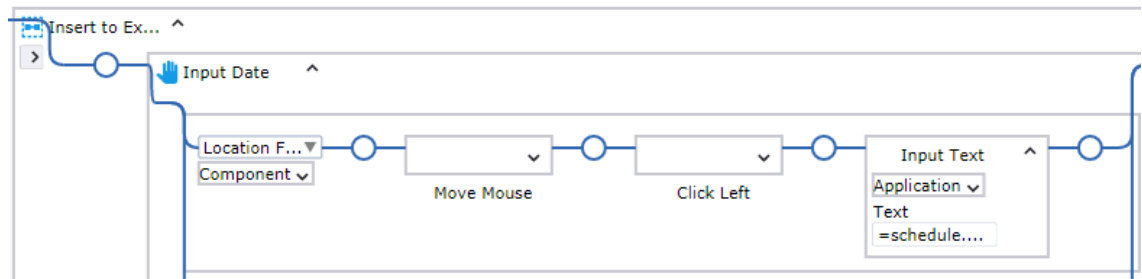

   b. Expand the step, expand the **Component** box, and then add **+ (i+1) + "\"** to the expression in the **Component** field to move to the next line in the spreadsheet after the current line. The final expression must look like the following: **"Cell[Address=\"A" + (i+1) + "\"]"**.

   Click the gray bar on the left so the equal sign appears.
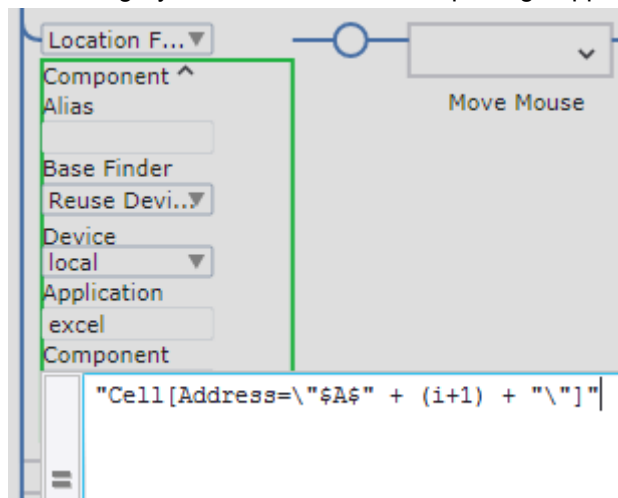
   c. Repeat these actions for the next five cells in the spreadsheet, but select the respective variable each time: **course**, **location**, **language**, **can_register**, and **link**.

   d. In the end, you have six "Input text from" steps. Group the steps. For example, you can name the group "Insert to Excel."
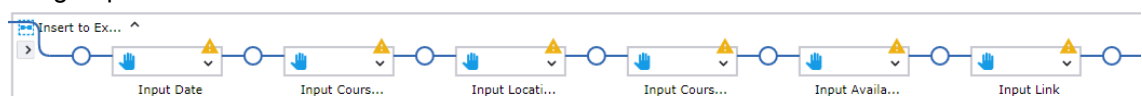
e. Double-click the program point next to the "For Exclude first row" step to execute the entire step.
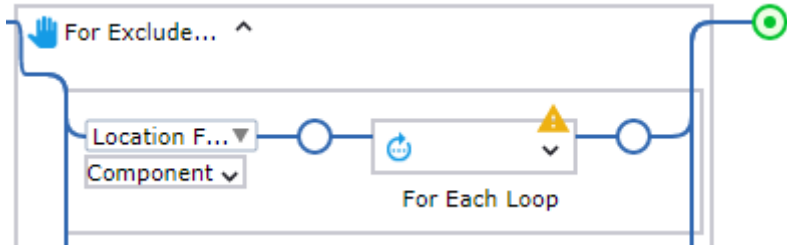
Observe how the information is being extracted from the web page and written into Excel.

When finished, proceed to the next section.

Save Excel file locally and close applications

1. **Save the Excel spreadsheet**

   a. In Excel, right-click the Save button and click **Click** > **Left**.

   The Left Click step is inserted in the workflow. Click **Step Over** to execute this step.

   b. In the Save As dialog box, select the "File name" text field. To do so, click inside the field and then use the **Select Next Node Matching Click** button to select the correct field.

   Then, right-click the field and click **Enter Text in Field** > **Manually Enter Text**.

   Specify the location where to save the file and the file name, such as `C:/Documents/KofaxTrainingSchedule.xlsx`, and click **OK**. Make sure the specified path exists.

   The Input step is inserted in the workflow. Click **Step Over** to execute this step.

   c. Select and right-click the **Save** button and then click **Click** > **Left**.

   The Left Click step is inserted in the workflow. Click **Step Over** to execute this step.

For convenience, you can rename the steps.

2. **Close built-in Excel and the built-in browser**

   To ensure that open windows are not duplicated when re-starting the robot, which may lead to an error, add steps that close the Excel window and the web page at the end of the run.

   a. Select the Excel tab, right-click the Close button in the upper right corner and click **Click** > **Left**.
       Click **Step Over** ↪ to execute this step. The Excel tab is now closed.

   b. Select the browser tab and perform the same action on the Close button.
       Click **Step Over** ↪ to execute this step. The browser tab is now also closed.

   For convenience, you can rename the steps.

   > **Tip** As an alternative, to close applications in your other Desktop Automation robots without recording the steps in the workflow, you can perform an instant mouse click on a respective Close button. To perform this action, right-click the element in the **Recorder View**, click **Unrecorded Instant Click**, and select the **Left** mouse click. The Unrecorded Instant Click action is also useful when you need to see available options in context menus and drop-down lists without recording these actions in the workflow.

Your Desktop Automation workflow is now ready for use. After you save the created Desktop Automation workflow, refresh it, and then click **Start Execution** ▶ to execute the workflow from the beginning. When the robot finishes executing, navigate to the selected location and review the results in your Excel file.