

Kofax RPA

Getting Started with Desktop Automation

Version: 11.0.0

Date: 2019-12-18

The KOFAX logo is displayed in a bold, blue, sans-serif font. The letters are uppercase and spaced evenly, with a slight shadow effect behind the text.

© 2019–2020 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

Table of Contents

Preface.....	4
Related Documentation.....	4
Training.....	5
Getting help with Kofax products.....	5
Chapter 1: Build a Desktop Automation robot.....	7
Desktop Automation overview.....	7
How to configure Desktop Automation Service.....	7
How to build a Desktop Automation robot.....	7
Open website and Excel.....	10
Extract information from website.....	12
Write extracted information to Excel.....	20
Save Excel file locally and close applications.....	22

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 RPA Installation 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/11.0.0_qrvv5i5e1a/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 Upgrade Guide

Contains instructions on upgrading Kofax RPA and its components to a newer version.

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 Best Practices Help for Robot Lifecycle Management

Offers recommended methods and techniques to help you optimize performance and ensure success while using Robot Lifecycle Management in your Kofax RPA environment.

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 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.

Kofax RPA Developer's Guide

Contains information on the API that is used to execute robots on RoboServer.

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.

Chapter 1

Build a Desktop Automation robot

Desktop Automation overview

With a Desktop Automation robot you can 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.

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.

When editing the Desktop Automation workflow, you are presented with 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 *Help for Kofax RPA*.

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 *Help for Kofax RPA*.

As the network environment and applications may vary with each computer, the tutorial "[How to build a Desktop Automation robot](#)" 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 Desktop Automation robot

This step-by-step tutorial shows how to create, edit and use a Desktop Automation robot. The tutorial covers some of the most commonly used Desktop Automation functions such as Loop steps, Extract Value steps, Conditional step, Assign step, Enter Text step, and others.

The tutorial consists of four main parts:

- Using the built-in browser, extract information from the Rankings page on the ATP Tour website (<https://www.atptour.com/en/rankings/singles>) about the top five tennis players in the world
- Using built-in Excel driver, write the extracted information to a spreadsheet
- Save the Excel file to a local folder
- Close the browser and Excel

Preliminary steps

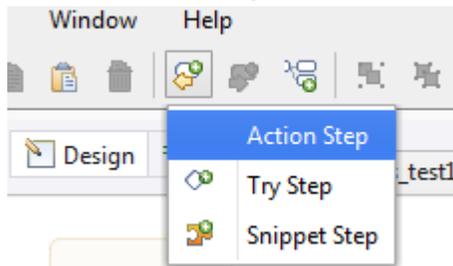
Before proceeding to the main sections of the tutorial, complete the following preliminary steps. First, you need to create a web automation robot in Design Studio, create a Desktop Automation robot, and then call it from the web automation robot. You also need create a type to store the extracted data.

1. Create a web automation robot

- a. Start **Design Studio**.
- b. Click **File > New Robot**.
- c. Name the robot **TennisPlayers**, select a project, and then click **Finish**.

The new robot appears on a new tab in the editor window. By default, the Smart Re-execution (Full) execution mode is selected and the End step is selected in the created robot.

- d. Insert an **Action Step** in the new robot.



- e. Save the changes.

2. Create a Desktop Automation robot

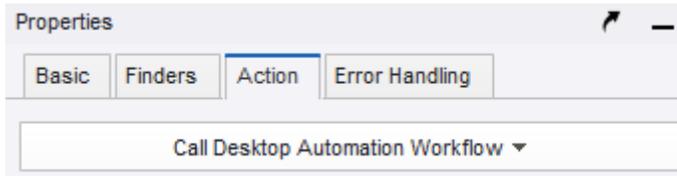
- a. Click **File > New Desktop Automation Robot**.
- b. Name robot **TennisPlayersDA**, select a project, and then click **Finish**.

The new robot appears on a new tab in the editor window. As opposed to web automation robots that are identified by a blue icon, Desktop Automation robots are identified by a green icon: 🟢

At this point, you cannot edit the workflow as you need to first call the new robot from the web automation robot.

3. Call a Desktop Automation robot from a web automation robot

- a. Open the tab with the **TennisPlayers** web automation robot.
- b. In the inserted step, click **Select an Action** on the **Action** tab and choose **Call Desktop Automation Workflow**.



- c. In the **Workflow** drop-down list, select the **TennisPlayersDA** robot.
- d. Save the changes.
- e. Allow execution for the web automation robot by clicking **Prepare Execution** on the toolbar.

Important To allow execution for a web automation robot or take the execution privilege from one web automation robot to another, you need to use the Prepare Execution function.

When a robot has an execution privilege, a red dot appears on the robot icon and the robot tab is highlighted. When a Desktop Automation robot has an execution privilege, the tab with the web automation robot calling it is also highlighted for convenience.



- f. When the execution is allowed, you can open the Desktop Automation workflow for editing. To do so, click **Step Into DA Robot** on the toolbar.
- The tab with your **TennisPlayersDA** robot is opened and the editor is now active.

4. Create a type

- a. Click **File > New Type**.
- b. Name the type **TennisPlayersType**, 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
name	Short Text
URL	Short Text
points	Short Text

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

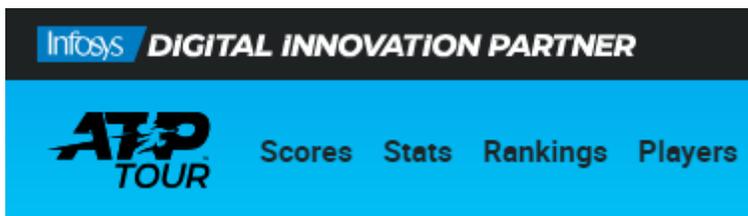
- d. Save the changes.

You are now ready to start designing the automation workflow. Proceed to the next section.

Open website and Excel

1. Open the ATP Tour website with the built-in browser

- a. To open the Rankings page of the ATP Tour website in the built-in browser, right-click the first flow point (small circle) and click **Open step**.
- b. Expand the inserted Open step and paste the following URL to the **URI** property:
`https://www.atptour.com/en/rankings/singles`
- c. Click **Step Over**  to execute this step. In the **Recorder View**, the website is opened in a new tab.



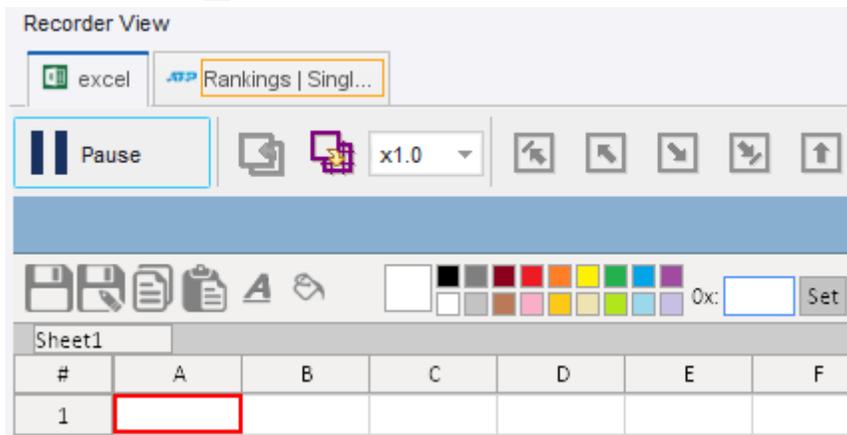
Tip If you have some information to note, you can leave comments to the steps in your automation workflow.

Comment
This is a link to the Singles page.

To enable comments, on the toolbar, click **Window > Comment**. To write or change a comment, click a step or a Group step and add/change your notes in the Comment window. You can use the Undo and Redo buttons here. The comment is automatically saved when you click outside the window.

2. Open the built-in Excel driver

- a. Right-click the next flow 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

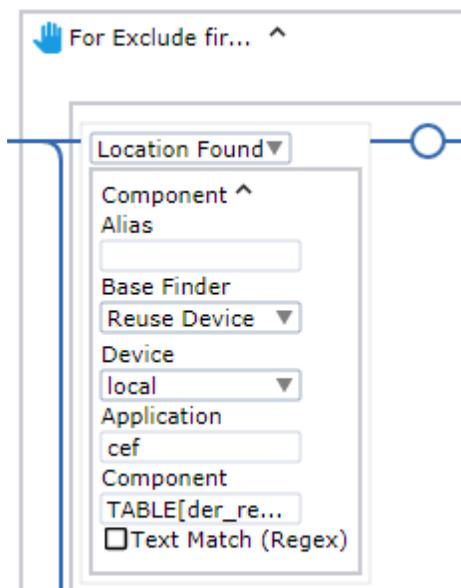
In this procedure, you extract information on the top five tennis players according to the Men's Tennis ATP Rankings, which includes player name, points, and URL to the player profile on the ATP Tour website.

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

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

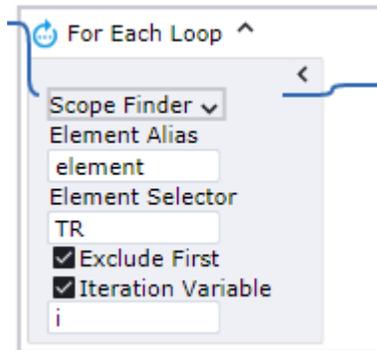
This action inserts in your robot a For Each Loop step that iterates over all table rows, except for the header row.

- 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.

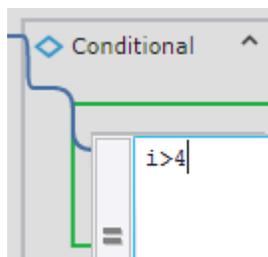


The Component field contains the specific upper-level table element found by the selector: `TABLE[der_rendered="y"]`. This component is used to find the correct elements when looping over the table rows.

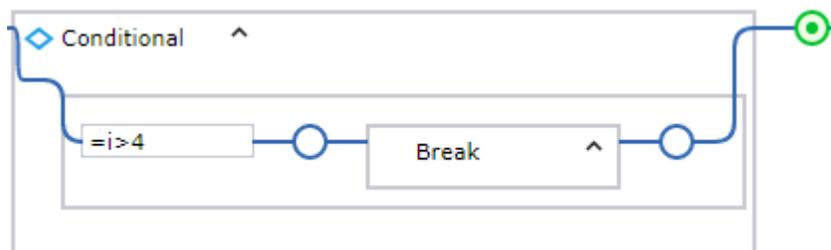
- Expand the **For Each Loop** box, then click **>** to open the loop property panel. Select **Iteration Variable** to store the iteration value, and enter **i** as the variable name. Also, verify that **Exclude First** is selected.



- To extract information only on the first five tennis players (from the first five rows), add a condition to the loop. Right-click the flow point to the right of the loop property panel and select **Conditional step**. In the step, click the plus sign then click the text field and 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 flow point next to the text field and click **Break step**. Double-click the flow point next to the Conditional step to execute to this point.



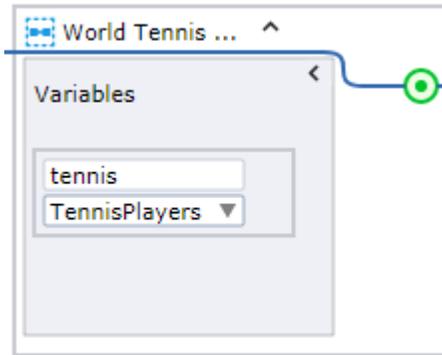
The loop now iterates over the first five rows and then stops.

2. Add steps to extract the information

- 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**

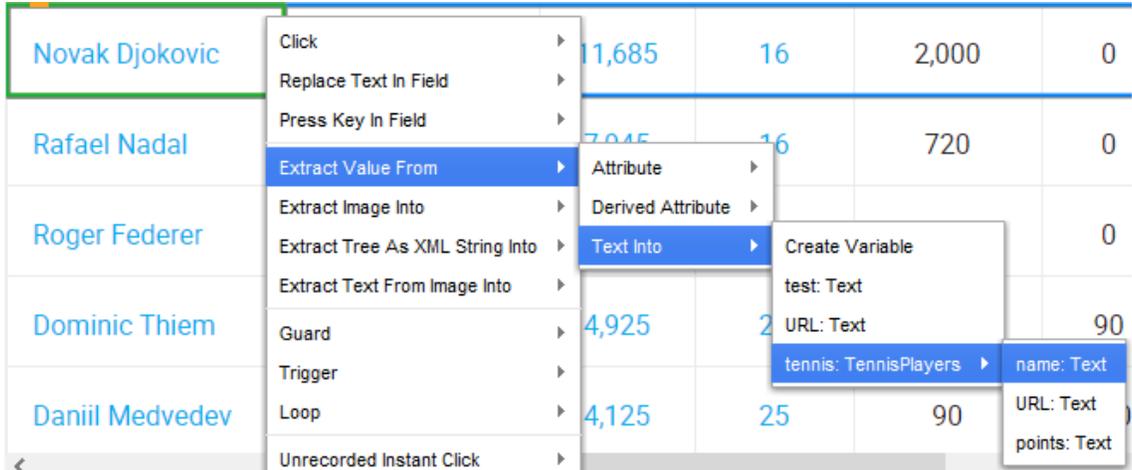
Loop box, right-click the flow point to the right of the Conditional step and click **Group step**. For example, you can name it "World Tennis Rankings."

- b. In the Group step, expand the **Variables** box, specify a name for the variable to store extracted information, such as "tennis," and from the drop-down list, select the type **TennisPlayers**. Double-click the flow point inside the created Group step to execute to this point.

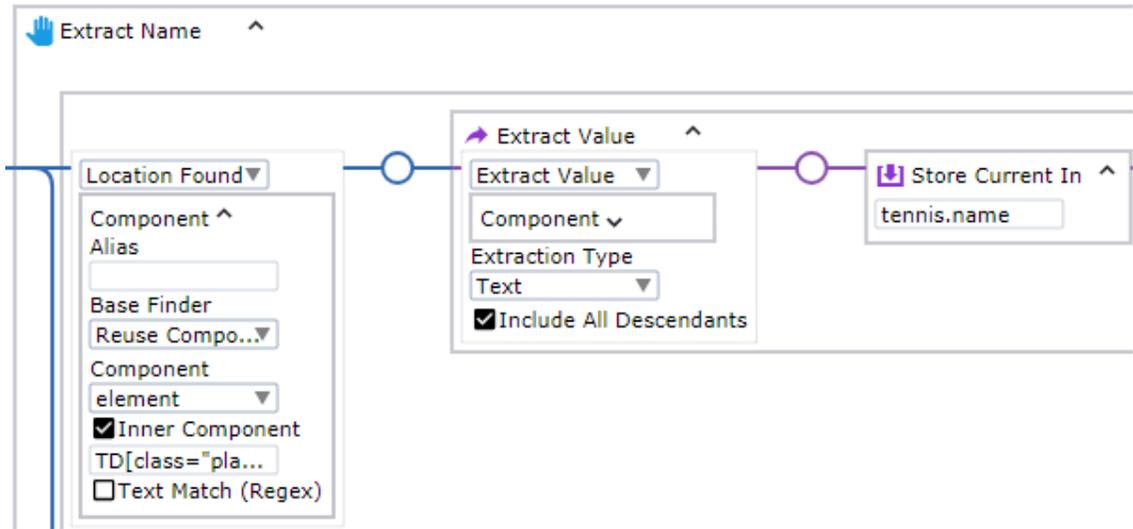


- c. In the built-in browser, in the table with tennis players, locate the second row and then right-click the cell with the name of the tennis player ranked first and then click **Extract Value From > Text Into > tennis: TennisPlayers > name: Text**.

To ensure that the cell is selected, not the name itself, click the name and then click  to select the parent element of the text element.

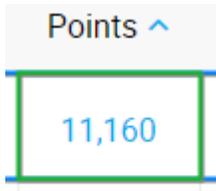


The Extract Value step is inserted in the workflow. For example, you can name it "Extract Name."

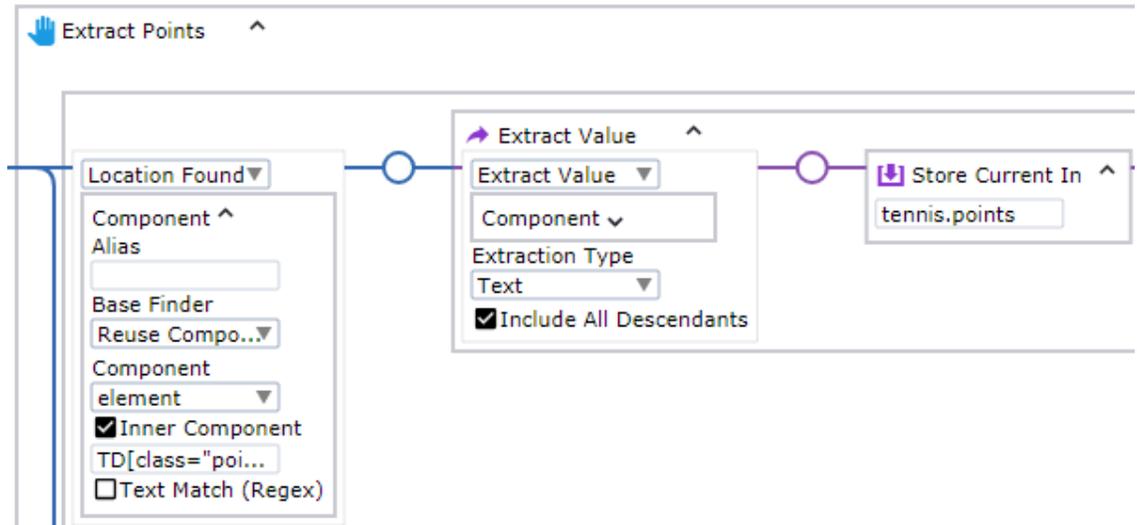


The Inner Component field is used to find components within the already found table component and contains the following lower-level table element: `TD[class="player-cell"][der_rendered="y"]`. TD denotes "table row."

- d. Click **Step Over**  to execute this step.
- e. In the same row, right-click the cell with the points achieved by the first player and click **Extract Value From > Text Into > tennis: TennisPlayers > points: Text**.



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



The Inner Component field contains the following lower-level table component:
TD[class="points-cell"][der_rendered="y"].

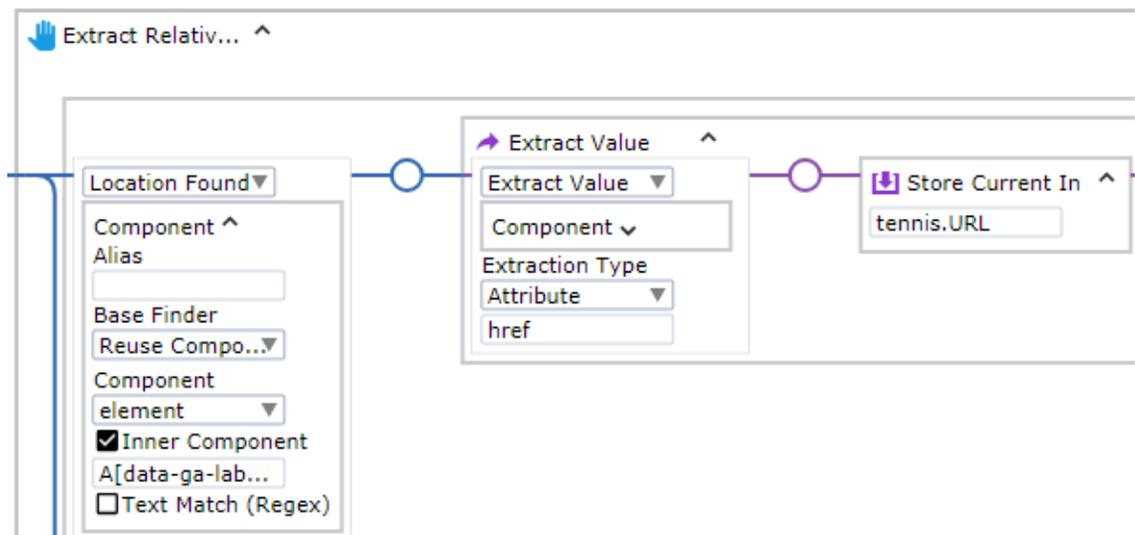
- f. Click **Step Over**  to execute this step.

3. Extract URL to the player profile on the ATP Tour website

- a. Right-click the points once again, this time selecting the figure itself, not the entire cell. Then click **Extract Value From > Attribute > href into > tennis: TennisPlayers > URL: Text**.



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

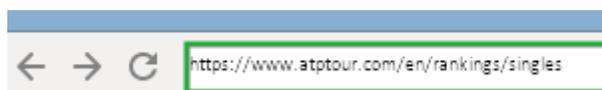


The Inner Component field contains another lower-level table element: `A[data-ga-label="rankings-breakdown"][der_rendered="y"].A` denotes "anchor," which is an element that usually contains hyperlinks.

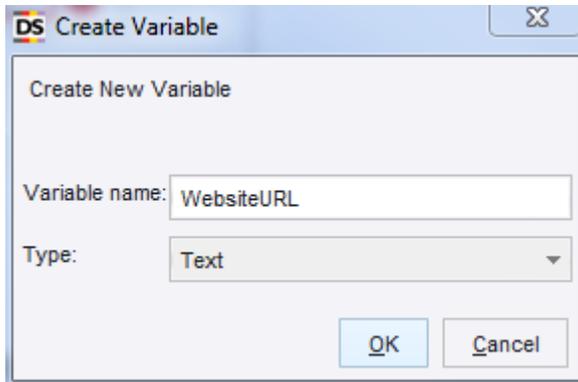
This step extracts the *relative* URL to the player profile on the ATP Tour website. For example, `/en/players/novak-djokovic/d643/overview`. This URL cannot be used independently as it does not contain the base URL `https://www.atptour.com`.

To convert it to an *absolute* URL that can be used separately, you need to extract the Rankings page URL, trim it to the base URL as shown below, and then concatenate it (link) with the relative URL as shown in [Write extracted information to Excel](#).

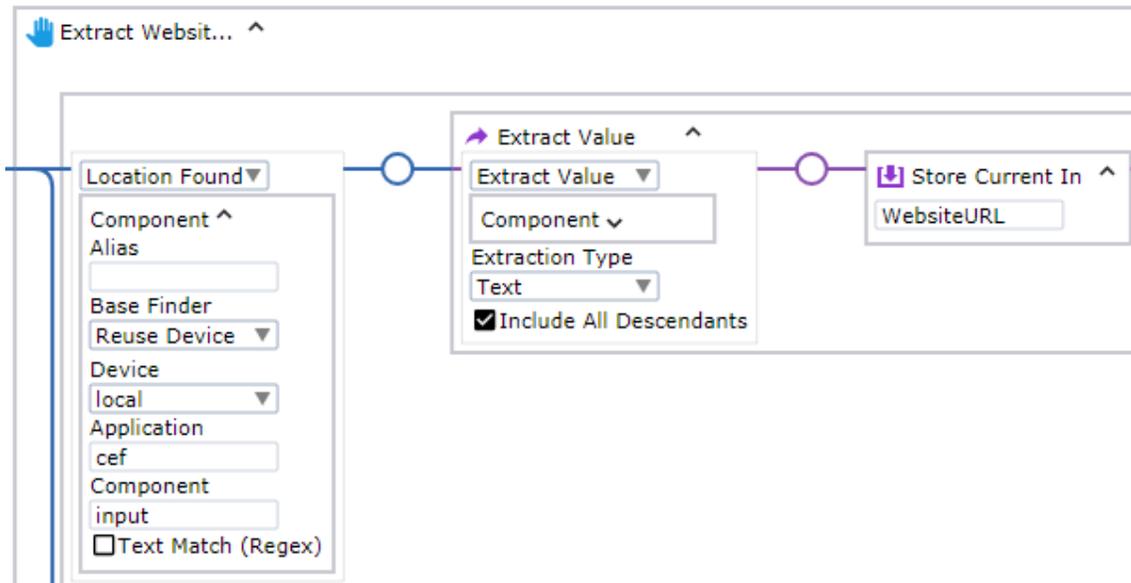
- b. Click **Step Over**  to execute this step.
- c. Right-click the website address at the top of the page and click **Extract Value From > Text Into > Create Variable**.



- d. In the new dialog box, assign a name for the variable to store the extracted URL. For example, name it **WebsiteURL** and then click **OK**.



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

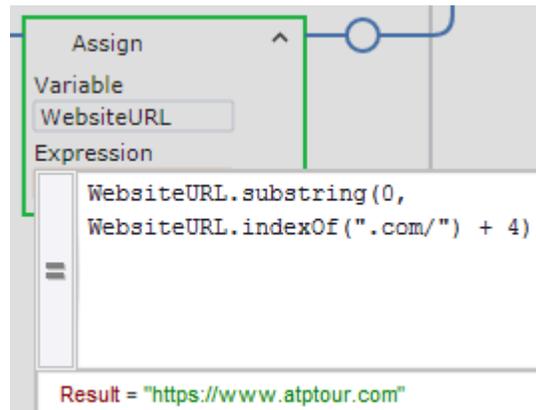


This step extracts the URL to the Rankings page on the ATP Tour website and stores it in a variable.

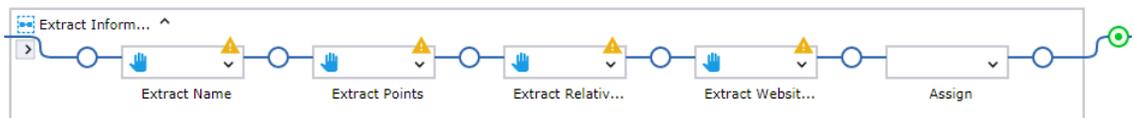
- e. Click **Step Over** to execute this step.
- f. Now you need to trim the Rankings page URL so it only contains the base URL.
 1. Right-click the flow point after the **Extract Website URL** step and click **Assign Step**. The Assign step is inserted in your workflow. Expand the step.
 2. In the **Variables** field, enter **WebsiteURL**. In the **Expression** field, enter the following expression: **WebsiteURL.substring(0, WebsiteURL.indexOf(".com/") + 4)**

With this expression, the string contained in the WebsiteURL variable is trimmed to a substring. In other words, the Rankings page URL is shortened to only contain the base part ending with ".com".

3. Click the gray bar on the left so the equal sign appears, and the expression can be evaluated.



- g. Click **Step Over** to execute this step.
- h. In the end, you have four Extract Value steps and one Assign step. Group the steps. You can name the group "Extract Information."



- i. Double-click the flow point next to this group 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.



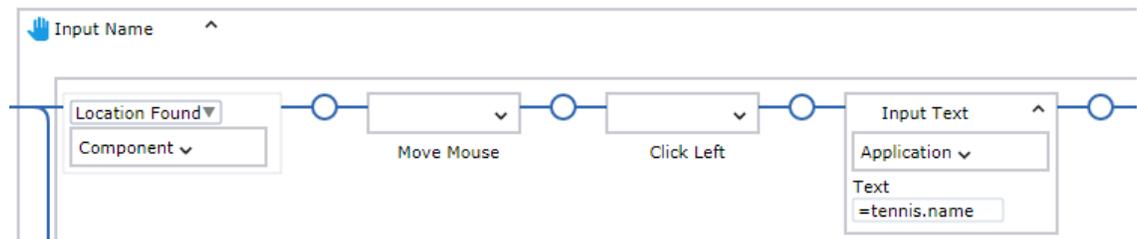
When finished, proceed to the next section.

Write extracted information to Excel

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

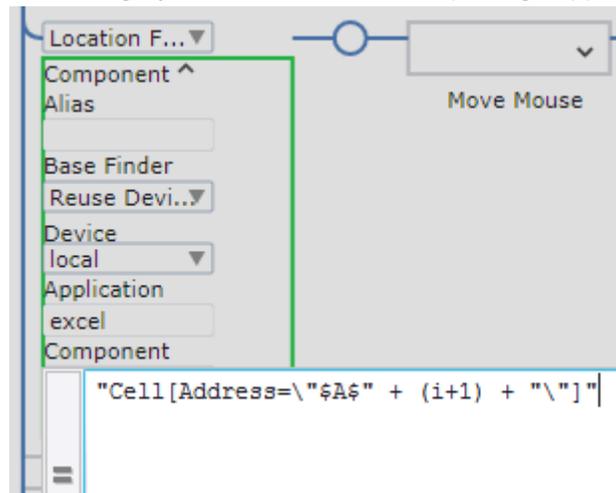
- In Excel, right-click the A1 cell, and then click **Replace Text In Field > From variable > tennis: TennisPlayers > name: Text**.

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

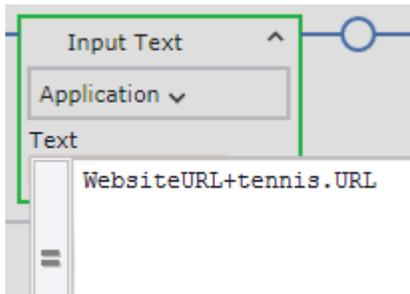


- Expand the step, expand the **Component** box, and then add **+ (i+1) + "\n"** to the expression in the **Component** field to move to the next line in the spreadsheet after the current line. The final expression must look like the following: **"Cell[Address="\\$A\$" + (i+1) + "\n"]"**.

Click the gray bar on the left so the equal sign appears.

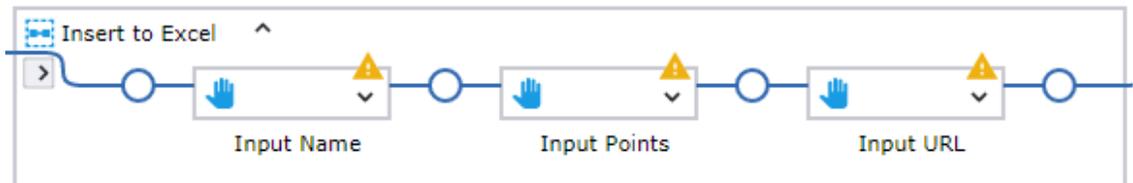


- Click **Step Over**  to execute this step.
- Repeat these actions for the B1 and C1 cells in the spreadsheet, but select the respective variables each time: **points** and **WebsiteURL**. Rename the steps to "Input Points" and "Input URL," respectively.
- Expand the **Input URL** step and then expand the **Input Text** box. In the **Text** field, enter the following expression: **WebsiteURL+tennis.URL**. Click the gray bar on the left so the equal sign appears, and the expression can be evaluated.

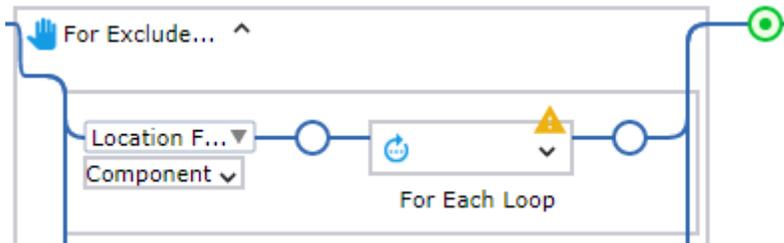


With this expression, the two URLs that you extracted in the previous section are concatenated to become an absolute URL to the tennis player profile on the ATP Tour website.

- f. In the end, you have three "Input text from" steps. Group the steps. For example, you can name the group "Insert to Excel."



- g. Double-click the flow 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.



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/WorldTennisRankings.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 driver 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 robot 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.

- To step out of the Desktop Automation robot and switch to the web automation robot, click **Step Out**  on the toolbar after the entire workflow is executed. In the web automation robot, the Call Desktop Automation Workflow step is now shown as executed.

- To close the robot without saving the result, click **Leave Robot**  on the toolbar. The tab with the web automation robot is now opened. The Call Desktop Automation Workflow step is now shown as *not* executed.