

Printing Multiple Drawings as PDF Files Using UiPath

Tomi Palomäki

2023-02-09

Contents

1 Automated Drawing Processing Report **1**

1.1 Introduction 1

1.2 Code Description 1

1.3 Code Ease of Use and Modification 4

1.4 Results and Conclusion 4

1.5 Additional Note 4

1 Automated Drawing Processing Report

1.1 Introduction

The purpose of this report is to detail the automation process that opens, prints, and converts drawings stored in a specified folder to PDF format. This process was created to streamline the manual process of converting these drawings and to save time for the user.

1.2 Code Description

The program flow for this automation process is as follows:

1. The automation process starts by opening the UiPath workflow file named "open-drawing-print-pdf.xaml".
2. The workflow file contains the main function, "open_drawing_print_pdf", which is executed to process the drawing files.
3. The first step in the process is to get a list of all the drawing files in the specified folder using an "Assign" activity. This list of files is stored in an array named "Files".
4. A For Each loop is then used to iterate over each file in the "Files" array.
5. For each file, a "Write Line" activity is executed to display the name of the file in the output console.
6. An application, "Paint", is opened with the file as an argument using the "Use Application" activity.
7. Within the Paint application, a set of actions are performed to print the file as a PDF. This includes the use of keyboard shortcuts and selecting the "Microsoft Print to PDF" option.
8. The PDF file is then saved to the same directory as the original drawing file using the "Save As" dialog box within the Paint application.
9. Once all the files have been processed, the automation process ends.

The Paint application was operated using a series of steps generated with the UiPath "App/Web recorder" tool (see Figure 13). These steps included the use of the "Use Application: Paint" activity, a "Do Sequence" activity, a keyboard shortcut for "Ctrl+P" in UiPath, an activity for selecting an item from a dropdown list, a click on the Print button, a "Type Into" activity for specifying the file save location, and a click on the Save button. Some of the parameters were modified after recording to tailor the automation process to the specific requirements.

1.2.1 Main

The screenshot in Figure 1 shows the visual representation of the UiPath workflow where the 'Invoke Workflow File' activity is used to open and print a drawing as PDF. The workflow file name is specified as "open-drawing-print-pdf.xaml". The activity has no input arguments as the 'ArgumentsVariable'

1.2.2 getting list of files in a directory

The screenshot in Figure 2 shows an Assign activity in UiPath, which is used to get a list of files in a specific directory. The activity takes the directory path as an input and returns an array of strings, which contains the names of all the files present in that directory. In this particular example, the directory path used is "C:/drawings" and the output of this activity is stored in the variable "Files". This list of files is then used in subsequent activities to open and process each file.

1.2.3 For Each Loop for Processing Drawing Files

This section shows the workflow activity of a For Each loop in UiPath. The loop iterates through a list of files (Files) and performs a set of activities for each file. Inside the loop, the activities are performed to process the current file. Once all the activities for the current file are executed, the loop moves on to the next file in the list. The screenshot in Figure 3 shows how the for each loop is implemented in UiPath.

1.2.4 Sequence Body and Variable Assignment

Sequence named "Body" contains an "Assign" activity that assigns a value to the "FileNameNoExtension" variable. The "Assign" activity has an input argument that uses the "path.GetFileNameWithoutExtension" method to retrieve the file name without its extension. The "Assign" activity also has an output argument that sets the "FileNameNoExtension" variable with the value obtained from the input argument.

The screenshot in Figure 4 shows the visual representation of the sequence and the "Assign" activity.

1.2.5 WriteLine Activity

The WriteLine activity is used to display a message in the output console during the automation process. In this case, the message displays the name of the file the automation process is currently working on. The screenshot in Figure 5 shows the Text property of the activity, which is set to a string containing the message to be displayed.

1.2.6 Use Application: Paint

The section describes the activity of using the Paint application, which is specified by the file path "C:/Windows/System32/mspaint.exe". The application arguments, specified by "[File]", are also shown in the Figure 6.

1.2.7 Do Sequence" activity

The Do Sequence activity is a container that enables the user to group a set of activities as a single unit of work that is executed sequentially.

The screenshot (Figure 7) shows the visual presentation of the child activities in the Do Sequence activity. Specifically, it shows the following actions: selecting an item from a list folder view, clicking a print button, typing into an edit field, and clicking an Print button.

1.2.8 Keyboard Shortcut for "Ctrl+P" in UiPath

The screenshot (Figure 8) depicts a visual representation of a sequence in UiPath, which includes the activity "Send Hotkey" with the key combination "Ctrl+P" selected. This activity is used to send a keyboard shortcut to an application.

In this particular case, the "Ctrl+P" shortcut is used to trigger the print command in the Paint application.

1.2.9 UiPath activity for selecting an item from a dropdown list.

The UiPath activity "SelectItem" with the Display Name "Select Item 'list Folder View' " is used to select item from a dropdown list. The dropdown list contains various items, and the code specifies the item to be selected using the "Item" attribute, which in this case is "Microsoft Print to PDF" (Figure 9).

1.2.10 Click Print button

The "Click" activity is configured with the "ClickType" property set to "Single" and the "MouseButton" property set to "Left". The target element to click on is defined using the "TargetAnchorable" element, which specifies the properties of the UI element such as its type, name, location, and other attributes. The "ScopeIdentifier" property identifies the scope of the UI element, and the "Version" property specifies the version of the activity. The corresponding visual representation of this code is show as a screenshot of the UI with the "Print" button highlighted or selected Figure 10.

1.2.11 UiPath Type Into Activity for File Save Location

This section of the UiPath workflow contains an NTypeInto activity that simulates typing a file path and name into a Windows Explorer file save dialog box. The activity is configured to activate the dialog box before typing, and it includes a dynamic file path generated by concatenating a fixed folder path ("c:/drawings-pdf") with the name of the file being saved (FileNameNoExtension). The target for the activity is an input box element within the Explorer pane of the save dialog, which is specified using a combination of selectors, including a full selector, fuzzy selector, and image-based selector. The scope selector for the activity indicates that the save dialog is part of the MSPaint application. An informative screenshot is included, which shows the MSPaint save dialog box with the input box for the file path and name selected Figure 11.

1.2.12 Click save button

This section of program flow contains UiPath workflow activity that clicks on the "unchecked" button in the "Save Print Output As" window of Microsoft Paint application. The activity is a NClick activity with ClickType set to "Single" Figure 12.

1.3 Code Ease of Use and Modification

The code is written in a way that makes it easy to understand the steps involved in the process, and it is also easy to modify if necessary. This means that the user can easily adjust the code to fit their specific needs, such as specifying a different folder to process or using a different application to open the drawings.

1.4 Results and Conclusion

The result of this automation process is a set of PDFs that have been converted from the original drawings stored in the `C:/drawings` folder. These PDFs are now easily shareable and can be used for a variety of purposes, including printing, emailing, or displaying on a website.

In conclusion, the automation process for converting drawings to PDF format has been successfully created and tested. The code is efficient, easy to understand, and can be easily modified to fit the user's needs. This process will save the user time and effort, allowing them to focus on more important tasks.

1.5 Additional Note

It is worth noting that the use of Paint in this example is just to demonstrate the steps needed to print a file as a PDF. The actual application used can be any software that is capable of opening and printing drawings, such as Vertex. The process remains the same, regardless of the specific application used. The main objective of this automation process is to streamline the conversion of drawings to PDF format and save time for the user.

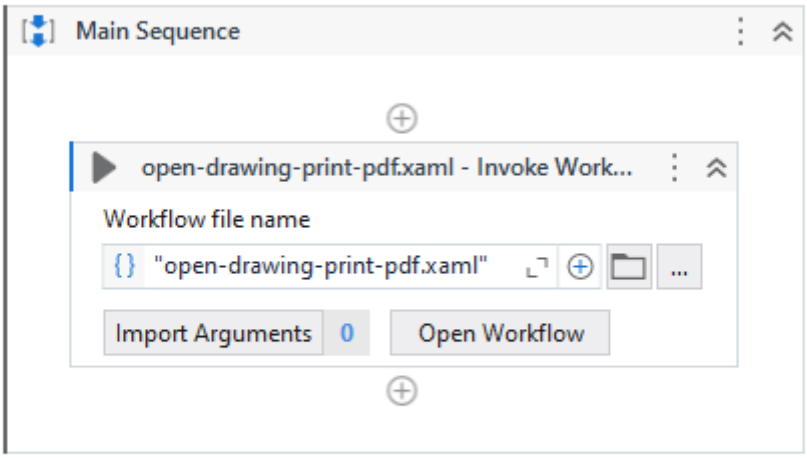

Figure 1: Screenshot of UiPath workflow diagram showing the 'Invoke Workflow File' activity for opening and printing a drawing as PDF

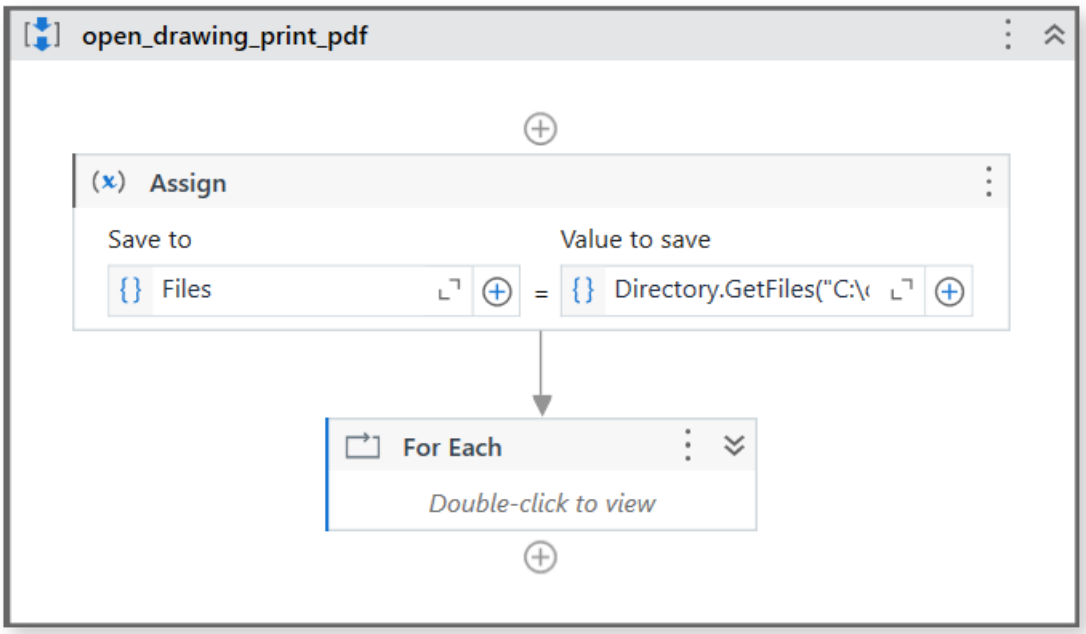

Figure 2: Assign activity for getting list of files in a directory

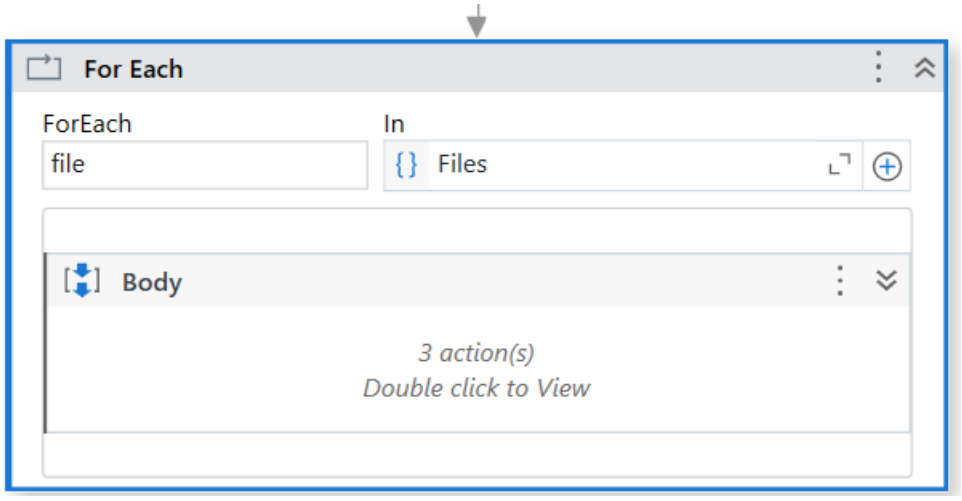

Figure 3: For Each Loop for Processing Drawing Files

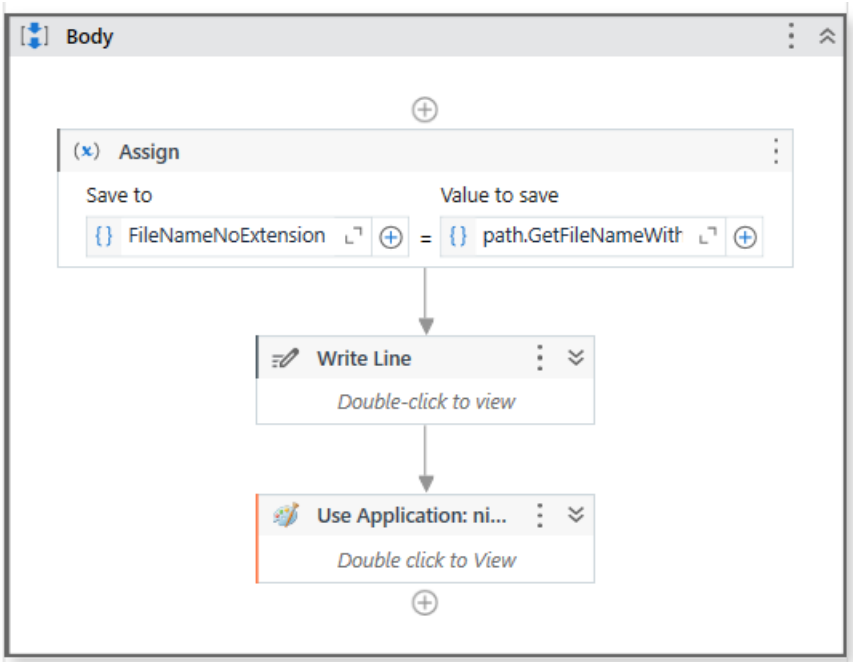

Figure 4: Sequence Body and Variable Assignment

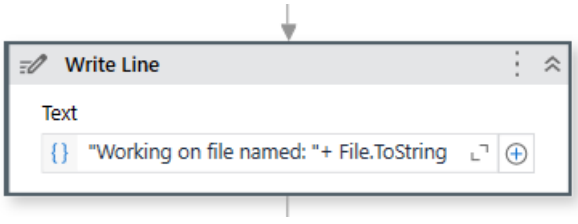

Figure 5: WriteLine Activity

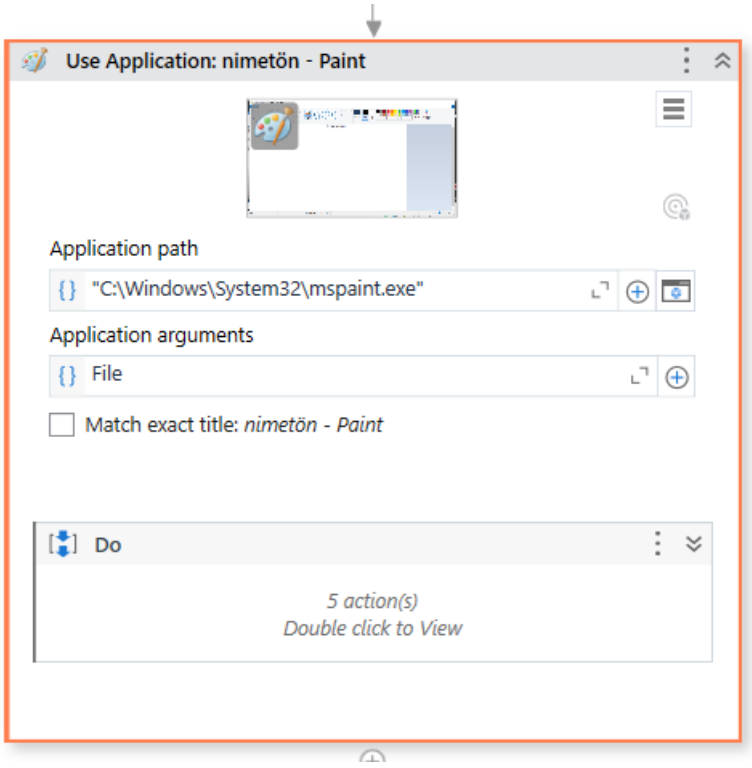

Figure 6: Use Application: nimetön - Paint screenshot

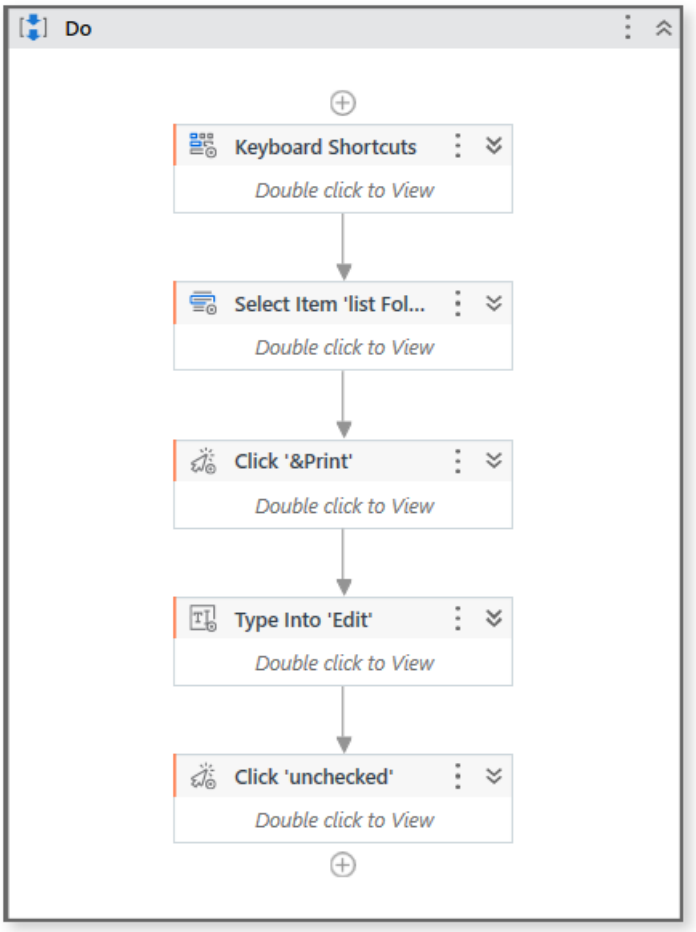

Figure 7: Do Sequence of activites recorded with App/Web recorder

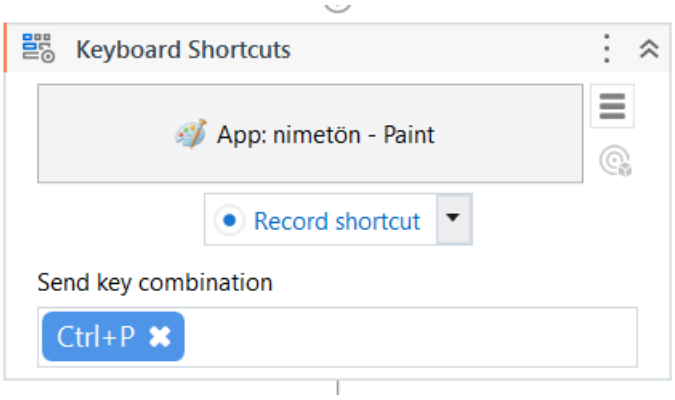

Figure 8: Keyboard Shortcut for 'Ctrl+P' in UiPath

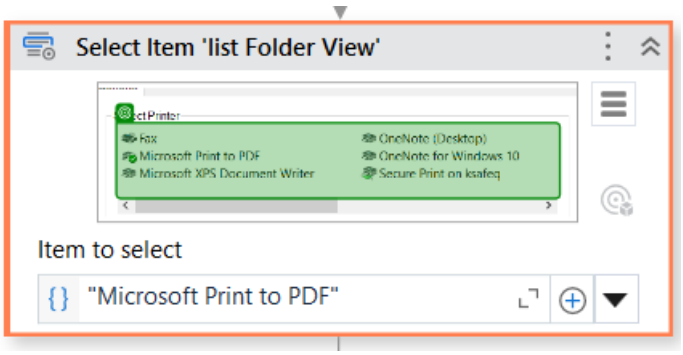

Figure 9: UiPath activity for selecting an item from a dropdown list.

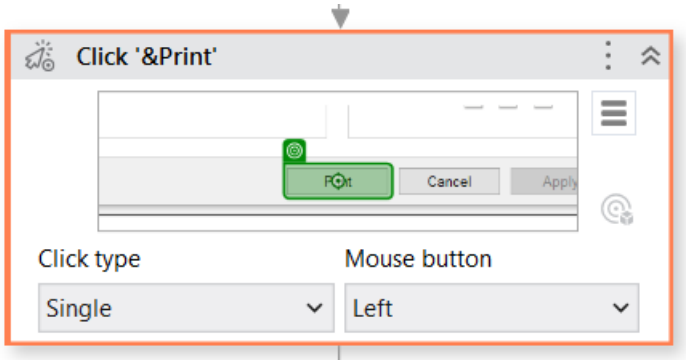

Figure 10: Click print button codeblock

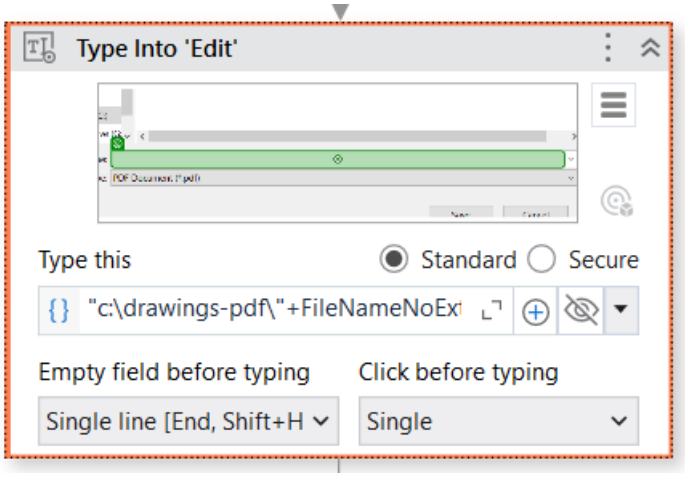

Figure 11: UiPath Type Into Activity for File Save Location

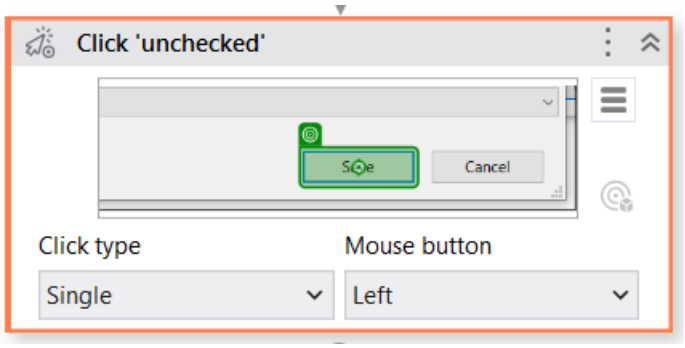

Figure 12: Codeblock for click save button

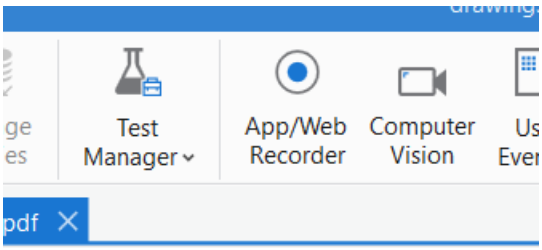

Figure 13: App/Web Recorder button in UiPath application