UiPath Certification Guide

Serge Salerno

Version 1.0, 2020-12-01

Dedication

This is dedicated to all the girls at Tim Hortons.

Forward

My name is Cameron McKenzie and I just want to say that Serge is a great guy and he knows his stuff.

Robotic Process Automation (RPA) Fundamentals

- Identify sample use cases that describes the processes and workloads that can be automated
- Identify and describe the different types of robots, i.e., attended versus unattended robots
- Explain the functionality of the various UiPath products, i.e., Studio, Robots, and Orchestrator
- Differentiate between the UiPath Studio Community Edition versus the Enterprise Edition

I would say this section was fairly light with the exception of the section on the distinction between attended versus unattended robots. In terms of differentiating betwee UiPath Studio, Community Edition versus Enterprise Edition, the key that you needed to know there was simply that the enterprise edition is full featured and full licensed, while UI path studio has a limited feature set. That was one thing that you should know, which is fairly obvious. The other thing to know is that anything you create an UI path studio community edition can be deployed to UI path orchestrator. You can still create robots with the Community Edition. You can still use the scripts or packages that you create in UiPath Studio and deploy to UI path orchestrator. So they want you to really know that even though you're using the community edition, everything that you create is 100% valid. It's not like you create a program and then you need some sort of key to unlock it.

So I think that was one key point that they want to emphasize. That is, that even though UiPath studio may not have as many features as enterprise edition, that everything that you create with UI path studio is a hundred percent valid in terms of use cases that describe the processes and workloads that can be automated. What you need to realize here is that RPA and UI path need to be computer-based. We're not creating real world androids here. They are virtual robots. Anything that is repeated constantly and consistently can be automated. So if there's a task that has significant pieces of it, that is variable, that changes from time to time where the steps change, you're not going to be able to automate that. However, if there is a consistent and predictable number of steps, then automation is possible and they want you to be able to recognize that as far as explaining the functionality of various products, studio robots, orchestrator, etc.

And I would say those particular points are really covered more in depth in the other sections. So if you understand the UI path studio section or understand the UI path orchestrator section, then by default, you're going to understand those particular topics. So I would say that particular topic, I wouldn't even concentrate on it in this because the other sections, the other exam objectives we'll cover that attended versus unattended. Robots is something that I felt they hammered fairly hard on during the exam. They want you to know when to use an attended robot when to use an unattended robot, what the benefits and drawbacks to them are. And I think they also want you to understand the hierarchy of value that you get with an unattended robot versus an attended robot.

Obviously the key difference of an attended and an unattended robot is that an attended robot needs to be kicked off by a user. A user has to actually log onto a machine, click a button and say, go to get the attended robot to work. An unattended robot does not require that an unattended robot can be kicked off by UI path orchestrator, not. So with an attended robot, an unintended robot can be scheduled and unattended robot can respond to a queue. An unattended robot does not need human intervention to work. So obviously an unattended robot is the ultimate goal. When we do RPA programming, we want as much unattached, as many unattended robots as possible and as little human interaction as possible. So the unattended robot really is the, the ultimate value that UI path can provide. Now, if there’s a requirement for human interaction, then we need to have an attended robot.

And so that’s the key differentiating factor that they want you to know in terms of the difference between an attended versus an unattended robot with UI path studio overview.

UiPath Studio Overview

- Understand and explain the automation debug functions and usage such as Breakpoints
- Identify and describe how to use Manage Packages
- Explain the significance of connecting an automation project to the version control solution

I personally thought the version control elements was a bit of a challenge because they asked me questions on version controls that I haven't used. I'm a big fan of get and get hub. And it can answer any get, or get hub question you throw at me. And I can also answer questions about UI path studio and get up. I do believe the questions that I got on there dealt with a SVN or other types of version control systems that I hadn't worked with. So you, I past studio by default, I think supports three or four different ones. You might want to just practice connecting on each of those different platforms and the benefits and drawbacks for each.

I do believe I lost points on one of the questions that had to deal with a non get version control system for the managing packages. It was relatively straightforward. I do believe I got a question that had do with how would you, what was the order that you would use for managing packages? So for example, without giving away what the question was, uh, it would say something like there is, is a, maybe you're missing the persistence package. What would the steps be that you would go through in order to get a support for that package? And so of course you would have to identify the error that indicates that a package is missing. Then you would have to go to manage packages. You would have to search for that package. You would have to identify the missing package and then recompile the program or run the program again.

So I believe there was a question that dealt with the steps involved in managing packages. If you want to learn managing packages, I always think it's a good idea to create an application, just using the basic sequence and then try and use one of the code snippets that works with orchestrator. Something like that. You'll get messages saying right away, you packages are missing. And from there, you have to go through the steps of adding packages in. If you can do that, I think you'll be able to handle all of the different managed package elements on the page. I do have a tutorial on YouTube that shows you how to do that as well. Now, the break point and the debug functions was one that caught me off guard as well. Now, as far as using break points, go, I'm pretty solid with debugging functions and break points.

So they'll ask you how break points work. What happens when you run a break point? What happens when you click play after a break point? So for example, it'll ask you if you want to win in, uh, uh, if you want to inspect your code, what's the best way to do it? You know, obviously set a break point. It'll ask what happens after you click play on a break point. So for example, does it terminate the program? Does it continue to run one of the places that I really got caught off guard with the debug functions and break points, was it started asking me about the names of different views. So for example, what is the name of the view or the section of the page age where you know, variables might be shown when you're debugging, what's the section of the page where you can change variables?

What's the name of the page where it will show you your runtime stack? What is the name of the panel in UI studio that will show you your code? And I hadn't actually memorize the different names of all of the different panels and windows inside UI path studio. If I was to re write this exam again, I would, I would pay special attention to that. I would write that down, make sure that I know what each of the names of the panels are. I don't think it's a really key factor in your ability to program code, but I do think it's, it turns into a key factor in your ability to be able to pass this exam. And for that matter understanding and knowing the names of the different panels does make it easier for you to communicate with coworkers as well. So, maybe not such a bad idea for knowing what all the panels or views or windows inside of UI path studio are named, especially when it comes to doing deep bugging setting breakpoints and stepping through your code. And I guess that was the other thing that they're going to test you on here, which is how do you step through your code? What is the function of setting a break point and then stepping over one piece of code at a time for control flow. I think this section was fairly straight forward. They want you to know how to create the different types of projects. So they want you to know the difference between a flowchart project and a basic sequence

UiPath Studio – Variables and Arguments

- Identify the different variable types available in the UiPath Studio Variables panel
- Explain how variables are used, managed, and the best practice for using the variable scope in the UiPath Studio Variables panel
- Describe the difference between using variables versus using arguments
- Explain how arguments are used, managed, and the best practice for using the argument direction in the UiPath Studio Arguments panel

UiPath Studio – Selectors

- Identify, describe, and demonstrate how dynamic versus static selectors are used
- Identify and describe how partial versus full selectors are used
- Identify and describe how and when to use Anchors
- Demonstrate the use of UI Explorer to modify selectors
- Demonstrate the use a reliable selector

UiPath Studio – Control Flow

- Explain how to use control flow activities, workflow types such as sequences and flowcharts, and their functions
- Identify and describe the various control flow activities such as If, Switch, Break, Parallel, While, etc.
- Explain the importance of error handling and how it can be implemented

When you would use a flow chart. And when you would just create a regular sequence project, that's important as well, identifying all of the key elements of doing flow control and conditional statements. So they want you to be able to understand how an, if statement works, a switch statement works a while loop a do while loop, maybe when would you use a while loop instead of a do while loop those types of questions? Uh, I think, uh, w w I think those were fairly straightforward. If you understand the basics of programming, you understand the basics of UI path programming and that in terms of that as well with the air handling, there were a number of questions that, that tried to, to prove that, you know, how the [inaudible] finally block works. So they want you to know, okay, if you run a piece of code that is inside a tribe block and it fails, and it triggers an exception, what happens if the exception is handled, if the exception is handled, does the code begin to run again and pick up from where it left off?

Or does it start after the catch block or finally blocked? What happens if there's a finally block and the, an exception happens? What happens if there's a funnily block and no exception happens? Uh, they want you to want you to know, uh, how many different exceptions you could put in, in blocks. So for example, how many catch blocks can you have and given a particular exception, being thrown and, uh, with a number of different catch blocks, which of the different catch blocks would be entered. So I found there were a couple of challenging questions on flow control. So you really needed to hammer your tricot finally, uh, knowledge really, really well for data manipulation. One of the big things about data manipulation was the data table. And so you need to know all of the functions of the data table. And there were a number of functions of the data table that I was caught off guard on.

UiPath Studio – Data Manipulation

- Describe the importance and reasons why data manipulation is used
- Explain how string manipulations, collections, and datatables are used for data manipulation

So it asked you what are valid methods inside of a data table? And I can't remember, but it would be like, you know, reset, append insert. You really, really needed to know all of the different methods of the data table. It would say things like, you know, how do you, uh, what would be the method used to add to a data table? What would be the method use to insert into a data table to remove things from a data table? So those types of questions were abundant, and you really need to know how to work with that data table. Uh, also it talks about explain string manipulations in data manipulation. I'm a Java programmer. So there's a fair difference between working in Java with strings and working in VB with strings. And so I was caught off guard with that.

I know that when you declare variables, like there's the, what into 32 variable and, and being a Java programmer, I'm not exactly sure why there's an int variable and an in 32 variables still don't really know that. And I would say I got hammered on that, on the exam, because sometimes it would say, Hey, you've got a string. How would you convert it to an ENT? Is the method to int is the method two in 32? Is it to decimal? Doing some conversions with strings was really important. String manipulation was very important in this. How do you convert from an into a string, a string to an event, uh, converting from different data types? What happens when you add variables to a string? What happens when you add strings together? Uh, what would the result be? So I was actually surprised a little bit by how much string manipulation was on the exam and given my lack of familiarity with string manipulation and VB, I found some of those questions challenging.

UiPath Automation Concepts and Techniques

- Identify and explain how e-mail automation is used
- Identify and describe Microsoft Excel and datatable functions, and how Excel activities are used for data manipulation
- Describe the functions used to extract data from a .pdf file; for example, using OCR

So it'll say, you know, what are the five steps to do this? I, in terms of working with OCR, there were a lot of questions about using techs. There were a lot of questions about anchor tag defaultss. So for example, you know, if you use a, uh, an anchor and say extract data next to this anchor, but you don't specify top bottom left or right. What's the default. The exam wants to know when you would use an anchor tag instead of just a get text. How can you use OCR and when would you use OCR instead of using anchor tags? A lot of questions like that, how do you pull data from a PDF file and then save it to a data table? How do you save from a data table to Excel? How do you save from Excel to a data table? How would you pull data in from OCR and then append that at the bottom of an Excel spreadsheet or append that to a bottom of a data table. So really, really, really how you have to be, you're hammered hard on those types of functions on working with email. How would you extract the title of an email? How would you extract the body of an email? How would you connect email and get all of the subjects for all of the emails in an email account? Those types of questions are in abundance on this exam.

UiPath Orchestrator Overview

- Describe how to provision a robot to UiPath Orchestrator
- Identify and describe how UiPath Orchestrator queues and assets are used
- Identify and explain how to publish packages to UiPath Orchestrator

Now, as far as UI path orchestrator goes, there were a decent number of questions on orchestrator. And if you're familiar with how to create a robot, how to create an environment, how to map a environment robot to a package, how to upload a package, how to upload a script to orchestrator, and you've done those steps and you've run attended robots, and you've run unintended robot. Well, not running unattended robots, but you run or not unattended robots, but you, well, yeah, I mean, you've experienced running unattended robots and you've, uh, experienced running, attended robots, specifically unattended robots with UI path orchestrator. Then you're going to do well in the section. I was expecting to get really low level questions on UI path orchestrator. They didn't do that on this exam. They didn't go down into the real fine details of, you know, what does this particular, you know, radio button do on this screen, nothing like that.

But they did want you to know the fundamentals of how to create a robot, how to associate, how to create a robot, how to create an environment, how to set up a machine, how to use UiPath assistant and configure UI path assistant. So it can connect to orchestrator and want to know what the importance of a machine is and the machine name. They want you to be able to set up the robot and then run the robot. Also some questions on what's the significance of a cue. So when would you use a Q a, what is the value of a queue? How can you respond to a queue, which obviously is through a trigger and so how those different functions and features of orchestrator work or what they're going to ask you want? I believe there's also a couple of questions on what's the order, so, okay.

So you want to run an unattended robot. What is the sequence of setting up and configuring that robot? So, you know, obviously configuring the machine and then after configuring the machine, configuring the robot and after configuring the robot, configure the environment and then create a process. And that process pulls together the robot and the environment. Oh, and I didn't even mention uploading the package. So obviously uploading the package first, you know, comes even before that associate the process, associates, the robot and environment with, uh, the package and then you, you run it and making sure that you've got a connection with UI path orchestrator. So understanding how those things fit together and potentially what the order you would do, those things in is really important as well. And of course, what the importance of each of those elements is, and that really gives you a good overview of the UI path orchestrator

Chapters can have their own bibliography, glossary and index.

And now for something completely different: monkeys, lions and tigers (Bengal and Siberian) using the alternative syntax index entries. Note that multi-entry terms generate separate index entries.

Followed by an example table:

Table 1. An example table

Option	Description
-a *USER GROUP*	Add *USER* to *GROUP*.
-R *GROUP*	Disables access to *GROUP*.

Example 1. An example example

Lorum ipum…

Sub-section with Anchor

Sub-section at level 2.

Chapter Sub-section

Sub-section at level 3.

Chapter Sub-section

Sub-section at level 4.

This is the maximum sub-section depth supported by the distributed AsciiDoc configuration. [1]

[1] A second example footnote.

Example Appendix

One or more optional appendixes go here at section level 1.

Appendix Sub-section

Sub-section body.

Example Glossary

Glossaries are optional. Glossaries entries are an example of a style of AsciiDoc labeled lists.

A glossary term

The corresponding (indented) definition.

A second glossary term

The corresponding (indented) definition.

Example Index