**RPA Interview Questions:**

**Introduce yourself?**

**🡪**

Hello, first of all, thank you for considering my application and giving me an opportunity to Introduce myself.

My name is Shubham Pawar, Basically I am from Chatrapati Sambhajinagar, Maharashtra, and now I am currently working in Accenture as an Application Development Analyst in Pune. Overall, I have 2.1 years of experience in IT. I have gained extensive experience in UiPath, as of now I have 1.5 years of experience in UiPath.

My experience with UiPath includes designing, developing, and implementing automation solutions to improve business efficiency. I am familiar with building and maintaining bots, orchestrator, RE Framework and how to use UiPath’s set of activities and features.

I have a strong ability to analyze complex requirements, break them down into manageable tasks, and deliver high-quality results within deadlines. I believe my skills and experience with UiPath, along with my IT knowledge make me well suited for any challenges.

Moving to my projects, as of now I’ve completed 2 automation projects using UiPath both are from banking domain. (1. Service now ticket🡪 creating, modifying, and deleting account. (**IAM Access Gateway**) 2. Gmail🡪 excel file for loan apply.)

I am delighted to have the opportunity to use my skills in UiPath and IT to contribute to your team. I look forward to sharing how I can add value to your organization.

**Explain your Project.**

**🡪**

1. **Banking Project** 🡪 (Using service now ticket tool)

In this project, we are receiving the tickets from users like to Create new account, modify the existing account, and delete the account permanently.

So, I have worked on modification and deletion part in this project.

In this project we used to get tickets in Service Now.

If we want to work in ServiceNow using UiPath then we must install some packages in workflow/project. Here we have installed **UiPathTeam.ServiceNow.activities** and **UiPath.ServiceNow.activities**

For any service now activity we need to use in **ServiceNow application scope** activity.

So, we are reading the tickets from service now tool using **Get Incident list**

activity. Here we need to write the one simple SQL command for new state(1) incidents.

**Command: SELECT number, description, short description, priority, state FROM Incident WHERE state=1;**

This data is stored in one **DataTable** variable. After that we processed this data.

From this data we are finding the account number for modify and delete request. That bank is using one application called **IAM Access Gateway.**

In this application we can get all the customer’s data.

So, using this application we are performing the creation modification and deletion as per user’s request.

After that we are updating those tickets.

**Difficulty faced in this project:** ServiceNow is the Dynamic tool/dashboard that’s why our most of the time is gone in selecting the stable selectors for this automation.

1. **Loan Process Automation:**

In this project We are getting Mails on Gmail, in that mail they are giving us one excel file in that we have the bank customer details is there for which is required for processing the loan.

And also, they gave us some conditions when we need to apply for the loan and when we don’t want to apply for loan.

After that we are processing on that data if customer is illegible for loan then we are applying for loan and giving them loan account number back in the same excel (we are storing details in one cell).

If customer is not illegible for loan, then we are storing the reasons in one cell in excel.

After that we send back same excel file to the client.

**Difficulty:** Before customer is not sending mails properly means they are not writing subject properly. So that’s why we are not able to identify which mail we have to read. So we gave them one template for mail subject like

**Loan Processing Data | Todays date.**

1. **Employee Access for Organization:**

In this project we are getting request through the outlook emails like Provide access to the newly joined employee, remove access from employee if he/she is rolled off from project and also Remove all access and delete profile from organization if the employee resigned from organization.

For Access purpose they are using **Employee Center in NetSuite.**

**Difficulty:** Before they are not sending mail in proper format, so we are facing issue to read the data, that’s why we gave them one proper mail Format/Template.

**1. What is RPA?**

Robotic process automation (RPA) is a software technology that makes it easy to build, deploy, and manage software robots that emulate human’s actions interacting with digital systems and software.

**2. How does RPA differ from traditional automation methods?**

Robotic Process Automation (RPA) is a form of automation that uses software robots to automate repetitive and rule-based tasks. Unlike traditional automation methods, RPA does not require any programming skills to automate tasks. RPA robots can be trained to perform tasks by simply demonstrating the steps to be performed on a computer interface, and the software then replicates those steps in an automated manner.

Traditional automation methods typically involve writing code to automate a process. This requires programming skills and expertise and can be time-consuming and expensive. In addition, traditional automation methods are often inflexible and can be difficult to modify if the process changes.

RPA, on the other hand, is designed to be flexible and easy to modify. RPA robots can be retrained quickly and easily to adapt to changes in the process or to handle new tasks. This makes RPA a more agile and adaptable solution than traditional automation methods.

Another key difference between RPA and traditional automation methods is that RPA robots can interact with applications in the same way that a human user would. This means that RPA can automate processes that involve multiple applications or systems, even if those applications were not designed to work together.

Overall, RPA offers a faster, more flexible, and more cost-effective approach to automation than traditional methods.

**3. What are the benefits of RPA?**

There are several benefits of Robotic Process Automation (RPA), including:

1. **Increased Efficiency:** RPA robots can work 24/7 and perform tasks at a faster speed than human workers, leading to increased efficiency and productivity.
2. **Cost Savings:** RPA can help businesses reduce costs by automating repetitive tasks that were previously performed by human workers. This can lead to significant cost savings in terms of salaries, benefits, and other expenses associated with hiring and managing employees.
3. **Accuracy and Consistency**: RPA robots are programmed to perform tasks with a high degree of accuracy and consistency, which can help reduce errors and improve the quality of work.
4. **Improved Compliance:** RPA can help ensure compliance with regulations and standards by automating tasks related to data entry, reporting, and auditing.
5. **Flexibility and Scalability:** RPA robots can be easily scaled up or down to accommodate changes in business demand, making it a flexible solution for businesses of all sizes.
6. **Improved Customer Service:** RPA can help improve customer service by automating tasks such as data entry and response to customer inquiries, leading to faster response times and improved customer satisfaction.
7. **Enhanced Analytics**: RPA can provide businesses with valuable insights into their processes by generating data and analytics that can be used to identify opportunities for improvement and optimize processes.

Overall, RPA can provide significant benefits to businesses by improving efficiency, reducing costs, and improving the quality of work.

**4. What is UiPath?**

UiPath is a leading Robotic Process Automation (RPA) software company that provides a platform for businesses to automate their repetitive and rule-based tasks. UiPath offers a range of tools and services to help businesses automate their processes, including a drag-and-drop visual designer, process recorder, and software robots called "UiPath Robots."

UiPath offers a wide range of pre-built activities and connectors, as well as integrations with third-party tools and technologies, making it a highly customizable and flexible solution for businesses of all sizes. UiPath also provides advanced features such as artificial intelligence (AI) and machine learning (ML) capabilities, enabling businesses to automate more complex tasks and processes.

In addition, UiPath provides training and certification programs to help businesses and individuals learn how to use their platform effectively. They also have a large community of developers and users who share best practices and provide support to one another.

Overall, UiPath is a powerful and popular RPA platform that enables businesses to streamline their operations, increase efficiency, and reduce costs through automation.

**5. What are the different components of UiPath?**

1. **UiPath Studio:** UiPath Studio is a visual designer tool used for building RPA workflows. It provides a drag-and-drop interface for building automation workflows, as well as a variety of pre-built activities and connectors for interacting with different applications and systems.
2. **UiPath Orchestrator:** UiPath Orchestrator is a web-based management console that provides centralized control and monitoring of UiPath robots. It allows businesses to manage and schedule robot activities, track performance and analytics, and automate processes across multiple environments.
3. **UiPath Robots:** UiPath Robots are software agents that perform the automation tasks defined in UiPath Studio workflows. Robots can be deployed on physical or virtual machines and can perform tasks on both attended and unattended modes.
4. **UiPath Activities:** UiPath Activities are pre-built components that enable robots to interact with different applications and systems, such as Excel, email, databases, and web applications. Activities can be combined to build complex automation workflows.
5. **UiPath Marketplace:** UiPath Marketplace is a platform for sharing and downloading reusable components, such as connectors, templates, and workflows. It allows developers to share their solutions with the UiPath community and helps businesses to accelerate their automation projects.
6. **UiPath AI Fabric:** UiPath AI Fabric is an add-on to the UiPath platform that provides advanced artificial intelligence (AI) and machine learning (ML) capabilities. It allows businesses to train, deploy, and manage AI models, and integrate them into their automation workflows.

Overall, these components work together to provide a comprehensive RPA platform that enables businesses to automate their processes and streamline their operations.

**6. Explain the architecture of UiPath.**

The architecture of UiPath is designed to provide a scalable and flexible Robotic Process Automation (RPA) platform that can be easily customized and integrated with other systems. It consists of the following components:

1. **UiPath Studio:** UiPath Studio is the development environment where RPA workflows are created using a drag-and-drop interface. The workflows are saved in a project file with the extension .xaml, which contains a sequence of activities that define the automation steps.
2. **UiPath Robot:** UiPath Robot is the software agent that executes the workflows created in UiPath Studio. The Robot can be deployed on a local machine or on a server and can run in attended or unattended mode.
3. **UiPath Orchestrator:** UiPath Orchestrator is a web-based application that provides centralized control and monitoring of the UiPath Robots. It allows users to manage and schedule robot activities, monitor their performance, and view analytics and reports. The Orchestrator also provides access control and security features to protect the RPA workflows.
4. **UiPath Activities:** UiPath Activities are pre-built components that can be used to interact with different applications and systems. They provide a range of functionalities such as file operations, email automation, data manipulation, and web automation. Activities can be customized and combined to create complex workflows.
5. **UiPath Libraries:** UiPath Libraries are reusable components that can be shared across multiple projects. They can be created in UiPath Studio and published to the UiPath Orchestrator, where they can be accessed by other users or projects.
6. **UiPath Marketplace:** UiPath Marketplace is a platform for sharing and downloading reusable components such as connectors, templates, and workflows. It provides a community-driven ecosystem where developers and users can share their solutions and collaborate on automation projects.

Overall, the architecture of UiPath is designed to provide a flexible and scalable RPA platform that can be easily customized and integrated with other systems. It enables businesses to automate their processes and streamline their operations, leading to increased efficiency and productivity.

**7. What are the activities available in UiPath?**

UiPath provides a wide range of pre-built activities that can be used to interact with different applications and systems. Some of the activities available in UiPath include:

1. **UI Automation:** These activities allow users to automate interactions with various graphical user interface (GUI) elements, such as clicking buttons, entering text, selecting options, and manipulating windows.
2. **System:** These activities provide functionality for working with files and folders, invoking command-line commands, manipulating system settings, and handling errors.
3. **Excel:** These activities allow users to automate interactions with Excel spreadsheets, including reading and writing data, formatting cells, and performing calculations.
4. **Database:** These activities provide functionality for connecting to databases, executing SQL queries, and retrieving or updating data.
5. **Email:** These activities allow users to automate interactions with email systems, including sending and receiving emails, and managing email folders.
6. **PDF:** These activities provide functionality for working with PDF documents, including reading, and extracting data, merging and splitting documents, and converting PDFs to other formats.
7. **Web Automation:** These activities allow users to automate interactions with web applications, including navigating to pages, filling out forms, and scraping data.
8. **AI:** These activities provide functionality for integrating artificial intelligence (AI) and machine learning (ML) capabilities into automation workflows, such as image recognition, text analysis, and language translation.

Overall, UiPath provides a comprehensive set of activities that enable users to automate a wide range of tasks and processes, leading to increased efficiency and productivity.

**8. Explain the difference between attended and unattended robots.**

Attended and unattended robots are two types of software agents used in Robotic Process Automation (RPA). The main difference between them is how they interact with human users.

Attended robots are designed to work alongside human users. They typically run on the same machine as the user and require human supervision to perform their tasks. Attended robots are usually used to automate tasks that require human input or decision-making, such as data entry, document processing, or customer service.

In an attended scenario, the robot may prompt the user for input or request authorization before performing certain tasks. Attended robots can also operate in a background mode, where they run silently in the background and do not interrupt the user's work.

On the other hand, unattended robots are designed to work autonomously, without human intervention. They typically run on a server or virtual machine and are scheduled to perform their tasks at specific times or intervals. Unattended robots are usually used to automate repetitive and rule-based tasks that do not require human input or decision-making, such as data extraction, report generation, or batch processing.

In an unattended scenario, the robot does not require human intervention and can run its tasks without any human interaction. Unattended robots can also monitor systems and processes in real-time and trigger alerts or notifications when certain conditions are met.

Overall, the main difference between attended and unattended robots is the level of human involvement required for their operation. While attended robots work alongside human users, unattended robots operate autonomously and can perform their tasks without any human intervention.

**9. How can you manage robots in UiPath?**

Robots in UiPath can be managed using UiPath Orchestrator, which is a web-based application that provides centralized control and monitoring of the robots. Some of the ways to manage robots in UiPath are:

1. **Adding Robots:** New robots can be added to the Orchestrator by providing the machine name and user credentials. The robots can be configured to run in attended or unattended mode and can be assigned to specific environments or processes.
2. **Monitoring Robots:** The Orchestrator provides real-time monitoring of the robots, including their status, performance metrics, and logs. Users can view the status of individual robots or groups of robots and track their activities over time.
3. **Scheduling Jobs:** Jobs can be scheduled to run on specific robots or groups of robots, at a specific time or interval. Users can configure the start time, recurrence, and priority of the jobs, and monitor their execution in real-time.
4. **Managing Packages:** UiPath packages contain the workflows, activities, and other resources needed for the robot to perform its tasks. The Orchestrator provides a centralized repository for managing the packages, including version control, publishing, and deployment to the robots.
5. **Assigning Roles and Permissions:** Users can be assigned different roles and permissions in the Orchestrator, depending on their level of access and responsibility. Roles include administrator, robot, and user, and can be customized to match the organization's security policies.
6. **Generating Reports:** The Orchestrator provides a range of reports and analytics on the robots' performance, utilization, and efficiency. Users can generate custom reports, view dashboards, and export data to other applications or formats.

Overall, UiPath Orchestrator provides a comprehensive set of tools and features for managing robots and automating processes. It enables organizations to optimize their automation workflows, improve their operational efficiency, and enhance their overall productivity.

**10. What is the workflow in UiPath?**

🡪  
 **Sequence:🡪** Simple linear representation of process.

* + - It will execute top to bottom one by one.

**Flowchart:🡪** It will more flexible for connecting activities.

* + - Here we can use multiple branching logical operator which enable us to create complex business process and connect activities with multiple ways.
    - It will have used in that process have multiple decision points.

**State Machine:🡪** It is the event driven process.

* + - It is based on finite number of states and transitions.
    - It is also called as finite state machine.
    - It is best suited for complex and continue process.
    - In state machine we can declare only one Initial state but we can add multiple Final states.

**Global Exception Handler:🡪** The Global Exception Handler is a type of workflow designed to determine the project’s behavior when encountering an execution error. Only one Global Exception Handler can be set per automation project.

-The Global Exception Handler is not available for library projects, only processes.

-The Global Exception Handler has two arguments, that should not be removed.

- The first argument is **errorInfo** with the **In** direction and it stores information about the error that was thrown and the workflow that failed. The level of the error to be logged can be set in the Log Message activity.

- The second argument, **result** has the **Out** direction and it is used for determining the next behavior of the process when it encounters an error. The following values can be assigned to the result argument:

- Any workflow may be flagged as a **Global Exception Handler** in Studio, except for library projects.

**11. What are the advantages of using UiPath?**

There are many advantages to using UiPath for Robotic Process Automation (RPA), including:

1. **User-Friendly:** UiPath has a user-friendly interface that enables users to create and manage automation workflows easily, even without extensive programming knowledge.
2. **High-speed Automation:** UiPath provides high-speed automation capabilities, enabling organizations to automate repetitive and time-consuming tasks quickly and efficiently.
3. **Scalability:** UiPath is a scalable platform that can handle a large number of robots and automate a wide range of processes, making it suitable for organizations of all sizes.
4. **Integration:** UiPath can integrate with a wide range of enterprise applications and systems, including SAP, Oracle, Salesforce, and more.
5. **Increased Productivity**: UiPath can automate a wide range of tasks, freeing up employees to focus on higher-value activities, which can increase overall productivity.
6. **Cost-Effective**: UiPath is a cost-effective solution for RPA, as it can reduce the need for manual labor and enable organizations to achieve cost savings in the long run.
7. **Accuracy:** UiPath robots perform tasks with a high degree of accuracy and consistency, reducing errors and improving overall quality.
8. **Security:** UiPath provides enterprise-grade security features to protect sensitive data and ensure compliance with data privacy regulations.
9. **Analytics:** UiPath provides analytics and reporting features that enable organizations to track the performance of their automation workflows and identify areas for improvement.

Overall, UiPath provides a comprehensive set of tools and features for RPA, making it an ideal choice for organizations looking to automate their business processes and improve their operational efficiency.

**12. What are the different types of variables available in UiPath?**

In UiPath, there are different types of variables that can be used to store and manipulate data during the execution of an automation workflow. The different types of variables available in UiPath include:

1. **String:** This variable type is used to store text data, such as names, addresses, and descriptions.
2. **Boolean:** This variable type is used to store logical values, such as true or false.
3. **Integer:** This variable type is used to store whole numbers, such as counts, quantities, and IDs.
4. **Double:** This variable type is used to store decimal numbers, such as currency values, percentages, and ratings.
5. **Object:** This variable type is used to store complex data structures, such as arrays, lists, and dictionaries.
6. **Generic:** This variable type is used to store any type of data, and it can be dynamically assigned based on the data being stored.
7. **Array:** This variable type is used to store a collection of related data items of the same data type.
8. **Dictionary:** This variable type is used to store a collection of key-value pairs, where each key is unique and associated with a specific value.
9. **DataTable:** It will store data like excel files.

Overall, UiPath provides a wide range of variable types to suit the different data storage and manipulation needs of automation workflows. The ability to work with different variable types allows for greater flexibility and customization of automation workflows.

**13. Explain the importance of selectors in UiPath.**

Selectors are a critical part of UiPath's automation capability. A selector is a string of characters that identifies a specific user interface (UI) element within an application, such as a button, text box, or menu option. Selectors are used by UiPath robots to interact with UI elements and automate tasks within an application.

The importance of selectors in UiPath can be summarized as follows:

1. Accuracy: Selectors ensure that UiPath robots interact with the correct UI elements within an application. This is critical for accurate automation, as it ensures that the robot performs the desired actions on the correct elements.
2. Resilience: Selectors are designed to be resilient to changes in the application's UI. Even if the UI element's location or properties change, UiPath's selector engine can still locate the element and interact with it.
3. Flexibility: Selectors are flexible, enabling UiPath robots to work with a wide range of applications and UI elements. UiPath supports a variety of selector types, including attribute-based, relative, and dynamic selectors, which can be used to target different UI elements.
4. Reusability: Selectors can be reused across multiple automation workflows, saving time and effort in designing and implementing automation processes.
5. Customization: UiPath provides selectors that can be customized to match specific automation requirements. This customization can include modifying the selector to target a specific attribute of a UI element or incorporating regular expressions to enhance selector accuracy.

Overall, selectors are a critical component of UiPath's automation capabilities, enabling robots to interact with the correct UI elements within an application, providing accuracy, resilience, flexibility, reusability, and customization.

**14. What are the different types of control flow available in UiPath?**

In UiPath, there are several types of control flow activities that can be used to control the sequence and flow of an automation workflow. These include:

1. Sequence: A sequence is a linear flow of activities that are executed in the order in which they appear in the workflow.
2. Flowchart: A flowchart is a graphical representation of a workflow that uses shapes to represent activities and arrows to indicate the flow between them. Flowcharts enable conditional branching and looping, allowing for more complex control flows.
3. If: The If activity is used to create conditional statements that execute specific activities based on a condition. If the condition is true, the activities within the If block are executed, and if it is false, the activities within the Else block are executed.
4. Switch: The Switch activity is used to create a set of conditional statements that execute specific activities based on the value of a variable. The Switch activity compares the variable against a set of possible values and executes the activities associated with the matching value.
5. While: The While activity is used to create a loop that executes a set of activities repeatedly while a condition is true. The loop continues until the condition is false.
6. For Each: The For Each activity is used to iterate over a collection of items and execute a set of activities for each item in the collection.
7. Do While: The Do While activity is used to create a loop that executes a set of activities repeatedly while a condition is true. The loop executes at least once, even if the condition is initially false.

Overall, these control flow activities provide a wide range of options for controlling the sequence and flow of an automation workflow, enabling UiPath developers to create complex and flexible workflows that can handle a wide range of automation tasks.

**15. What is a package in UiPath?**

In UiPath, a package is a bundle of reusable components that can be used to extend the functionality of UiPath Studio. Packages can include activities, libraries, templates, workflows, and other components that can be added to a UiPath project to simplify and accelerate the development of automation workflows.

Packages can be created by UiPath developers or downloaded from the UiPath marketplace, where a wide range of packages is available for free or for purchase. UiPath packages are distributed in the form of NuGet packages, a standard format for .NET libraries and components.

Once a package is added to a UiPath project, the activities, libraries, and other components within the package are available for use within the project. This enables UiPath developers to take advantage of pre-built components that have already been tested and optimized, reducing the time and effort required to develop automation workflows.

UiPath provides several built-in packages, including the UiPath.System.Activities package, which includes a range of core activities for working with files, strings, and other data types. Additionally, UiPath developers can create custom packages containing components tailored to their specific automation needs.

Overall, packages are an essential feature of UiPath's automation capabilities, providing a powerful mechanism for extending the functionality of UiPath Studio and accelerating the development of automation workflows.

**16. What is the role of orchestrator in UiPath?**

The Orchestrator is a web-based management console that provides a centralized platform for managing and monitoring UiPath robots, automation workflows, and assets. It is a critical component of UiPath's enterprise-grade automation capabilities, enabling organizations to scale and manage their automation efforts effectively.

The role of the Orchestrator in UiPath includes:

1. Robot management: The Orchestrator enables organizations to manage their UiPath robots from a central location. This includes configuring robot settings, monitoring robot performance, and managing robot licenses.
2. Workflow management: The Orchestrator provides a central location for managing UiPath automation workflows. This includes deploying workflows, scheduling workflows to run at specific times, and managing workflows across multiple environments.
3. Asset management: The Orchestrator enables organizations to manage their UiPath assets, including credentials, queues, and reusable components such as libraries and templates.
4. Monitoring and analytics: The Orchestrator provides real-time monitoring and analytics capabilities that enable organizations to track the performance of their automation workflows and identify potential issues.
5. Security and access control: The Orchestrator includes advanced security features, including access control, audit logging, and encryption, that help organizations protect their automation workflows and data.

Overall, the Orchestrator plays a critical role in enabling organizations to effectively manage and scale their automation efforts, providing a central location for managing UiPath robots, automation workflows, and assets.

**17. Explain the difference between UiPath and Blue Prism.**

UiPath and Blue Prism are two of the most popular robotic process automation (RPA) platforms, both designed to automate repetitive and manual tasks. While both platforms have similar functionalities, there are some key differences between UiPath and Blue Prism, including:

1. Ease of use: UiPath is generally considered to be more user-friendly and easier to learn than Blue Prism. UiPath has a visual, drag-and-drop interface that enables developers to create automation workflows quickly, whereas Blue Prism requires more technical expertise and has a steeper learning curve.
2. Licensing model: UiPath follows a per-user licensing model, whereas Blue Prism follows a per-robot licensing model. This means that organizations using UiPath pay for the number of users accessing the platform, while with Blue Prism, organizations pay for the number of robots they use.
3. Integration capabilities: Both UiPath and Blue Prism offer strong integration capabilities, but UiPath has more pre-built integrations with third-party applications than Blue Prism. This makes it easier for UiPath users to integrate their automation workflows with other enterprise applications and systems.
4. Customizability: Blue Prism is considered to be more customizable than UiPath. While UiPath provides a range of pre-built activities and components that can be used to build automation workflows quickly, Blue Prism allows developers to build more customized workflows from scratch.
5. Support and community: Both UiPath and Blue Prism have active user communities and offer support to their users, but UiPath has a larger community and more comprehensive support resources.

Overall, both UiPath and Blue Prism are powerful RPA platforms with similar functionalities, but there are some key differences in ease of use, licensing model, integration capabilities, customizability, and support resources that may influence an organization's decision when choosing between them.

**18. What is the role of UiExplorer in UiPath?**

UiExplorer is a powerful tool in UiPath that allows developers to inspect user interface elements in various applications and extract their attributes. The role of UiExplorer in UiPath includes:

1. Inspecting user interface elements: UiExplorer enables developers to inspect elements in various applications, including web, desktop, and Citrix applications. It allows users to view the properties and attributes of UI elements, such as buttons, text boxes, and dropdown lists, and to select the elements they want to interact with in their automation workflows.
2. Extracting attributes: UiExplorer allows users to extract the attributes of UI elements, such as name, ID, class, and value. These attributes can be used in automation workflows to interact with the elements, such as clicking a button or entering data into a text box.
3. Customizing selectors: UiExplorer allows users to customize the selectors of UI elements, which are used by UiPath to identify and interact with the elements. Users can modify the selector attributes to make them more specific or to handle dynamic elements that may change over time.
4. Testing automation workflows: UiExplorer can also be used to test automation workflows during development. By inspecting the UI elements and their attributes, developers can ensure that their automation workflows are properly interacting with the user interface.

Overall, the role of UiExplorer in UiPath is to enable developers to inspect and extract the attributes of user interface elements in various applications and to customize selectors to ensure that their automation workflows are accurate and reliable.

**19. What is the use of screen scraping in UiPath?**

Screen scraping is a technique used in UiPath to extract data from the user interface of an application. It involves using software tools to capture and interpret information that is displayed on the screen, and then extracting the relevant data for use in automation workflows. The use of screen scraping in UiPath has several benefits:

1. Extracting data from legacy systems: Screen scraping can be used to extract data from older systems that do not have APIs or other methods for data extraction.
2. Automating manual data entry: Screen scraping can be used to automate manual data entry tasks, such as copying data from one application and pasting it into another.
3. Handling unstructured data: Screen scraping can be used to extract data from unstructured sources, such as PDF documents, images, or web pages.
4. Enhancing data accuracy: Screen scraping can improve data accuracy by reducing the risk of human error that may occur during manual data entry.

UiPath provides several screen scraping techniques, including OCR (Optical Character Recognition), data scraping, and screen scraping with native Citrix support. These techniques allow developers to extract data from various types of applications, including web, desktop, and Citrix applications. Additionally, UiPath provides a range of pre-built activities and components that can be used to automate screen scraping tasks quickly and easily.

**20. Explain the difference between data scraping and screen scraping.**

Data scraping and screen scraping are two techniques used in UiPath to extract data from applications, but they differ in how they extract the data.

Screen scraping is a technique used to extract data from the user interface of an application by capturing and interpreting the visual elements displayed on the screen. It involves using software tools to analyze the pixel values, coordinates, and attributes of the user interface elements, such as buttons, text boxes, and dropdown lists, to extract the relevant data. Screen scraping is typically used for applications with graphical user interfaces (GUI), such as web and desktop applications.

Data scraping, on the other hand, is a technique used to extract structured data from web pages or other sources that contain structured information, such as tables or lists. It involves using software tools to locate the relevant HTML tags or XML elements and extract the data they contain. Data scraping is typically used for web scraping applications or other structured data sources, such as XML or JSON files.

**The main difference between data scraping and screen scraping is that data scraping is focused on extracting structured data from structured sources, while screen scraping is focused on extracting data from the user interface of an application, which may or may not be structured.** Data scraping is typically used for applications that involve web scraping, while screen scraping is used for GUI-based applications. Additionally, screen scraping is more complex than data scraping, as it involves analyzing the visual elements of the user interface, which may vary based on the application and its settings.

Table

Description automatically generated

**21. Difference between native, full text and OCR: (Output 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.

| Capability Method | Speed | Accuracy | Background Execution | Extract Text Position | Extract Hidden Text | Support for Citrix |
| --- | --- | --- | --- | --- | --- | --- |
| **FullText** | 10/10 | 100% | yes | no | yes | no |
| **Native** | 8/10 | 100% | no | yes | no | no |
| **OCR** | 3/10 | 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.**

To start extracting text from various sources, click the **Screen Scraping** button, in the **Wizards** group, on the **Design** ribbon tab.

The screen scraping wizard enables you to point at a UI element and extract text from it, using one of the three output methods described above. Studio automatically choses a screen scraping method for you, and displays it at the top of the **Screen Scraper Wizard** window.

**Note:** When we use scrapping methods then it will give us these three options for o/p.

In UiPath, "native", "full text", and "OCR" refer to different methods of extracting text from documents.

* Native extraction: This method involves accessing the document's internal structure to extract its text. Native extraction is the fastest and most accurate method for extracting text from digital documents such as PDFs, Microsoft Word documents, and web pages. Native extraction works best when the document is in a format that UiPath can natively read.
* Full text extraction: This method involves using OCR (Optical Character Recognition) to recognize and extract text from an image of a document. Full text extraction is useful when dealing with scanned images or documents with complex layouts, but it can be slower and less accurate than native extraction. Full text extraction is also more resource-intensive and can take longer to process.
* OCR extraction: This method is similar to full text extraction, but only extracts the text that the OCR engine recognizes. OCR extraction is useful when dealing with documents that are not in a format that UiPath can natively read, but it may require additional processing to obtain accurate results.

In summary, native extraction is the preferred method when working with digital documents, while full text extraction and OCR extraction are useful for extracting text from scanned or non-native documents.

**22. Output methods available in UiPath:**

1. Write Text File: This activity is used to write text to a file, and it can be used to create, update, or overwrite files. The text can be written to a specified path or to a file selected using a file picker.
2. Write CSV: This activity is used to write data to a CSV (Comma Separated Values) file. The data can be written to a specified path or to a file selected using a file picker.
3. Write Range: This activity is used to write data to an Excel worksheet. The data can be written to a specified range of cells, and the activity can be used to create new worksheets or overwrite existing ones.
4. Append Line: This activity is used to add a new line of text to an existing file. The text can be appended to a specified path or to a file selected using a file picker.
5. Message Box: This activity is used to display a message box on the screen, which can be used to display information or ask for user input.
6. Log Message: This activity is used to write a message to the output panel, which can be used for debugging purposes or to log information during the automation process.
7. Send Email: This activity is used to send an email with specified content and subject to one or more recipients.
8. HTTP Request: This activity is used to send HTTP requests to a web server and retrieve the response.

These are some of the output methods available in UiPath.

**23. Difference between Default, Simulate Type, and Simulate Click:**

**(Input Methods in UiPath)**

In UiPath, "Default", "Simulate Type", and "Simulate Click" are three different input methods that can be used to simulate user actions on a computer screen. Here are the differences between them:

1. **Default:** The default input method uses the standard Windows API to send keystrokes or mouse clicks to an application. It is the most reliable input method but can be slower than the other two methods. It is used when the other two methods cannot be used.
2. **Simulate Type:** This input method uses low-level keyboard hooks to simulate keystrokes. This method is faster than the default input method and can be used in situations where the default method may not work, **such as when an application has complex keyboard shortcuts or when there are conflicts with the keyboard layout.** However, the simulate type method may not work correctly with some applications, and there may be issues with international keyboard layouts.
3. **Simulate Click:** This input method uses low-level mouse hooks to simulate mouse clicks. This method is faster than the default input method and can be used in situations where the default method may not work, **such as when an application has complex mouse events or when there are issues with the mouse driver.** However, the simulate click method may not work correctly with some applications that use custom controls or have complex mouse interactions.

In summary, the default input method is the most reliable but can be slower, while the simulate type and simulate click methods are faster but may not work correctly with all applications. The appropriate input method to use depends on the specific requirements of the automation task and the application being automated.

**- Ans from UiPath**

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 **SendWindowMessages** method and then the **Default** one, as it is the slowest.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Capability Method | Compatibility | Background Execution | Speed | Hotkey Support | Auto Empty Field | Design experience |
| **Hardware Events** | 100% | no | 50% | yes | no | Classic/Modern |
| **SendWindowMessages** | 80% | yes | 50% | yes | no | Classic/Modern |
| **Simulate Type/Click** | 99% - web apps  60% - desktop apps | yes | 100% | no | yes | Classic/Modern |
| [ChromiumAPI](https://docs.uipath.com/studio/standalone/2023.10/user-guide/chromium-api) | 100% - Chrome, Edge browsers | yes | 50% | yes | yes | Modern |

**NOTE:** When the browser is started with **ChromiumAPI**, a ribbon shows up stating that the browser started in debug mode. This message does not show up if the extension is installed via [policy](https://docs.uipath.com/studio/standalone/2023.10/user-guide/extension-for-edge-chromium).

- The input method can be changed at any point from the **Properties** panel of the selected activity. If the **SimulateType** or **SendWindowMessages** check boxes are not selected, then the **Default** method is applied.

- 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.

**Important documents in SDLC (s/w development life cycle) in RPA UiPath.**

**KDD** - Knowledge Discovery Document:

**PDD:** Process Definition Document.

**SDD:** Solution Design Document.

**Difference between KDD, PDD and SDD in uipath automation:**

1. **KDD - Knowledge Discovery Document:** The Knowledge Discovery Document (KDD) is the initial phase in the automation development process. It involves understanding and analyzing the business process to be automated. During this phase, the automation team gathers information by interacting with subject matter experts (SMEs) and stakeholders to document the current manual workflow, pain points, exceptions, rules, and other relevant details.

The KDD is focused on capturing the "as-is" state of the process. It helps identify the tasks that can be automated, potential improvements, and the overall feasibility of automation. The KDD acts as a blueprint for the subsequent phases of the development process.

1. **PDD - Process Definition Document:** The Process Definition Document (PDD) is the second phase of the automation development process. It builds upon the KDD and defines the "to-be" state of the automated process. The PDD specifies the step-by-step workflow of the automated process, including inputs, outputs, decision points, and exception handling.

In the PDD, the automation team outlines the specific automation requirements, data inputs, expected outcomes, error handling strategies, and user interactions. The document also describes the expected behavior of the automation in various scenarios.

The PDD is reviewed and approved by stakeholders before proceeding to the next phase.

1. **SDD - Solution Design Document:** The Solution Design Document (SDD) is the final phase of the automation development process. It elaborates on the technical details of the automation solution. The SDD provides an in-depth description of the architecture, components, integration points, data flow, and security considerations of the automation project.

In the SDD, the automation team defines how they plan to implement the automation based on the information gathered in the KDD and PDD phases. It includes a breakdown of the automation into smaller components or modules, detailing how they will interact and function together.

The SDD serves as a reference for developers to build the automation solution according to the specified design.

In summary, KDD is about understanding the current process, PDD is about defining the automated process, and SDD is about designing the technical solution to achieve the automation goals. These three documents form a crucial foundation for successful and efficient automation development in UiPath.

Top of Form

**Wipro Questions from ChatGpt:**

1. **What is UiPath? Explain its key features**.

🡪UiPath is a leading Robotic Process Automation (RPA) tool that enables organizations to automate repetitive and rule-based tasks. Its key features include a visual designer interface, drag-and-drop functionality, support for a wide range of applications and technologies, robust automation capabilities, and extensive integration options.

1. **Can you explain the different types of workflows in UiPath?**

🡪Workflows in UiPath can be categorized into three types:

Sequence: It executes activities in a linear manner, one after another.

Flowchart: It provides branching and looping capabilities, allowing for more complex logic and decision-making.

State Machine: It enables the creation of complex workflows with multiple states and transitions.

1. **What are the core activities in UiPath?**

🡪Core activities in UiPath are pre-built automation actions that perform specific tasks. Some examples include:

Click: Simulates a click on an element in a target application or webpage.

Type Into: Sends keystrokes to a target application or field.

Read Range: Reads data from a specified range in an Excel spreadsheet.

If: Executes a set of activities based on a specified condition.

1. **How would you handle exceptions in UiPath?**

🡪Exception handling in UiPath involves using Try-Catch blocks to catch and handle errors that may occur during automation execution. By enclosing activities within a Try block, we can specify actions to be taken in case of exceptions using Catch blocks.

1. **How do you extract data from a web page using UiPath?**

🡪To extract data from a web page using UiPath, we can use the Data Scraping wizard. This tool allows you to select the data elements you want to extract and generates an output with structured data.

1. **What are selectors in UiPath? How do you use them?**

🡪 Selectors in UiPath are XML-based expressions used to identify UI elements within target applications or web pages. They consist of various attributes such as tag names, attribute values, and positions. Selectors help UiPath locate and interact with specific elements accurately.

1. **Can you explain the difference between a queue and a list in UiPath?**

🡪 In UiPath, a queue is a data structure used to store and manage work items. It follows the First-In-First-Out (FIFO) principle, where the oldest item is processed first. On the other hand, a list is a collection of items that can be accessed in any order, based on the requirements of the automation process.

1. **How do you automate email processing using UiPath?**

🡪 To automate email processing using UiPath, you can use the Outlook or Gmail activities. These activities allow you to perform actions such as sending emails, reading email messages, retrieving attachments, and managing folders.

1. **How do you handle data in Excel using UiPath?**

🡪 UiPath provides various activities to handle data in Excel. You can use activities like Read Range, Write Range, Add Data Row, and For Each Row to read, write, append, and iterate through Excel data.

1. **What is the purpose of the Get Full Text activity in UiPath?**

🡪 The Get Full Text activity in UiPath is used to extract the complete text from an application or window. It captures all the visible text within the specified region or scope.

1. **How do you handle dynamic selectors in UiPath?**

🡪 To handle dynamic selectors in UiPath, we can use wildcard characters such as asterisks (\*) or question marks (?) to replace dynamic or changing parts of the selector. This allows UiPath to match the dynamic element and interact with it reliably.

1. **What is the use of the Anchor Base activity in UiPath?**

🡪 The Anchor Base activity in UiPath is used to locate a target UI element relative to a fixed or stable anchor element. It helps in automating scenarios where the position of the target element may vary but is always relative to a known anchor element.

1. **Can you explain the concept of asset management in UiPath?**

🡪 Asset management in UiPath involves storing and managing credentials, settings, or other sensitive data securely. Assets are stored in the Orchestrator, allowing easy access and secure sharing across multiple automation projects.

1. **How do you handle data scraping in UiPath?**

🡪 Data scraping in UiPath refers to the process of extracting structured data from web pages or applications. It involves using the Data Scraping wizard or activities to identify and extract data elements based on their patterns and properties.

1. **What are arguments in UiPath? How do you use them?**

🡪 Arguments in UiPath are variables that allow you to pass data between workflows, sequences, or activities. They enable modularity and reusability by allowing you to customize the behavior of a workflow or activity based on input values.

1. **How would you automate the process of reading a PDF file using UiPath?**

🡪 To automate the process of reading a PDF file using UiPath, We can use activities like Read PDF Text or Read PDF with OCR. These activities extract the text content from PDF files, which can be further processed or manipulated as required.

1. **What are queues in UiPath? How do you use them?**

🡪 Queues in UiPath are used for managing and distributing work items in a queue-based manner. They ensure that the automation processes work on the right item at the right time and provide built-in features for prioritization, transaction handling, and reprocessing of failed items.

1. **Can you explain the difference between attended and unattended automation in UiPath?**

🡪 Attended automation in UiPath involves human interaction and collaboration. It assists humans in their tasks by automating certain steps, while allowing them to make decisions or perform manual actions when needed. Unattended automation, on the other hand, operates without human intervention and can run scheduled or triggered tasks autonomously.

1. **How would you handle pop-up windows in UiPath?**

🡪 Pop-up windows in UiPath can be handled using activities like Attach Window or Element Exists. These activities allow you to interact with or detect the presence of pop-up windows, enabling automation to handle them accordingly.

1. **Can you explain the concept of reusability in UiPath? How do you achieve it?**

**🡪** Reusability in UiPath refers to the ability to create reusable components or workflows that can be used across multiple automation projects. It promotes code efficiency, maintenance, and standardization. To achieve reusability, you can create libraries, workflows with input/output arguments, or custom activities that can be easily integrated into different automation processes.

**Note:**

1. Screen scrapping will help us to store the selected data from the screen.
2. From Scrapping only we have two options like **Data Scrapping** and **Table Extraction.**

3)Using table extraction we can store data in datatable variable as a table format.

4) Data Scrapping is used for extract the structured data from structured source

**OCR Engines:**

1. Google cloud vision OCR
2. Microsoft Azure vision OCR
3. Tesseract OCR
4. UiPath OCR

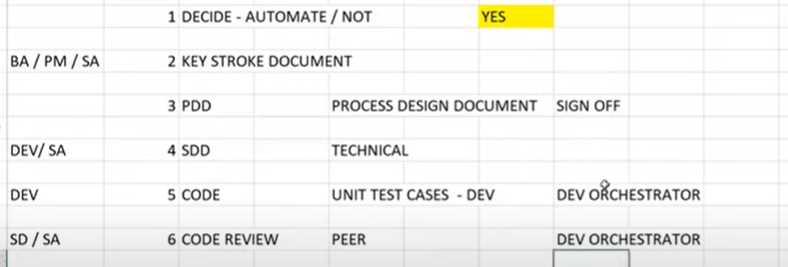
**Current Version of UiPath:** 2023.8

**3 Orchestrator is there for company.**

1. Dev (Developer) team
2. NFT Team (Nonfunctional testing)
3. Production

**KSD: Key Stroke Document.**

**RPA Lifecycle:**



A picture containing text, number, font, screenshot

Description automatically generated

**BA –** Business Analyst

**PM –** Project Manager

**SA –** Solution Architect

**Dev –** Developer

**SD –** Senior Developer

**TM –** Test Manager

**TL –** Test Lead

Wipro Questions:

Difference between classic and modern?

Default robot and floating robot?

Can we store multiple credentials in same asset?

UI Explorer

**Differences Between Modern Experience and Classic Experience**

**Recorders**

* Modern experience - The [App/Web](https://docs.uipath.com/activities/docs/app-web-recorder) and [Computer Vision](https://docs.uipath.com/activities/docs/computer-vision-recorder)recorders are available.
* Classic experience - The [Basic, Desktop, Web, Image, Native Citrix](https://docs.uipath.com/studio/standalone/2023.4/user-guide/about-recording-types), and [Computer Vision](https://docs.uipath.com/activities/docs/computer-vision-recorder) recorders are available.

### Scraping Wizards

* Modern experience - The **Table Extraction** wizard is available for data scraping.
* Classic experience - The **Data Scraping** and **Screen Scraping** wizards are available.

### Activities

A separate set of activities is available by default in each experience.

Regardless of the experience selected for a project, you can also enable the activities that are only available by default in the other experience. Click **View Options** at the top of the **Activities** panel, and then select:

* **Show Modern** to view modern activities in a project that uses the classic experience
* **Show Classic** to view classic activities in a project that uses the modern experience.

| Modern Activities | Classic Activities |
| --- | --- |
| [Check App State](https://docs.uipath.com/activities/docs/n-check-state) | [On Element Appear](https://docs.uipath.com/activities/docs/on-ui-element-appear), [On Element Vanish](https://docs.uipath.com/activities/docs/on-ui-element-vanish), [On Image Appear](https://docs.uipath.com/activities/docs/on-image-appear), [On Image Vanish](https://docs.uipath.com/activities/docs/on-image-vanish), [Wait Element Vanish](https://docs.uipath.com/activities/docs/wait-ui-element-vanish), [Wait Image Vanish](https://docs.uipath.com/activities/docs/wait-image-vanish), [Find Image](https://docs.uipath.com/activities/docs/wait-image-appear), [Image Exists](https://docs.uipath.com/activities/docs/image-found), [Element Exists](https://docs.uipath.com/activities/docs/ui-element-exists) |
| [Check/Uncheck](https://docs.uipath.com/activities/docs/n-check) | [Check](https://docs.uipath.com/activities/docs/check) |
| [Click](https://docs.uipath.com/activities/docs/n-click) | [Click](https://docs.uipath.com/activities/docs/click), [Click Image](https://docs.uipath.com/activities/docs/click-image) |
| [Extract Table Data](https://docs.uipath.com/activities/docs/n-extract-data) | [Extract Structured Data](https://docs.uipath.com/activities/docs/extract-data), [Get Full Text](https://docs.uipath.com/activities/docs/get-full-text), [Get Visible Text](https://docs.uipath.com/activities/docs/get-visible-text) |
| [Get Text](https://docs.uipath.com/activities/docs/n-get-text) | [Get Text](https://docs.uipath.com/activities/docs/get-value) |
| [Go to URL](https://docs.uipath.com/activities/docs/n-go-to-url) | [Navigate To](https://docs.uipath.com/activities/docs/navigate-to) |
| [Highlight](https://docs.uipath.com/activities/docs/n-highlight) | [Highlight](https://docs.uipath.com/activities/docs/highlight) |
| [Hover](https://docs.uipath.com/activities/docs/n-hover) | [Hover](https://docs.uipath.com/activities/docs/hover), [Hover Image](https://docs.uipath.com/activities/docs/hover-image) |
| [Keyboard Shortcuts](https://docs.uipath.com/activities/docs/n-keyboard-shortcuts) | [Send Hotkey](https://docs.uipath.com/activities/docs/send-hotkey) |
| [Navigate Browser](https://docs.uipath.com/activities/docs/n-navigate-browser) | [Close Tab](https://docs.uipath.com/activities/docs/close-tab), [Go Back](https://docs.uipath.com/activities/docs/go-back), [Go Forward](https://docs.uipath.com/activities/docs/go-forward), [Go Home](https://docs.uipath.com/activities/docs/go-home), [Refresh Browser](https://docs.uipath.com/activities/docs/refresh-browser) |
| [Select Item](https://docs.uipath.com/activities/docs/n-select-item) | [Select Item](https://docs.uipath.com/activities/docs/select-item) |
| [Take Screenshot](https://docs.uipath.com/activities/docs/n-take-screenshot) | [Take Screenshot](https://docs.uipath.com/activities/docs/take-screenshot) |
| [Type Into](https://docs.uipath.com/activities/docs/n-type-into) | [Type Into](https://docs.uipath.com/activities/docs/type-into), [Type Secure Text](https://docs.uipath.com/activities/docs/type-secure-text) |
| [Use Application/Browser](https://docs.uipath.com/activities/docs/n-application-card) | [Open Application](https://docs.uipath.com/activities/docs/open-application), [Open Browser](https://docs.uipath.com/activities/docs/open-browser), [Attach Window](https://docs.uipath.com/activities/docs/window-scope), [Attach Browser](https://docs.uipath.com/activities/docs/browser-scope), [Element Scope](https://docs.uipath.com/activities/docs/element-scope), [Close Window](https://docs.uipath.com/activities/docs/close-window)  [Start Process](https://docs.uipath.com/activities/docs/start-process) |
|  | [Anchor Base](https://docs.uipath.com/activities/docs/anchor-base), [Context Aware Anchor](https://docs.uipath.com/activities/docs/anchor-context-aware) |

**Which programming standards your organization is using when developing the project?**

🡪

"Our organization follows the UiPath best practices and guidelines for our projects. We adhere to the recommended programming standards and design principles set forth by UiPath. This ensures that our workflows are maintainable, scalable, and consistent throughout the development process. Additionally, we regularly review and update our practices based on the latest updates and recommendations from UiPath to ensure we stay up-to-date with industry standards."

We are analyzing our code using code analyzer in uipath.

**Log levels in Log Message activity:**

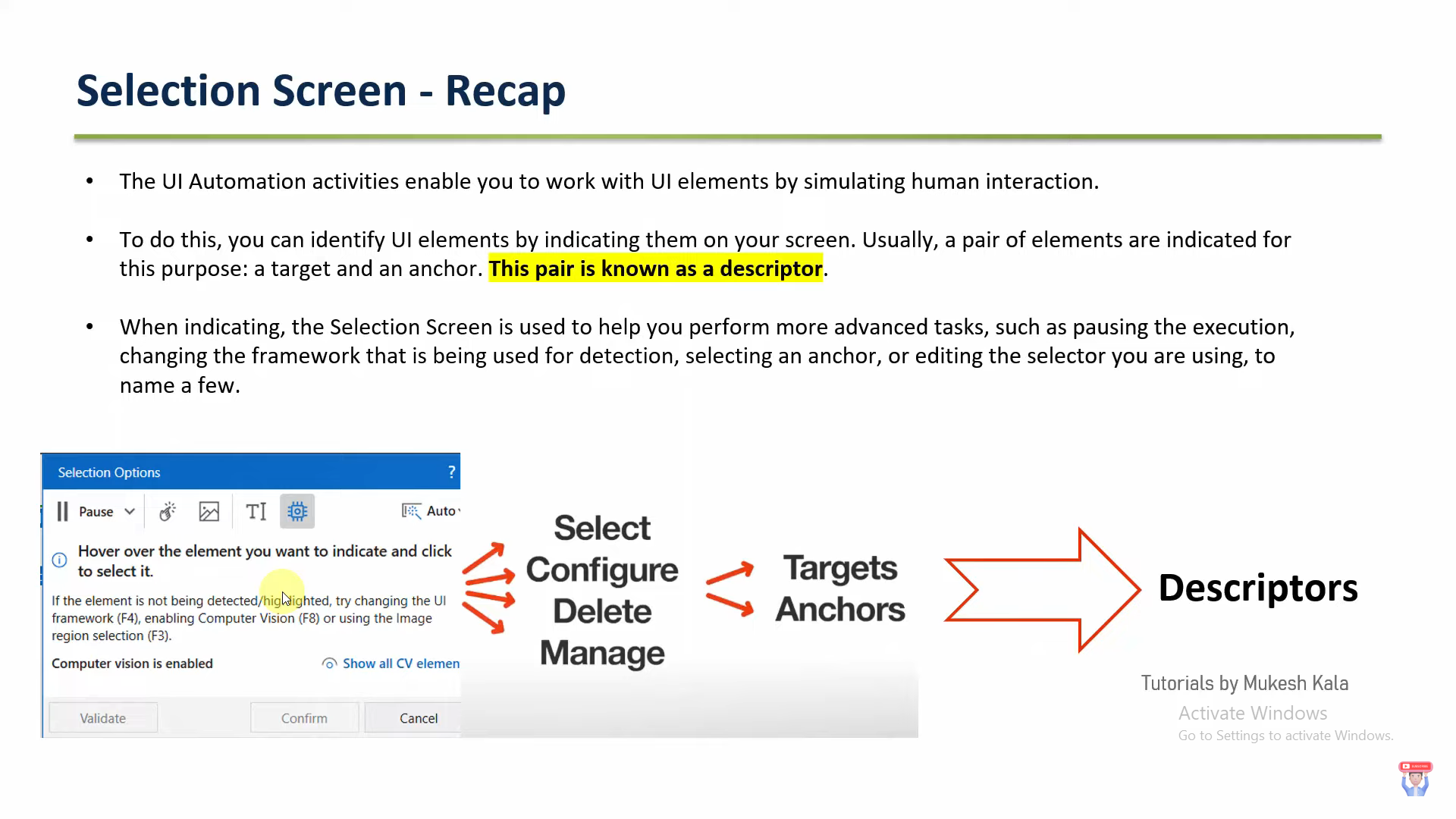
* 1. Fatal
  2. Error
  3. Warn
  4. Info
  5. trace

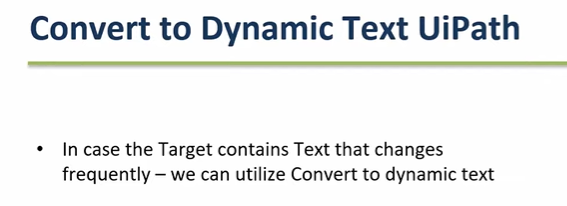
Note:

1. DataTable variable index starts from 0
2. Excel index starts from 1.

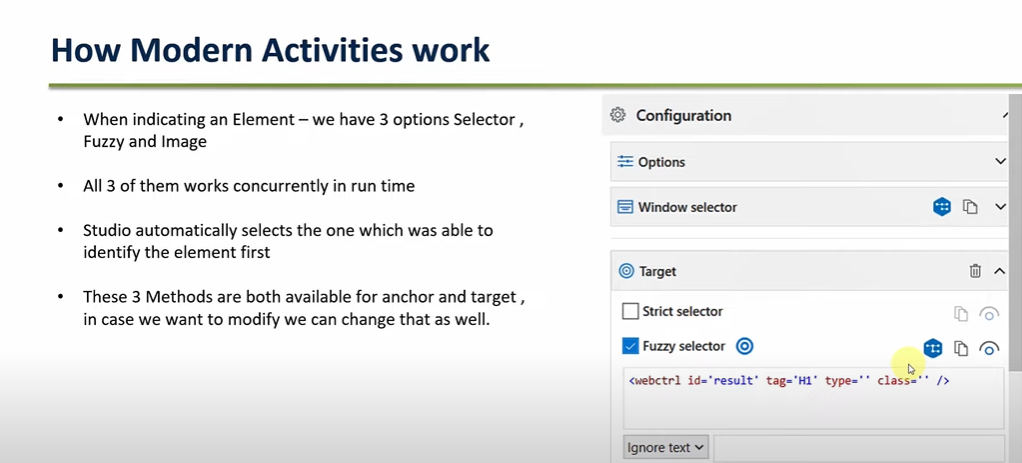
# **Why UiPath modern design experience is better ? Difference from Classic**

**🡪**

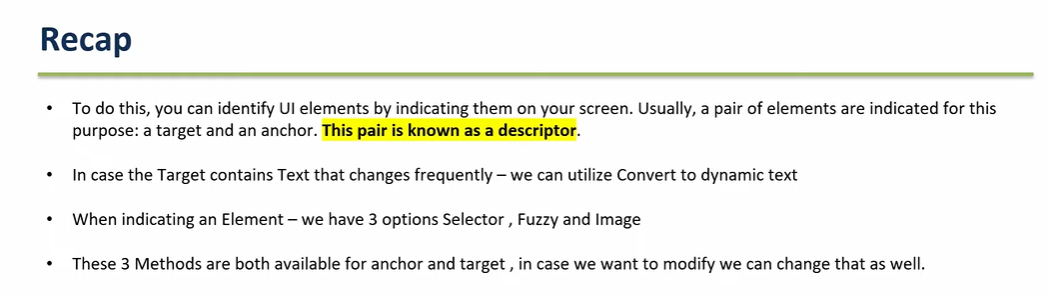




* 1. We have one option in Modern Design when we selecting the element in UI called **Convert to Dynamic Text.**
  2. It will help to identify the dynamic element. Like dynamic selector.
  3. In classic when we are selecting the element in UI then it will auto select the Anchor and target (descriptor).



* 1. Modern design will generate all the selectors for one single element and choose the best one from them.
  2. We can add Max 3 anchors for one target element.



## Automation Lifecycle: Steps

## Requirement gathering

## Process Understanding.

## Documenting the Process

## Development and code review.

## Testing

## Release.

## Process Understanding

Deciding between an automation for Attended Robots or Unattended Robots is the first important decision that impacts how developers build the code, as the general running framework (Robot triggering, interaction, exception handling) is different. Switching to the other type of Robots later may be cumbersome.

For time-critical, live, humanly triggered processes, like in a call center an Attended Robot working side by side with a human might be the only possible answer.

But not all processes that need human input are supposed to run with Attended Robots. If a purely judgmental decision (not rule-based) during the process could not be avoided, evaluate if a change of flow is possible, like splitting the bigger process into two smaller sub-processes, when the output of the first sub-process becomes the input for the second one. Although human intervention takes place in between, such as validating/modifying the output of the first sub-process, both sub-processes could be triggered automatically and run unattended.

A typical case would be a process that requires a manual step somewhere during the process, such as checking the unstructured comments section of a ticket and, based on that, assign the ticket to certain categories.

Generally speaking, going with an Unattended Robot ensures a more efficient usage of the Robot load and a higher ROI, a better management and tracking of robotic capacities.

But these calculations should take into consideration various aspects, such as the fact that an Attended Robot could usually only run in the normal working hours of a person, or it may keep the machine and user busy until the execution is finished. Input types, transaction volumes, time restrictions, the number of Robots available, and others also play a role in this decision.

## Documenting the Process - DSD

The process documentation guides the developer's work and provides help in tracking the requests and the application maintenance. Of course, there might be lots of other technical documents, but one is critical for a smooth implementation, namely **DSD** (Development Specification Document).

The Development Specification Document should contain the automated process details and focus on two main categories: **Runtime Guide** and **Development Details**.

The **Runtime Guide** should contain a high-level runtime diagram, as well as details about the functionality of the Robot, such as sub-processes, schedules, configuration settings, input files, output files, temporary files, and performed actions. Additional details about the master process should be specified, such as prerequisites, automatic and manual error handling, process resuming in case of failure, Orchestrator usage, logging and reporting, credential management, and any other relevant information related to security or function.

The **Development Details** should contain information about the packages in use, the development environment, the logging level, the source code repository and versioning, a list of workflow components with their description and argument list, a list of reusable components, the workflow invoke tree, defined custom logs and log fields, relevant snapshots of the process flowchart, the level of background versus foreground automation, and any other relevant or outstanding development items.

## Development and Code Review

The RPA Solution Architect is responsible for continuously coaching developers on the best practices. Hence, frequent and thorough code reviews are a must, to enforce a very high quality of the developed workflows. This way, the developers are motivated to build robust workflows and to follow the best practices guide.

## Test

After each component is built, unit testing should be conducted. If every component is thoroughly tested, the integration runs more smoothly, and debugging lasts for a shorter period of time. The **REFramework** contains a **Test\_Framework** folder where all the test files should be placed. Using the RunAllTests.xaml file, a developer can test a sequence containing a lot of .xaml files automatically, thus being able to try out small integrations between components and to run stress tests. A report is generated at the end of each test. Typically, these kinds of tests should be executed outside of office hours, in testing environments, to optimize the developer’s time.

The recommended UiPath architecture includes **Dev** and **Test** environments that allow the processes to be tested outside the live production systems.

Sometimes applications look or behave differently between the **Dev**, **Test**, or **Production** environments and extra measures must be taken, sanitizing selectors or even conditional execution of some activities.

Use the UiPath.config file or Orchestrator assets to switch flags or settings for the current environment. A test mode parameter (Boolean) could be checked before interacting with live applications. This could be received as an asset (or argument) input. When it is set to **True**, during debug and integration testing, it follows the test route, does not execute the case fully. For example, the test patch can skip sending notifications, skips the **OK** or **Save** button or press the **Cancel** or **Close** button instead. When set to **False**, the normal **Production** mode route is followed.

This allows you to make modifications and test them in processes that work directly in live systems.

## Release

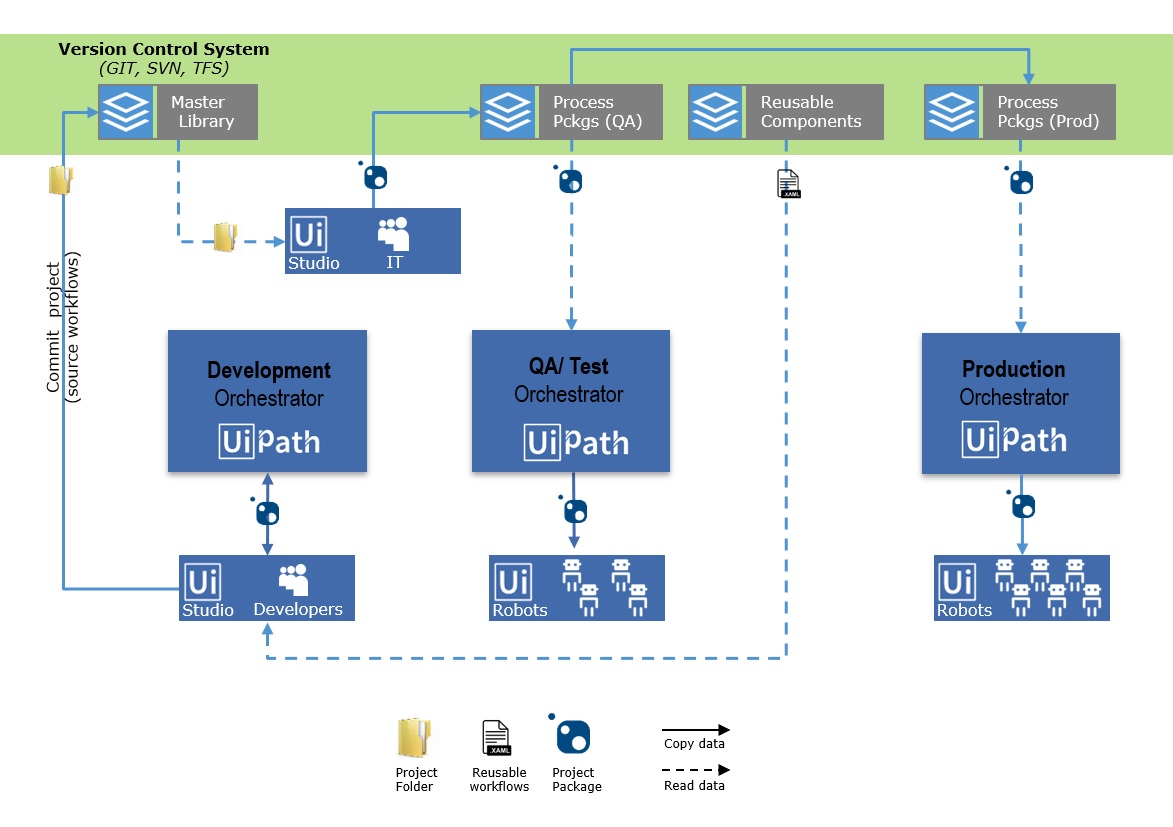
There are various ways of designing the architecture and release flow, considering the infrastructure setup, concerns about the segregation of roles, etc.

In this proposed model UiPath developers can build their projects and test them on **Development** environments in Orchestrator. They are allowed to check in the project to a drive managed by a version control system (VCS), such as GIT, SVN, or TFS.

Publishing the package and making it available for **Test** and **Production** environments is the work of a different team, such as IT.

The deployment paths on Orchestrator have been changed from their default to folders managed by the **VCS**, by changing the **Storage.Location** value in the UiPath.Orchestrator.dll.config file in the **Deployment** section..

The model also contains a repository of reusable components.



Here is the project publishing flow, step by step:

1. Developers build the process, test & debug pieces of it locally (Studio).
2. Once the automation development is completed, the process is published to the **Development** Orchestrator and tested again end-to-end.
3. The project folder is committed (not packaged) to a **Master Library** folder (on **VCS**).
4. The IT/ RPA Operations team creates the project package for QA. This step is intended to be an additional safety measure: the automation source code is inspected (by a different entity) before being packaged and run by robots.For example, the packaged process is stored in the **Process Pckgs (QA)** folder on **VCS**, from which it is deployed to the QA robots and executed.
5. If any issue is revealed during the tests, the steps above are repeated.
6. Once all QA tests are passed, the package is pushed to a **Production** environment - **Process Pckgs (Prod)**.
7. When the Process goes live, the process package is deployed to the production Robots and executed.

**Reusable Content** is created and deployed separately, as UiPath code (**Reusable Code Library**) and **Invokes** (**Invokes Repository**).

Workflows with source code are .xaml files containing activities for automating common processes, such as Log in to SAP:

**I have one data table variable having some duplicate value. I want to just take unique value using linkq. in uipath give me query**

**🡪**

Assign activity:

**uniqueValuesDataTable = YourDataTableVariable.DefaultView.ToTable(True, "ColumnName")**

Replace **YourDataTableVariable** with the actual name of your DataTable variable and **"ColumnName"** with the name of the column from which you want to extract unique values.

Here's a breakdown of what the LINQ query does:

1. **YourDataTableVariable.DefaultView**: This gets a DataView from your DataTable, which provides a customizable view of the data.
2. **.ToTable(True, "ColumnName")**: This method of the DataView allows you to create a new DataTable containing only unique rows. The first parameter (**True**) specifies that only distinct rows should be included. The second parameter (**"ColumnName"**) specifies the name of the column from which you want to extract unique values.

**How to add UiPath Project to GitHub?**

**🡪**

There are two ways available in UiPath.

1. Add to source control button in status bar in UiPath
2. Using Team tab

In both times we have to Init the repository.

* + - We need to give Github repository link.
    - We need to give private key path and password.

**UiPath Assistant:**

The UiPath Assistant is a tool created specifically to turn the user's interaction with our robots into a great and enjoyable experience from the comfort of their desktops. It's the place where individuals can easily access, manage and run automations with just a couple of clicks. The interface can be customized to better suit the person behind it by choosing an avatar and a name for the robot, organizing processes in custom folders on the launchpad, or maybe choosing another theme. All of this makes the UiPath Assistant our bridge between humans and Robots.

The first time Assistant is started, you are presented with an interactive guided tour which takes you through its main features. This can always be restarted from the **Help** section in the **Preferences** menu.

**What is the difference between UiPath robot and UiPath assistant?**

**🡪**

The UiPath Robot is the underlying service that connects to Orchestrator and does the actual processing. The UiPath Assistant is just the visual application that allows you to control the Robot.

what is the difference between compatibility windows, cross-platform and windows legacy in UIpath?

🡪

1. **Compatibility Windows:**
   * Compatibility Windows in UiPath refer to the automation of Windows-based applications.
   * This mode is designed for automating traditional **Windows desktop applications** that run on the Windows operating system.
   * It is typically used for applications that have a graphical user interface (GUI) and can be interacted with using mouse clicks, keyboard input, and screen scraping.
2. **Cross-Platform:**
   * Cross-platform automation in UiPath refers to the ability to automate applications and processes that run on multiple platforms, not just Windows.
   * UiPath offers cross-platform capabilities to automate web applications, virtual desktops, and other technologies that may run on Windows, macOS, or Linux.
   * It allows for automating a wider range of applications and systems beyond just traditional Windows desktop applications.
3. **Windows Legacy:**
   * Windows Legacy automation in UiPath is a more traditional approach to **automating Windows applications**.
   * It is used for applications that may not be fully compatible with the latest UiPath automation features.
   * Windows Legacy automation methods might include using older technologies like the Windows Workflow Foundation (WWF) or more basic automation techniques.

what is the difference between add queue item and add transaction item activity in Uipath?

🡪

1. **Add Queue Item Activity:**
   * The "Add Queue Item" activity is used to add individual items to a queue in Orchestrator.
   * It allows you to define and specify the data that you want to add to the queue. This data can be any information you need to process later, such as work items, tasks, or data related to a process.
   * The data is typically added one item at a time, and each item can have its own set of custom properties to store additional information.
   * This activity is often used when you want to add data to a queue as part of a process, and you have control over what specific data should be added.
2. **Add Transaction Item Activity:**
   * The "Add Transaction Item" activity is used specifically in the context of REFramework, which is a widely used framework in UiPath for building robust, stateful automation processes.
   * It's used to add a transaction item to the Orchestrator queue but is typically used within the REFramework's predefined workflows like the "Get Transaction Data" state.
   * The purpose of this activity is to create a structured transaction data object that contains all the necessary information related to the transaction, including its status, priority, and specific data relevant to the process. It's often used to create a transaction before adding it to the queue.
   * The REFramework provides a structured way of processing transactions, and this activity fits within that framework.

In summary, while both activities are used for interacting with queues in UiPath Orchestrator, "Add Queue Item" is a more general activity for adding data to a queue, while "Add Transaction Item" is specifically designed for use within the REFramework and is used to create and add structured transactions to the queue as part of a stateful, transaction-based process. The choice between these activities depends on your specific automation requirements and the framework you are using.

**what is the difference between simple OCR and Cloud OCR in UiPath?**

**🡪**

1. **Simple OCR:**
   * Simple OCR is an on-premises OCR engine that is installed and runs locally on the machine where UiPath is installed.
   * It is used for basic OCR tasks and is suitable for small-scale automation projects where OCR accuracy requirements are not very high.
   * Simple OCR may not be as accurate as cloud-based OCR engines, and its performance can vary based on factors like image quality and text complexity.
2. **Cloud OCR:**
   * Cloud OCR, also known as Cloud-Based OCR, utilizes OCR engines hosted in the cloud.
   * It offers more advanced OCR capabilities and typically provides higher accuracy when extracting text from images or scanned documents.
   * Cloud OCR is suitable for more complex automation tasks, especially those involving large-scale document processing or where high accuracy is required.
   * UiPath offers integrations with various cloud-based OCR providers, such as Microsoft Azure OCR, Google Cloud Vision OCR, and Abbyy OCR, allowing you to choose the one that best suits your needs.

what is the use of schema when creating the queue in uipath?

🡪

When creating a queue in UiPath Orchestrator, defining a schema is an important step in structuring the data that will be stored and processed within that queue. The schema essentially defines the structure of the data that can be added to the queue, and it serves several important purposes:

1. **Data Structure Definition:** The schema allows you to specify the types of data fields that will be associated with each queue item. These fields define what information can be stored for each item in the queue. For example, you might define fields like "InvoiceNumber," "CustomerName," "DueDate," etc., depending on the nature of the data you are processing.
2. **Data Validation:** By defining a schema, you can enforce data validation rules for the information added to the queue. You can specify whether certain fields are required, set data type constraints, and establish field-specific validation rules. This ensures that only valid and properly formatted data is added to the queue.
3. **Data Consistency:** A well-defined schema helps maintain data consistency across all queue items. It ensures that each item in the queue adheres to a consistent data structure, making it easier to process and analyze the data downstream.
4. **Ease of Integration:** When working with queues in UiPath, you often use them as part of larger automation processes. Having a defined schema makes it easier to integrate the queue with other parts of your automation workflow. Robots can expect a consistent data format when processing items from the queue, which simplifies automation design.
5. **Reporting and Analysis:** With a structured schema in place, it becomes easier to generate reports and perform data analysis within Orchestrator. You can filter and aggregate data based on the defined fields, which can be valuable for tracking and monitoring the progress of your automation processes.
6. **Data Retrieval and Processing:** When you retrieve items from the queue for processing, the schema helps ensure that you can access and work with specific fields of data easily. This is crucial for automations that rely on accurate data extraction and manipulation.

In summary, defining a schema when creating a queue in UiPath Orchestrator is essential for organizing and managing the data you intend to process. It provides structure, validation, and consistency to the data stored in the queue, which is important for successful automation and reporting. The schema you define should align with the data requirements of your automation process and help achieve your automation objectives effectively.

**how much maximum timeout in uipath studio?**

🡪

he maximum timeout is **999,999 seconds** while the **minimum one is 1 second**. Open Project Behavior - Select what files to open by default when reopening a project. Pick from Keep Last Opened Tabs, Open Main Entry Point or No Tabs.

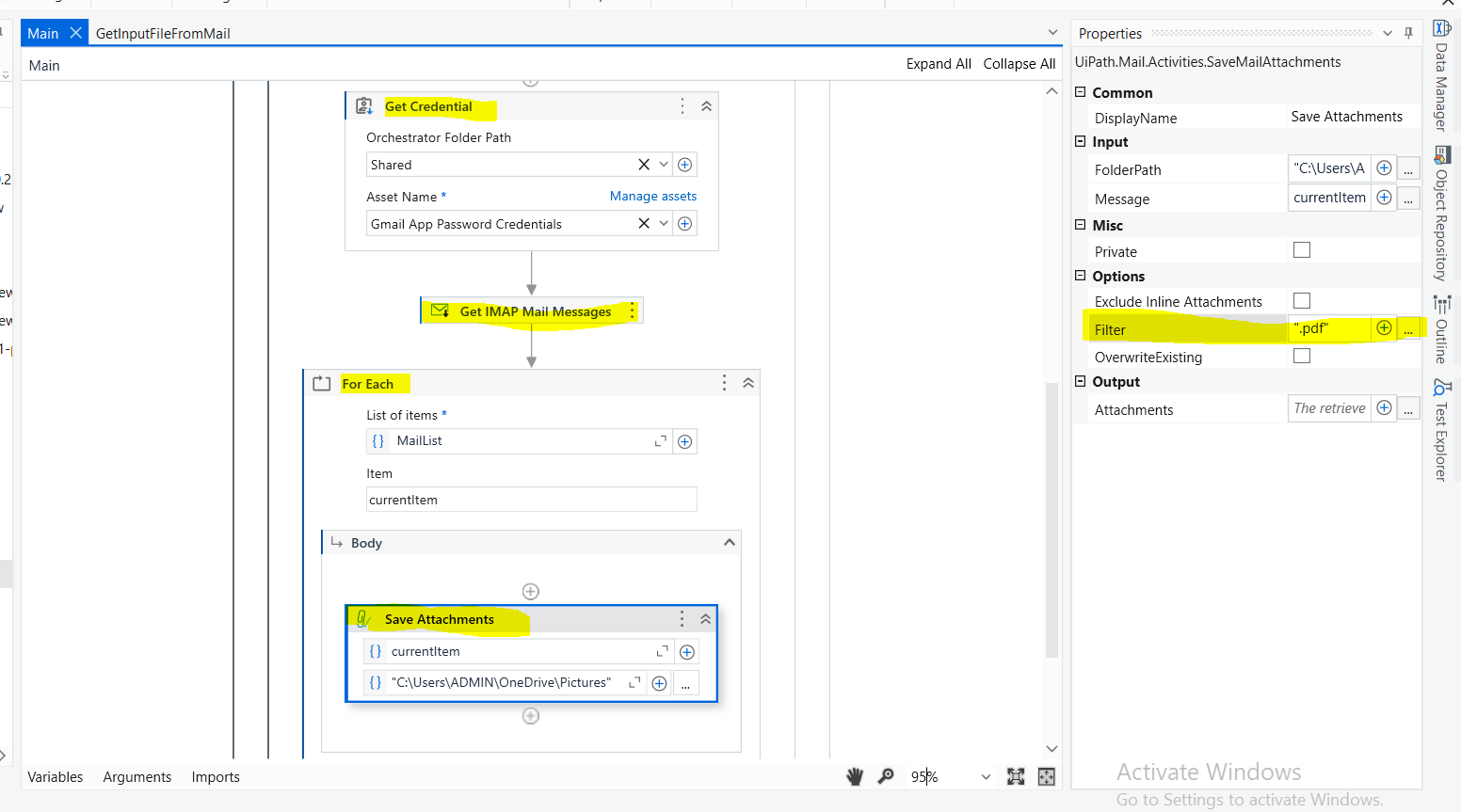
Top of Form

**Note:** The default value is 30000 milliseconds (30 seconds).

1 Second == 100 Millisecond.

**How do Read Mails from Gmail and how to save only .pdf file in local machine?**

🡪



In filter property give value **“.pdf”**.

Fill all the Properties value in Get IMAP mail messages. Like

1. Mail folder 🡪 From which folder you want to read the mails
2. Port 🡪 IMAP server port. (993)
3. Server 🡪 "imap.gmail.com"
4. Email 🡪 Which mail ids mail you want to read.
5. Secure password 🡪 we have to create this password if you want to use your mail id for automation.
6. Mark as Read 🡪 select the check box. It will mark as read mail which is read from automation
7. Top 🡪 From top how many mail you want to read.

**if we gave MaxRetryNumber 2 in config file and from orchestrator we gave MaxRetryNumber is 3 then what automations will do? in RE framework uipath?**

🡪

In UiPath's REFramework, which is a framework for building robust and scalable automation projects, the **MaxRetryNumber** is typically used for setting the maximum number of retries for a specific transaction or a process.

In the event of a conflict between the configuration file and the Orchestrator queue item regarding the **MaxRetryNumber**, REFramework will typically prioritize the configuration defined in the Orchestrator queue item over the one in the config file.

Here's how it generally works:

1. **Config File:** If you have set **MaxRetryNumber** to 2 in the config file, this value serves as a default for all transactions or processes unless overridden.
2. **Orchestrator Queue Item:** If you have set **MaxRetryNumber** to 3 when queuing a specific transaction in Orchestrator, this will override the default value from the config file for that specific transaction.

In UiPath's REFramework, the Orchestrator queue item configuration typically takes precedence over the configuration file settings. So, in your example:

* If you have **MaxRetryNumber** set to 2 in the config file but specify **MaxRetryNumber** as 3 when queuing a transaction in Orchestrator, the automation will use 3 as the maximum number of retries for that specific transaction.

This allows you to have flexibility in setting different retry counts for different transactions or processes without needing to modify the config file for each one. It's a way to customize the behavior of your automation at the transaction level while maintaining a default configuration in your config file.

Top of Form