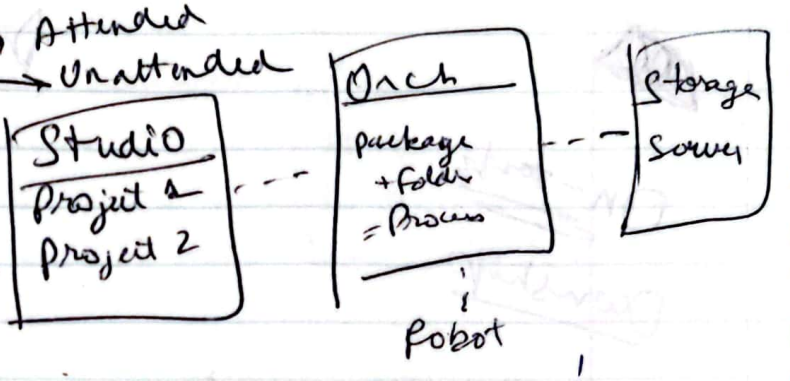
RPA Developer Foundation

RPA components of UI Path Platform

- Studio
- Orchestrator
- Robot → Attended / Unattended
- Assistant



UI Path Studio

Tool for building automation processes.

UI Path Robot

Executes the process built in UI Path Studio published to Orchestrator or locally.

UI Path Orchestrator

A Web application that acts as the interface b/w Studio and Robot. and manages, controls, and monitors automation tasks.

The automation projects can be published as NuGet packages either locally or in Orchestrator.

NuGET Packages

To run automation projects, published packages need to be associated with folders

Orchestrator

Packages + Folders
= Process

A process can be accessed by Robots

Human can access Robot with UIPath Assistant Tool to execute a Process on a machine

↓

Attended Robotes

No Human interaction

↓

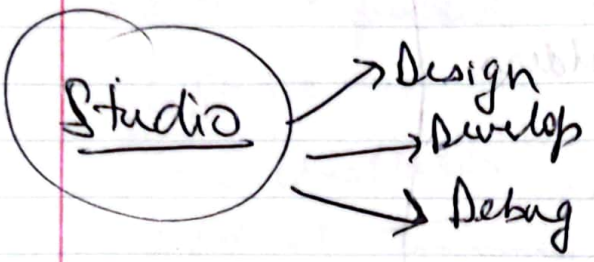
Unattended robots.

These are managed by Orchestrator.

* In both cases, process execution is done through jobs

* A communication b/w robot & Orchestrator is always triggered by the robot through the Robot Service component of the robot through a heart beat mechanism. every 30 seconds & waits for a response.

* In case of multiple packages a storage server is also involved for storing packages



Studio ——via dedicated feeds——→ Orchestrator (NuGet Packages)

Studio X → Business users looking to automate tasks for themselves & teams

Studio → Build complex process automations

Studio users can switch b/w studio & studio X

Robots

An execution agent that runs automation

↳ Attended Robots → Human intervention using UiPath Assistant

↳ Unattended Robots → No intervention
→ Triggered from Orchestrator
→ They can work on other tasks while waiting for input from user.

Orchestrator

↳ manage, control & monitor automation

↳ Repository for libraries, reusable components, assets & process by robots / developers.

✓ - Provisioning
✓ - Control & License Distribution
✓ - Automation storage & distribution
✓ - Running automation jobs in unattended mode
✓ - Monitoring

Studio

- Any automation is a sequence of building blocks called activities.

- Robot is a mixed role and includes both tenant and folder permissions.

- An automation developer - Named User license provides a unique user with the right to use Studio, StudioX, attended robot, Data Service, Action center and Task Capture

What the user can & can not do within each product is further controlled using roles & permissions.

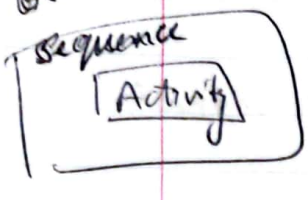
- By being signed in to Studio, users can publish package to Orchestrator. The processes will appear in assistant & jobs can be run.
  - Jobs cannot be done from automation cloud.

- We can provision only attended robots using the sign in feature

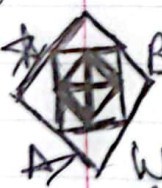
Studio

MARK

- In the center there is a designer pannel
- Activities are automatically surrounded by Sequence
  - Allow us to go from one activity to another seamlessly
  - Act as a single block activity



- Step Into debugging opens & highlights the activity before it is executed
- Debugging of a single file / whole project is available from Design & Debug ribbon tabs but debugging is not available if there are validation errors.



- Breakpoints are used to pause debugging process
- We can show / hide messages having log levels, errors / warnings in Studio from the Output panel

Business Analyst (as-is & to-be) → Process Definition Document (PDD) → Validated by Solution Architect → Project Manager Signs

Layout in a project → Sequences
→ Flowcharts
→ State Machines

OPTIONS

Filter
↳ "[Subject] = 'Course Invoices'"

Mark As Read → This will mark the emails as read

Output

Messages Set Var: Courses_Emails.

↳ right click then create variable

Annotate the top sequence

Activities → Filter (right click) → show classic

* Excel application scope activity

Path → invoices\ + Retrieved Email. Attachments.
FirstName

* Read cell (From Classic)
Activity

the type

Client code → Convert ~~it~~ into System.Double

→ Create a variable in expression editor to store the client's code

→ Add a logg message Activity

→ Add another sequence

→ Add 'Use application/Browser' activity

(indicate ACME Web app

↓ Do

Click Activity

↓ (Part 2)

↓

We must specify the (Anchor)

↓

A nearby element that helps to uniquely identify an element (Blue color)

→ 'Type into' activity

↳ select the text field & the anchor.

↳ Click confirm.

In the 'type this' field → Client Code.ToString

↓

Click activity → Send Button

* Get Text Activity

↳ Anchor.

↳ Output → New Var → 'DiscountValue'

* 'write cell Activity'

↳ Cell Range 'E18'

↳ Sheet Name "Invoice"

↳ Discount Value Variable

* write cell Activity

↳ Cell Range A31

↳ Signature

↳ Goto variable and add a difrent value text in double quotation

★ Right Click → Remove surrounding sequences.

★ If activity

. Contains("$")

★ "Use Desktop" Outlook Activity

(Outside everything (before the last)

↳ DO → Sent Email

↓

Account → Outlook

→ Outlook. Select Account

Body → Text

Attachments → Folder "Invoices" ↵

★ Message Box Activity

☆ When a process is created in studio, a folder is created with all files except the test cases files. The folder is archived to be sent to the robot.

☆ To ignore a file, → right click on a file
↓
ignore from publish

☆ We cannot publish a process if the project.json file is on a read only mode.

☆ When studio is connected to orchestrator, our publish options differ.

Publish
To ⇒ Orchestrator Personal Workspace Feed.

Assistant → Orange Dot on top right means assistant connected to orchestrator
↳ Clicking on it will open orchestrator

Process → Activities that transforms inputs to outputs

(Diagram) Components
- Inputs
- Process Flows
- Source Applications
- Outputs

Procedure → (Complex Document)
- Who is responsible for each part of process
- When the process needs to occur
- Handling exceptions
- Specifications of each process

★ Criteria for automation:
- Process Fitness [If can be automated]
- Automation complexity [No. of screens]

RPA Developer Tasks

Development & Unit Testing
- Build code modules as outlined & assigned
- Conduct unit tests
- Code is in line with development practices

Integration & UAT
- Break Fix per test results
- Process logging & Post deployment care

Deployment & Hypercare
- Hypercare period fixes
- Monitor logs & exceptions

L.01
v 2
3.8.2
5.6
4.38

Automation Categories

- No PPA
  - Semi Automation
- High Cost RPA
  - Zero Touch Automation

Data types of for Variables

- Boolean
- Int32
- String
- Object
- System.Data.DataTable
- Array of [T]
- Browse for Types [From imported dependencies]

Ctrl+R

→ Scope of Variable
→ Name
→ Type

☆ Variable with same name
↓
Most inner scope tops the priority list

☆ Input Dialogue Activity
↓
Takes input from user

★ DataTypes are borrowed from Visual Basic .NET language

☆ Generic Value: Variable type when we are unsure (usually temporary)

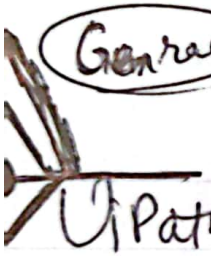
WVE - College States

2000 mile

3;2 ₹$₹

Texas Trinity Joshi

Ankita



UiPath Studio has an automatic conversion mechanism of GenericValue variables & arguments. First element is used as a guideline.

Arguments (ctrl + M, dirⁿ - IN) (ctrl + shift + M, Dirⁿ → OUT)

- ↳ store data dynamically
- ↳ Same data type
- ↳ Same methods & properties

Arguments pass data b/w workflows with direction

In / Out / In/Out / Property → Argument not currently in used

Syntax: in_DefaultTime, io RetryNumber

Direction ↵ ↳ Pascal case ↳ in/out

Invoke Workflow Activity

↓ has a log property where log msg's are integrated in the workflow

↓ To take in a whole workflow as an activity.
↳ We can add from dropdown
↳ Drag & Drop

★ For arguments of the type 'out' we need a ★ to assign it a variable value in the workflow (c++ like)

Array

To access an element → StringArray(0) ← Variable name

Concatination → String.Join(" ", StringArray)

★ When "+" is used b/w Generic value variables it is interpreted based on the value of first variable

VB.net

<> in VB.net means not equal to

★ Work flow layouts
- ↳ Sequence
- ↳ Flowchart
- ↳ State Machine → Uses finite no. of states → Triggering of an activity
- ↳ Global Exception Handler → In case of error

★ Control flow statements
- → If (For flowcharts, Flow decision Activity is used)
- → While
- → Do while
- → For Each
- → Switch
- → Parallel → Executes child activity in parallel

★ Input Dialogue Activity to get user's input

★ Right Click → Disable activity ★

★ Flowchart
↓
Right click, Paste the activity

★ If statement Syntax in VB

```
if(((Year mod 4=-0 and Year mod 100 <>0) or
    (Year mod 400 ==0)), 'Leap Year',
                         "not a leap year")
```

```
if((cond), True, false)
```

For Each
While
Do While
→ Only Loops

Break is used to stop the loops

☆ Browse for folder Activity → To access a folder

We can create a variable and assign the folder path to it

☆ Directory.GetFiles(FolderPath)

This gets all the files in a folder

☆ ForEach Activity

→ First thing we do is change the Type Argument

Variable

Converts to a string

★ FileName.Replace(".pdf", Now.ToString("_yyyy_MMddHH") + ".pdf")

☆ Move File Activity

↳ Moves a file from one folder to another.

Doubt

Loops video
if we are providing the path, why do I have to manually select the path ?!

* Switch → Atleast 3 actions, Not Boolean
* Parallel [Top to Bottom, Left to right]

↓

Finishes after all child activities are complete
OR Complete Condition is true

✓ EAD Recipt
✓ DL
✓ Resume ✓
✓ Prof. Photo ✓
Credential form ✓

* UI Automation
↳ UI Automation Activities
↳ Activity properties (How the robot performs)
↳ Targeting methods (Subset of properties)
↳ Input & Output methods
↳ Recorders and Scraping Wizards (Translates user's actions into seq's)
↳ The object repository (capturing as obj's in repos)
↳ AI CV
↓
Only in modern design

UI Automation Activities
↳ Containers
↳ Input
↳ Output
↳ Synchronization

Screen Scrapping & Data Scrapping are available in Classic

Targeting Methods
↳ Selectors
↳ Fuzzy Selectors
↳ Image
↳ Native Text

Input Method
↳ Hardware Events
↳ SendWindowsMessage
↳ Simulate (Fastest)
↳ Chromium API (Only Modern)

OutPut Methods
↳ Full Text (Fastest)
↳ Native (GDI)
↳ OCR (VM's)

Simulate input is not used to send hotkeys

CPU memory Activity Performance

- Classic & Modern Design
  - Uses mostly Selectors as Targeting methods → only selectors
  - Uses Unified Target as Targeting methods → Selectors, Fuzzy Selectors, Image

Project → General → Settings → Classic ↔ Modern

Container Activities.

- Open browser Activity
- Open Application

- The anchor & target together are called descriptors.
- Anchor's should be kept minimum.
- Argument Property of Input
  - If we want to open another application with the use browser application

Foreground → In the front

★ **Chromium API** Activities
- ↳ Use Application/Browser
- ↳ Click
- ↳ Type Into
- ↳ Hover
- ↳ Keyboard Shortcuts

★ Image targets doesn't work in background mode → Only in Use application browser activity

★ Output Activities for modern design
- → Get Text Activity
- → CV Get Text
- → Get OCR Text

★ Actions not compatable with background mode
- → Activities using images as target
- → Native text automation
- → Keyboard Shortcuts
- → Minimizing open applications
- → Screenshot activities

These will run in the foreground and then return to the background

★ PIP → Run attended process with a robot in an isolated window

Input Activity → Get User Input

Extract Table Data Activity

Write Range Activity → Opens excel activity

UI Synchronization Activities

- Check app state Activity
- Verify Execution feature ('Wait for' option increases the time to search for the UI element to be found)

If we are not in the correct tab, we can use the check app state activity.

For Verify Execution → Right Click, add verification
↳ Appears
↳ Disappears
↳ Text changed
↳ Visually changed

If the clickable activity changes with every automation, when we are selecting the target, we click on 'convert to dynamic text'

F2 → Pause ~~automation~~ UI Explorer for 3 seconds

1800

- Attach window activity → To open a window
- Minimize window Activity, Close Activity
- To send hotkeys, In the Type into activity, click dropdown and select 'Enter'
  - Also an activity
- Options → Send Window messages → Enabled
  - To run in the background.
- CInt (Variable) → Converts to integer
  - String
- Output method is the technology behind output activities.
  - Full text
  - Native
  - OCR
- Data Scraping → Stores in a variable called DataTable
  - Can be used with write CSV Activity.
- Image exists Activity → When an image doesn't move from a position

Tomato onion Milk

Excel application scope activity

- **Excel**
  - ↳ .xls, .xlsx, .xlsm, some .csv also
  - ↳ File can be open at runtime
  - ↳ MS Excel must be installed
- **Workbook**
  - ↳ only .xls & .xlsx
  - ↳ File not open at runtime

Excel works in a container ('Use Excel File') but workbook do not

- **Read Range Activity** → To read workbook
- **Filter Data Table** → To use operations on DT's (Double)
- **Build Data Table** → MAKE ~~both~~ DT's.
- **Add Data row** → ~~Converts data in a row~~ Adds data in a row

Row.ItemArray → To convert the data from each row to an array. Item, we used the row item method

Pension-
Success factor
N. Carolina Amit-
2001 Tiwari.

→ Read Range → Used to read entire sheet from Excel
→ Create DT variable
→ Filter Data Table
→ Build Data Table
→ For Each Row
→ Init Row Item → We can also use assign activity

↓

Variable = Row.ItemArray(4).ToString

→ Add Data Row → {Name, Age, Income}

→ Excel Application Scope
→ Write Range Activity

Output Data Table activity can be used to write a data table to a String variable

MS/SOLELY

Uses of REGEX
↳ Input Validation
↳ String parsing
↳ Data Scrapping
↳ String manipulation

Activities for Regex :
↳ Matches
↳ IsMatch
↳ Replace

4

* When we create an output variable for regex, it is of the type IEnumerable<Match>

→ Type Argument in Properties of For Loop:

System.Text.RegularExpressions :Match
Windows & Windows.Legacy Differ

* StrConv → converts first letter to upper and the rest to lower in a string for every word

* Invoke method → To implement a code outside of built in activities
  - Instance method → we specify Type as owner & pass Target Object
  - Static method → Present in the signature for static, not in instance
    → we mention TargetType & not TargetObject

Lists & Dictionaries are reference type variables.

Process:
→ Declare
→ Instantiate
→ Assign

[Basically make an object for it and assign it to list/Dict]

} System.Collections.Generic Type of String

Industrial Machinery System SDE

→ Papaya
→ Potato
→ Meat
→ Bread

Write Line Activity

Initialize a string

Added in default → ① new List (of String) from { " ", " ", " " }

② Using 'Add to collection' Activity.

→ To add two lists :

Enumerable.Concat (Cities.AsEnumerable, UKCities.AsEnumerable).ToList

Dictionary

Default Value as:

new Dictionary (of String, String)

Then use assign activity.

Add to collection, Remove from Collection, Clear collection, Exists in Collection

	Speed	Accuracy	Background Extraction	Text Position	Hidden Text	Support for Citrix
Full Text	10/10	100%	Yes	No	Yes	No
Native	8/10	100%	No	Yes	No	No
OCR	3/10	98%	No	Yes	No	Yes

Multi Threading
Except Handling
API storage
12 Factor Design an API
] - Interview sugg → Java

9:30

★ Selectors
- Full Selectors [Top level window is also used] [Multiple windows]
- Partial Selectors [Same window]

↳ Helps in identification of UI elements & its parents
↳ Stored as XML fragments.

Can be seen in UI Explorer window / selector property

idx : Index of current element

issues:
↳ idx, dynamic elements, selectors are too specific, system changes
↳ We use fine tuning for this

★ Frameworks
↳ Default
↳ Active Accessibility [old Microsoft]
↳ UI Automation [new Microsoft]

Selector editor in properties only takes values in string, → it is automatically converted in the backend

★ Activities → SetAttribute, GetAttribute

Full Selectors

- Not in Containers
- Multiple windows
- ~~Classic Desig~~

Basic Recorder generates Full selectors

Partial

- Containers
- Single Window
- Classic Design
- Desktop Recorder generates Partial Selectors

Fine Tuning

Wild cards

* → replaces zero / more characters ← Anything after this
  example : Hello*
? → Replaces single characters.
  Hello?
  ↳ Just 1 character

Nicer Variable Support

Variables → Edit selector → Find
Use Variable ← Right click ← attribute

Index Variables : like an index of array

ctrl + space → Shortcut to put our variable

Handling difficult situations

- Anchor Base Activity → Find element, Find image
- Relative Selector → Uses UI Explorer
- Visual Tree → Selecting the attribute from parent class
- Find Children → Find children Activity

We can either use a selector of type string in the Selector property or an object of type UI Element in the Element property but not both to identify a target.

Selection option window – Advanced Options

→ Enforce Visibility
→ Dynamic Text target
→ Native Text Target
→ Image Selection Mode [F3]

9-6

A robot will find the UI element (target or anchor) with all three methods in parallel

Selectors Image Fuzzy selectors

Then uses the one which finds it first.

Process is done again at runtime.

Anchor & Target together forms a unique pair & is called a Descriptor.

To improve reliability of UI elements

1. Show all matches (eye icon)
2. Target element validation.

Symbols while identifying Targets.

Blue → First one to identify

Blue → Successfully identified in combination with one of the non equivalent methods (Not the fastest)

Blue → Duplicates.

Red → Method fails

Get Attribute Activity

↳ ① See the ID name in the selector,
② Right click
③ Use Dynamic 'Variable' in the for each loop

We can unselect the CSS selector in the UI Explorer, as it is not recommended

Object Repository

↳ Creating & using UI Taxonomies
↳ Build UIAPI for apps & Share

Breakpoint → Click Once → Enable
→ Click Twice → Disable
→ Click Thrice → Remove

Symbol Red

- ○ Simple BreakPoint – Pauses
- ⊕ Conditional BreakPoint – Pause when a cond^n is met
- ◇ Simple Trace Point – Simple breakpoint with logging
- ◈ Conditional Trace Point – Cond^n with logging

BreakPoint → Right Click → **Settings**

- Step Into
- Step Over
- Step Out
- Run to this activity
- Run from this activity
- Test Activity
- Test Bench

Try → Activities which throw error

Catches → Activities to be performed when error (or exception) occurs

Rethrow → Causes exception to be rethrown to handle at an upper level (inside catch)

Finally → Activities to be performed after all try & catches blocks OR no errors

* **Invoking** → Activities panel / right click menu
* System.Exception type catch handles all exceptions
* If there are multiple same exceptions, we throw the most specific one
* exception.Message, exception.Source ⇒ Both of these are stored as exception objects.

If statement

'variable' is Nothing

★ Element Exists activity → Checks if UI element is present/not

★ Kill process activity ⇒ Closes an open application

If we want to take a cell from the excel:

Convert.ToInt32(row(1).ToString) > 100

★ In throw exception:

new BusinessRuleException("Values > 100")

★ Retry Scope

Retries until the cond^n is not met
OR error is thrown

If no termination cond^n → Retries until no exception
→ OR No. of attempts exceeded

No Of Retries property → Default value is 3

Retry interval property → HH MM SS format
00:00:05

☆ Continue On Error → If this is set true on the parent container then all child activities are also true

If it is set true in the try section, no error is caught

Two Ways: ① Click New File → Global Handler
② Right Click on XAML File in project → Get as Global Handler

Global Exception Handler

- Only one per project
- Ignore, retry, abort or continue.
- Two arguments → errorInfo IN, result OUT

Choose Next Behavior: Continue, Ignore, Retry, Abort

Example

Condition:
errorInfo.RetryCount < 3

Then:
Assign
result = ErrorAction.Retry

Else:
Assign
result = ErrorAction.Abort

Logging ctrl+L

1. Studio Logs
2. Setup Logs
3. Orchestrator Diagnostic Logs
4. Robot Logs
   - Process logs → Default Logs
   - Robot execution logs → User Defined Logs
   - Robot Diagnostic logs

ORCHESTRATOR

Job started from orchestrator can only be stopped from an orchestrator ← Robots
↳ Folders are subdivision of tenants

→ Packages are published to orchestrator using feeds.
→ These feeds can be tenant level / folder level
→ Packages in tenant level can be used in a process in any folder.
→ If a package is published in a folder level, it cannot be used for processes in other folders.

Entities that exist at tenant level
↳ User
↳ Machine
↳ License
↳ Webhooks

Folder Entities:

Types
- Classic → Created by default
- Modern

Process & Jobs are folder entities

↳ Asset
↳ Storage Bucket
↳ Queue
↳ Trigger → Time Trigger, Queue trigger

Personal Workspace

↳ Modern folder for a dedicated User.

Roles → Permissions

Role	Level
Tenant Admin	Tenant
Allow - Folder Admin	Tenant
Folder Admin	folder
Allow - Automation User	Tenant
Automation User	folder.

Logs

- → Diagnostic Logs (UI Path Orch)
- → Execution logs (Process Execution)

Process, User & machine template need to be mapped to the folder in order to run an unattended job.

The package is a tenant entity

⇒ Background Process is also called Headless Process

refers to a job that runs without interacting with any sort of UI.

⇒ Foreground Process interacts with UI & can run with multiple background processes.

Unattended robots don't exist as independent entities.

* Unattended robots are a combination of User / Robot & a machine & Process.

* Another name for a robot is execution slot

~~For unattended robots~~, &
For background process, we don't need to set up login credentials for a robot

Allocating ~~more~~ than one runtime & executing several processes on the same machine is applicable only for Windows Server machines

Jainil Solanki

Jainil

To get files in Array → Var = Directory.GetFiles ("Path")

- The folder should be atleast assigned an automation user role to run automations
- Jobs in foreground run in a sequential manner one at a time but in background process they run at the same time
- Licenses are also called Runtimes.
- Robot creation can be done from Active Directory for both attended & Unattended robot

Scroll Simulate Actiu

- Resources in Orchestrator

  Assets → To store data
  Queues
  Processes

  [Create ACME Activity]
  [Add Queue Activity item]

In Studio, in the bottom right we can see these 3 resources when connected

- Resources are stored at folder level.

Switching folder means different resources related to that folder.

In Queue Resource → Input → Item Information

'Indira' ← Put items from DT & Convert to string

- Drag & drop resources to sequence

Libraries

Create new → Library

Get Credential Activity

↓

Create Argument 'in_ACMEloginCred' & pass it as input to the activity & create two output variables

- Tenant → Settings → Deployment → Libraries

↓

Both host & tenant feeds.

Save ←

- Libraries are published as nupkg packages. just like automation projects
- We can use libraries as activities now.
- New → Make new Template → Orch Tenant → Select library

Save ← Install ←

Drop the activity in the sequence & put the name of the asset in its input field.

Water Video

Storage Bucket

Activities → Available → Orchestrator → Storage

↓

Upload storage file

Works with Dispatcher-Performer model

Queues — Folders entity

↳ Unlimited items in free form

↳ If specific schema is needed, can be done at the start in form of JSON

↳ As soon as the queue items enter processing they become transactions

Bulk Add Queue Activity → Add DT to queue

Type into Activity

↓ To get items from queue

Output Variables of the queue

↓

Transaction Item . Specific Content ("Sender").ToString

We can also do a set Transaction Status.

Business Process → Linear
→ Iterative
→ Transactional
↳ Similar to iterative but each process is individual

Mail Activities Pack

↳ Covers various protocols like IMAP, POP3 & SMTP.

The retrieved messages (Output) are stored as a collection of MailMessage objects.

Outlook
Exchange
IMAP
GMAIL
] Activities with Inbuilt filters

IMAP

* Get username/password Activity → Retrieve email & pass from windows Credential Manager
MailFolder: "INBOX"
PORT: 993
Server: "imap.gmail.com"
Email: GmailCredentials.Username (Variable)
Password: " " . Password (Variable)
Messages: EmailsList

IMAP ⇒ GMAIL

Properties → Type Argument for retrieved email list

(Object Type): System.Net.Mail.MailMessage

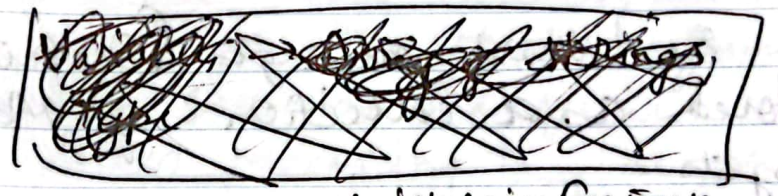
* To use the list in a for each

If cond^n

Retrieved Email.Subject.ToLower.Contains("invoice")

* Save mail msg Activity

OUTLOOK



Variable in For Each

Log in For each → Current Mail.Attachments.Item(0).Name.ToString

- Use GMail Activity
- Use Desktop Outlook Activity
- for each email
- Delete Email
- Save Email Attachments

To Save current Time Assign : Start = DateTime.Now

To Send an Email

→ Credentials from username

→ Send SMTP Mail Message activity

→ Properties.

Attachments (...) → Dict" → In
Value → Path of local file

Port → 465

Server → " Smtp.gmail.com"

Logon → Email → Gmail Credentials.Username
" " . Password

* For multiples file attachments.

For each file in folder Activity
↳ In filter, we can use wild cards *, $ within " ".

Then use Add to Collection Activity Inside Do

Prop: Collection → Variables
Item → Current File . FullName

* A send calender Activity must be used inside Outlook App Activity

PDF

Native (Original) Text can be selected

Scanned (Images)

TextActivity / OCRActivity

Pakages → PDFActivities

Tesseract OCR

☆ Anchor Base Activity → Anchor + Action Block

Same concept as Anchors.

→ Find Element Activity
→ Find Image Activity → Should be open at runtime

2012
2013 & Express
2015

Version Control - Git ; TFS & SVN

Bottom Right Corner

Studio
↓
Backstage View
↓
Team

Git → Distributed Version Control

TFS & SVN → Centralized Version control

In the project tab, right click on project folder, we get version control options

pencil icon → Files that have been edited.

Right Click Folder → Undo → Retrieves previous version from local repo

Right click File → Show changes → Workflow level

Yellow → modified changes
Green → Added changes.

Solve Conflict Window → When we push with some changes but in the backend someone already made changes.

Amend Commit → Overwrite last commit to a Git repository

UI Path Testing Package

RPA Testing with Studio

Unit Testing
Integration Testing
System Testing
Acceptance Testing

Stability of robots is affected by:

Environmental issues
Application issues
Automation issues

★ Behavior Driven Development (BDD)

Given
When
Then

① Given – Test Data
② When – RPA Workflow
③ Then – Check Activities

Activities:

Verify Expression
Verify Expression with Operator
Verify Control Attribute

★ Right click on project file & click "create test case".

★ By default, test cases are not publishable to orchestrator.
Right click → Set as publishable

Can be easily converted

Basic Test Case → Single Set of Data

Data Driven Test Case → Different Scenarios

Verify control Attribute Activity has a 'Alternative Verification Title' in its properties which overrides the title in the orchestrator.

Test Explorer contains Test Results & Activity Coverage Panel

Questions

→ In Orchestrator, asset values can be defined for each user.

→ Retrieve all UI elements from form on a filter condition → Find Children Activity

→ Workbook read Range Activity works with .xlsx files and excel does not need to be installed

→ Feature of Native Screen Scrapping method is to extract the text position.

→ Attended robots can be connected to UiPath Orchestrator without Windows Credentials

→ When working with the show changes option it's good to know that it works at workflow file level.

→ Supported variable type in output property field of get mail activities → List (Mail Message)

→ A table cannot be stored in orchestrator asset

→ Orchestrator admins can toggle between read only and write privileges for each storage bucket.

→ Assets, Process & Queues are available in resources tab

→ The retry Scope activity can be used without a termination condition. In this case it will retry the activities until no exception occurs or the provided number of attempts is exceeded

→ Adding variables can be done to the watch panel through locals panel, variables panel & watch panel

→ Changing a variable in the immediate panel will be reflected in local, watch & debug mode

→ Call stack panel displays the next activity to be executed and its parent containers.

DOUBT, OR OUTPUT DATATABLE

→ Generate Data Table activity can be used to write the content of a datatable into a String.

→ You can manage your variables from the Design panel > Manage Variables > Remove Unreferenced

→ Global exceptional handler can't be used in a library.

→ Execute Macro is an Excel Activity.