Integrating Human-in-the-Loop Automation with UiPath Action Center

Himadeep Movva

movvahimadeep@gmail.com
Independent Researcher

Abstract

The Action Center provides business users with a mechanism to manage actionable tasks and give business-related input to Robots. It facilitates support for extended, unattended workflows that need human involvement. The execution of extended workflows is broken into segments, letting the Action Center suspend and resume the workflow once human feedback is received. This paper explores using the UiPath's Action Center for data scraping and extracting content for various business processes across different domains or departments of businesses. UiPath's AI center is a one-stop solution for setting up feedback mechanisms where human-in-the-loop solutions could be configured. Human intervention – approval or validation may sometimes be needed before going to the next step in the process, and the AI center enables this feature. This literature explores the immense potential of UiPath's ActionCenter while analyzing several use cases and problems solved by AI centers. Also, this paper would be a great way to learn about and explore the different features provided by the UiPath Action Center.

Keywords: Action Center, Uipath Studio, Machin Elearning, OCR, Persistence, Orchestration Process, AI, Human In The Loop, Classification, Validation, Document Object Model, Tenant, and Taxonomy

I. Introduction:

A lengthy workflow requiring human approval is set up in Studio utilizing the Orchestration Process template and specific activities.Such a workflow produces Actions that users can complete. After a human submits an Action, the flow returns to the process, as indicated by the relevant activity in the workflow, and continues on an available UiPath Robot. In the realm of extended business processes, this facilitates improved resource distribution and minimizes execution delays, particularly since any free Robot can carry out portions of the tasks. Actions are activities assigned to users that they can complete while managing lengthy workflows, and user input is needed before moving on to the subsequent step in the workflow. When an Action is created, it appears on the Actions page with the status of Unassigned. Details such as priority, title, or action catalog are filled in based on how the Action was defined in Studio using the appropriate activity. This research paper explores the nuances of action center and itsusage within the context of the orchestration processes and invoice processing.

II. Why UiPath Action Center:

When there is a need to review data extracted by the bot, often there is no scope- either the bot doesn't process the records in question, or the bot sends a report with those records such that those records are resent. Often, during data extraction from digital documents in finance, accounting, pharmacy, hr, or supply chain departments, the bot wouldn't be able to extract content reliably with reasonable confidence due to

various reasons including but not limited to low-quality images in documents, non-standard structure, or varying layouts with different fields. This requires additional overhead and resources to perform the tasks manually. Partial automation is when a human manually starts the process in attended mode. End-to-end automation is when the bot is run in an unattended mode without human action or human trigger. In attended automation, where there is a need for user input, the bot stops processing until manual intervention is completed, and only then will the bot be resumed, blocking resources until the user acts. Using AI action center in the unattended automation Figure 1, Resources or bots are not blocked until user actions; resources are assigned to other tasks. As and when the manual action is completed, the bot resumes the process again without any manual intervention. In other words, until process execution is stopped, robots are freed, letting us utilize the bot elsewhere. This has been a major benefit of the UiPath action center. These kinds of processes are called orchestration processes, which are designed to handle long-running workflows with a combination of human intervention and background processes. Such processes allow workflows to wait for API responses, human input, or other triggers and then resume from that point.[1]

III. Using UiPath Action Center:

a. Enabling Action Center in orchestrator:

Actions need to be enabled at the tenant level.[2]

1. Log in to orchestrator.
2. Navigate to Admin > Tenants.
3. Click the vertical ellipsis button on the desired tenant line and select Edit Services. The Edit Services window will then appear.
4. Select Actions and click Save. The changes are reflected in your tenant Action status

The available action statuses are as follows as mentioned in Figure 1:

Unassigned—Although the action has been created, no user has been assigned to it.

Pending—After being assigned, the action is awaiting user confirmation.

Completed—The action has been finished and verified. Validation cannot be undone once it is finished.

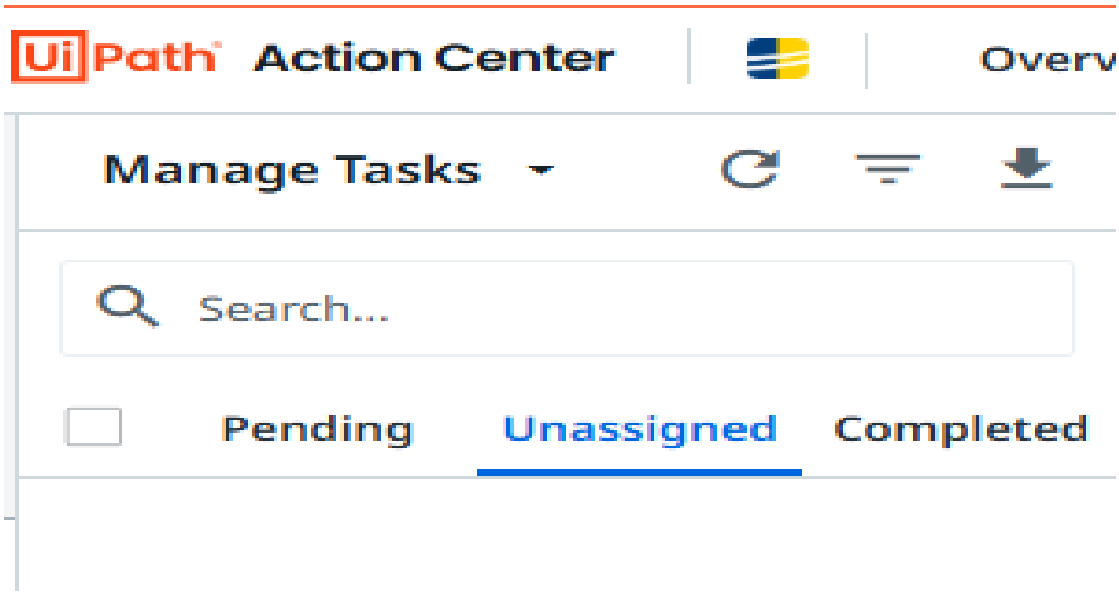

Figure 1- Status of Items assigned

b. Enabling action center in UiPath studio:

Go to Manage Packages and install UiPath.Form.Activities and UiPath.Persistence.Activities.UiPath.Form.Activities are used to create and use forms with buttons and interactive options, enabling decision flows based on user input. The bot execution will be continued after the form is completed. Some of the form activities include create form, show form, get form data, validate form, and close form. The UiPath.Persistence.Activitieslet workflows persist in their state, allowing users to interact with the Action Center and resume execution from the point of suspension. In case of simple tasks where user data is needed, use forms and build long-running workflows that may require human verification and action such as approval and use persistence activities. As mentioned in Figure 2 , enable supports persistence option in the studio design settings.

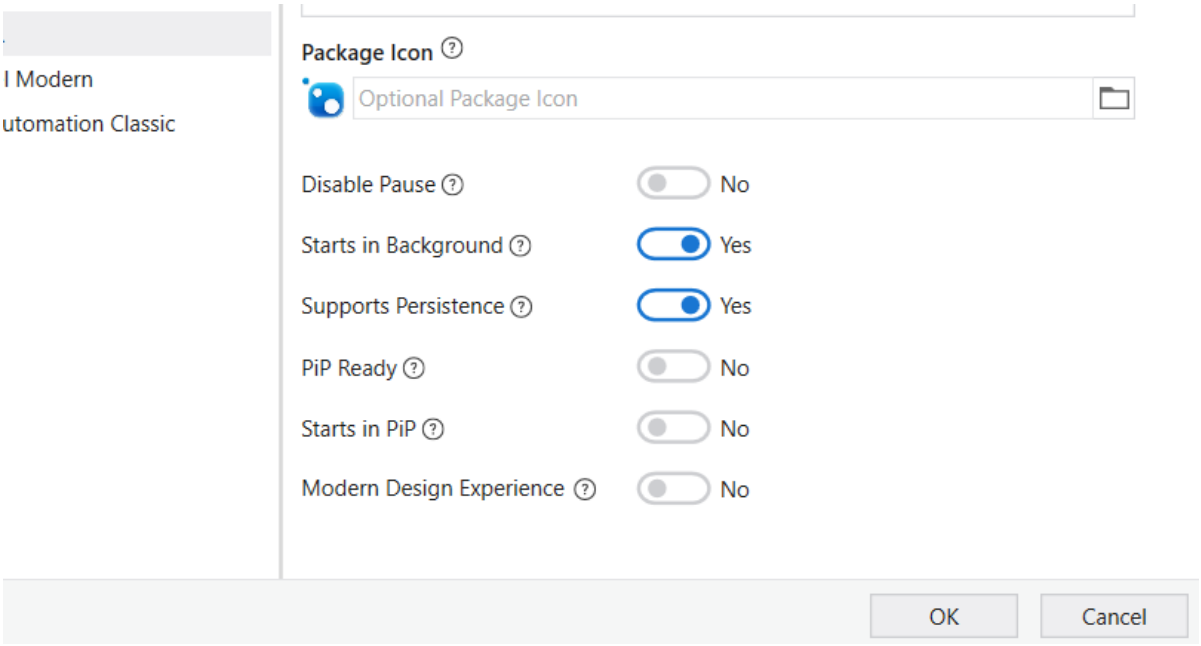

Figure 2- Enable persistence

c. Creating form tasks and beyond in the studio:

Drag and drop Create Form Task activity in the studio workflow. This is used to design UI for tasks in the Action Center, and forms can be configured to include textboxes, dropdowns, checkboxes, or fields for the user to validate or modify data. Mention the Task Title, which is a mandatory field;task priority that has the option to choose from low, medium, high, and critical;task catalogthat provides an option to classify items of tasks;form data that has the option to send and receive data as in/out objects of type Dictionary<String, Argument>; and task object that will be returned from orchestrator as a form task data object after the action is created. Upon clicking the designed form, there will be an option to customize the form that will be used.[3]TaskObject has several attributes that hold information about the output action object.ActionUrl attribute gives you the URL to the action inside the Actions tab in Automation Cloud. To access the action from the Action Center on-premises, add /actions_/tasks/taskID to the Action Center base URL, for example https://laptop-name:port-number/actions_/tasks/taskID

The components in the form designer can be seen in Figure 3.The left side of the form designer shows all the basic components of the form, which can be dragged and dropped onto the form. CountryList contains the values passed as an in-argument, and a value can be selected as an option. For example, if a fieldnumber is required, it can be dragged and dropped into the form. The display name can be defined in the Label,and

in_argument/out_argument can be assigned to the Field Key. For instance, if the use case is about expense approval and if the approved amount can be different from the applied amount, the form in

Figure ***4*** can be used

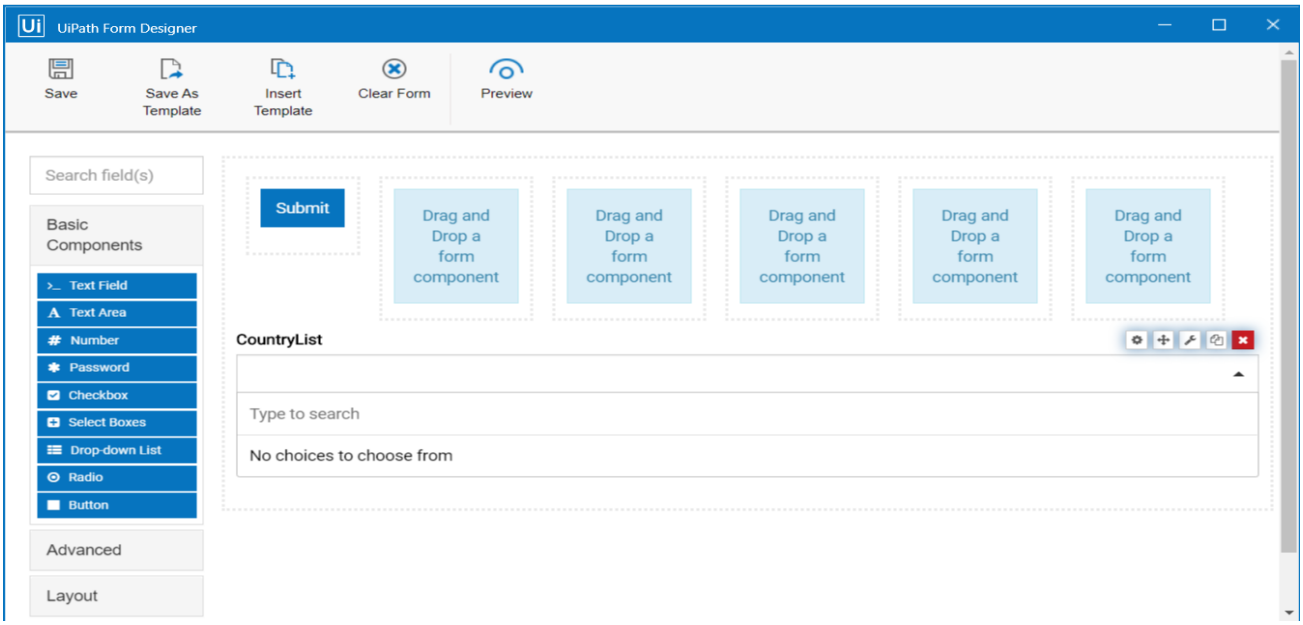

Figure 3-Form Designer

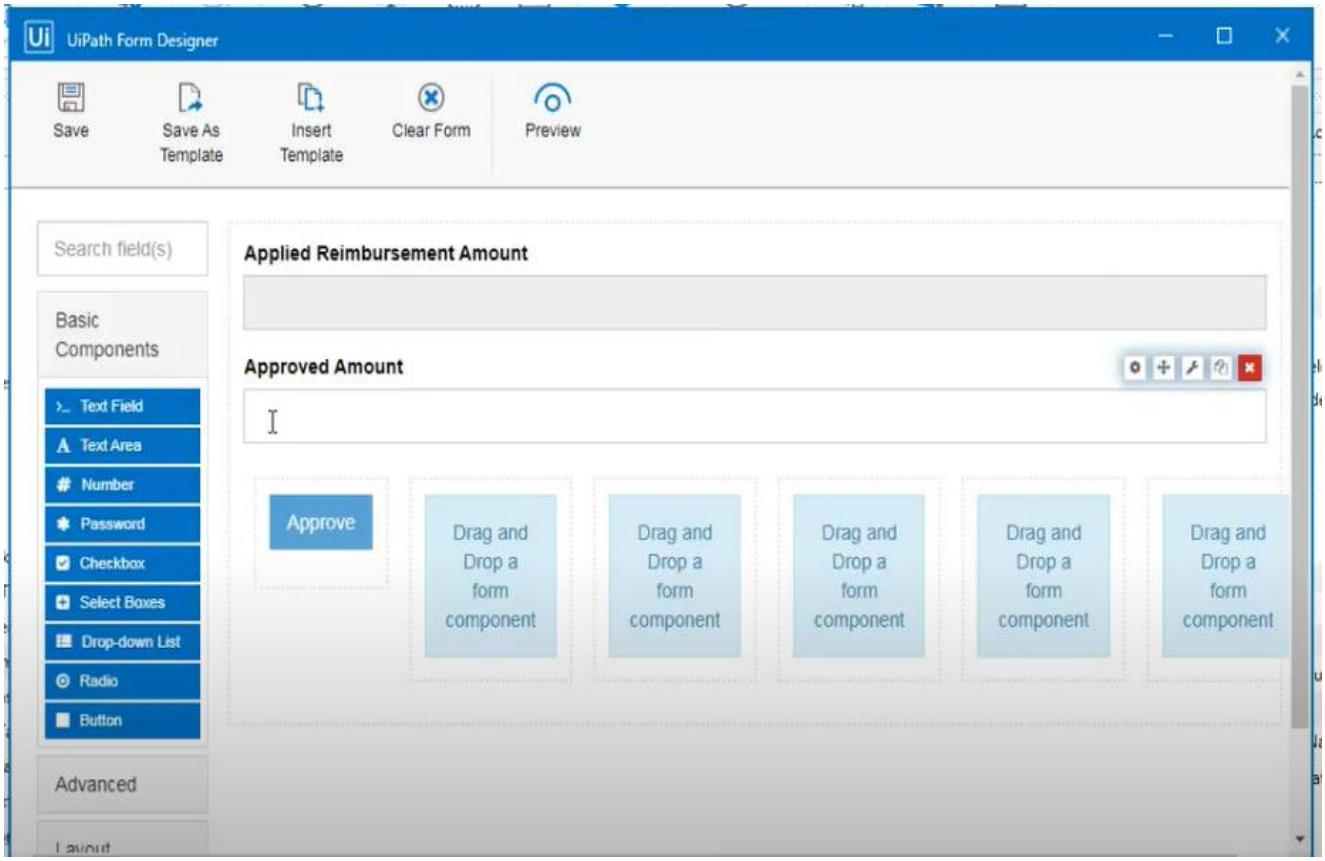

Figure 4- Drag and drop a form component

After this component, define wait for form task and resume activity. The variables in/out from the previous task, whichcreates form activity, will be captured by workflow variables. The input for this activity is the output of the previous activity, which is the task object. The output will be task action, and both are of

type form task data.Wait for the form task, and resume activity stops the execution of running workflow until a specific action is completed. Post completion of action, the execution of the workflow will be resumed.

There are three action statuses: unassigned, pending, and completed. When the create task form is executed, the action is created and will be in the unassigned state. After assigning the task to a user, the task will be moved to a pending state. Once the action has been completed, it will be marked as complete. The action filter options are priority, time, type, action catalog, label, and forwarded actions.

Action catalogs separate or group actions into different categories, helping users organize actions by departments or teams. The action will be added to the corresponding folder based on the catalog mentioned in the form-based activities. For instance, tasks related to human resources can be routed to the human resources catalog, tasks related to the supply chain can be routed to the supply chain catalog, and tasks related to the pharmacy can be routed to the pharmacy catalog. It is possible to edit the catalog nameby navigating to the hosting folder, clicking the catalog, and clicking the edit icon. Also, ensure that corresponding changes in task catalog names are reflected in the code, and if a task catalog needs to be deleted, ensure that the deleted task catalog is referenced now, here in the code. In the actions tab, the approvers can filter the actions based on their department and work on them instead of skimming through all the action requests across the departments.

Instead of manually assigning tasks, there is an option to assign tasks while they are being created automatically. To do so, UiPath's Assign Task activity is used.It assigns one or more tasks, using an email address registered in the orchestrator, to a user or a list of users. To enable this action by the bot, it is essential to set create folder permission for the robot. It has three fields: task id, username or email, and failed task arguments. Additionally, the AddToCollection activity can be used to define a list of users, and for each iteration in the loop, the users can be assigned tasks.In the case of assigning different user'stasks, effectively distributing the workload among different users. This can be achieved using the round-robin principle, where each user gets a fair chance before repeating the execution of the cycle. The following snippet depicts an algorithm related to this logic.

```
START
  DECLARE TaskList[] ← Collection of tasks to be assigned
  DECLARE UserList[] ← Collection of available users
  SET TotalUsers ← COUNT(UserList)

  FOR i = 0 TO COUNT(TaskList) - 1 DO
    NewTask ← Execute(CreateFormTaskActivity)  // Generate a new task
    TaskID ← NewTask.TaskId  // Retrieve the assigned Task ID
    AssignedUser ← UserList[i MOD TotalUsers]  // Distribute tasks cyclically

    Execute(AssignTask, TaskID, AssignedUser)  // Assign task to selected user
  END FOR
STOP
```

There's an option to get form tasks using get form tasks activity. Using the query, tasks from a certain catalog can be filtered based on the task status: completed, pending, or unassigned.A query needs to be given in the filter area. The query would be "Status eq Completed" to search for completed tasks. In the select field, the columns to be retrieved can be selected. The columns can be ID, Status, priority, or any other field that the activity supports. The result will beof type List<FormTaskData> object.

IV. Usage of UiPath Action Centerin invoice extraction:

Often, while extracting data from invoices, we encounter challenges in documents such as semi-structured or unstructured data, image-based data, varying layouts,handwritten data, poor image quality, missing fields, or formatting issues in data.UiPath document understanding allows using AI, OCR, and rule-based extraction to handle unstructured, semi-structured, or structured invoices. ML extractors and pre-trained Machine learning models can be configured. If the extractionresult's confidence score is low or if there are incorrect or missing fields, documents can be sent to the action center for manual review. The combination of Action center and document understanding provides high accuracy with AI-based invoice extraction, exception handling, and configuration of human validation when confidence scores of extraction results are low, and most importantly, scalable for huge volumes of data across multiple vendors.

To use the document understanding framework, UiPath.IntelligentOCR.Activities package needs to be installed from manage packages in the studio. Once installed,the Taxonomy Manager wizard appears in the studio.Apart from the scope activities that allow usingany document classification and data extraction algorithms, the document understanding framework can be used with built-in classifiers, extractors,and customized ones.[4]

a. Load Taxonomy:The taxonomy manager can configure which data is processed from which documents. Document types and fields that need to be extracted from those documents will be defined here. This information is stored as metadata and will be used throughout the process. Field names can be customized, and types can be text, number, date,name,address,keyword, set, Boolean, or table. The .json file created with load taxonomy activity will be used across the project scope.

b. Digitize document: This is the step where a document is digitized—the process of converting it into a machine-readable format. This way, the robot can understand the file and process it further. Input for this activity would be the document path, while output variables are document text, a string, and document object model, a JSON object for that file.By default, this activity requires configuring the OCR engine's usage. If not required, however, the OCR usage is skipped in the case of native documents. However, if force apply ocr flag is set to true, every time ocr will be applied. Choosing an ocr engine is recommended based on testing the best engine for the use case. If the plan is to use document understanding in conjunction with Action Center, certain OCR engines such as Microsoft OCR, Microsoft Azure Computer Vision OCR, or Google Cloud Vision OCR could be avoided as Microsoft OCR doesn't report confidence, and the other two OCRsreport confidence only after a certain setting is configured.

c. Document Classification: This step is used to classify what type of document is being processed. For this, classify document scope is used, and inputs are DOM variable of type document and document, both retrieved from digitize document activity and document taxonomy variable obtained from load taxonomy manager. Also, document classifier needs to be chosen from keyword-basedclassifier, intelligent keyword classifier,flexicapture classifier or machine learning classifier based on the use case. For each document type, in the document classification, minimum confidence threshold can be set. The output will result from running the classifier files on the specified file, stored in a IReadOnlyList<ClassificationResult> object. This field supports only IReadOnlyList<ClassificationResult> variables. The classification result object has information related to confidence, displayed as a numeric value between 0 and 1, and document ID, among other information. If a file contains a single document type, then the classification component should be configured, and it will return a single classification result. In case there would be multiple documents, such as 3 pages for the invoice, or 3 pages for an invoice, and 2 pages for a medical record, the classification results are expected to be retrieved, each representing the right page range from the input file. Document Types that classification is attempted on are the ones defined in the project Taxonomy.In case there is only

one type of document, document classification is not necessary, but in case of multiple document types, document classification is necessary.

d. Data Extraction: It helps in identifying information from the predefined document types.This information is defined in Taxonomy as a list of fields for each document type. Any field that is not defined in the taxonomy cannot be retrieved during data extraction. If there are two document types in a document, document extraction needs to be run, based on ranges of page numbers, for those two document types, and one classification result will be obtained each time. The extraction is done through Data Extraction Scope activity, and several extractors can be chosen based on requirements. The inputs for this activity are document path, taxonomy, document text, classification results, and document object model object.Extractors are built with predefined machine learning models, and the required fields can be obtained using those ML packages. Forunstructured and semi-structured documents with high variability, Machine Learning extractor can be used. The output is an extraction results object that contains fields as key-value pairs, confidence, document type ID, tables, and validation errors. We can loop through fields, access individual values, and check confidence, helping automate decisions based on data reliability.

e. Document Validation: UiPath document extraction can be validated using two ways: one is attended using the present validation station, and another is action center tasks, which is unattended. The attended mode works only if there is the bandwidth to assign a user while running the bot. Once the users confirm the data in the UI, the bot resumes. As extracted data is shown alongside the document, the values can be corrected if required and submitted. This requires User action during execution, making this unsuitable for unattended automation

The following steps are followed for UiPath document validation using the action center in unattended mode. In unattended mode, the Action Center task uses Create Document Validation Action and Wait for Document Validation Action and Resume activities.

- When using document validation action, a storage bucket must be created as files,and extracted data will be stored temporarily. This is primarily because, during human-in-the-loop validation, documents need to be accessed by both the bot and human user.A storage bucket is a cloud storage in the orchestrator. It ensures that the document is available to human users even after the bot picks up the next transaction. Hence, the first step would be to create a storage bucket.
- Also create a catalog name that can be used to separate and group documents based on departments or functions.
- Drag and drop Create Document validation action activity into the studio. Action title, Action priority, document path, document text, action catalog name, DOM variable, action folder path, taxonomy variable, and storage bucket details must be provided. The output will be an action object, which should be passed in as an input to the Wait for Document Validation Action and Resume activity. Ensure the robot create and edit permissions are given to storage files, view permission is given for storage buckets, and create permission is given for actions.
- Drag and drop Wait for Document Validation Action and Resume into the studio. This activity ensures that the bot pauses for human input and that the bot resumes after a human user completes the validation task in the action center. The inputs for this activity are action objects from the create document validation action activity, validated extracted results variable, document path, document text, DOM, taxonomy, and extracted results. The output is an action object.
- Drag and drop export extracted results activity. The input will be extracted results, and the output will be the dataset. The dataset variable is the array of tables and contains the results.
- Output data into Excel

Start

For each dataset.Tables
Write to Excel

End

- How to display pdf in action center- Drag and drop create form task, click open form designer, pass the variable as argument for file path, drag HTML element, and click refresh on change after entering HTML content <embed src={{data.pdf_storage}} width = “1000px” height=”600px”></embed>

V.Overview of Orchestration processes

Long-running workflows support orchestration, humans in the loop, and long-running transactions in unattended environments. UiPath is a component of an ongoing workflow.perseverance.Activities can be paired with non-user interaction and invoke process activities to coordinate human and robot duties.Long-running processes are of three types:background processes that may contain API activities and running in session 0; processes with user interaction activities started by start job inthe user session; and human in the loop process: User interacts with the form that needs to be completed in orchestrator by using create form task and wait for form task and resume.

There’s a template for the orchestration process in the templates available in Uipath Studio. Usually, in unattended mode, the bot picks up the next transaction once the current transaction is queued for manual review in the action center. Additionally, using a resume after a delay, the execution can be suspended. Reusable components developed in long-running workflows can only be referred to within another long-running workflow. The queue item will stay in an in-progress state for more than 24 hours until the workflow sets the transaction status upon resuming if a lengthy workflow is linked to queue processing and the workflow is suspended. If queue item IDs are included in the same scope as Wait and Resume actions, they will be a part of the persistent workflow context.

Ensure that persistence points such as resume, and wait are not included in for each loop[5]. Otherwise, after the execution of one loop or first iteration, the process gets suspended until the human user takes action, and this may not be a desired behavior.Alternatively, Parallel for each can be used.Ensure that All variables used in the scope of a long-running activity are serializable. The Delay and Retry Scope activities are unsupported and do not work properly when used in the Main workflow of an Orchestration Process. They should be placed inside a No Persist Scope activity in such cases.

VI. Conclusion:

UiPath's Action Center, Document Understanding, and Orchestration Processes improve automation by facilitating smooth communication between humans and robots. Action Center is essential to human-in-the-loop automation because it enables validation, approvals, and exception handling. Custom form design makes it possible for business users to engage with automation in an efficient manner, enhancing data compliance and accuracy. Action Center is a UiPath Orchestrator component that enables human-in-the-loop automation by allowing users to review and validate data before automation resumes. Document Understanding enhances automation even more by utilizing various extractors and AI-powered models to extract both structured and unstructured data from documents.

The orchestration process is key to automating the human-in-the-loop process in unattended mode.These workflows ensure that robots work in parallel while awaiting human input, optimizing efficiency in business-critical operations.

References

[1] [Online]. Available: https://docs.uipath.com/studio/standalone/2022.4/user-guide/orchestration-process[Accessed Jul,2022].
[2] [Online]. Available: https://docs.uipath.com/action-center/automation-suite/2022.4/user-guide/exploring-actions[Accessed Jul,2022].
[3] [Online]. Available: https://docs.uipath.com/action-center/standalone/2022.4/user-guide/create-form-task[Accessed Jun,2022].
[4] [Online]. Available: https://docs.uipath.com/document-understanding/automation-suite/2022.4/classic-user-guide/introduction[Accessed Aug,2022].
[5] [Online]. Available: https://docs.uipath.com/studio/standalone/2022.4/user-guide/orchestration-process[Accessed Aug,2022].