UiPath Validation Control in Apps

Public Preview Documentation

Step by Step Guide for building your Validation Apps

Change Log 3

Introduction 3
Public Preview Feature List 4
Forward Demo and Walkthrough of the feature 4
Acessing updated Public Preview Documentation, Starter Kit and Updates 4
Limitations 5
Prerequisites 5
Starter Kit / Assisting Material 5
Demo Scenario and Workflow 7
Environment Setup 9
Assistant Setup 9
Studio Desktop Setup 9
Studio Web – Web App Project Setup 11
Document Understanding Setup 13
Comms Mining Setup (optional) 13
Orchestrator and Storage Bucket Setup 13
Serverless Robot template and machine Setup 14
Components of Validation App 16
Validation Control Details 17
General Properties 18
Accessing and Interacting with the Validation Control in Apps and Robot / VB Properties 19
Validation Control Events 24
Validation Control Specific Activities 24
Styling Properties 25
Common Expressions 27
New Document Understanding Activities Details (UiPath.IntelligentOCR.Activities pack) 30
Create Document Validation Artifacts 30
Retrieve Document Validation Artifacts 31
New Communications Mining Activities Details (UiPath.CommunicationsMining.Activities pack) 32
Create Communications Mining Validation Artifacts 32
Retrieve Communications Mining Validation Artifacts 32
Step by Step Guide – Design Time Setup 33
Agent App Setup 33
- Prerequisite 33
  - Web Apps Setup 33
  - Process Setup 34

Debugging the validation control directly in Apps instead of Action Center runtime 40
Studio Desktop Process Setup 42
Prerequisite 43
- Orchestrator and Storage Bucket Setup 43
- Studio Packages 44
- Taxonomy Setup 44
- Sample File to be used 45
- Argument as File Name for Process 47
- Process Setup 48

Step by Step Walkthrough – Run Time 60
Testing the App 60
Best Practices 61
Tips and Tricks 61
Validation Control Cheat Sheet 62
References and Other Learning Material 62

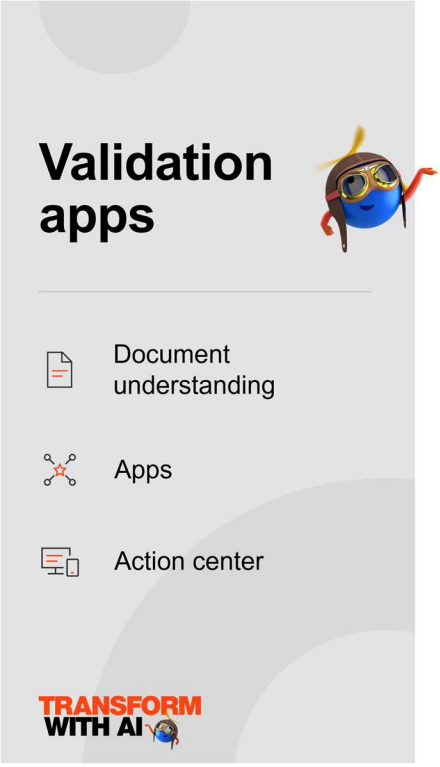
Change Log

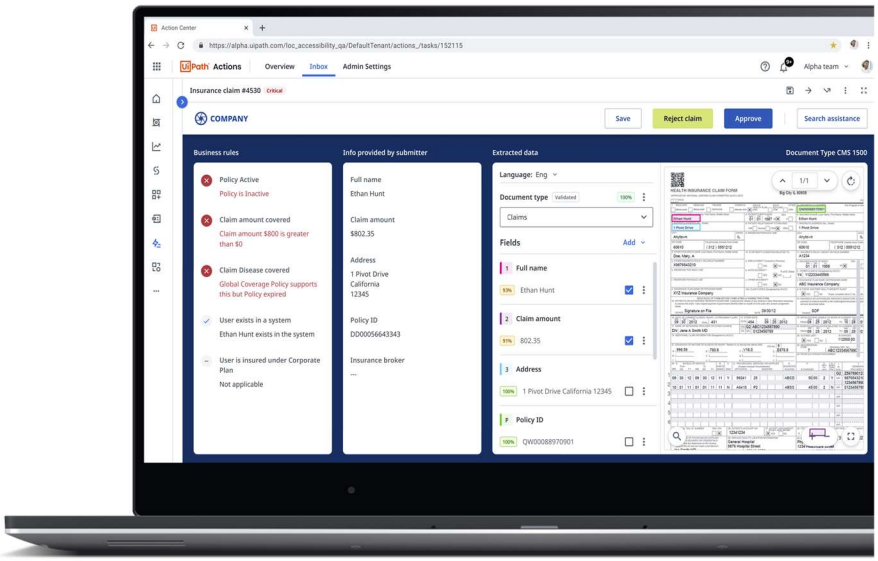
Date	Version	Changes	Description	Change Owner
Private Preview				
18 Dec 2024	1.3	Table and Multivalue field Support	VB expressions for table and multivalue field support Also added best practice on multiple content validation data	Anil Kumar Alanka, Udit Chandna
22 Nov 2024	1.2	Debugging & prerequisite Robot units	Added debugging section & prerequisite Robot units	Udit Chandna
20 Nov 2024	1.1	Prerelease checks	Check for prerelease in Studio Desktop setup	Udit Chandna
18 Nov 2024	1	-	First Draft	Udit Chandna Ioana Gligan
Public Preview				
1st Mar 2025	1.4	Public Preview Feature List Properties and Expressions update Contracts Details addition Updated Starter Kit		Anil Kumar Alanka, Udit Chandna

Introduction

A part of your document processing automations can go straight-through without any human intervention. For some, though, we need manual intervention. While few of these are basic field validation review tasks, others require a more complex system to support the human validator in making the right decisions related to the task at hand.

This private preview is centered around a new approach that will allow you to address such complex validation scenarios by leveraging the combined capabilities of Validation Station (Document Understanding), Apps, Action Center and Studio.





Public Preview Feature List

List of new features released as part of Public Preview (Geo wise deployments from 24th Feb till 3rd March)

1. Set field value and multi-value properties - confidence, confirm, keepReference
2. Set Table cell value & properties
3. Delete value and table cell value - DeleteValue property
4. New Style property - Fields panel position
5. Bug fixes, performance improvements
6. Related - dependent Apps in Studio has gone GA with revamped documentation to help users

Forward Demo and Walkthrough of the feature

Please refer to the video below for understanding the feature value proposition and demo. This is from Forward 7

https://vimeo.com/1024045825/b36c67a9c0

Accessing updated Public Preview Documentation, Starter Kit and Updates

All information to be made available in Insider Portal including documentation. **Official UiPath Documentation portal updates to be made available with GA only**

https://insider.uipath.com/#/portal/382

For feedback, suggestions or issues, please use the insider portal for reporting.

Limitations

- Automation Cloud only
- Web App Project support in Studio Web only [Please do not try to create Validation Apps in VB Apps]
- Document Validation Task support only (no support for document classification in this preview)
- Action Center based Action Apps only, and not standalone Apps (to be added later)
- No Public Apps support
- Studio Web Template support for Web App Project to be available later

Prerequisites

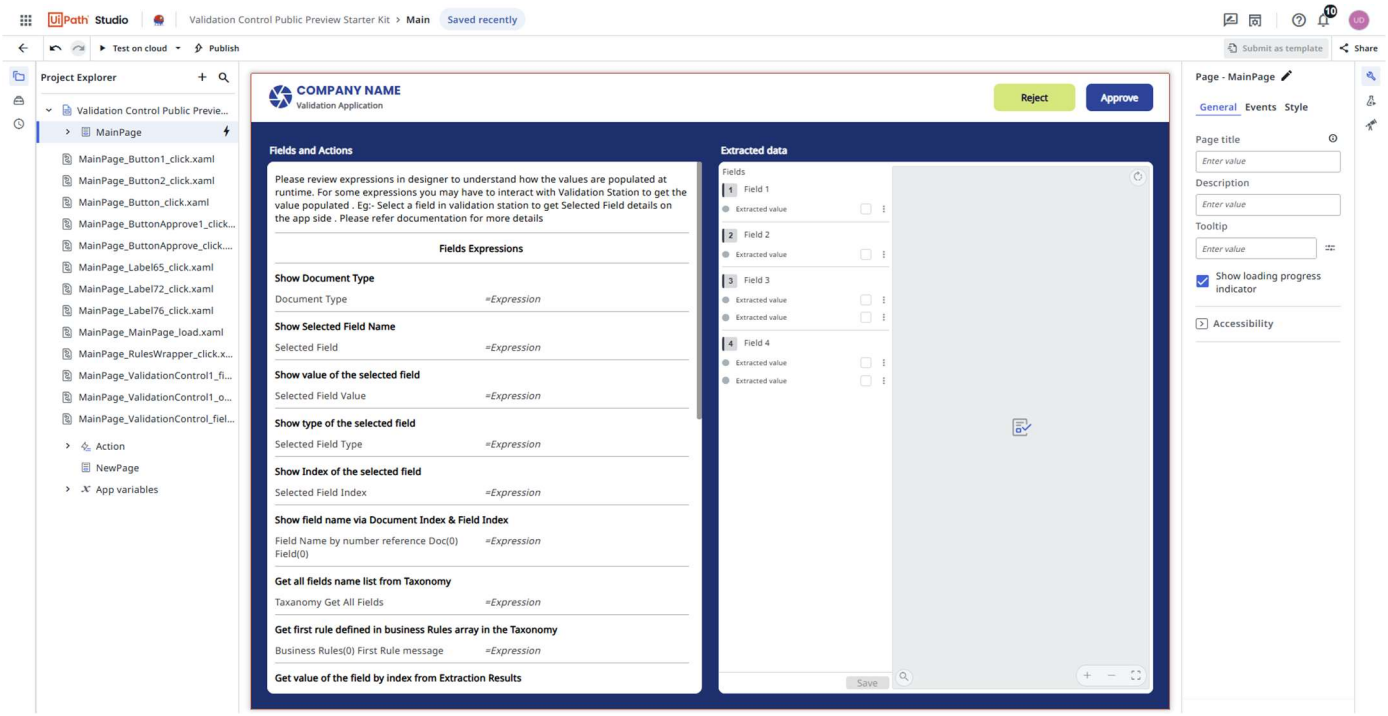
To get started with building your first Validation App, you would need the following applications

- Automation Cloud
- Studio Desktop
    - For building DU automations with suspend/resume App Tasks using the IntelligentOCR activity package in Windows projects (Studio Web/Cross-platform automation support will be available for Public Preview)
- Studio Web
    - Validation Control is currently available as part of the new Web App Project type in Studio Web
- Action Center
- Document Understanding
- Communications Mining (optional)
- Licenses – enough Robot Units for Serverless and AI units for DU extractor and OCR

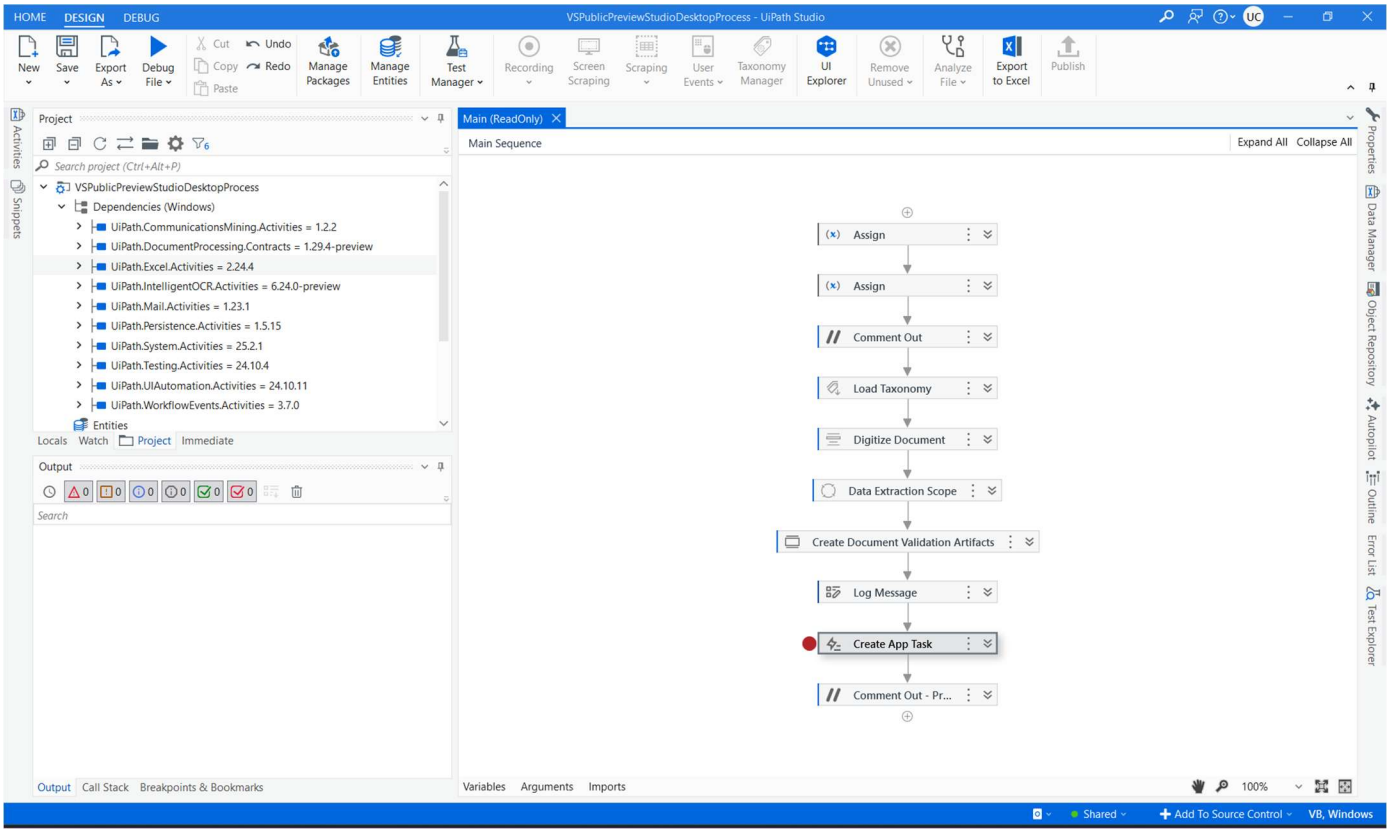
Starter Kit / Assisting Material

We will be providing a starter kit which includes the following assets to get you started quickly as starting from scratch can have a steeper learning curve

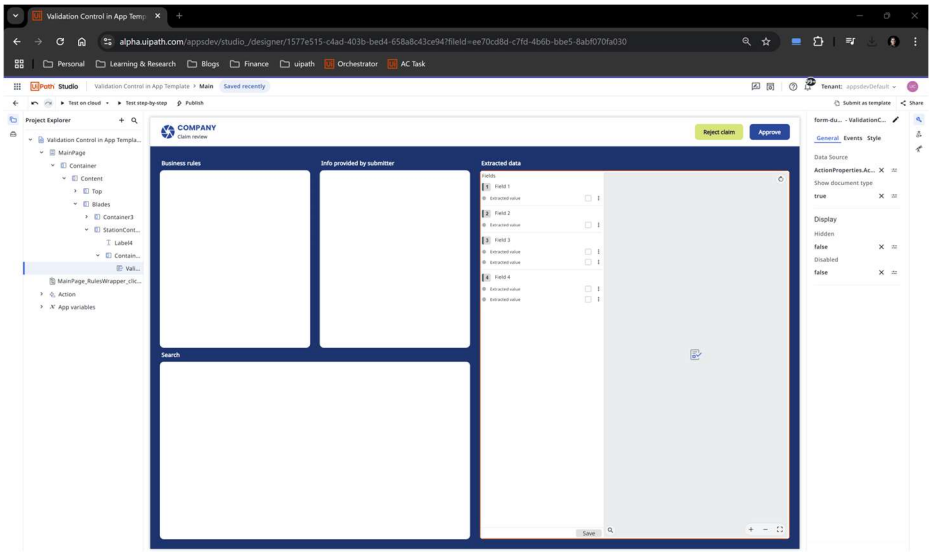
1) Sample Web App project containing a Validation Control and an Action type that is usable as a base for Action Center tasks **(Validation Control Public Preview Starter Kit.uip)**



2) Sample Studio Desktop Process to showcase how to create the required artifacts, create an app task and then consume it (**VSPublicPreviewStudioDesktopProcess**)



3) Step by step guide below on how to deploy the web app project, configure and run the studio process (as subset of this documentation)
4) Activity packages that are in private preview and that will enable the new features of Validation Control in Apps (as part of the process).
5) Studio Web Template as importable project named – **"Validation Control in App Studio Web Template.uip"**



Demo Scenario and Workflow

Demo Scenario

For non-straight through processing, we need Human in the Loop validations and for complex scenarios we can use Action Apps embedding validation control. Below is a health insurance review scenario for CMS 1500 Document submission and human review

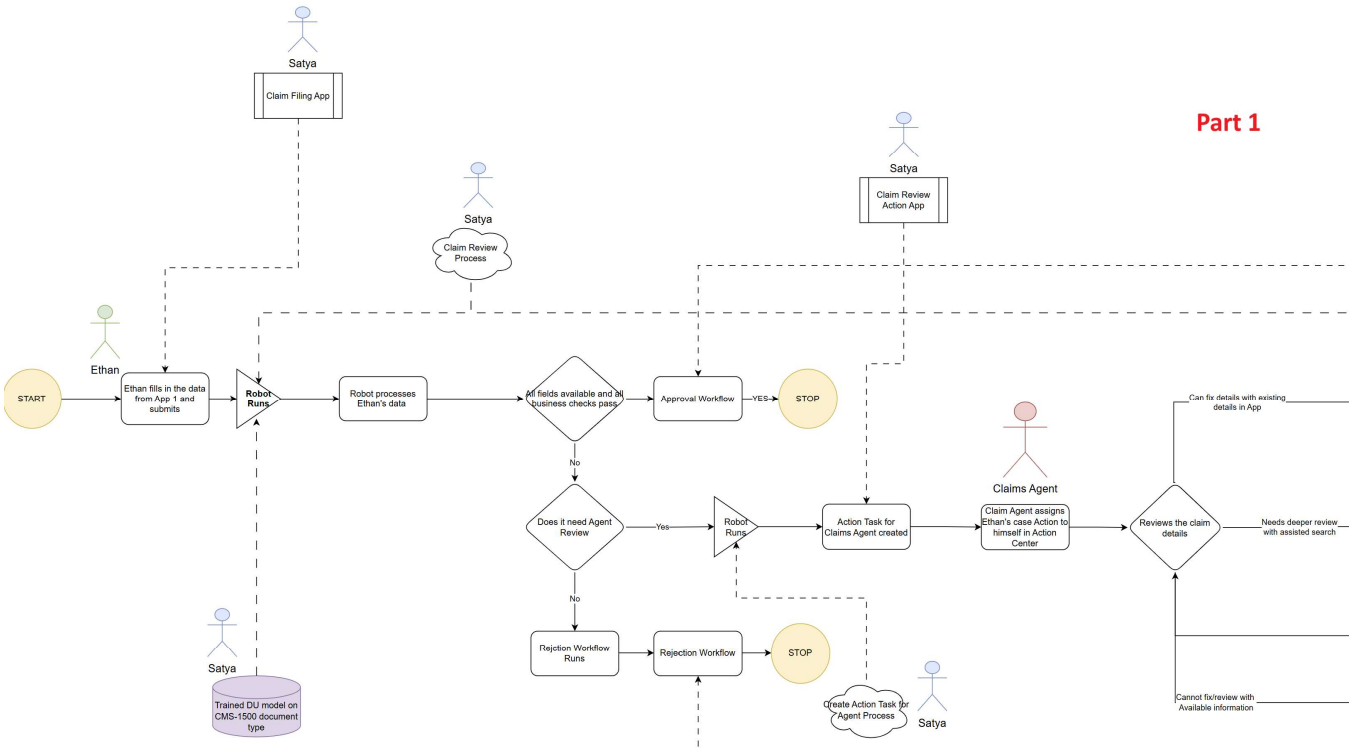
Developer Workflow

Create and Deploy Action App-> Create Studio Desktop Process

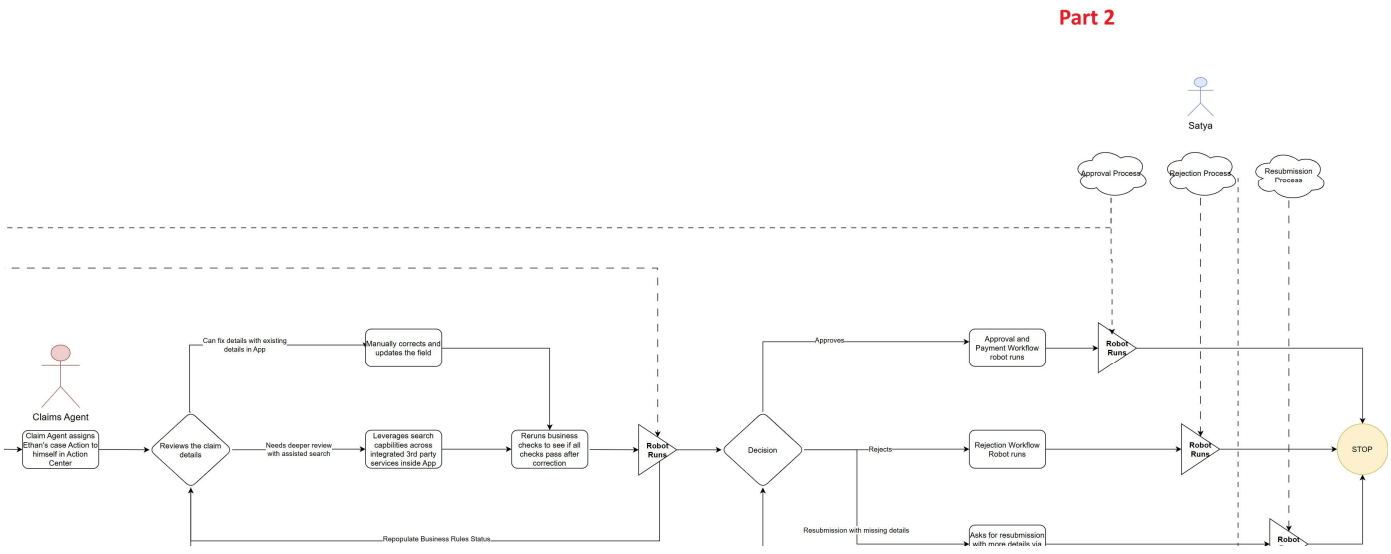
Scenario based Workflow (THIS WAS FOR PRIVATE PREVIEW – FOR PUBLIC PREVIEW WE USE INVOICE PROCESSING SCENARIO)

The workflow below shows how Ethan who submitted his CMS1500 form is flagged for human in the loop validation due to name mismatch or any other claim issue and Claims Agent reviews the same in Action Center. Post review it goes for final submission. Satya is the developer who builds all the components in UiPath. We are going to show partially the Agent App and Studio Process (Happy Path) from the complete scenario

Custom Validation Station Scenario



Station Scenario



Environment Setup

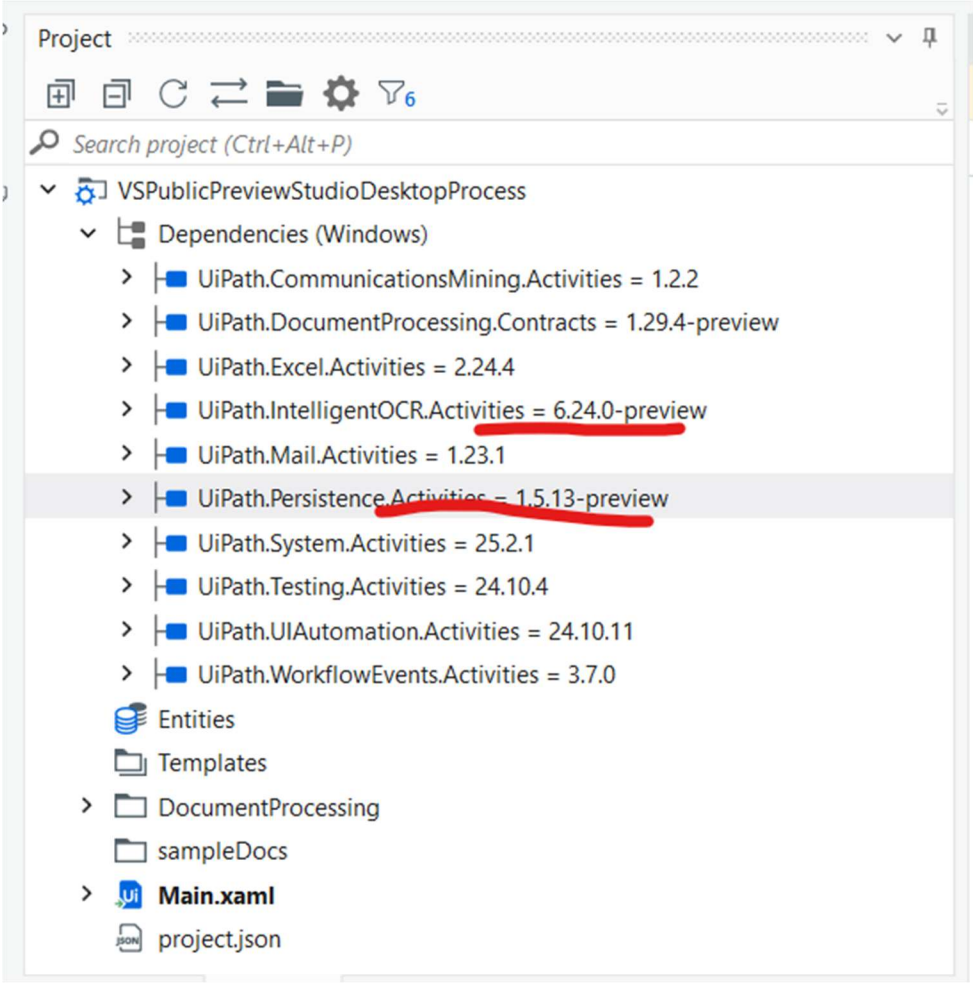
Assistant Setup

Please make sure you have the Assistant setup and connected to the right organization, tenant and folder.

Studio Desktop Setup

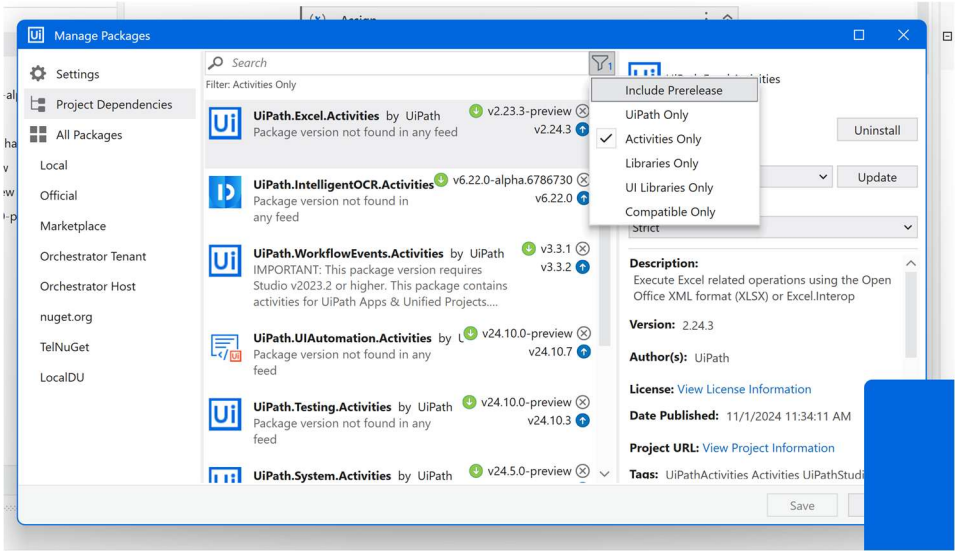
Make sure you have Studio Desktop installed and that you have access to the following packages:

- UiPath.IntelligentOCR.Activities -preview-6.23.4.-preview
- UiPath.CommunicationsMining.Activities
- UiPath.Persistence.Activities 1.5.13-preview
- UiPath.WorkflowEvents.Activities
- UiPath.DocumentProcessing.Contracts

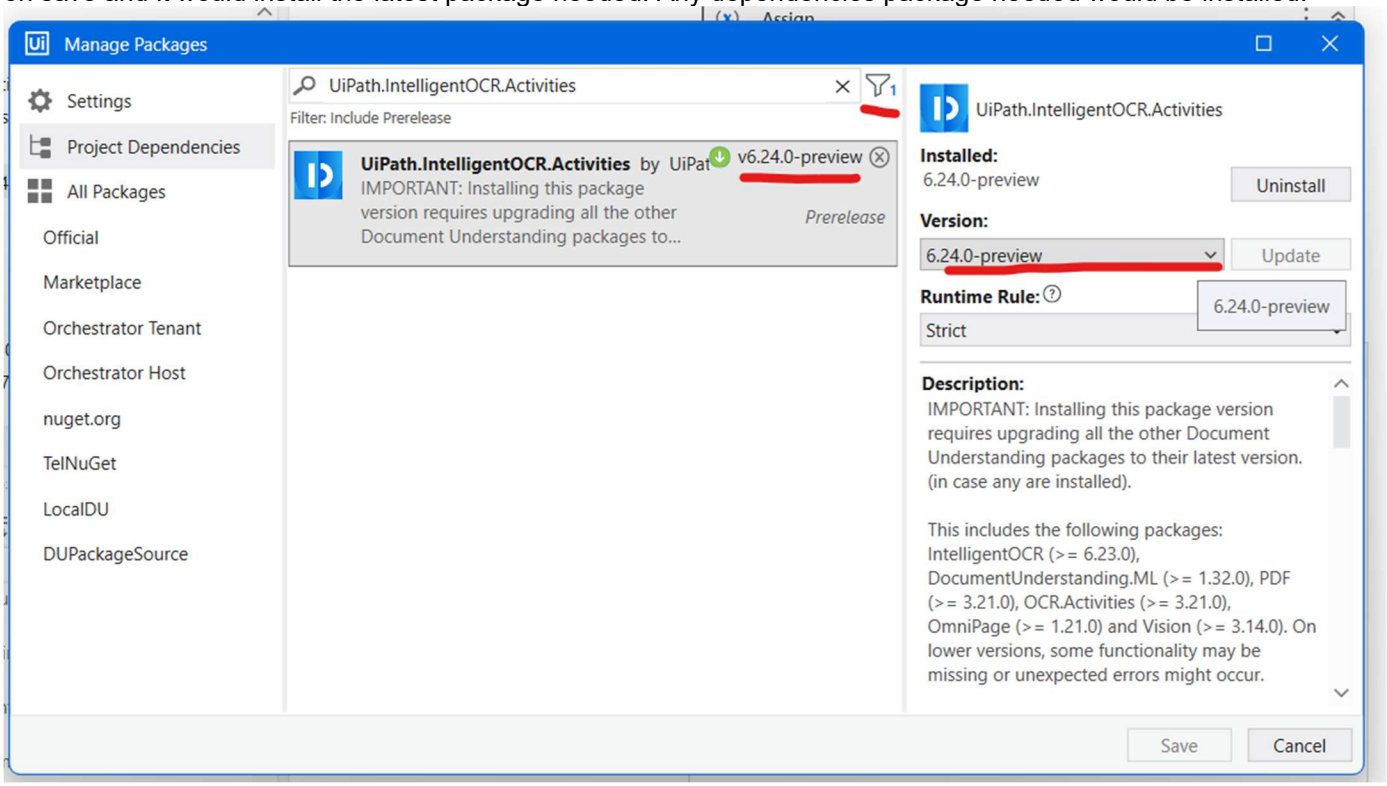


Customers need to add Intelligent OCR, Comms Mining and Persistence Activities package from the starter kit following the steps below. Also enable Prerelease activities setting in Studio Desktop. Some of them are released as updated preview version, so please use the same

1) Enable Prerelease option

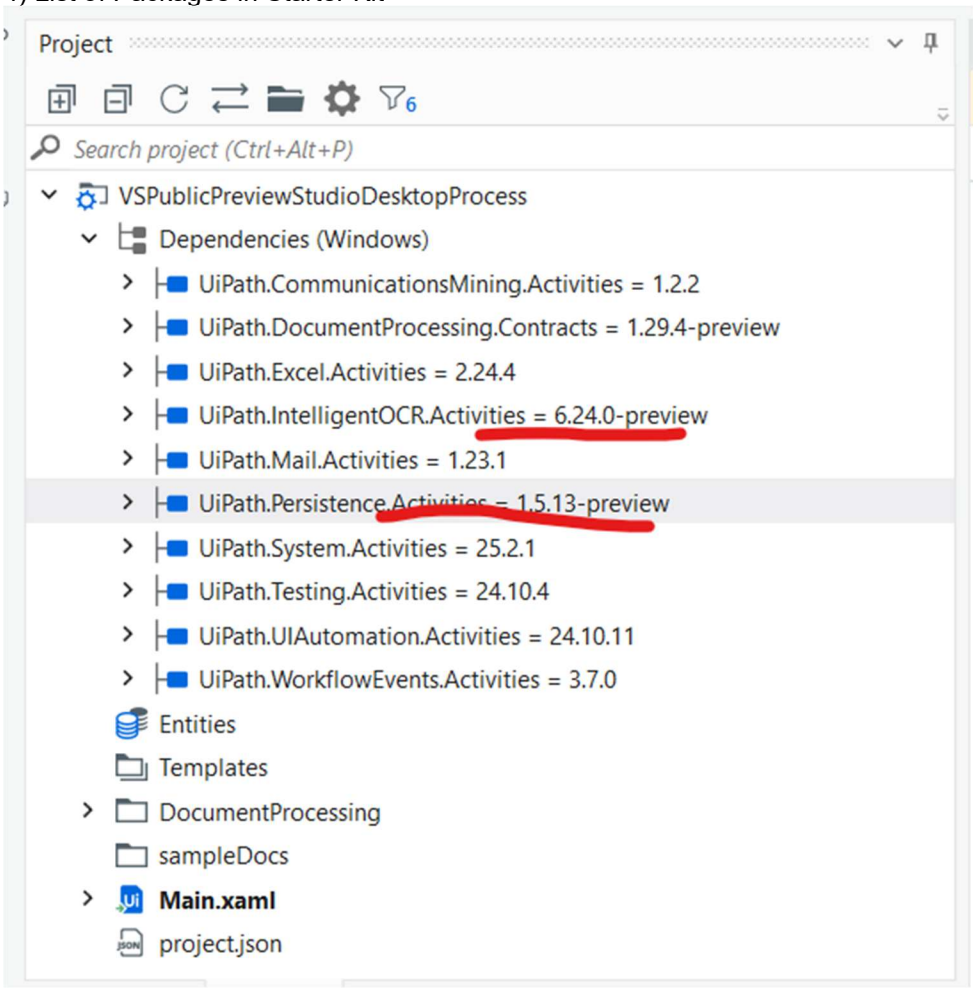


2) Select the package UiPath.IntelligentOCR.Activities v6.2.4.0-preview from All Packages if not already added, click on save and it would install the latest package needed. Any dependencies package needed would be installed.



3) Similarly upgrade UiPath.persistance Activities to v1.5.13-preview

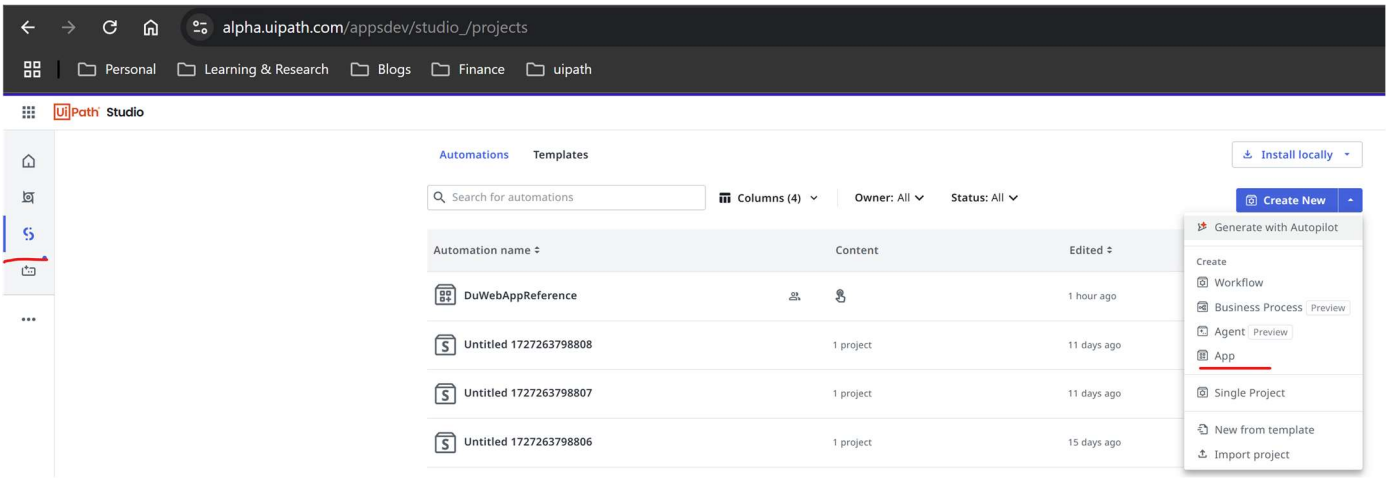
4) List of Packages in Starter Kit



Studio Web – Web App Project Setup

Check the following Studio Web prerequisites here
https://docs.uipath.com/studio-web/automation-cloud/latest/user-guide/prerequisites-for-using-studio-web

Check that you have access to Studio Web and that you can create a new " App Project " from the dropdown next to the "New Project" option.
If you do not see this option, please reach out to udit.chandna@uipath.com to check your org and tenant enablement for this feature.



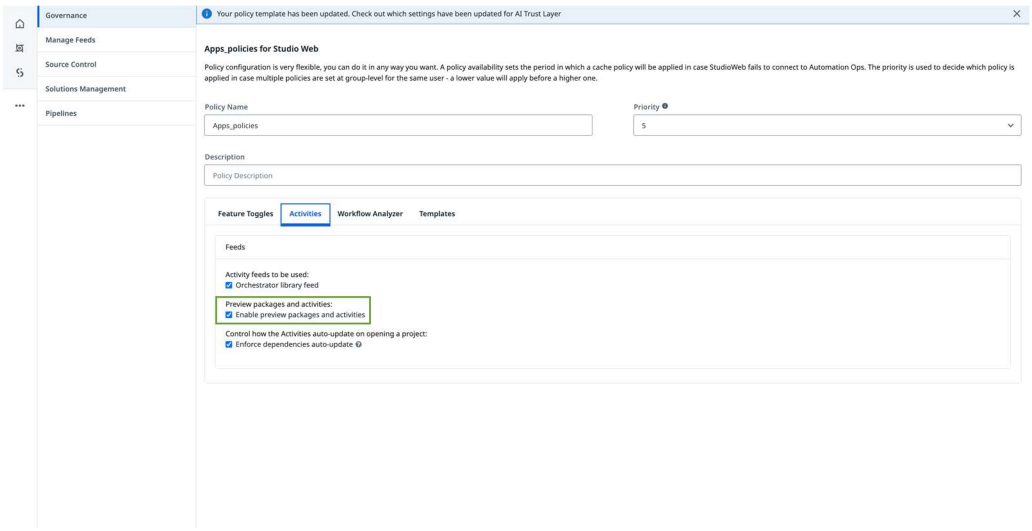
Also make sure that the UiPath Activities package latest version is uploaded in your orchestrator feed – this package contains some activities that allow you to save the state of a given validation control or set the focus on a particular field.

Customers need to enable Preview activities in Governance Policies as per the below instructions: (NO LONGER NEEDED FOR PUBLIC PREVIEW as APPS IN STUDIO HAS GONE GA)

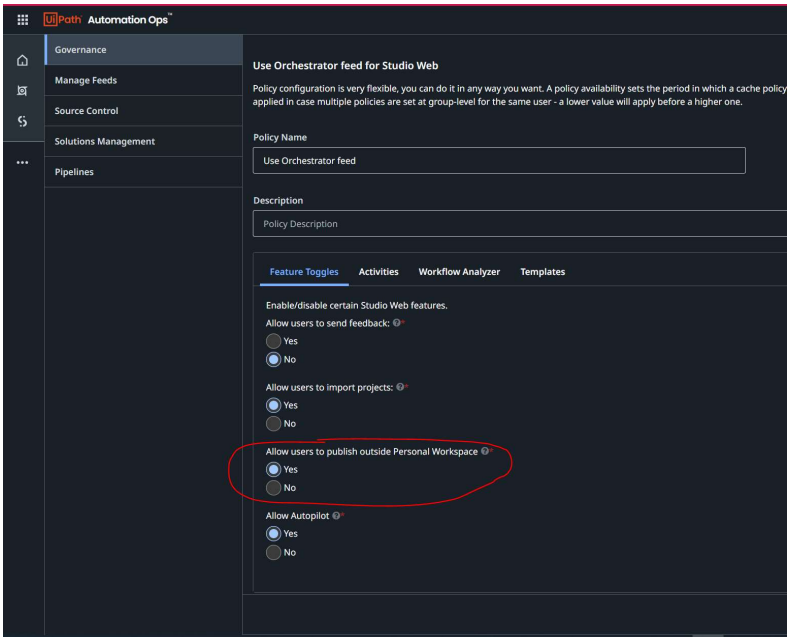
> For using Apps activities in Studio Web, users need to enable this governance policy for Studio Web: **Preview packages and activities.**
> This can be enabled from Automation Ops; the exact steps can be found here:

https://docs.uipath.com/automation-ops/automation-cloud/latest/user-guide/define-governance-policies
https://docs.uipath.com/automation-ops/automation-cloud/latest/user-guide/settings-for-studio-web-policies
https://docs.uipath.com/automation-ops/automation-cloud/latest/user-guide/deploy-governance-policies

> The following setting should be enabled in your tenant's policy:



For publishing outside your personal space, please enable the following setting (NEEDED)



Note: - If you don't see Validation Station Activities, this step was missed or done incorrectly

Document Understanding Setup

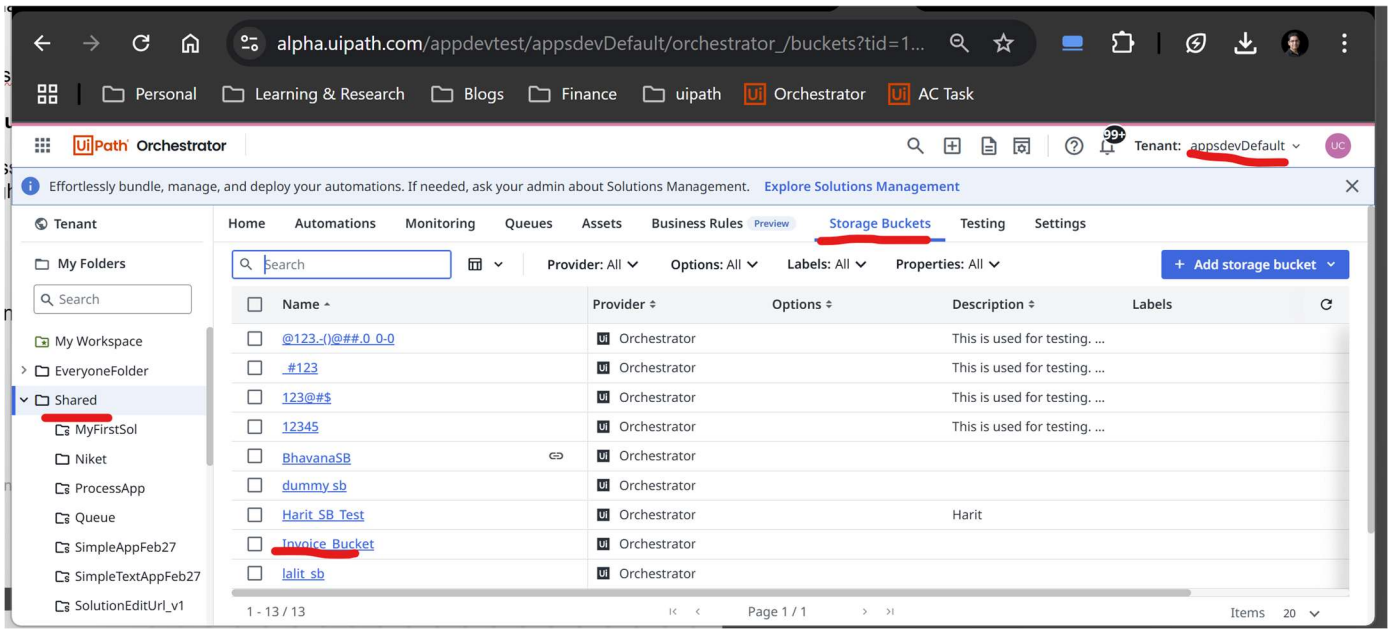
Check that you do have AI Units available so that you can build a simple document processing automation and run it on the files of your choice.

Comms Mining Setup (optional)

In case you are also using Communications Mining and want to try using the Validation Control in Apps with CM stream results, make sure you have the right permissions.

Orchestrator and Storage Bucket Setup

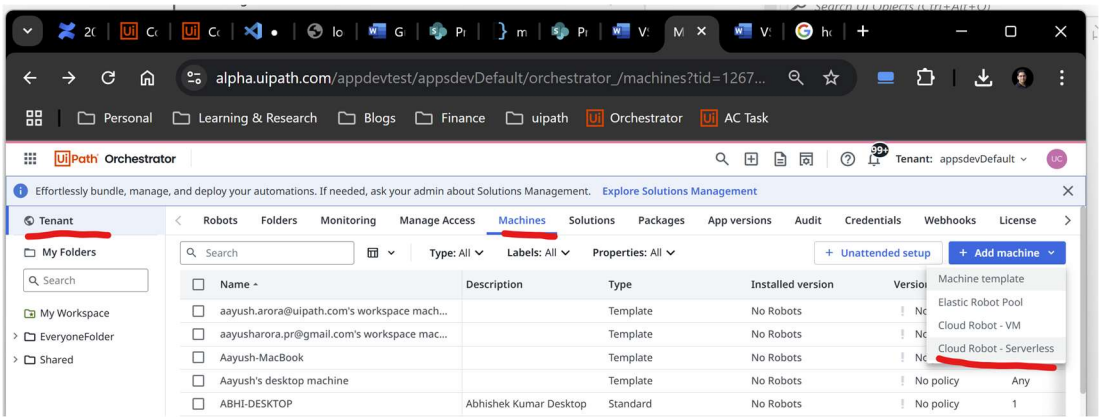
Data that is required for the Validation Control to load is stored in storage buckets. As a prerequisite to building your own custom validation app, you will need to create a storage bucket (from Orchestrator) in the right tenant, under the Shared folder of that tenant. You will be using this storage bucket to store the data required for validation controls in app tasks.



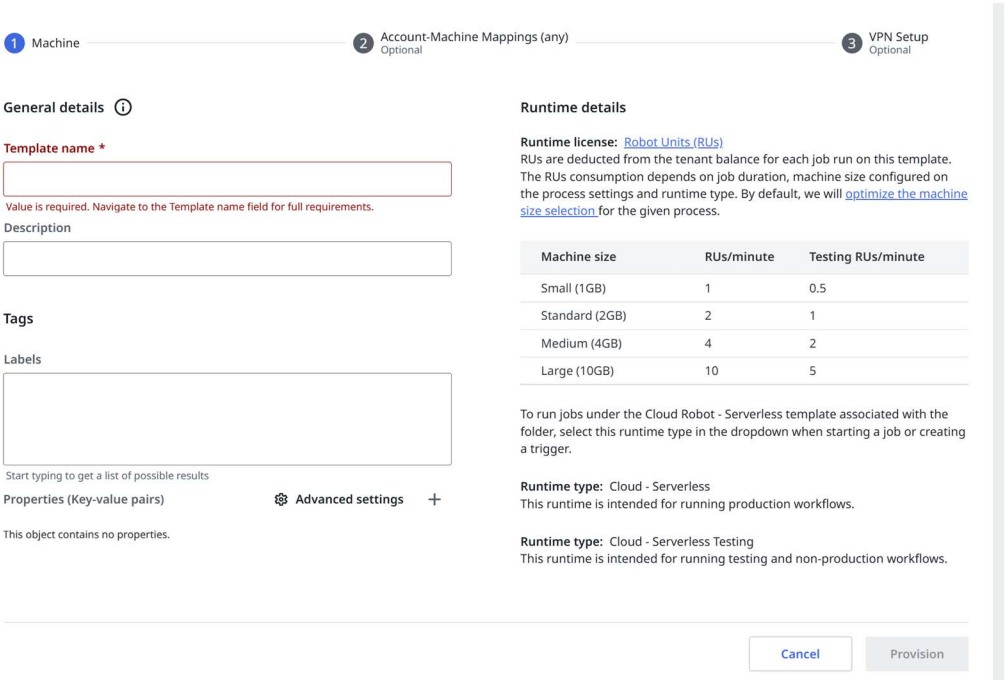
Serverless Robot template and machine Setup

For web app projects to run, we need to make sure we have Serverless Cloud Robot machine setup (if not available already). You might need Admin access to perform the following steps:

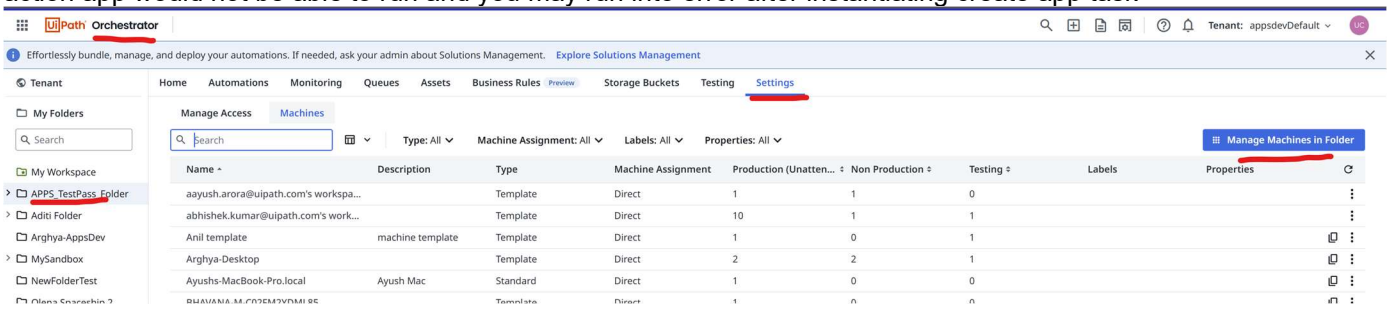
- Select Orchestrator
- Select Tenant
- Select Machines
- Select Add machine dropdown and then Serverless – Cloud Robot



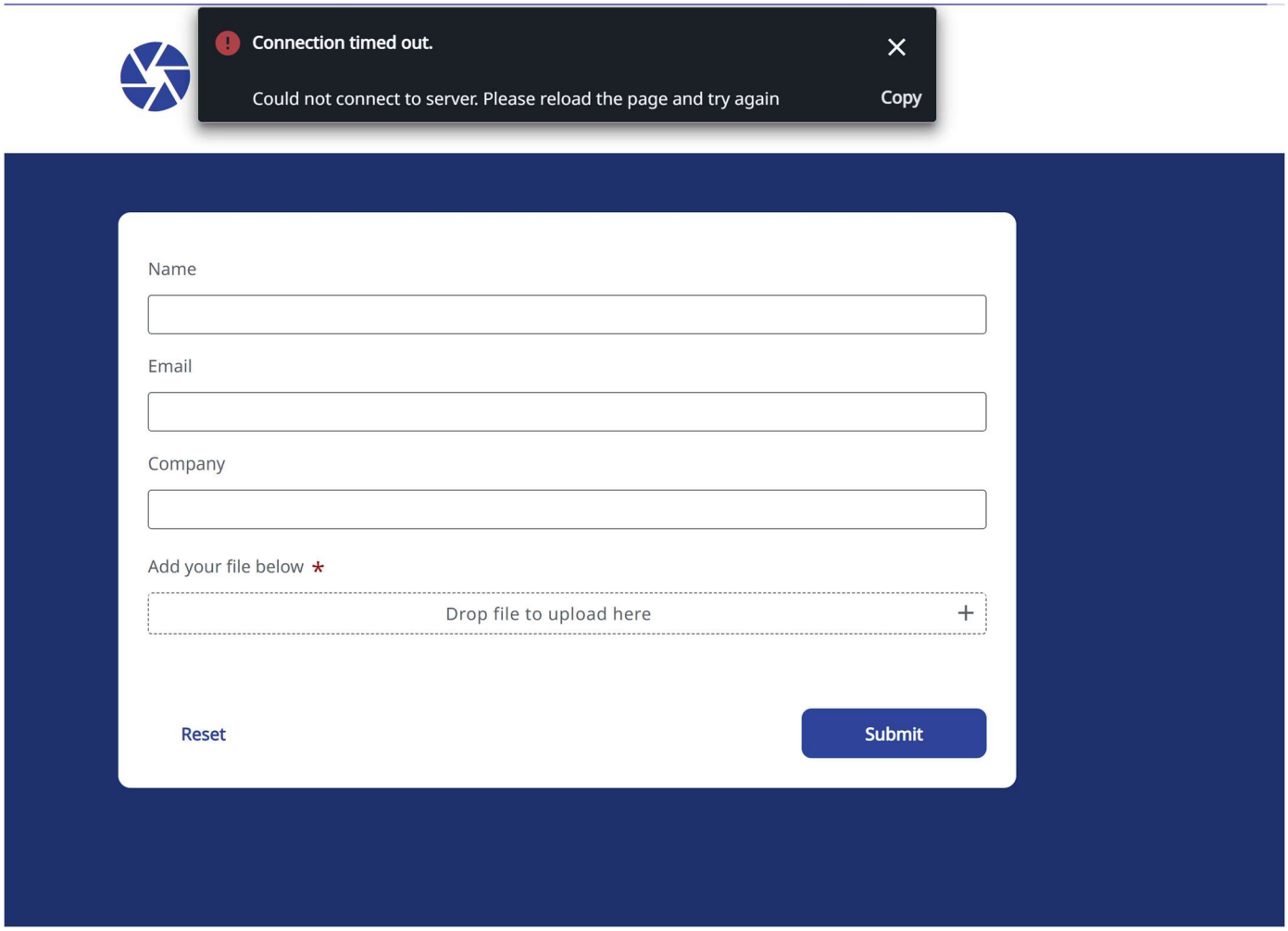
- Provide Template details below and click on Provision



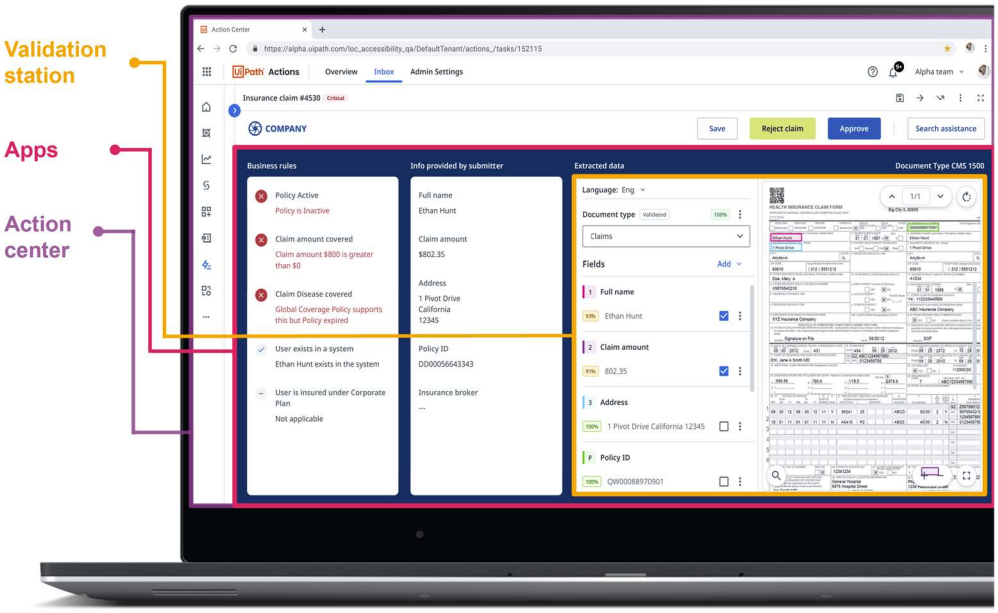
- Important Note : - If you plan to deploy it to specific folder please also add the machine to that folder, else action app would not be able to run and you may run into error after instantiating create app task



In case you do not follow the above step, you may see the below error



Components of Validation App



This private preview focuses on using the Document Validation functionality (as you can experience it directly in Action Center when you create a Document Validation task), as a component available for use within Web App projects. The component is named "Validation Control". Web App projects can then be used to build App Tasks (human tasks that use the user interface and interactions defined within your web app project). This way, you have full flexibility in creating Action Center tasks, with all the power of task management offered by Action Center as well.

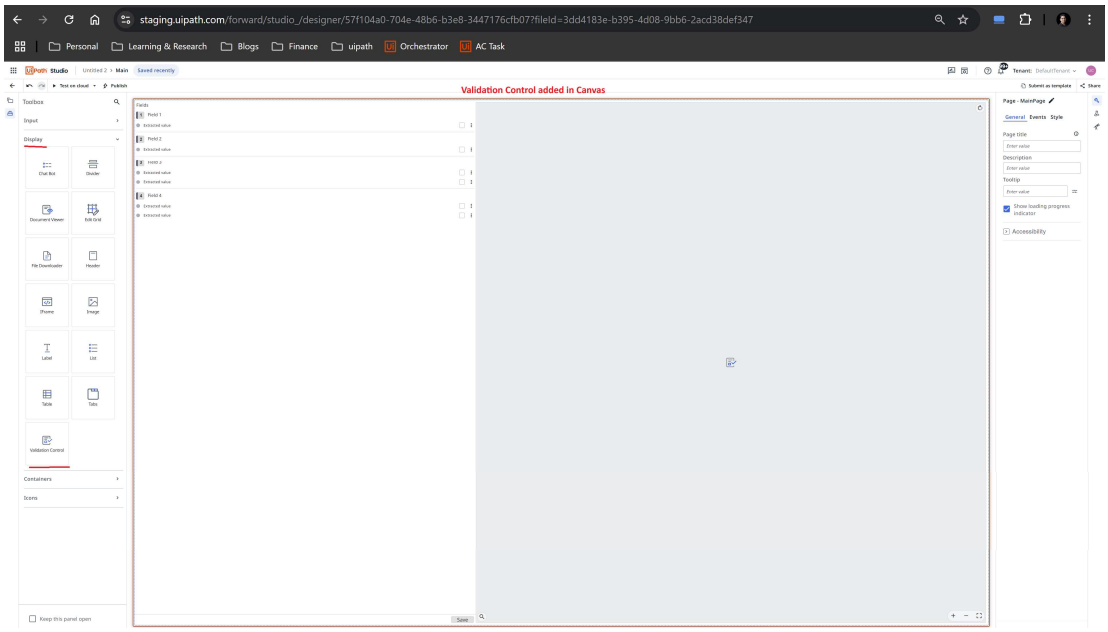
We will now detail how you can build your first end-to-end experience using the new features.

Learning about Apps in Studio

Studio in Apps has gone GA and refer available documentation to learn more on working with other App components https://docs.uipath.com/studio-web/automation-cloud/latest/user-guide/apps-in-studio-web

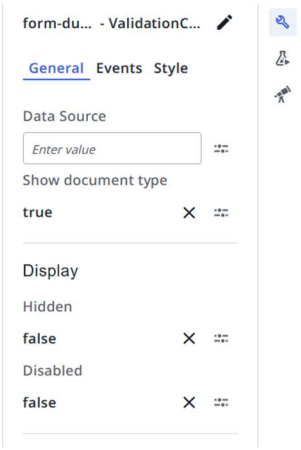
Validation Control Details

Validation Control can be added to the page by going to Toolbox, selecting Display Controls and then dragging Validation Control to the Canvas



Validation Control in Studio Web

General Properties



General Properties

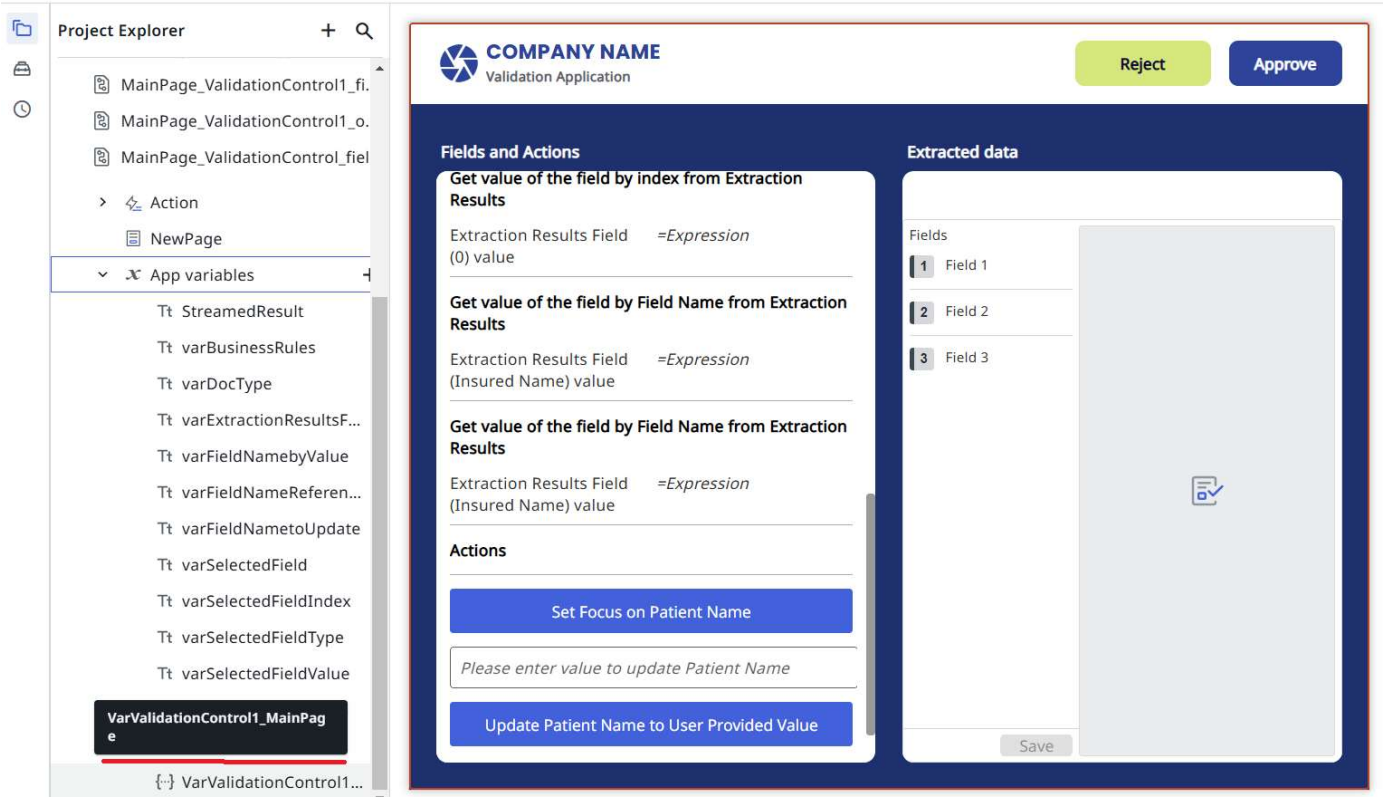
Property	Data Type	Description
Data Source	ContentValidationData	This is to specify which content is to be loaded in the Validation Control. The object required as input in this property can be generated using the “Create Document Validation Artifacts” activity and is passed as input to the Action Schema of the Web App, using the Create App Task activity. The action schema ContentValidationData input can be bound directly to this property.
Show Document Type	Boolean	This allows the configuration of whether the Document Type is visible or hidden. When

		visible, users of the app can change the document type, while when hidden, fields will be directly in focus.
Disabled	Boolean	This property controls whether the control is read only or editable.
Hidden	Boolean	This property allows you to show or hide the validation control.

Fields Position details mentioned in Styling properties

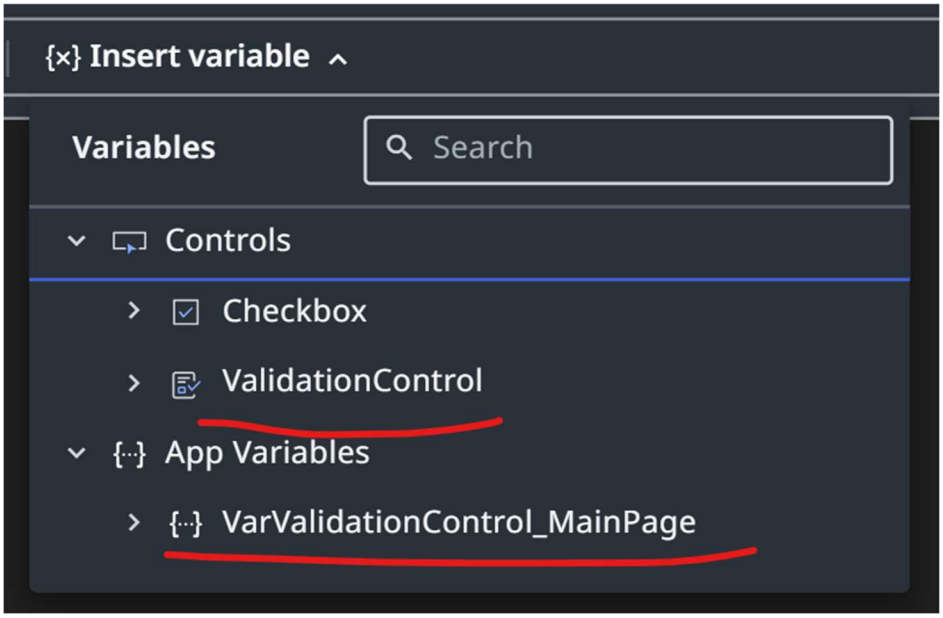
Accessing and Interacting with the Validation Control in Apps and Robot / VB Properties

Whenever a Validation Control is added to the canvas, you will notice a variable bound to that control is automatically created under App Variables, its name syntax being "Var" + "<Validation Control Name>"+"<PageName>" . When you rename a Validation Control, the associated variable will also be renamed, and when a Validation Control is deleted, the associated variable is also deleted. You ~~cant~~can't delete this variable as it's a restricted variable.



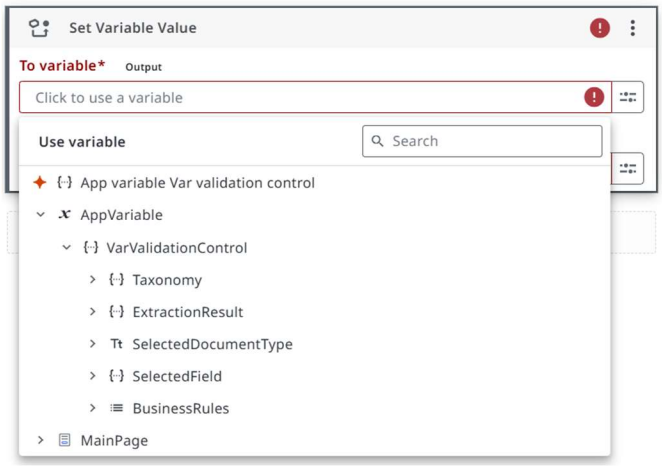
VarValidationControl Main Page variable created in App variables section

Important: To READ information from the Validation Control, you can use both the Control itself as well as the App Variable created that is attached to the control. To WRITE information to the Validation Control (overwrite values, manipulate the extraction results), you will need to use the App Variable attached to the Validation Control.



Variables to work with for DU Control

Accessing Control Data via the Attached App Variable



VarValidationControl properties exposed for consumption

Property	Data Type	Description
Taxonomy	UiPath.DocumentProcessing.Contracts.Taxonomy.DocumentTaxonomy	The taxonomy object with which the content was prepared for display. It contains information about document types, fields, display properties, etc. Refer to Class & properties here
Taxonomy Properties and Methods		

Methods

- `Serialize()` - Returns a string with the serialized contents of the `DocumentTaxonomy` object.
- `Deserialize(string SerializedTaxonomy)` - A static method that returns a new instance of the `DocumentTaxonomy` class containing the information serialized in the input string.
- `GetFields(string DocumentTypeId)` - Returns a list of `UiPath.DocumentProcessing.Contracts.Taxonomy.Fields` objects containing the field definitions associated with the document type identified by using the `DocumentTypeI`

Properties

- `DataContractVersion string` - The package version with which the `DocumentTaxonomy` object was generated. The default value is `1.0`.
- `DocumentTypes` - A list of defined `DocumentType` objects as found in the Taxonomy file referenced in `DocumentTaxonomy`.
- `Groups` - A list of `DocumentGroup` objects defined in the `DocumentTaxonomy`.
- `SupportedLanguages` - A list of languages supported by the `DocumentTaxonomy`.

ExtractionResult	UiPath.DocumentProcessing.Contracts.Results.ExtractionResult	The extraction results, as they are (with or without human changes), from the Validation Control. Refer to Class & properties here

Extraction Result Properties and Methods

Methods

- `AsClassificationResult()` - Returns a `UiPath.DocumentProcessing.Contracts.Results.ClassificationResult` generated using the information from the `ExtractionResult` object.
- `AsDataSet(<bool>)` - Returns a `System.Data.DataSet` containing multiple `System.Data.DataTable` in the `.Tables` collection. This collection contains a table called **Simple Fields**, that contains all fields as columns and all reported values for a field as values in the table. If a field has no value in the `ExtractionResult`, then no value is reported. If a table field has a value, it is reported as the string `table`, which is a marker of the fact that there is a `DataTable` with the field name that contains column fields as columns, and values for each row as values in the `DataTable`.
- `AsDataSet(<bool>, <bool>)` - Returns a `System.Data.DataSet` containing multiple `System.Data.DataTable` in the `.Tables` collection. This collection contains a table called **Simple Fields**, that contains all fields as columns and all reported values for a field as values in the table. If a field has no value in the `ExtractionResult`, then no value is reported. If a table field has a value, it is reported as the string `table`, which is a marker of the fact that there is a `DataTable` with the field name that contains column fields as columns, and values for each row as values in the `DataTable`.
- `Deserialize(<string>)` - A static method that returns a new instance of the `ExtractionResult` class with the contents serialized in the input string.
- `EnumerateValues(System.Func<UiPath.DocumentProcessing.Contracts.Results.ResultsValue, <bool>)` - Allows you to iterate over all values in the results document, including table cell values.
- `FlattenFields(System.Func<UiPath.DocumentProcessing.Contracts.Results.ResultsDataPoint, <bool>)` - Flattens the fields in `UiPath.DocumentProcessing.Contracts.Results.ExtractionResult.ResultsDocument` and filters them based on the condition.

  WARNING:
  This method is obsolete.

- `GetDocumentType()` - Returns a string containing the `DocumentTypeId` of the `ExtractionResult` object.
- `GetFields(System.Func<UiPath.DocumentProcessing.Contracts.Results.ResultsDataPoint, <bool>)` - Filters the fields in `UiPath.DocumentProcessing.Contracts.Results.ExtractionResult.ResultsDocument` that satisfy the condition.
- `GetField(<string>)` - Returns the field with the specified field ID or `null` if it doesn't exist.
- `GetSimpleFieldValues(<string>)` - Returns a list of UiPath.DocumentProcessing.Contracts.Results.SimpleFieldValue for the input `fieldId`.
- `GetTable(<string>)` - Returns the table with the specified id or null if it doesn't exist.
- `GetTableFieldValue(<string>, <int>)` - Returns a UiPath.DocumentProcessing.Contracts.Results.TableFieldValue containing cell values for the required `fieldId`. By default, this field is set to 0.
- `IsAnyFieldOrTableCellMissing()` **bool** - Determines whether there is any data point or table cell which doesn't have values.

  NOTE:
  Only a single value for a table field is supported. The `valueIndex` can therefore only be set to 0.

- `GetValueCount(<string>)` - Returns an integer containing the number of values associated with the provided `fieldId`.
- `Serialize()` **string**. Returns a string containing the serialized representation of the `ExtractionResult` object.
- `GetFieldValidatorNotes(<fieldId>)` - Returns the field value containing validator notes with the specified field id, or null if it doesn't exist.
- `SetFieldValidatorNotes(<fieldId>, <validatorNote>)` - Set the field value for validator notes with the specified field ID, if notes are activated for that field.

Properties

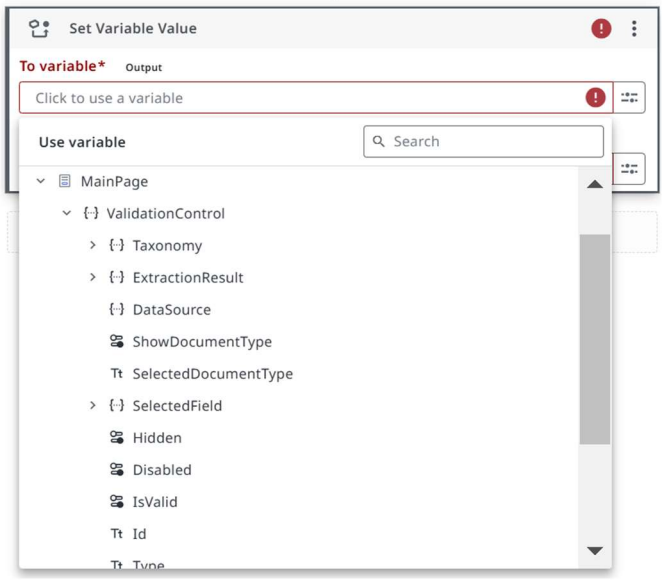
- `DocumentId` **String** - The name of the document for which the extraction results are reported.
- `ExtractorPayloads` **String** - Additional information provided by extractors.
- `ResultsDocument` **UiPath.DocumentProcessing.Contracts.Results.ResultsDocument** - The details of the classification and data extraction structure.
- `ResultsVersion` **Int** - The version of the `ExtractionResult` reported for the document. Starts at 0 when automatic extraction is performed and should be increased by one each time the contents of the object are changed, for tracking purposes.

Field	UiPath.DocumentProcessing.Contracts.Taxonomy.Field	Refer to Class & properties here
SelectedDocumentType	System.String	The name of the document type currently selected in the Validation Control.
SelectedField	UiPath.DocumentProcessing.Contracts.Apps.FieldValueDetails	The field currently selected in the Validation control along with its value.
BusinessRules	UiPath.DocumentProcessing.Contracts.Apps.EvaluatedBusinessRulesForFieldValue [] (array)	The list of all business rules defined for the document type in the Validation Control, organized by field, and the state of each (passed or failed).
Table Field Class		
Table Field	UiPath.DocumentProcessing.Contracts.Results	Refer to Class & properties here

Please Note: Syntax for Property assignment vs referencing in Activity may be different. Please refer example app for same

For Document Understanding Contract updated class details – please refer the below documentation section **https://docs.uipath.com/activities/other/latest/document-understanding/document-processing-contracts-classes**

2) Accessing Control Data and its Properties via the Control Variable (page scoped)



Property	Data Type	Description
Taxonomy	UiPath.DocumentProcessing.Contracts.Taxonomy.DocumentTaxonomy	The taxonomy object with which the content was prepared for display. It contains information about document types, fields, display properties, etc. Refer to Class & properties here Please refer above section for details
ExtractionResult	UiPath.DocumentProcessing.Contracts.Results.ExtractionResult	The extraction results, as they are (with or without human changes), from the Validation Control. Refer to Class & properties here

		Please refer above section for details
SelectedDocumentType	System.String	The name of the document type currently selected in the Validation Control.
SelectedField	UiPath.DocumentProcessing.Contracts.Apps.FieldValueDetails	The field currently selected in the Validation control along with its value.
DataSource	UiPath.DocumentProcessing.Contracts.Actions.ContentValidationData	The content validation data that is currently the source of the validation control.
ShowDocumentType	boolean	The state of the control property for showing or hiding the document type section of the validation control.
Hidden	boolean	The state of the control property governing whether the control is visible or hidden
Disabled	boolean	The state of the control property governing whether the control is editable or not
IsValid	boolean	This property is used when you want to set status of control as invalid (not related to document understanding validations or business rules)

Editing Control Data via the Attached App Variable

You can change values from the Extraction Result displayed in the validation control, using the App Variable pertaining to that control.

To add or update a field value (for regular fields, not multi-value), you need to:

- Use a "Set Variable Value" activity in an automation that you trigger, and use:
  - AppVariable.VarValidationControl.Field("<field name>").Value (in the To Variable input)
  - The string that you want to set as value for the field identified by <field name> above (in the To input)

We are working on providing the same functionality to allow for table cell editing, confidence and confirmation flag editing (incoming in the following few weeks)

Editing Control Data and its Properties via the Control Variable (page scoped)

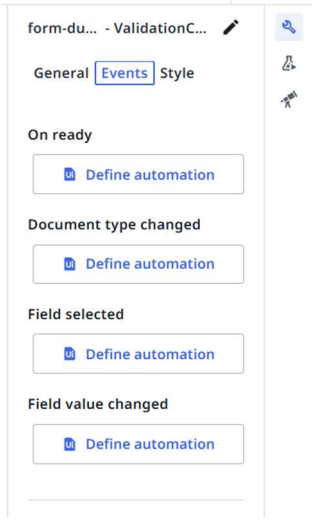
You can change some properties of the Validation Control using the Control variable itself. These are:

- DataSource
- Hidden
- Disabled
- ShowDocumentType

If you want to change any if these properties dynamically at run-time, you need to bind them to AppVariables:

- create an app variable that you want to use with the correct type
- Bind the newly created variable to the validation control property
- Change the value of the app variable at run-time through an automation that performs a "Set Variable Value" on the binded variable, to the new value you wish to have.

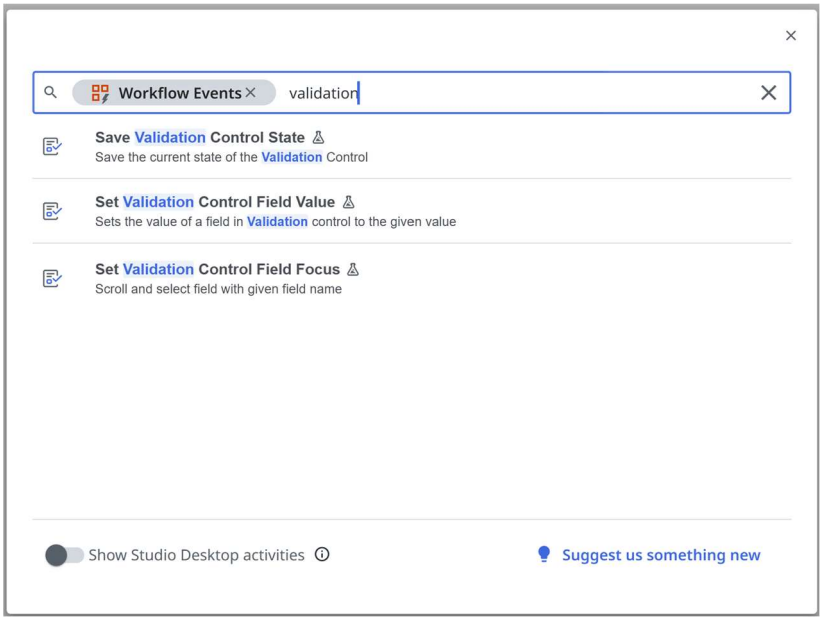
Validation Control Events



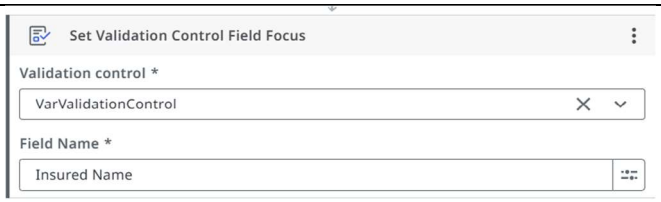
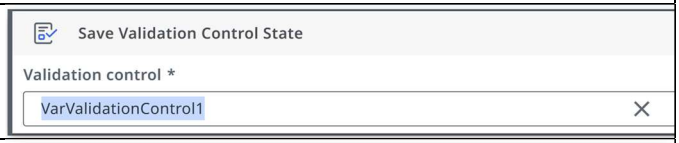
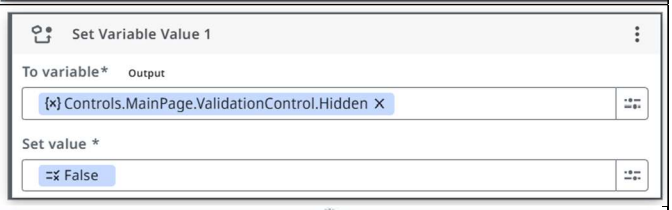
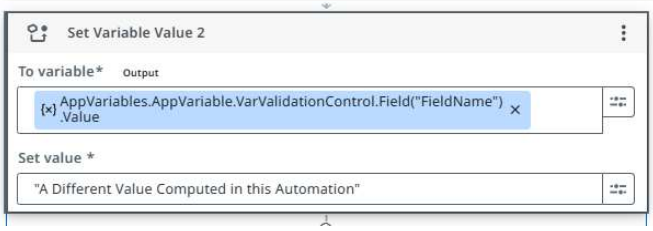
Events Section

Events	Description
On Ready	Triggered when the Validation Control is fully loaded (all data, including taxonomy, extraction results, displayed document are available and loaded)
Document Type Changed	Triggered when the user changes the document type from the Validation Control
Field Selected	Triggered when the user selects one of the fields or the value of a field in the validation control
Field Value Changed	Triggered when the user adds / removes / replaces a field value in the validation control

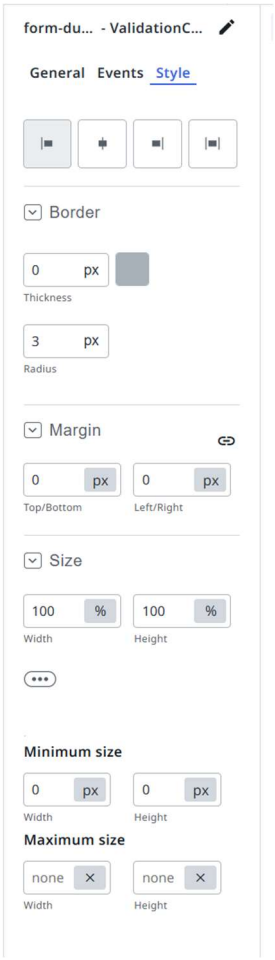
Validation Control Specific Activities



Validation Control Activity List

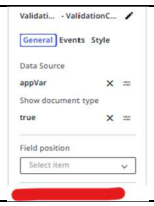
Activity	Description	Details
Set Validation Control Field Focus	Allow focusing on field based on event user want	 If you want to get Field Name dynamically from selected item in DropDown, then you can pass following in FieldName string.Format("{0}", Controls.MainPage.Dropdown.SelectedItem.FieldName) Before using above expression make sure to bind data source of dropdown as below VarValidationControl_MainPage.Taxonomy.GetFields(VarValidationControl_MainPage.SelectedDocumentType).ToListSource
Save Validation Control Sate	Save state of Validation Control in case you want to revisit and complete the validation later	
Set Variable Value (the generic activity)	Assign values to the Validation Control variable properties	
Set Variable Value (the generic activity)	Assign values to a Field within the validation control, using the attached auto-generated app variable **<Do not use this expression for reading value as for reading we get values as an array with separate confidence values>**	

Styling Properties



Styling properties

Property	Values/Choice Set	Description
Alignment	Left, Right, Center, Stretch	Aligning the control inside page/container
Border	Thickness – Numeric and Color, Radius – Numeric	Having Border and what size
Margin	Top – Numeric pixels Bottom – Numeric pixels Right – Numeric pixels Left – Numeric pixels	Having spacing margins (separate from padding)
Size	Width – Numeric/ Choice Set Values Height – Numeric / Choice Set Values Supports pixels, %, auto, fill Also option to set Minimum and Maximum size	Setting size of controls
Field Positions	Left/Right Choice Set	You can now decide if you want fields on left and document on right

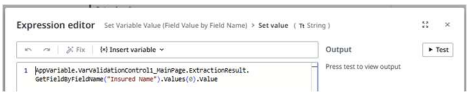
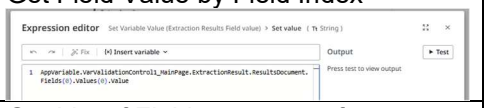
		or fields on right and document on left

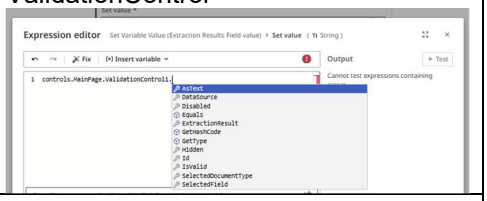
Note: Please refer to containers and other styling properties as well for better learning and also to learn more about styling. Also, we currently support design time styling only and not runtime styling

Common Expressions

Please refer to the table below for common expressions you can you in developing your web app project. Also please refer to the attached starter kit app that demonstrates the use of these expressions.

Note: Set value and Get Value do not use same epressions

Purpose	Syntax	Example Expression
Accessing ExtractionResults displayed in a Validation Control	AppVariable.VarValidationControl.ExtractionResult OR MainPage.ValidationControl.ExtractionResult	Get Field Value by Field Name (Remember values is an array as we get multiple probability based extracted outputs indexed by highest probability)  Get Field Value by Field Index 
Accessing Taxonomy loaded in a Validation Control	AppVariable.VarValidationControl.Taxonomy OR MainPage.ValidationControl.Taxonomy	Get List of Fields as source for dropdown control VarValidationControl_MainPage.Taxonomy.GetFields(VarValidationControl_MainPage.SelectedDocumentType).ToListSource
Accessing the selected document type in the validation control	AppVariable.VarValidationControl.SelectedDocumentType OR MainPage.ValidationControl.SelectedDocumentType	VarValidationControl_MainPage.SelectedDocumentType
Accessing the selected field in the validation control	AppVariable.VarValidationControl.SelectedField OR	Selected Field Name Selected Field's First value (Value with highest probablity)

	MainPage.ValidationControl.SelectedField The properties of the selected field: .Field.FieldName ; .Field.FieldType; .Field.ValidatorNotes etc All values of the selected field: .Field.Values Currently selected value within the selected field: .FieldValue.Value, .FieldValue.Confidence, .FieldValue.OperatorConfirmed, etc ; .FieldValueIndex (for multi-value fields – what index is the selected FieldValue pertaining to)	 Selected Field Data Type Selected Field ‘s Index
Accessing other validation control properties	Controls.MainPage.ValidationControl.<PROPERTY_NAME> Or MainPage.ValidationControl.<PROPERTY_NAME>	List of Properties of the ValidationControl 
Methods available for reading data from the ExtractionResult	(please refer to the documentation here for details: https://docs.uipath.com/activities/other/latest/document-understanding/extraction-result-class) .GetDocumentType() .GetField() .GetFieldByFieldName() .GetFields() .GetFieldValidatorNotes() .GetFieldValidatorNotesByFieldName() .GetSimpleFieldValues() .GetSimpleFieldValuesByFieldName() .GetTable() .GetTableByTableName() .GetTableFieldValue() .GetTableFieldValueByTableName() .GetValueCount() .GetValueCountByFieldName() ... and more to come!	
Methods available for reading data from the Taxonomy	(please refer to the documentation here for details: https://docs.uipath.com/activities/other/latest/document-understanding/document-taxonomy-class)	

	.GetFields()	
Accessing a field value with the intent of adding/overwriting the validation control current value	AppVariable.VarValidationStation_MainPage.Field(<FIELDNAME>).Value(<INDEX>)	AppVariable.VarVSDoc.Field("Patient Name").Value(0)
Other useful expressions		
Getting the list of field names displayed in the validation control	AppVariable.VarValidationControl.ExtractionResult.GetFields().Select(Function(f) f.FieldName)	.AppVariable.VarValidationControl.ExtractionResult.GetFields().Select(Function(f) f.FieldName)
Table support		
Read field value from table	AppVariable.VarValidationControl_MainPage.ExtractionResult.GetTableFieldValueByTableName(<TABLENAME>)(<COLUMN INDEX>)(<ROWINDEX>).Value	VarValidationControl_MainPage.ExtractionResult.GetTableFieldValueByTableName("Invoice Items")(0)(0).Value
Write value to table cell	AppVariable.VarValidationControl_MainPage.Field(<TABLENAME>).Field(<COLUMN INDEX>,<ROW INDEX>).Value Note: here column needs to be one from Taxonomy. Table Column Name in the document may vary	In To Variable property of Set Variable Value, you can provide following expression In Set Value, provide what Value you want to update. Refer Starter Kit
Multi Value Field support		
Read multi value field	AppVariable.VarValidationControl_MainPage.Field("<COLUMN NAME>", 2).Value	AppVariable.VarValidationControl_MainPage.Field("MultiFieldValue", 2).Value
Write to multi value in Field	AppVariable.VarValidationControl_MainPage.Field(<FIELD NAME>, <INDEX>).Value	AppVariable.VarValidationControl.Field("MultiValueField", 2).Value
Set Confirm property	AppVariable.VarValidationControl_MainPage.Field(<FIELD NAME>).Confirm	AppVariable.VarValidationControl_MainPage.Field("Patient Name").Confirm
Set Confidence property	AppVariable.VarValidationControl_MainPage.Field(<FIELD NAME>).Confidence	AppVariable.VarValidationControl_MainPage.Field("Patient Name").Confidence
DeleteValue of Cell	AppVariable.VarValidationControl1_MainPage.Field(<FIELD NAME>).DeleteValue	 Expression in To Variable

		AppVariable.VarValidationControl1_MainPage.Field("Vendor Name").DeleteValue

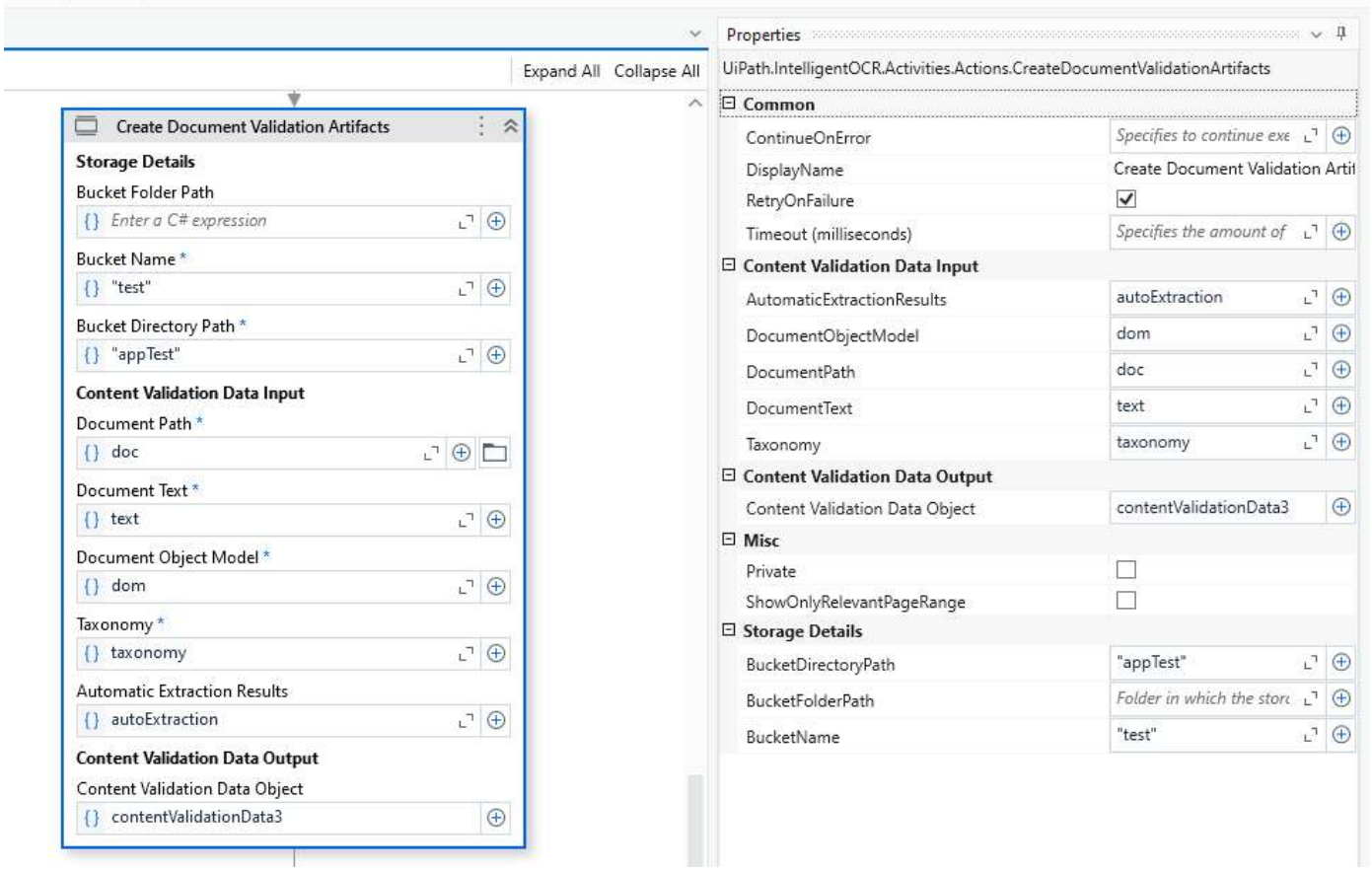
New Document Understanding Activities Details (UiPath.IntelligentOCR.Activities pack)

Two new activities allow you to create App Tasks based on Web App Projects containing Validation Controls for documents: Create Document Validation Artifacts and Retrieve Document Validation Artifacts.

These are pair activities: one stores the data required for display in a validation control and returns a reference to it, while the second retrieves the validated information once the task is completed.

You can use these activities as many times as you want for a single task – meaning you can display multiple documents and messages within the same task!

Create Document Validation Artifacts



Purpose:
This activity takes as input all the data required to prepare the document for display in a validation control, prepares it for display, and stores it in the storage bucket of your choice.

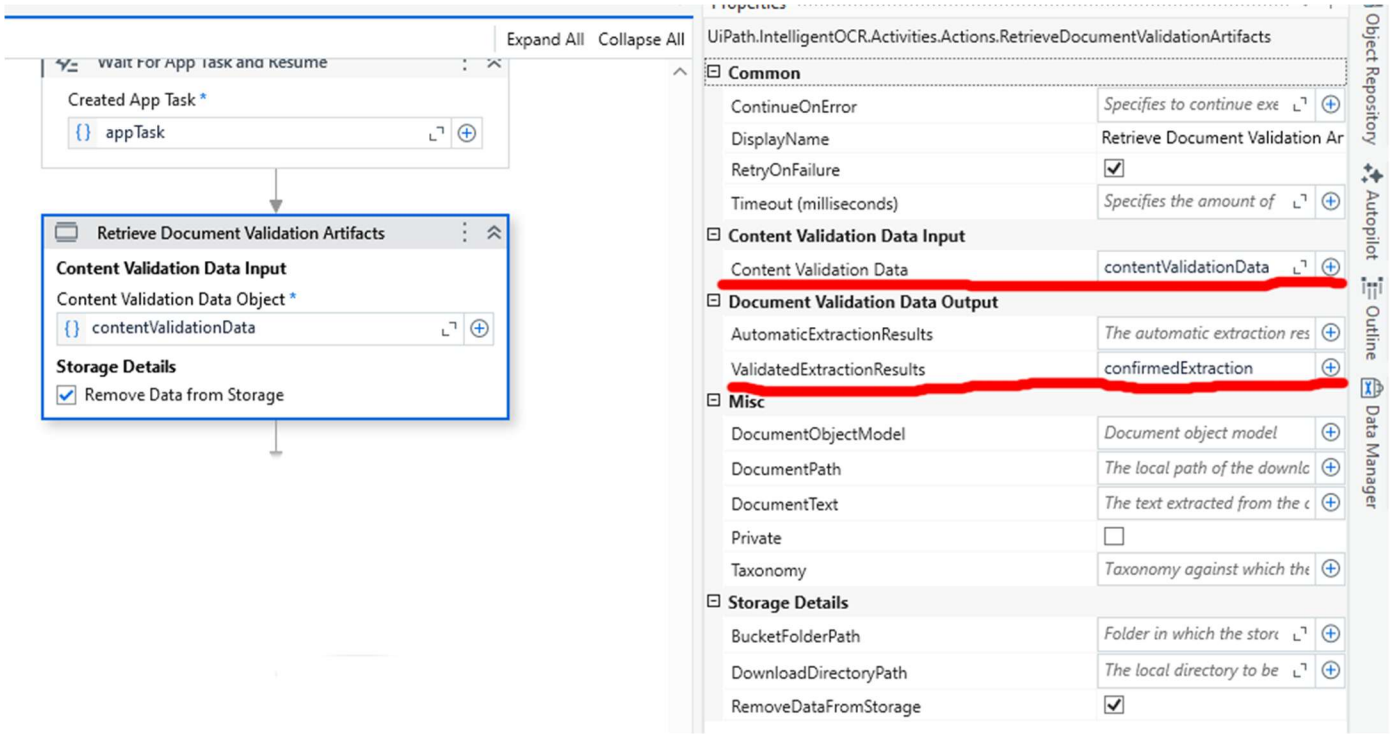
Essential Inputs:

- Document path, Text, DOM, Taxonomy and Automatic Extraction Results – to capture the data for the processed document to be reviewed,
- Bucket Folder Path, Bucket Name, Bucket Directory Path – to capture the storage bucket details about where the task data should be stored,
- ShowOnlyRelevantPages –flag governing display of the validation control – if it shows the entire file or only the part processed for data extraction (for cases of multiple documents within the same file)

Output:

- Content Validation Data Object (of type ContentValidationData) - this object will be passed on to the Create App Task activity, as part of the task payload as defined in the Action Schema.

Retrieve Document Validation Artifacts



Purpose:
This activity is used to retrieve the Document and Validated information after the task is completed and make them available to the robot so that the automation can continue.

Inputs:

- Content Validation Data object – the one that was used in the Create Document Validation Artifacts – this input will ensure the retrieval of the validated extraction result for that particular document.

Essential Outputs:

- Validated Extraction Result – the data confirmed by the human user
- All document related artifacts (document, text, dom, taxonomy, etc) in case the workflow is resumed on a different machine or context needs to be rebuilt once the task is completed.

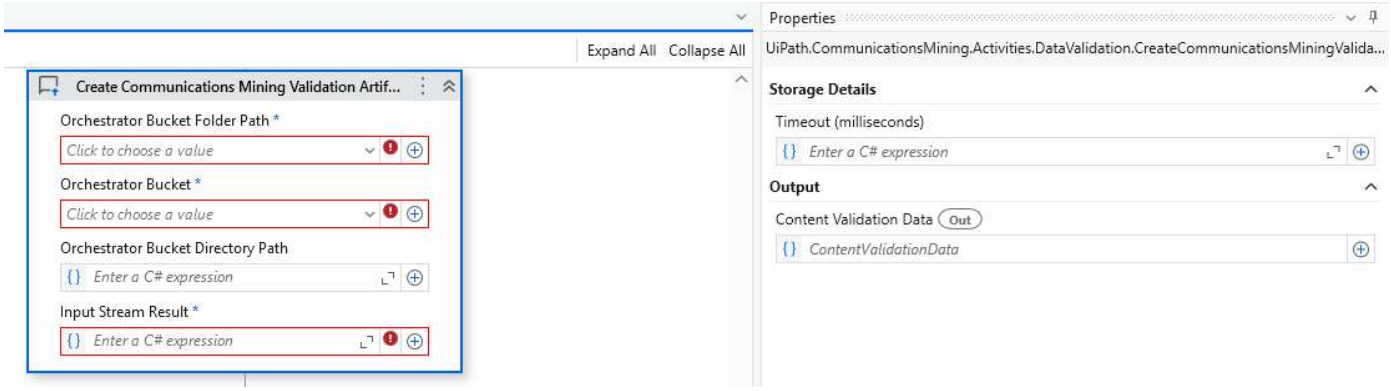
New Communications Mining Activities Details (UiPath.CommunicationsMining.Activities pack)

Two new activities allow you to create App Tasks based on Web App Projects containing Validation Controls for messages processed through Communications Mining: Create Communications Mining Validation Artifacts and Retrieve Communications Mining Validation Artifacts.

These are pair activities: one stores the data required for display in a validation control and returns a reference to it, while the second retrieves the validated information once the task is completed.

You can use these activities as many times as you want for a single task – meaning you can display multiple documents and messages within the same task!

Create Communications Mining Validation Artifacts



Purpose:
This activity takes as input a stream result, stores it in the storage bucket of your choice and generates the Content Validation Data required for loading a validation control.

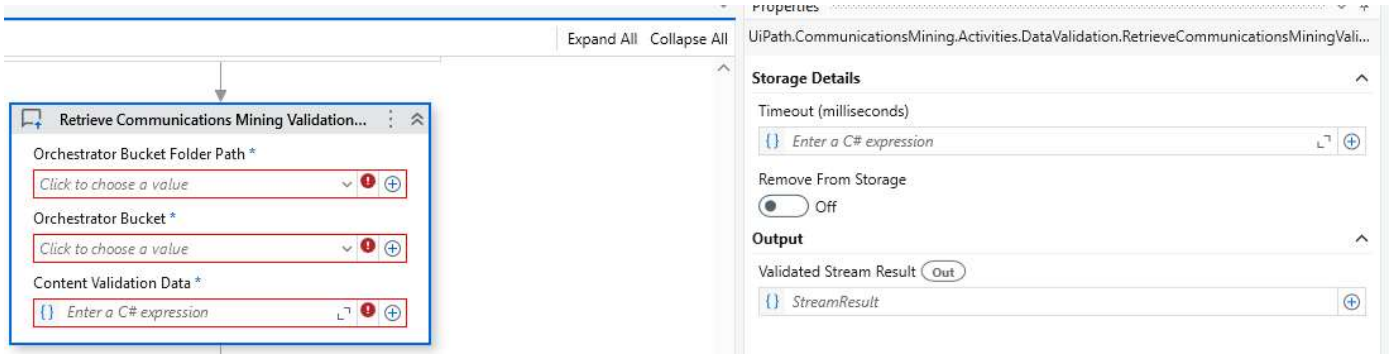
Input:

- Bucket Folder Path, Bucket Name, Bucket Directory Path – to capture the storage bucket details about where the task data should be stored,

Output:

- Content Validation Data Object (of type ContentValidationData) - this object will be passed on to the Create App Task activity, as part of the task payload as defined in the Action Schema.

Retrieve Communications Mining Validation Artifacts



Purpose:
This activity is used to retrieve the Validated information after the task is completed and make it available to the robot so that the automation can continue.

Inputs:

- Content Validation Data object – the one that was used in the Create Communications Mining Validation Artifacts – this input will ensure the retrieval of the validated extraction result for that particular message.

Outputs:

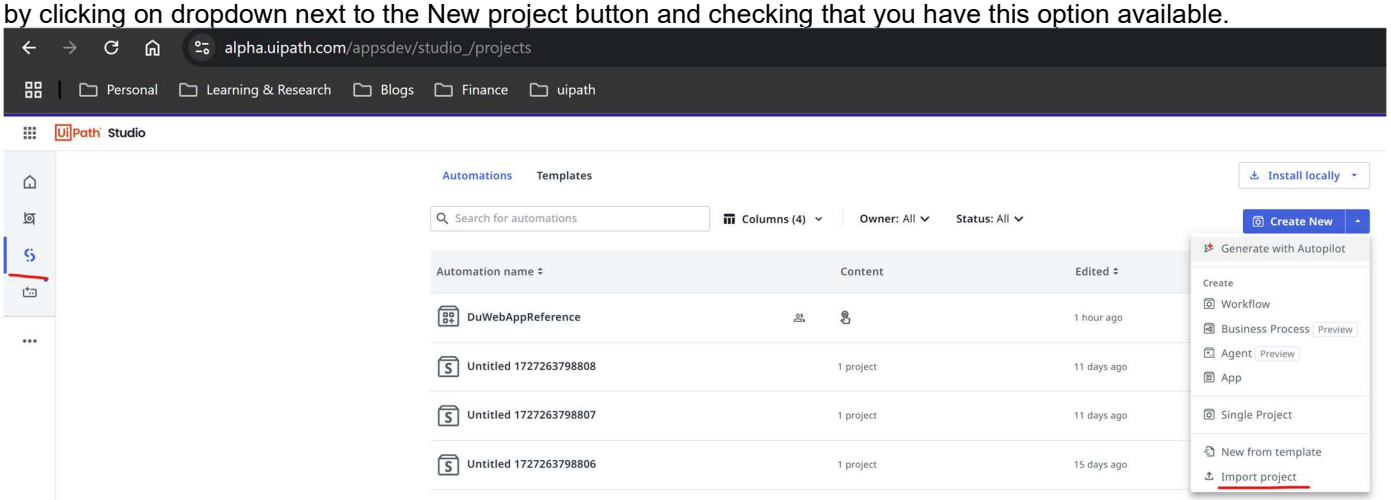
- Validated Stream Result – the data confirmed by the human user

Step by Step Guide – Design Time Setup

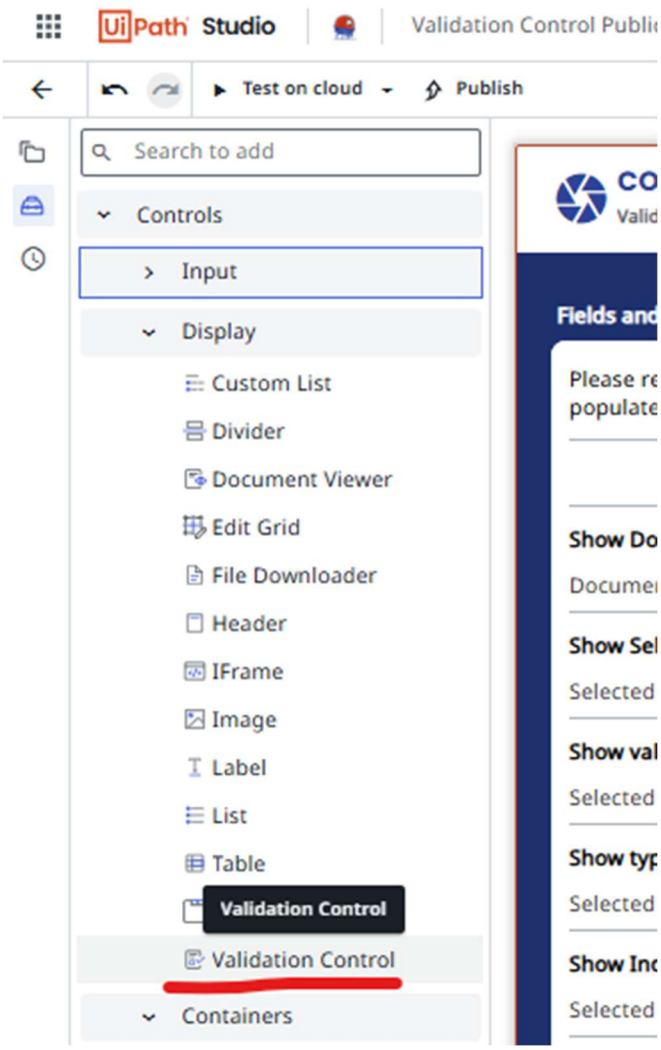
Agent App Setup

- **Prerequisite**
    - **Web Apps Setup**

1) As a prerequisite to building your own Validation App, please check you can create a new Webb App Project, by clicking on dropdown next to the New project button and checking that you have this option available.



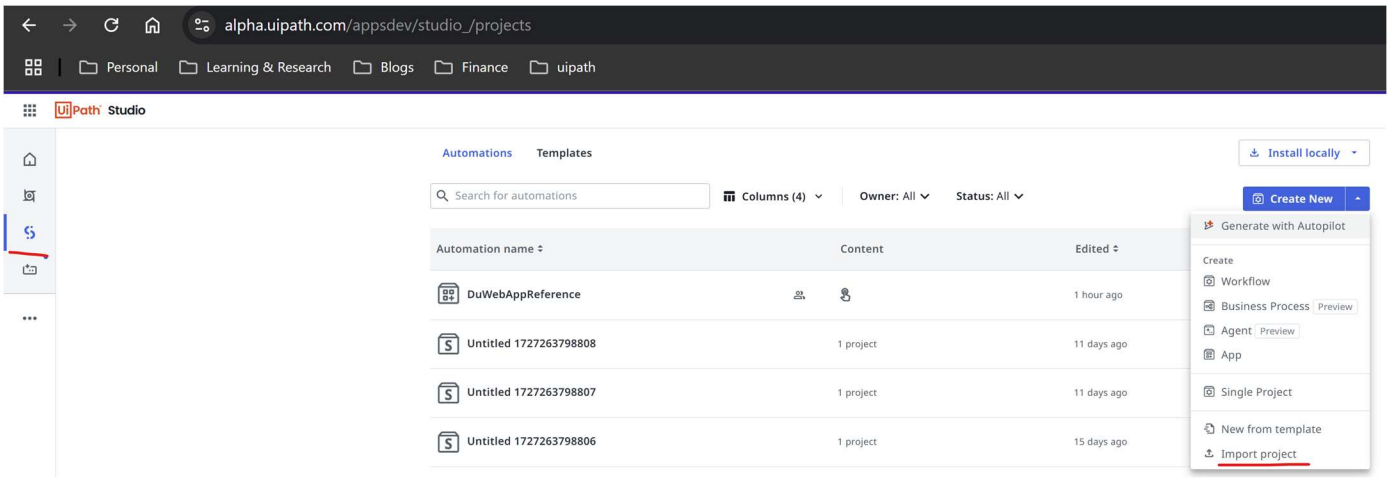
2) Check for Validation Control available as part of your Controls in a newly created Web App



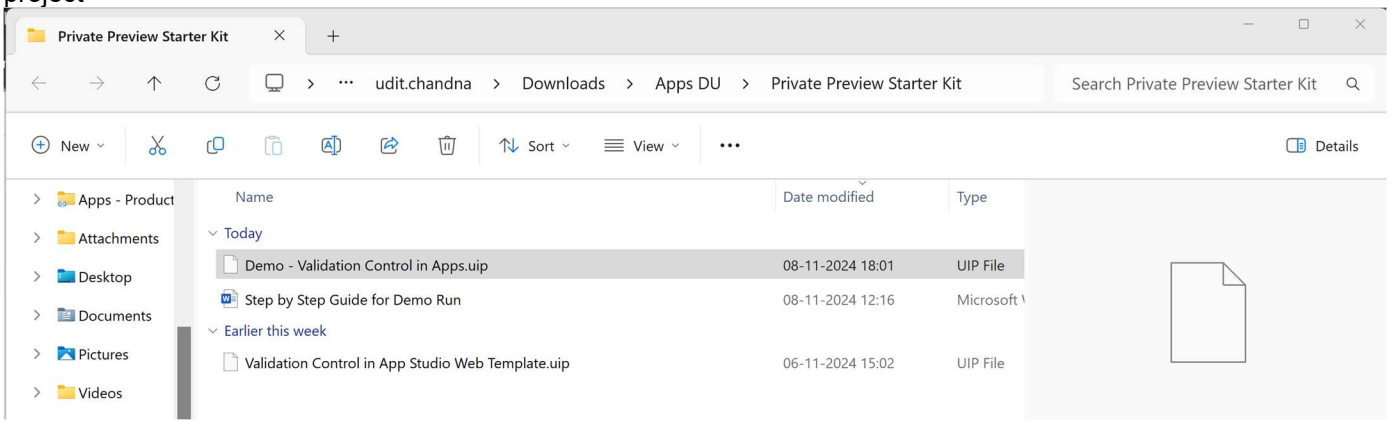
3) Check that, when you create automation on any control, you can find the "Save Validation Control State" and "Set Validation Control Field Focus" activities. If you do not have access to this, you need to enable, in Automation Ops, a policy to allow, on your tenant, the use of preview packages. (SHOULD NOT BE NEEDED IN PUBLIC PREVIEW)

- **Process Setup**

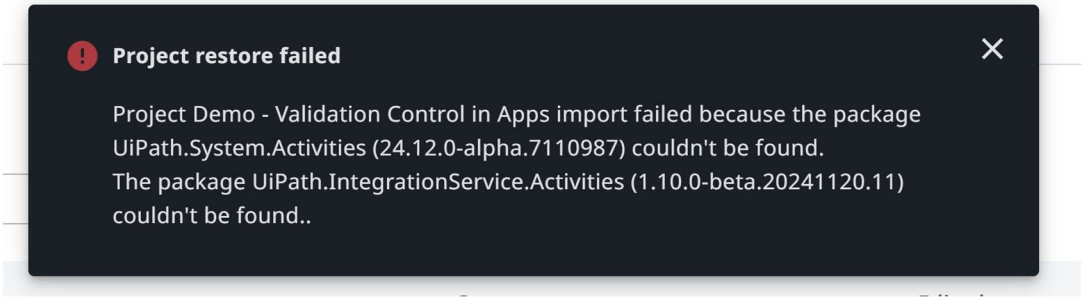
1) Go To Studio and Select Import Project



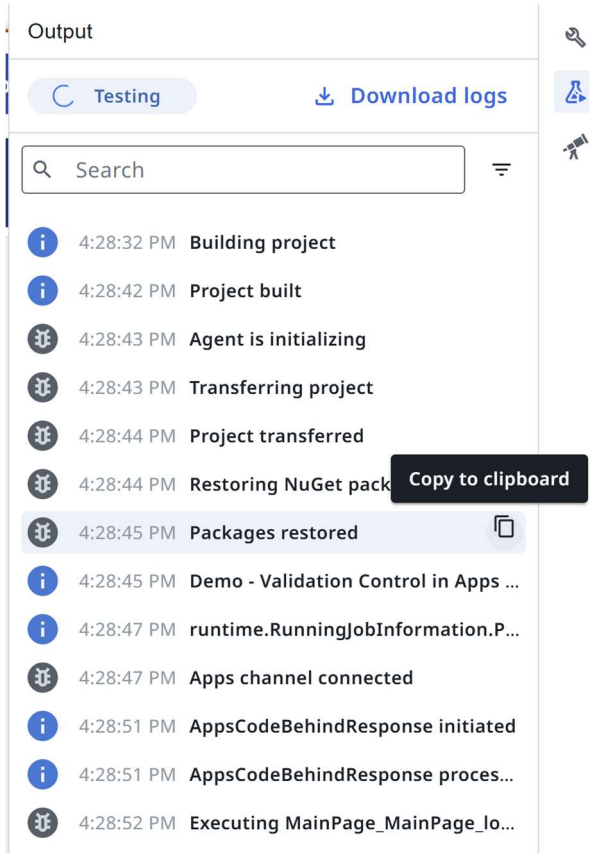
2) Select the file in **“Validation Control Public Preview Starter Kit.uip”** file in Starter Kit and import the project



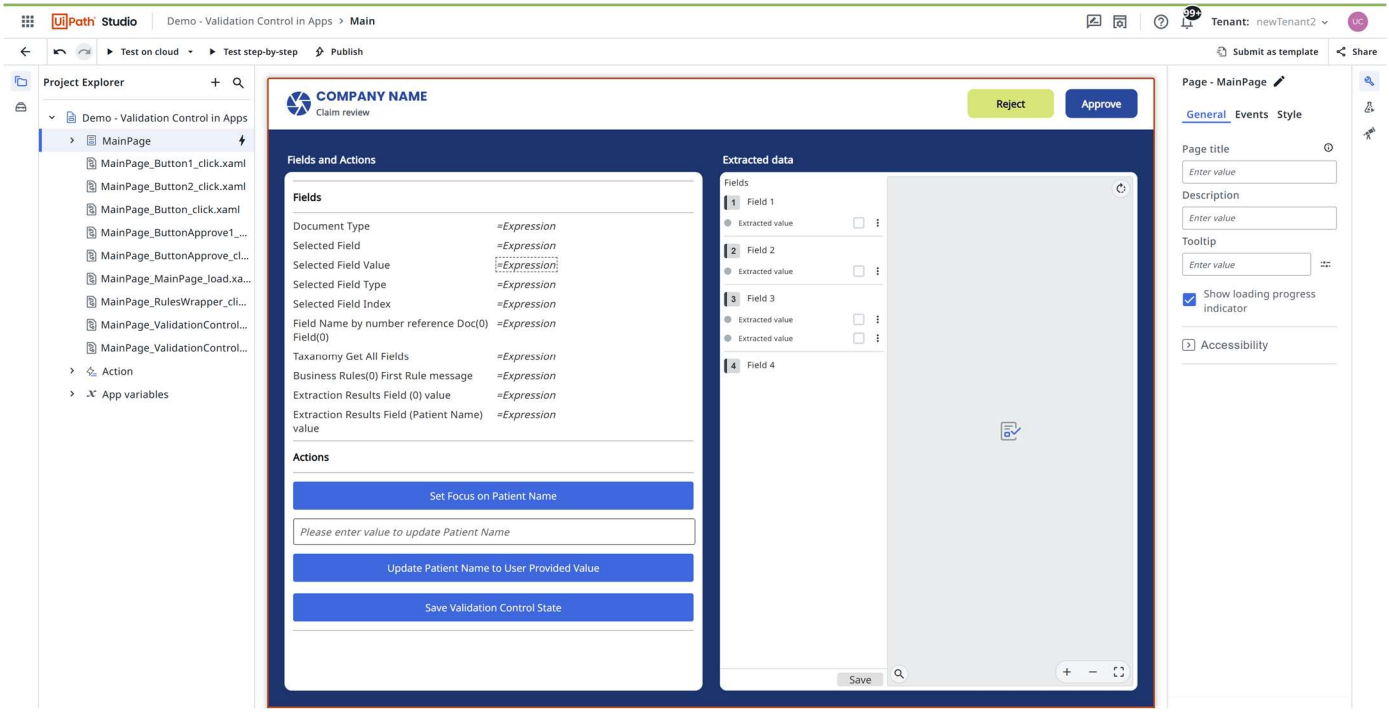
2a) In case you face error like below on import, please ignore and open the app and let it run by clicking on “Test on Cloud”, the app would run and fix all dependencies issue. In case still not working, please reach out to us at udit.chandna@uipath.com, loana.gligan@uipath.com



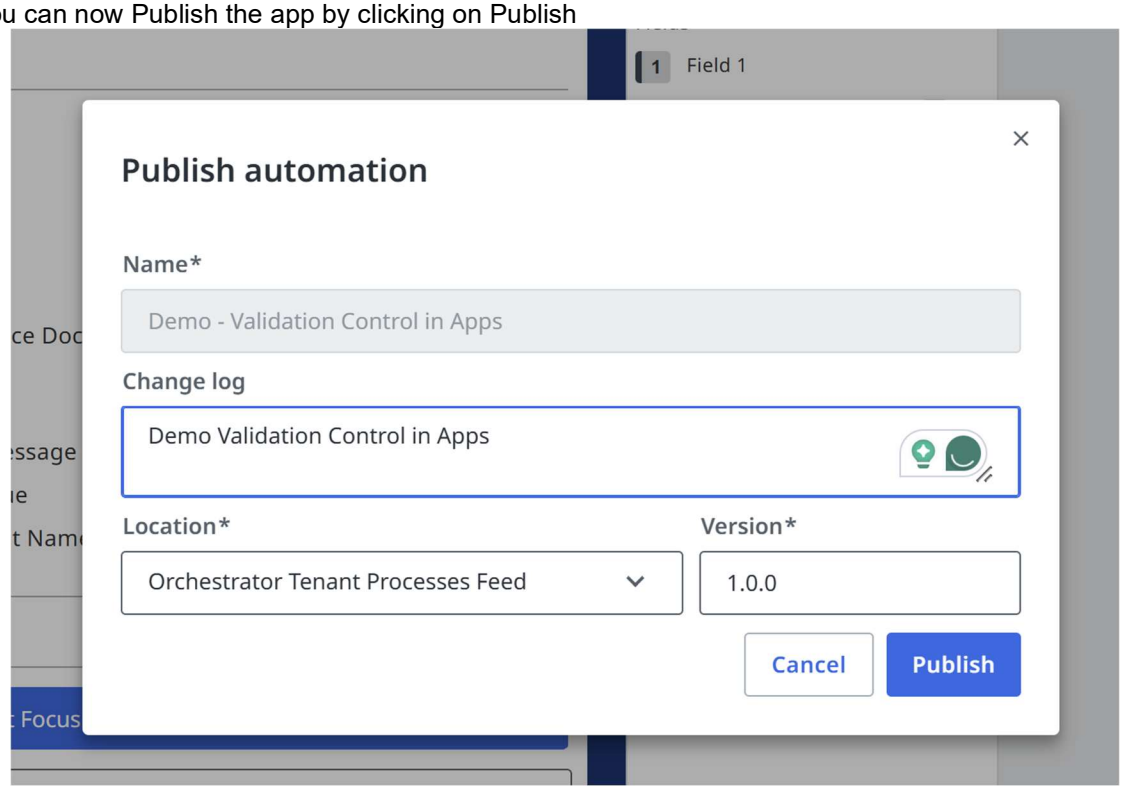
2b) Once you click on Test on Cloud, it should fix these package errors



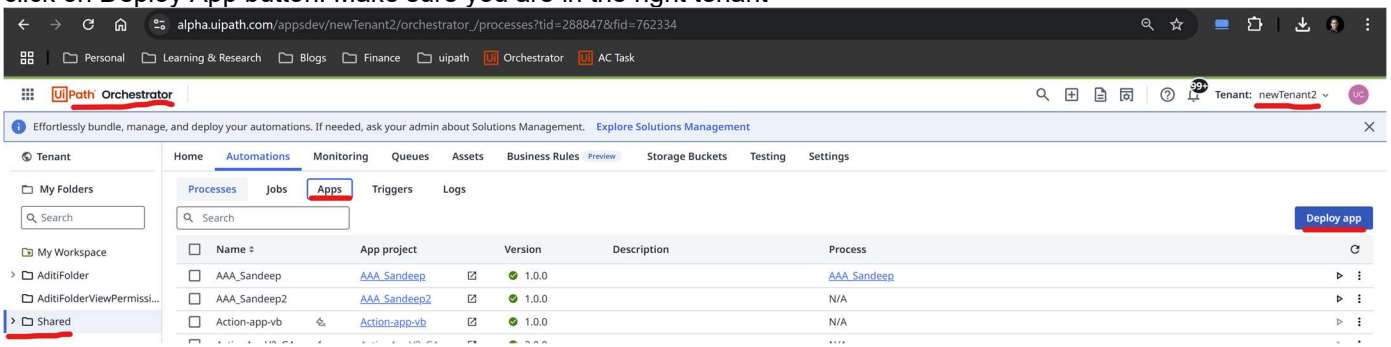
3) You can now see the app available in Studio Home page under Automations for your access. You can open it and would see the same below



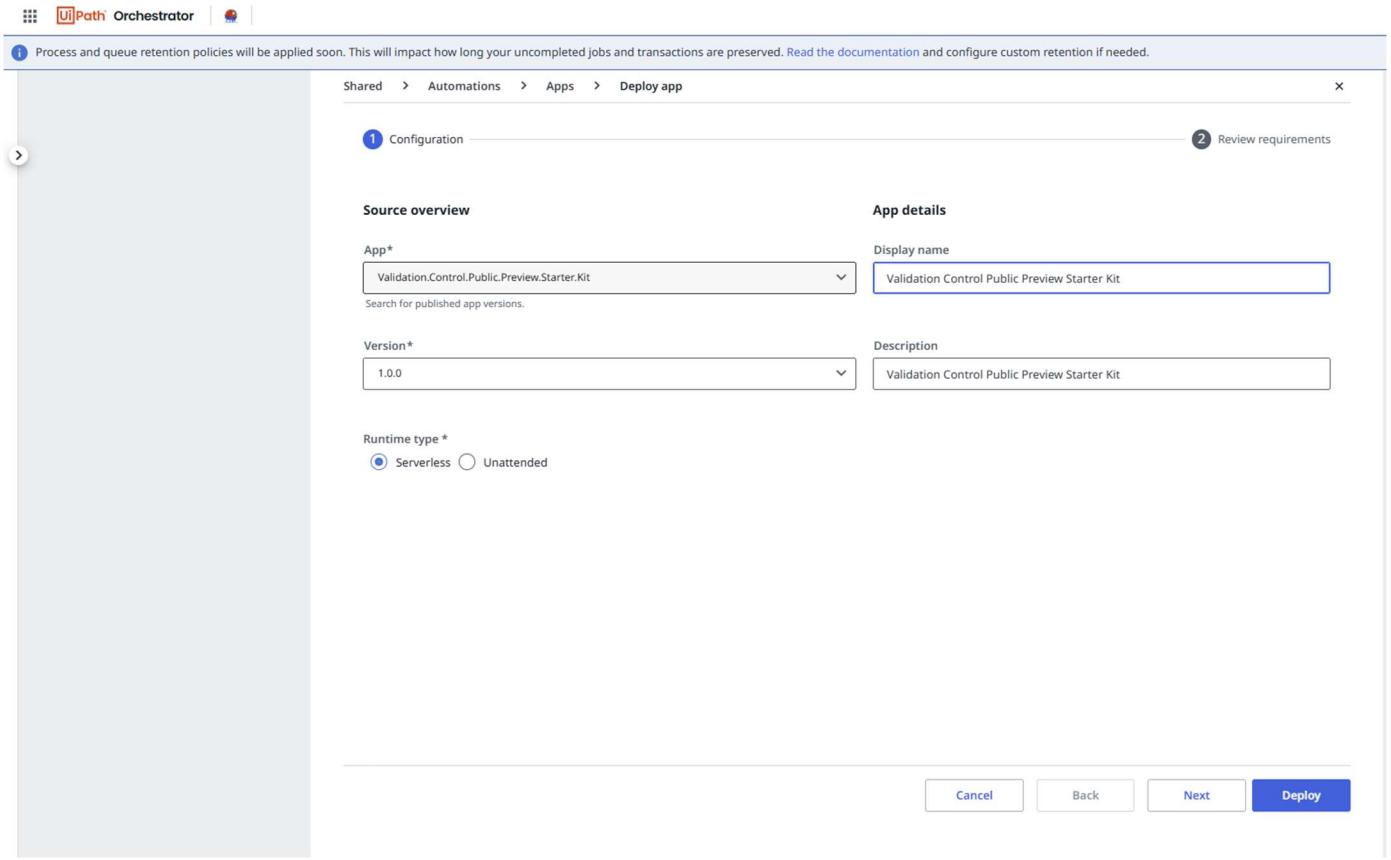
4) You can check the app and change the UI or activities according to your needs.
    a. Note: - Being an action app with Validation Station source available at runtime you can only test it after deploying, Preview will only help with App specific widgets preview but not rendering the Validation control
5) You can now Publish the app by clicking on Publish



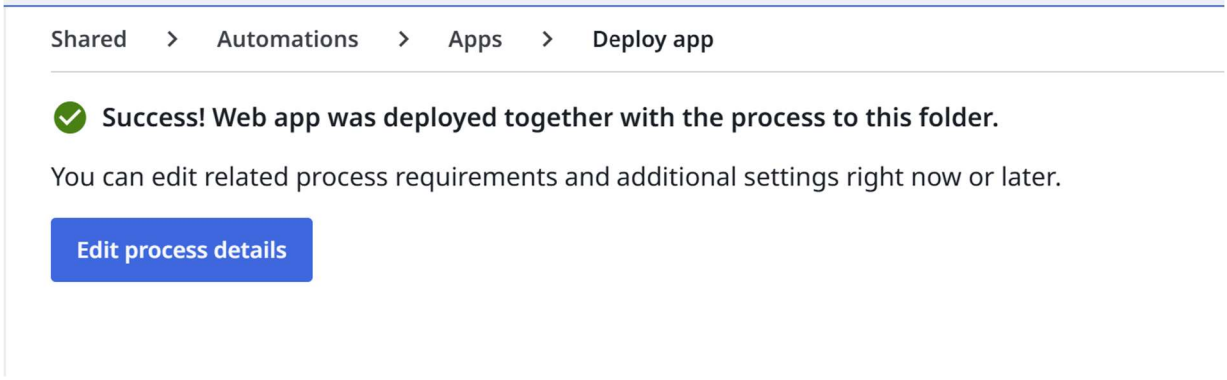
6) You can then Deploy the app as below, Go to Orchestrator, select automations, select the correct folder and click on Deploy App button. Make sure you are in the right tenant



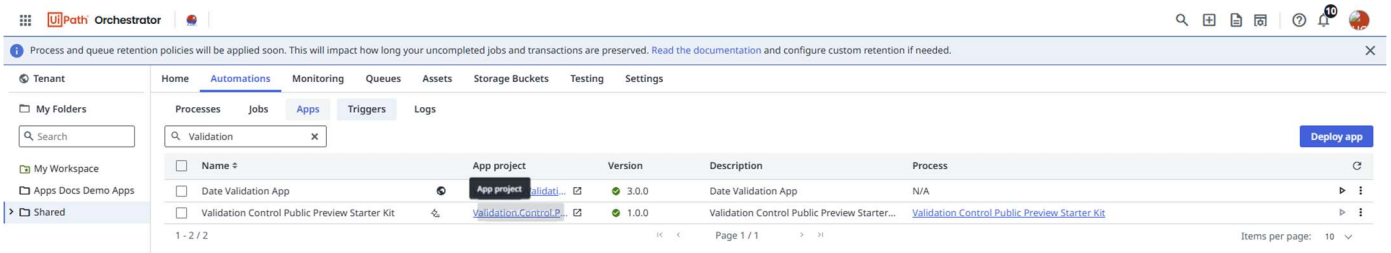
7) Once you click Deploy App, select the app you recently, published, if the app is not available, please check your tenant and folder settings. For our app , we had published as “Validation Station Public Preview Starter Kit”



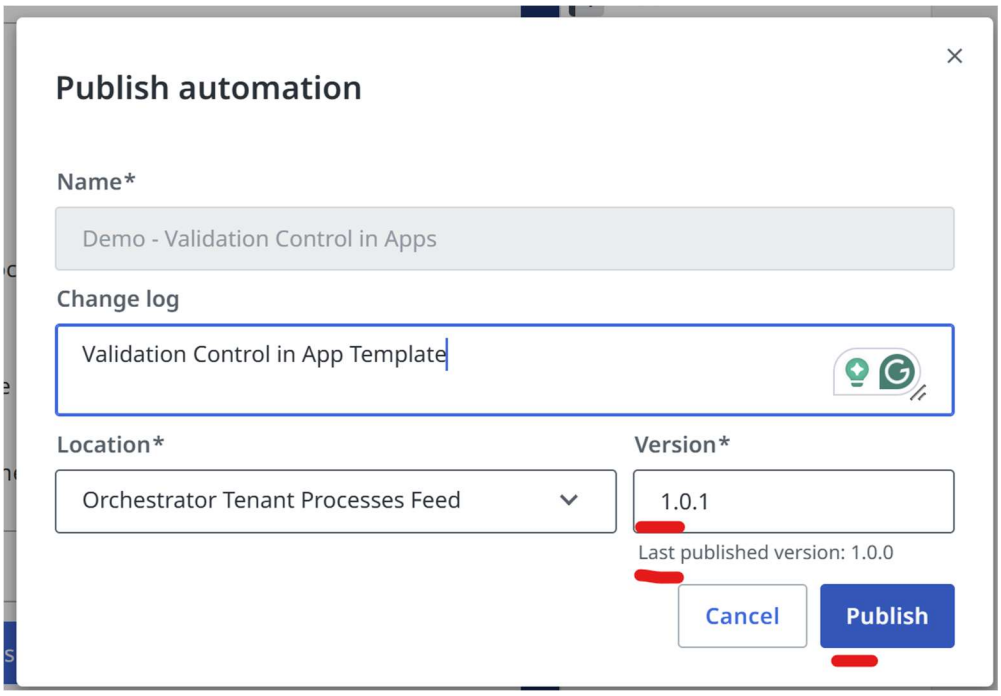
8) Click on Deploy

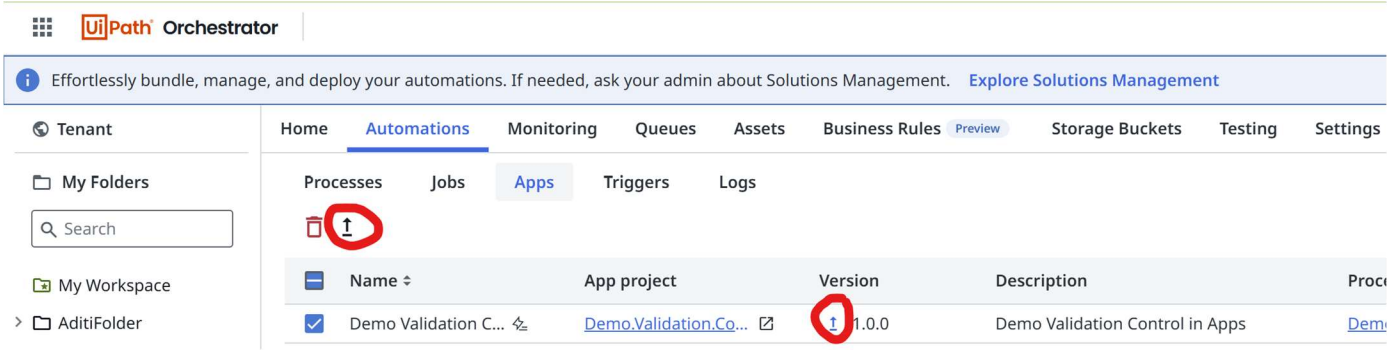


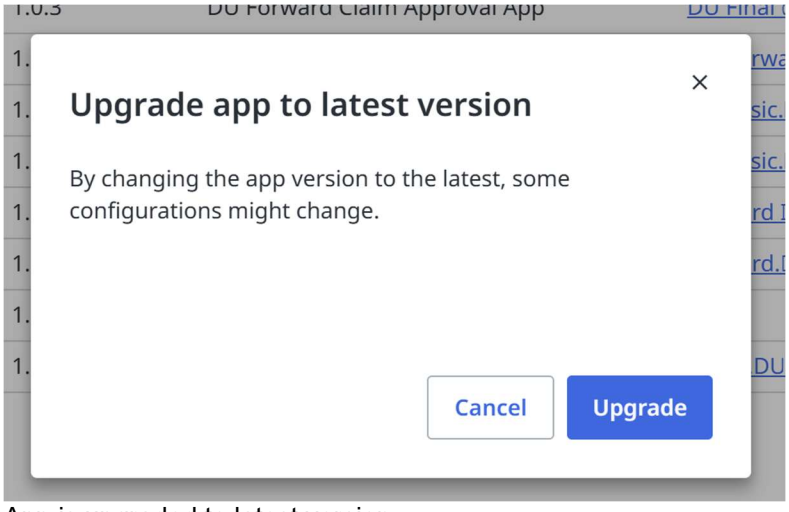
9) You should see the same available in Processes as well in Orchestrator



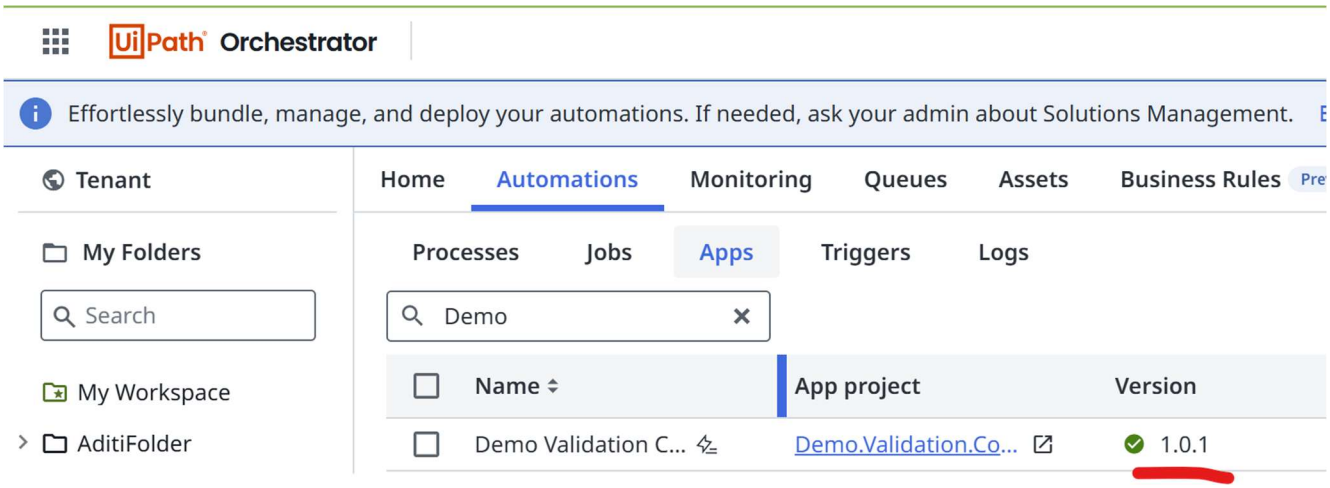
10) Every time you publish a new version of your app, you need to upgrade your deployed app to the latest version, by going to orchestrator/Processes/Apps and using the Upgrade to Latest Version from the three-dot menu.



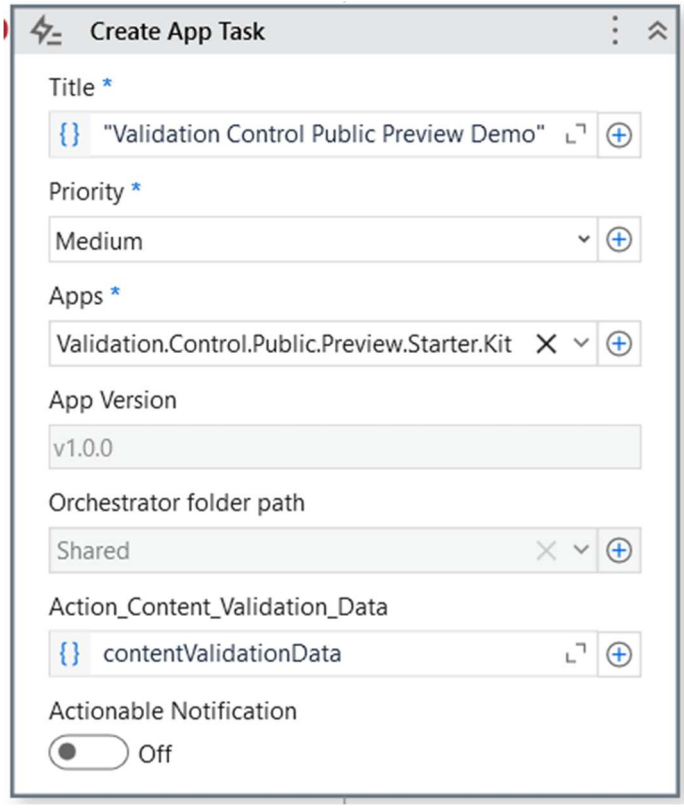




11) App is upgraded to latest version

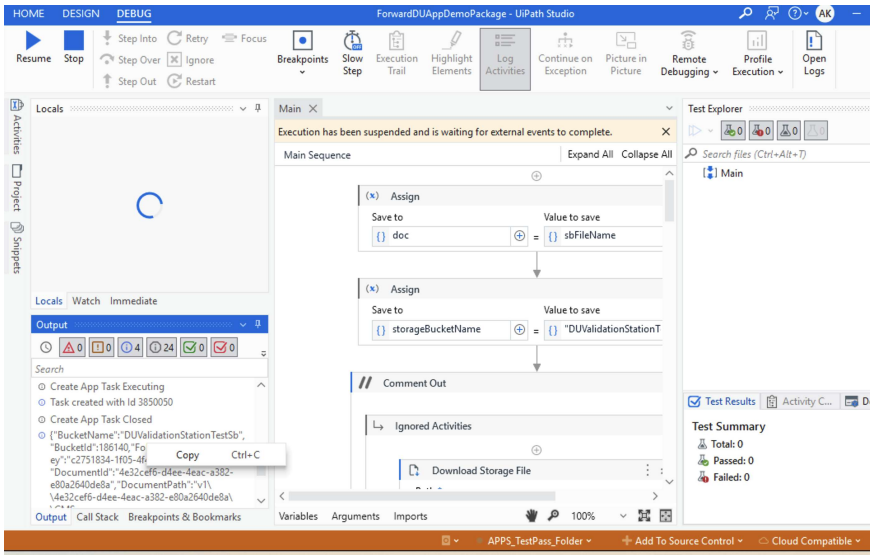


12) Now this app is available in create app task as below



Debugging the validation control directly in Apps instead of Action Center runtime

1) In Studio desktop, Enable Log activities > Debug file > copy the cvd data from log (screenshot below)



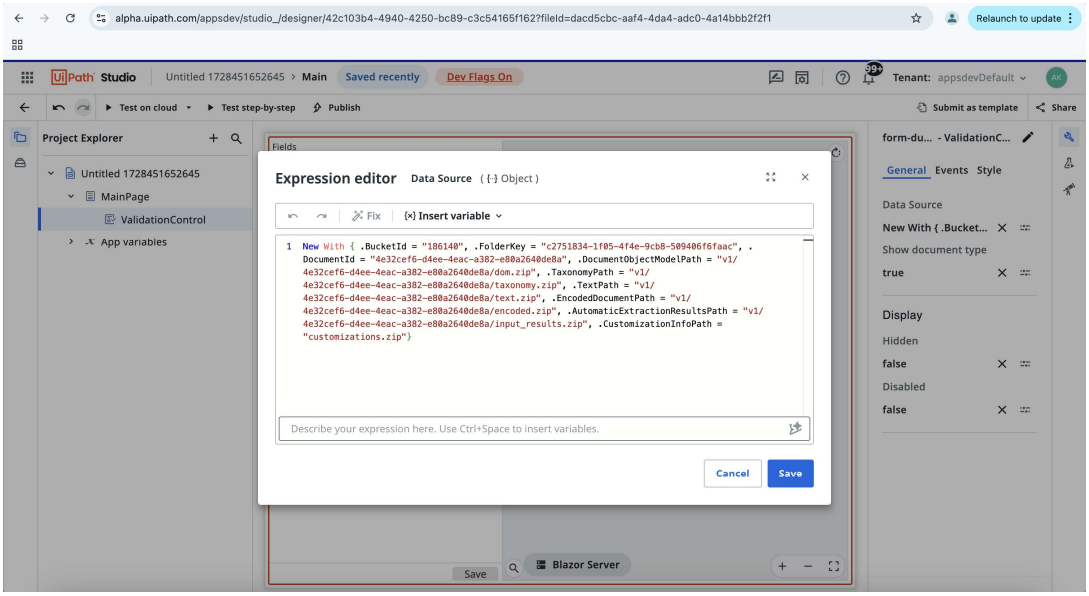
2) It should be in format like below

{"BucketName":"DUValidationStationTestSb","BucketId":186140,"FolderId":756377,"FolderKey":"c2751834-1f05-4f4e-9cb8-509406f6faac","DocumentId":"4e32cef6-d4ee-4eac-a382-e80a2640de8a","DocumentPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\CMS 1500.zip","EncodedDocumentPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\encoded.zip","TextPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\text.zip","DocumentObjectModelPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\dom.zip","TaxonomyPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\taxonomy.zip","AutomaticExtractionResultsPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\input_results.zip","ValidatedExtractionResultsPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\output_results.zip","ExtractorPayloadsPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\extractor_payloads.zip","ShowOnlyRelevantPageRange":false,"AdditionalDataPath":"","OriginalDocumentFileName":"CMS 1500.pdf","CustomizationInfoPath":"v1\\4e32cef6-d4ee-4eac-a382-e80a2640de8a\\customization_info.zip"}

3) Use the bucketId, FolderKey, DocumentId to create below VB expression and assign to Data Source for Validation Control (screenshot below)

New UiPath.DocumentProcessing.Contracts.Actions.ContentValidationData With { .BucketId = 186140, .FolderKey = new Guid("c2751834-1f05-4f4e-9cb8-509406f6faac"), .DocumentId = "4e32cef6-d4ee-4eac-a382-e80a2640de8a", .DocumentObjectModelPath = "v1/4e32cef6-d4ee-4eac-a382-e80a2640de8a/dom.zip", .TaxonomyPath = "v1/4e32cef6-d4ee-4eac-a382-e80a2640de8a/taxonomy.zip", .TextPath = "v1/4e32cef6-d4ee-4eac-a382-e80a2640de8a/text.zip", .EncodedDocumentPath = "v1/4e32cef6-d4ee-4eac-a382-e80a2640de8a/encoded.zip", .AutomaticExtractionResultsPath = "v1/4e32cef6-d4ee-4eac-a382-e80a2640de8a/input_results.zip", .CustomizationInfoPath = "customizations.zip"}

Tip: Please ask ChatGPT/Gemini to give transformed version for your contentvalidationdata serialized object based on above example

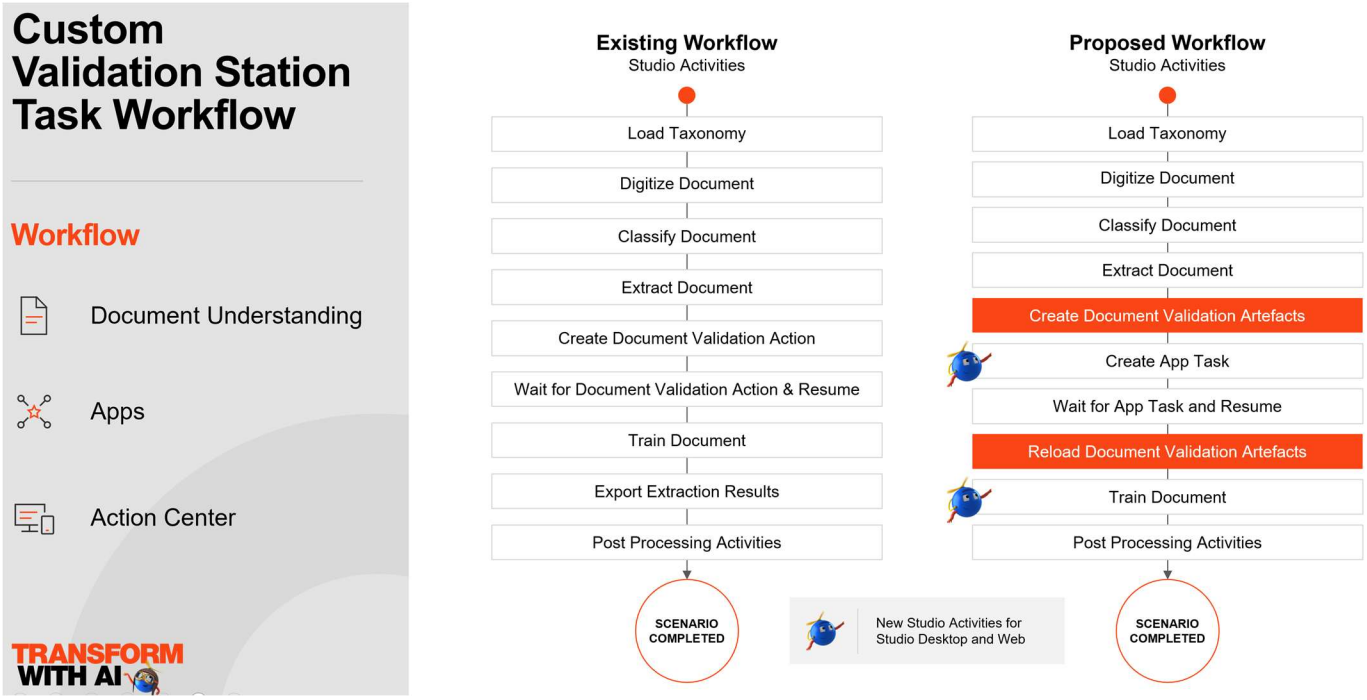


By doing this you can test the app directly in Studio Web without creating app task every time in Action Center

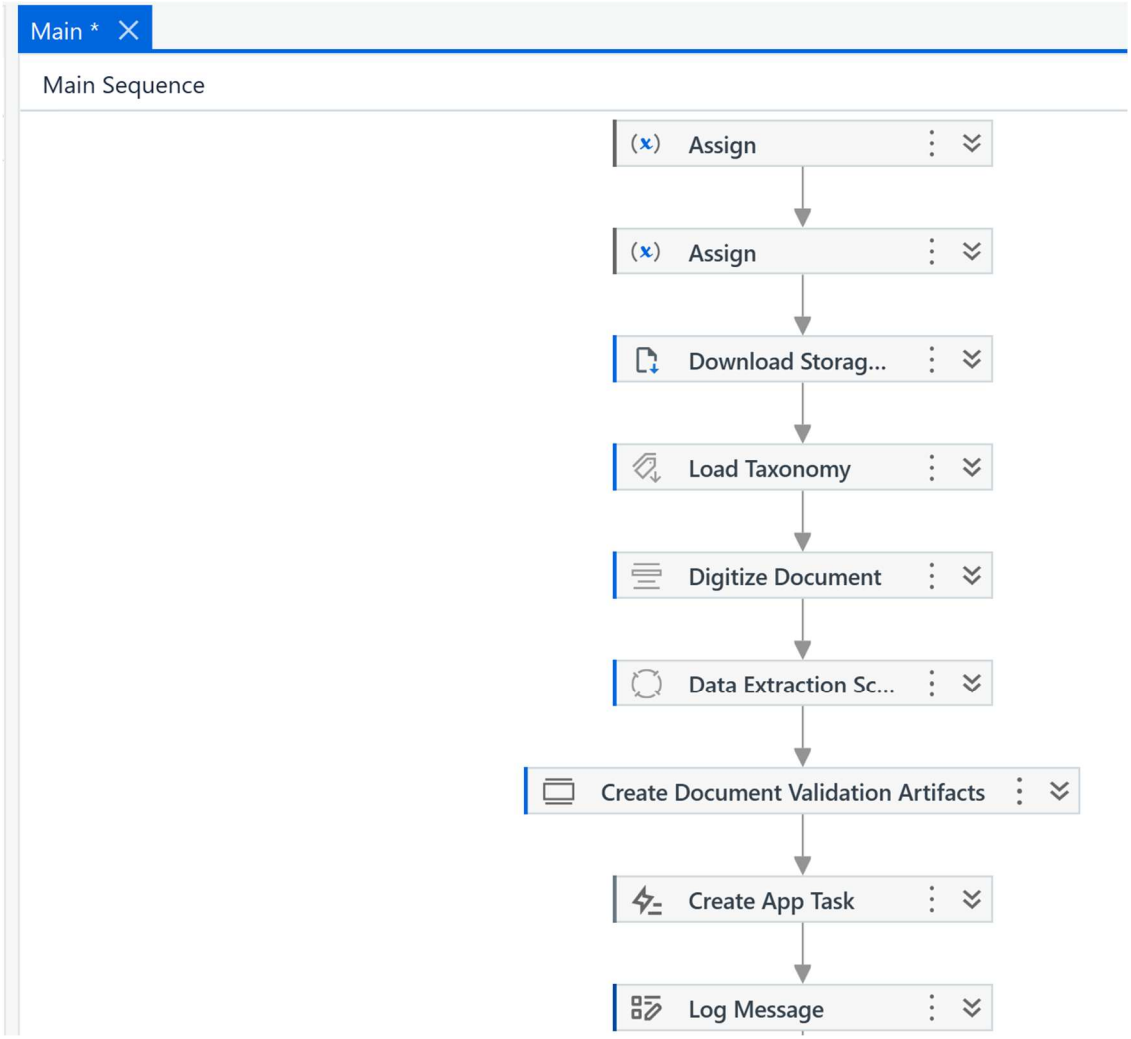
Note: Standalone Validation Apps (Validation Apps without Action Center) would come in later releases

Studio Desktop Process Setup

The following is the list of newly proposed workflow for the new Validation Control in App. You need to remember 2 major changes – “Create Document Validation Artefacts” and “Create App Tasks” are the ones we will use



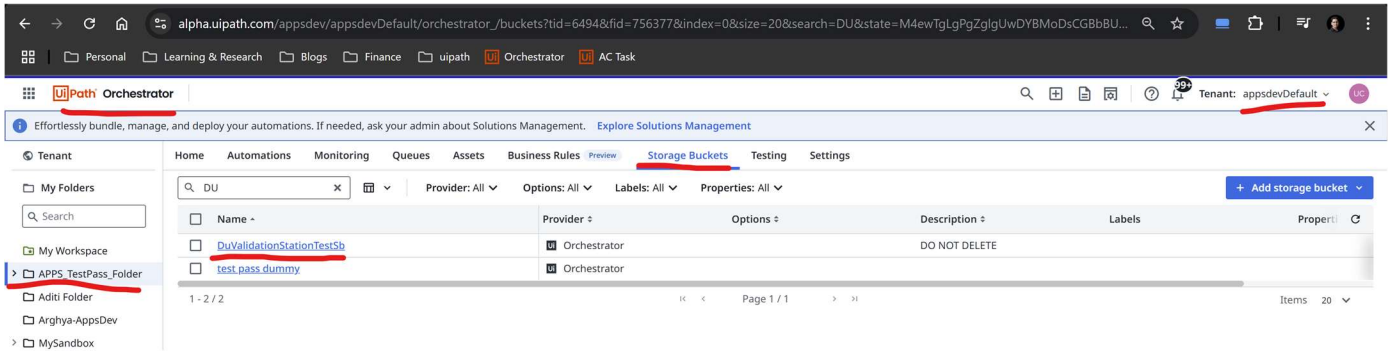
New Sample sequence from the process shared in Starter Kit



Prerequisite

- **Orchestrator and Storage Bucket Setup**

Data that is required for the Validation Control to load is stored in storage buckets. As a prerequisite to starting building your own custom validation app, you will need to create a storage bucket (from Orchestrator) in the right tenant, under the Shared folder of that tenant. You will be using this storage bucket to store the data required for validation controls in app tasks.

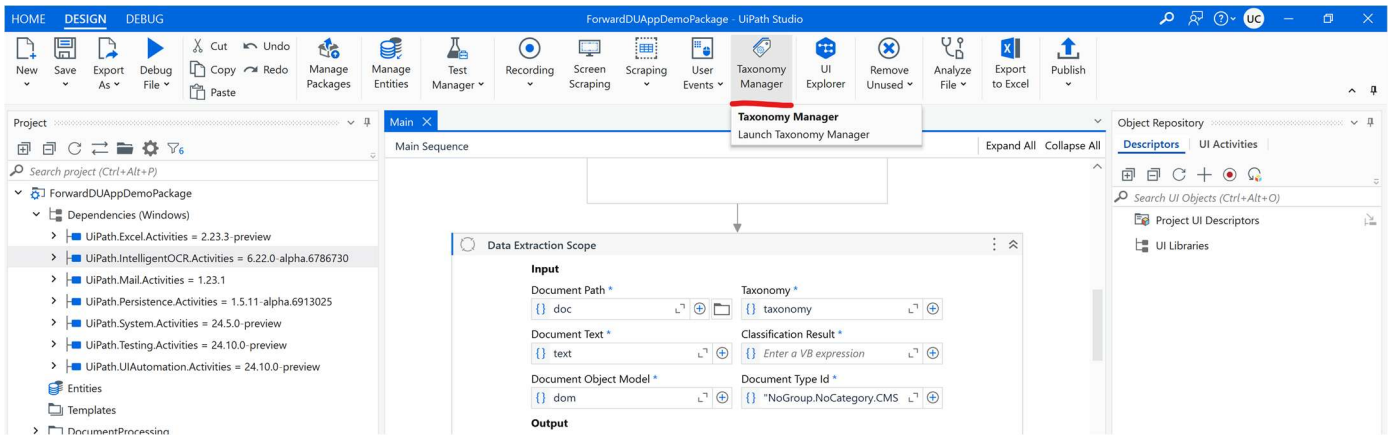


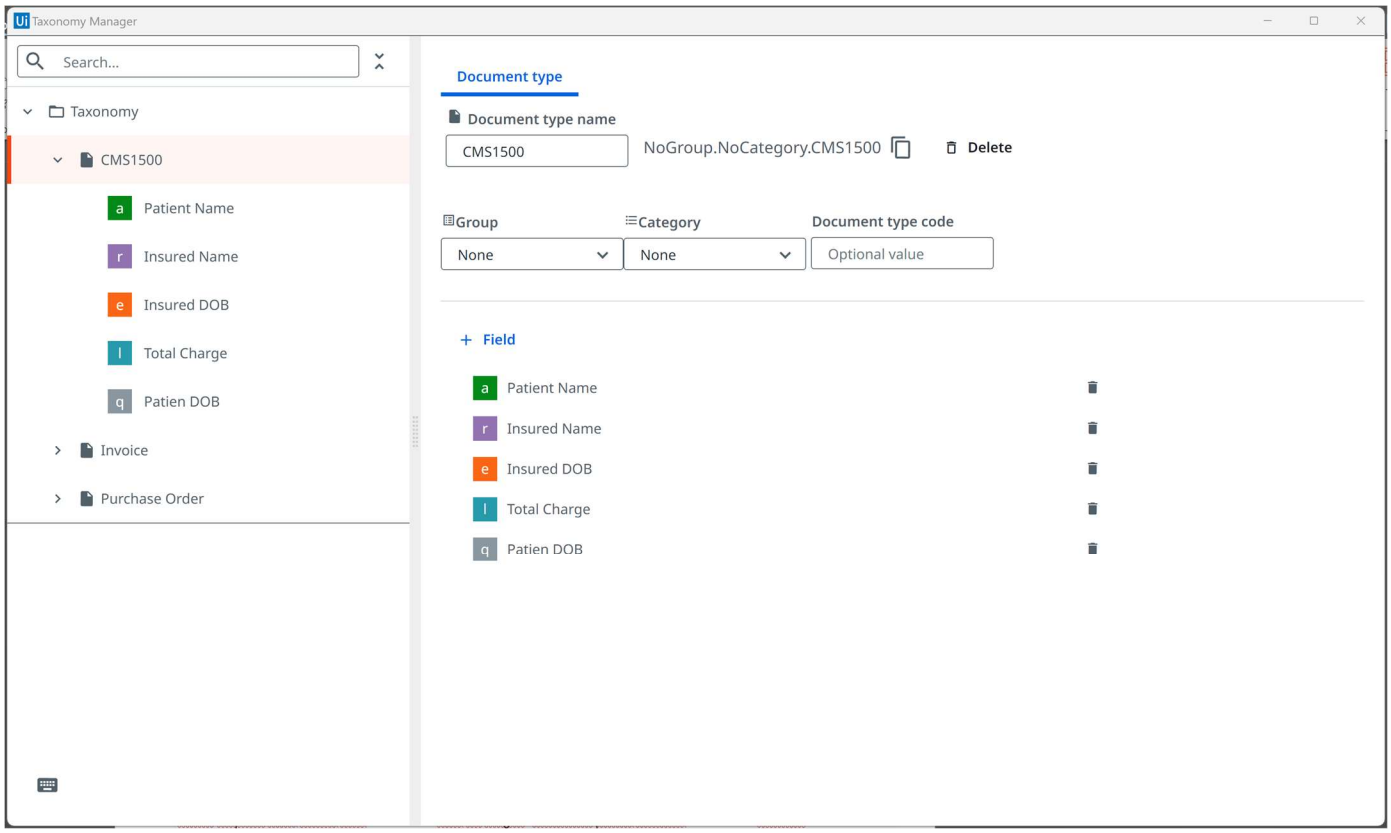
- **Studio Packages**

Make sure all the packages installed have the required version as documented at the beginning of this documentation.

- **Taxonomy Setup**

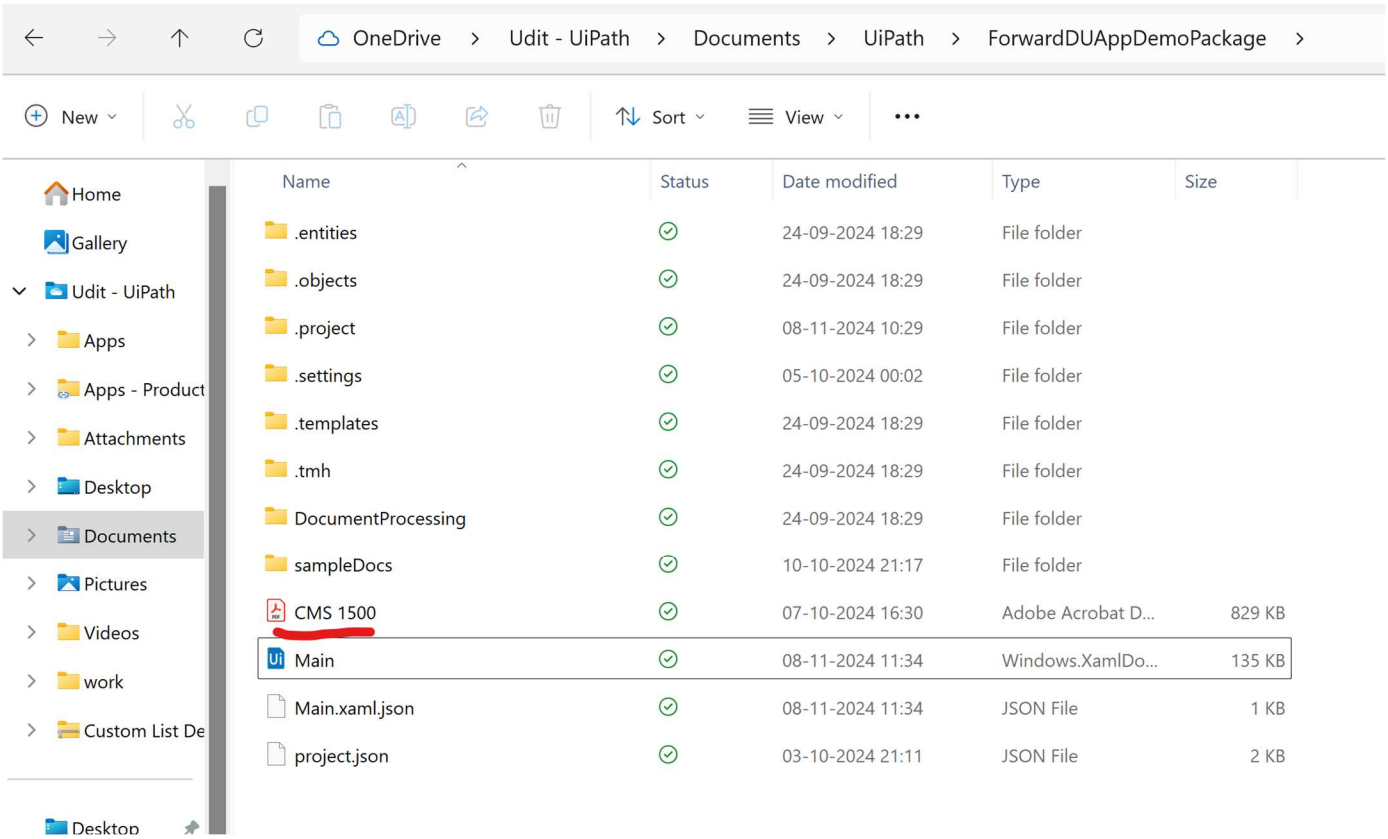
Build your document processing taxonomy using the Taxonomy Manager (for documents). This step is not required if you wish to process Communications Mining stream results.

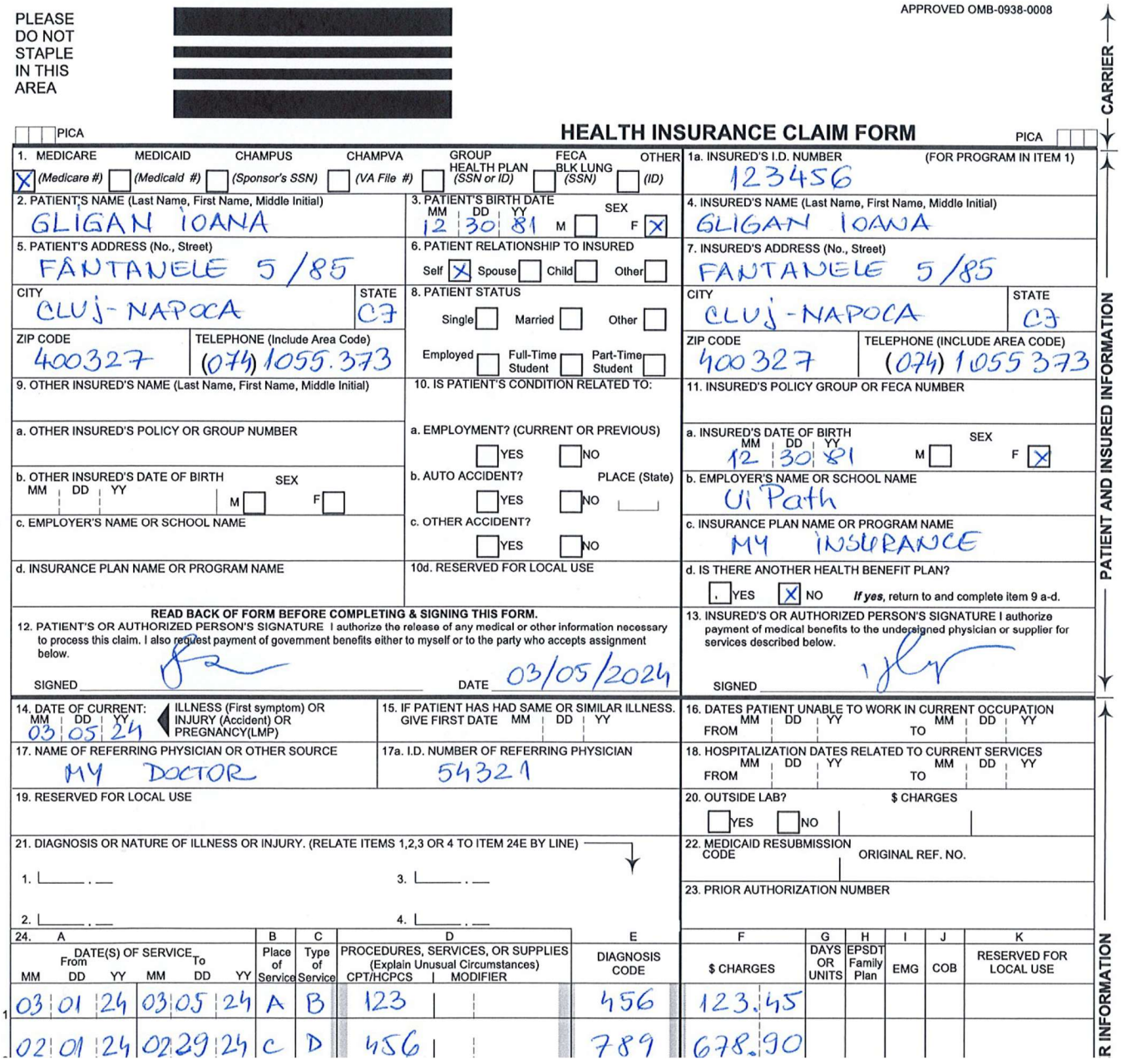




- **Sample File to be used**

The following parent folder stores the CMS1500 file and should be checked in case File not found


OneDrive
Udit - UiPath
Documents
UiPath
ForwardDUAppDemoPackage
New
Sort
View
Home
Gallery
Udit - UiPath
Apps
Apps - Product
Attachments
Desktop
Documents
Pictures
Videos
work
Custom List De
Desktop
Name
Status
Date modified
Type
Size
.entities
24-09-2024 18:29
File folder
.objects
24-09-2024 18:29
File folder
.project
08-11-2024 10:29
File folder
.settings
05-10-2024 00:02
File folder
.templates
24-09-2024 18:29
File folder
.tmh
24-09-2024 18:29
File folder
DocumentProcessing
24-09-2024 18:29
File folder
sampleDocs
10-10-2024 21:17
File folder
CMS 1500
07-10-2024 16:30
Adobe Acrobat D...
829 KB
Main
08-11-2024 11:34
Windows.XamlDo...
135 KB
Main.xaml.json
08-11-2024 11:34
JSON File
1 KB
project.json
03-10-2024 21:11
JSON File
2 KB



PLEASE DO NOT STAPLE IN THIS AREA

APPROVED OMB-0938-0008

PICA

HEALTH INSURANCE CLAIM FORM

PICA

1. MEDICARE [X] (Medicare #) MEDICAID [ ] (Medicaid #) CHAMPUS [ ] (Sponsor's SSN) CHAMPVA [ ] (VA File #) GROUP HEALTH PLAN [ ] (SSN or ID) FECA BLK LUNG [ ] (SSN) OTHER [ ] (ID)

1a. INSURED'S I.D. NUMBER (FOR PROGRAM IN ITEM 1): 123456

2. PATIENT'S NAME (Last Name, First Name, Middle Initial): GLIGAN IOANA

3. PATIENT'S BIRTH DATE MM DD YY: 12 30 81 SEX M [ ] F [X]

4. INSURED'S NAME (Last Name, First Name, Middle Initial): GLIGAN IOANA

5. PATIENT'S ADDRESS (No., Street): FANTANELE 5/85

6. PATIENT RELATIONSHIP TO INSURED: Self [X] Spouse [ ] Child [ ] Other [ ]

7. INSURED'S ADDRESS (No., Street): FANTANELE 5/85

CITY: CLUJ-NAPOCA STATE: CJ

8. PATIENT STATUS: Single [ ] Married [ ] Other [ ]

CITY: CLUJ-NAPOCA STATE: CJ

ZIP CODE: 400327 TELEPHONE (Include Area Code): (074) 1055.373

Employed [ ] Full-Time Student [ ] Part-Time Student [ ]

ZIP CODE: 400327 TELEPHONE (INCLUDE AREA CODE): (074) 1055 373

9. OTHER INSURED'S NAME (Last Name, First Name, Middle Initial):

10. IS PATIENT'S CONDITION RELATED TO:

11. INSURED'S POLICY GROUP OR FECA NUMBER:

a. OTHER INSURED'S POLICY OR GROUP NUMBER:

a. EMPLOYMENT? (CURRENT OR PREVIOUS) YES [ ] NO [ ]

a. INSURED'S DATE OF BIRTH MM DD YY: 12 30 81 SEX M [ ] F [X]

b. OTHER INSURED'S DATE OF BIRTH MM DD YY: SEX M [ ] F [ ]

b. AUTO ACCIDENT? YES [ ] NO [ ] PLACE (State):

b. EMPLOYER'S NAME OR SCHOOL NAME:

c. EMPLOYER'S NAME OR SCHOOL NAME:

c. OTHER ACCIDENT? YES [ ] NO [ ]

c. INSURANCE PLAN NAME OR PROGRAM NAME: MY INSURANCE

d. INSURANCE PLAN NAME OR PROGRAM NAME:

10d. RESERVED FOR LOCAL USE

d. IS THERE ANOTHER HEALTH BENEFIT PLAN? YES [ ] NO [X] *If yes*, return to and complete item 9 a-d.

READ BACK OF FORM BEFORE COMPLETING & SIGNING THIS FORM.

12. PATIENT'S OR AUTHORIZED PERSON'S SIGNATURE I authorize the release of any medical or other information necessary to process this claim. I also request payment of government benefits either to myself or to the party who accepts assignment below.

SIGNED _____ DATE 03/05/2024

13. INSURED'S OR AUTHORIZED PERSON'S SIGNATURE I authorize payment of medical benefits to the undersigned physician or supplier for services described below.

SIGNED _____

14. DATE OF CURRENT: MM DD YY: 03 05 24 ILLNESS (First symptom) OR INJURY (Accident) OR PREGNANCY(LMP)

15. IF PATIENT HAS HAD SAME OR SIMILAR ILLNESS. GIVE FIRST DATE MM DD YY:

16. DATES PATIENT UNABLE TO WORK IN CURRENT OCCUPATION MM DD YY FROM TO MM DD YY

17. NAME OF REFERRING PHYSICIAN OR OTHER SOURCE: MY DOCTOR

17a. I.D. NUMBER OF REFERRING PHYSICIAN: 54321

18. HOSPITALIZATION DATES RELATED TO CURRENT SERVICES MM DD YY FROM TO MM DD YY

19. RESERVED FOR LOCAL USE

20. OUTSIDE LAB? YES [ ] NO [ ] $ CHARGES

21. DIAGNOSIS OR NATURE OF ILLNESS OR INJURY. (RELATE ITEMS 1,2,3 OR 4 TO ITEM 24E BY LINE)

1. _____ 3. _____

2. _____ 4. _____

22. MEDICAID RESUBMISSION CODE ORIGINAL REF. NO.

23. PRIOR AUTHORIZATION NUMBER

24. A DATE(S) OF SERVICE From MM DD YY	To MM DD YY	B Place of Service	C Type of Service	D PROCEDURES, SERVICES, OR SUPPLIES (Explain Unusual Circumstances) CPT/HCPCS	MODIFIER	E DIAGNOSIS CODE	F $ CHARGES	G DAYS OR UNITS	H EPSDT Family Plan	I EMG	J COB	K RESERVED FOR LOCAL USE
1 03 01 24	03 05 24	A	B	123		456	123.45					
02 01 24	02 29 24	C	D	456		789	678.90					

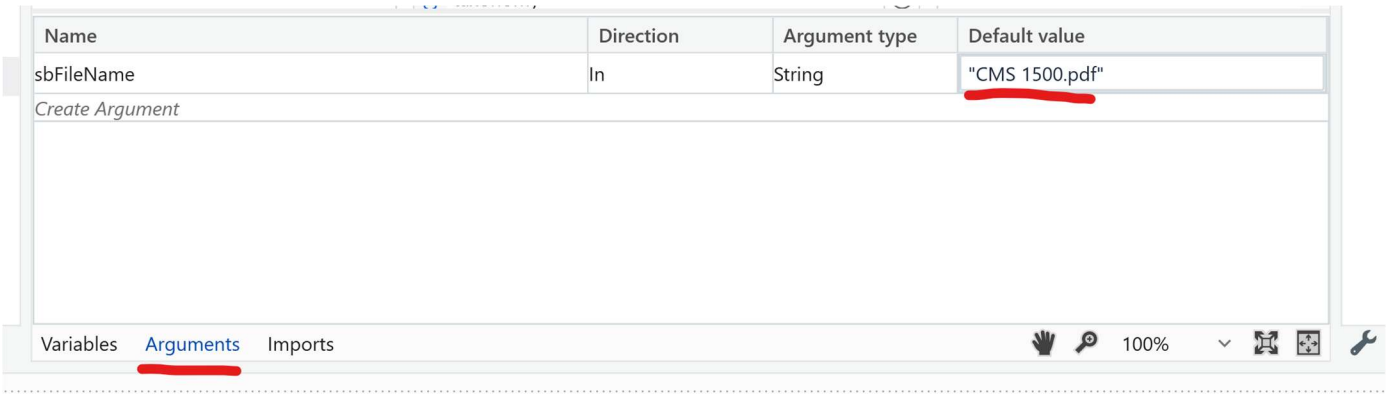
CARRIER

PATIENT AND INSURED INFORMATION

R INFORMATION

- **Argument as File Name for Process**

  The process should have the following sbFileName as Argument pointing . We have kept file CMS 1500 in the parent folder, in case you keep it in subfolder, change the file name value as "<FolderName>/<FileName>"



For Public Preview this would be the invoice file

Company Name: **Beta Technologies**
Address.: **One Drive, Arizona**

Invoice

Bill To: **Alpha Technologies**

Billing Address: 221 Baker Street

Invoice Number.: **1000**
Date: **14/10/2024**

PO Number: **9876**
Date: **09/09/2024**

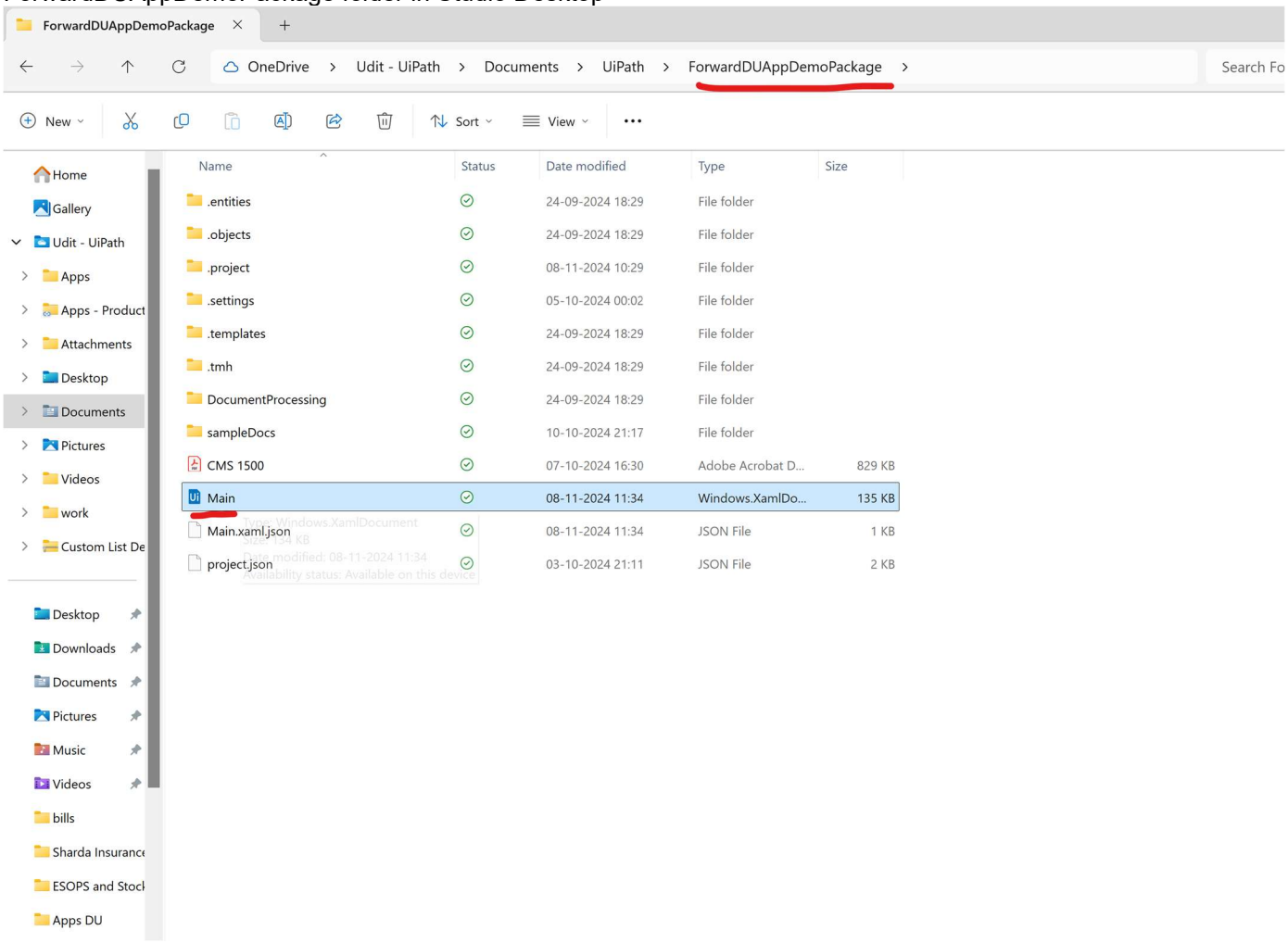
Sl. No.	Product Name	Quantity	Unit	Price/Unit	GST	Amount
1	Brake Pads (Set)	200	1	$150		$30,000
2	Alternators	50	1	$400		$20,000
3	Car Batteries	50	1	$200		$10,000
4	Headlights	100	1	$100		$10,000
5	Fuel Pumps	40	1	$500		$20,000
6	Oil Filters	400	1	$25		$10,000
7	Late Penalty: Net 30		**1**	$20,000		$20,000
					Sub Total	**$120,000**

Sub Total: **$120,000**
Discount: **$ 0**

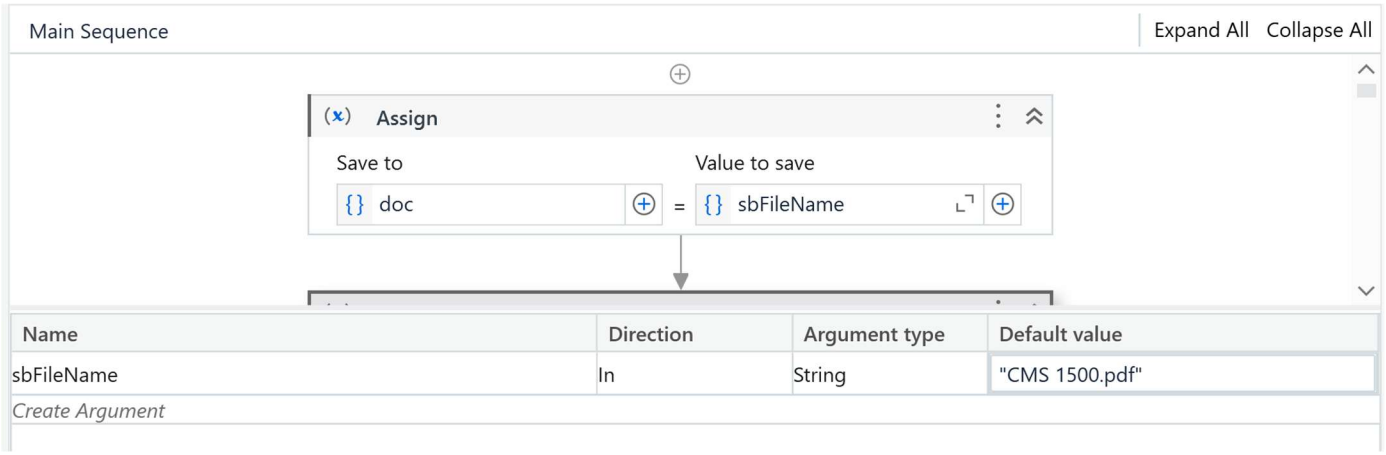
Total: **$120,000**

- **Process Setup**

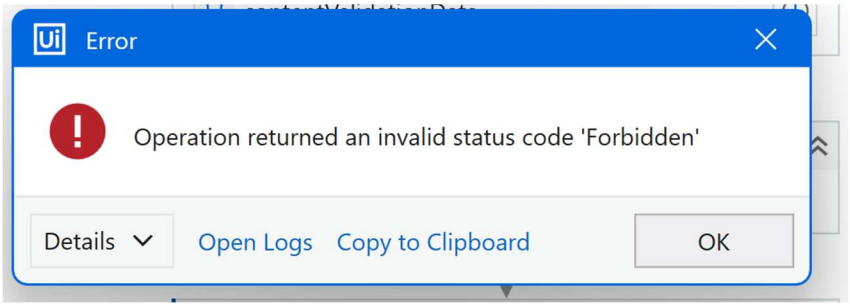
1) Extract the Studio Process from “Private Preview Studio Desktop Process” zip file and open the Main file in ForwardDUAppDemoPackage folder in Studio Desktop



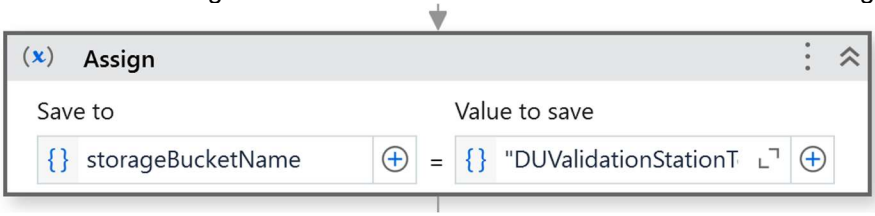
2) Once the project loads, Make sure file name is correct in the first Assign activity, it should be the one as we looked in the prerequisite folder. If you face an error, check if the folder contains the file. We have kept the file in the parent folder, in case you keep it in subfolder, append the file name value as “<FolderName>/<FileName>”



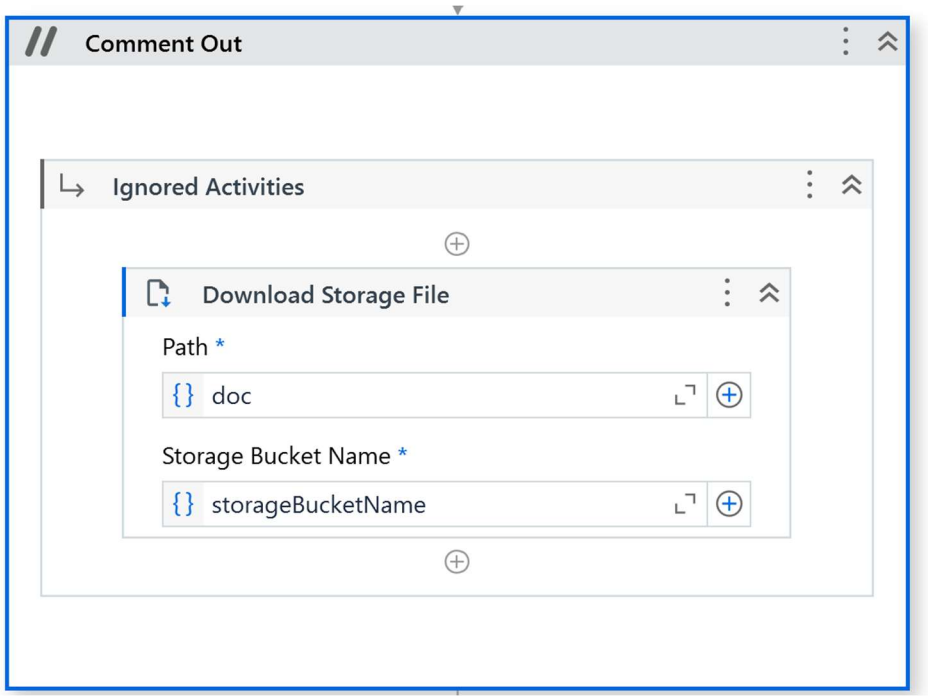
a. Please ignore if you get this error, it will not have any problem in executing and should be fixed with next version



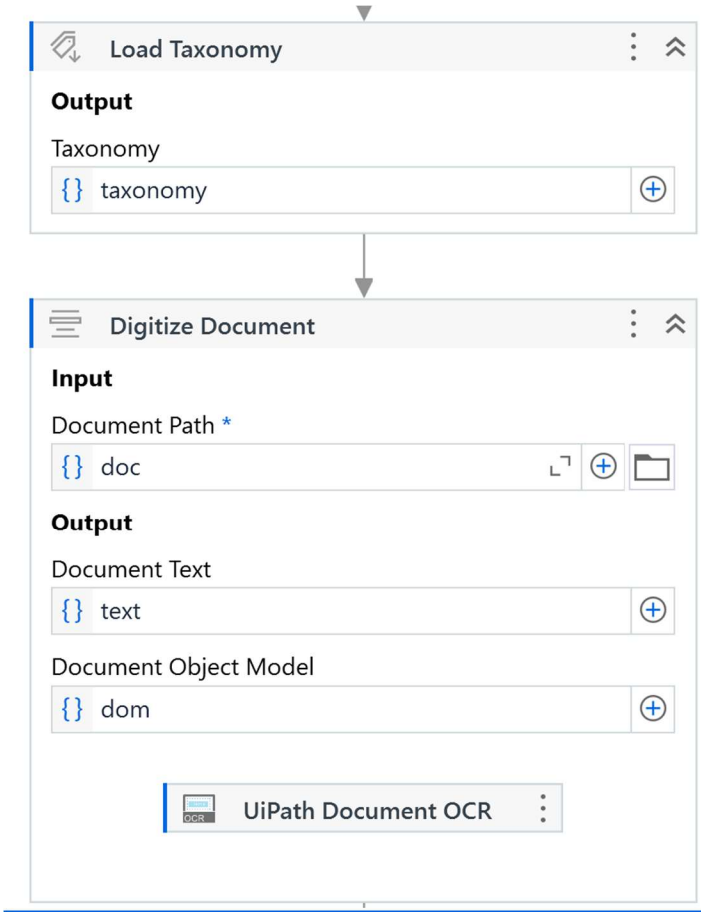
3) Make sure Storage Bucket created above matches the name in the Assign activity

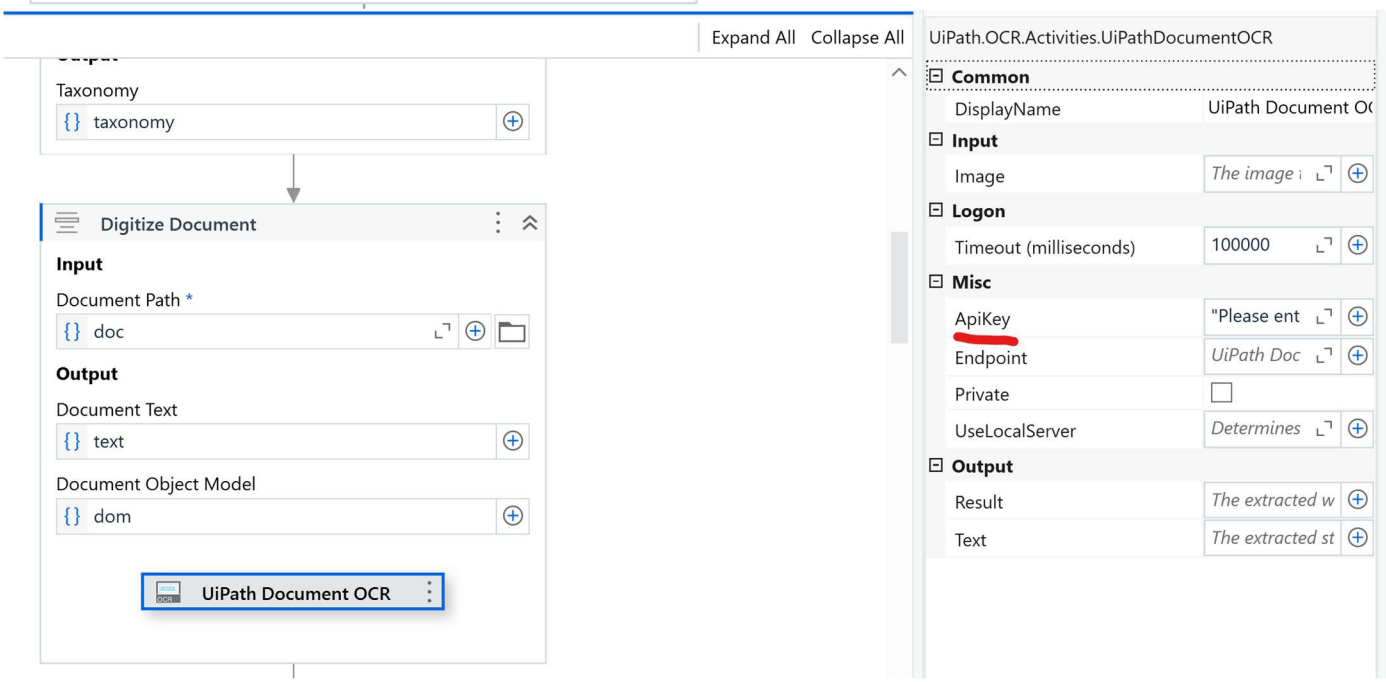


4) In case you wish to use file from storage bucket, you can use the commented out activity instead of the local file we used

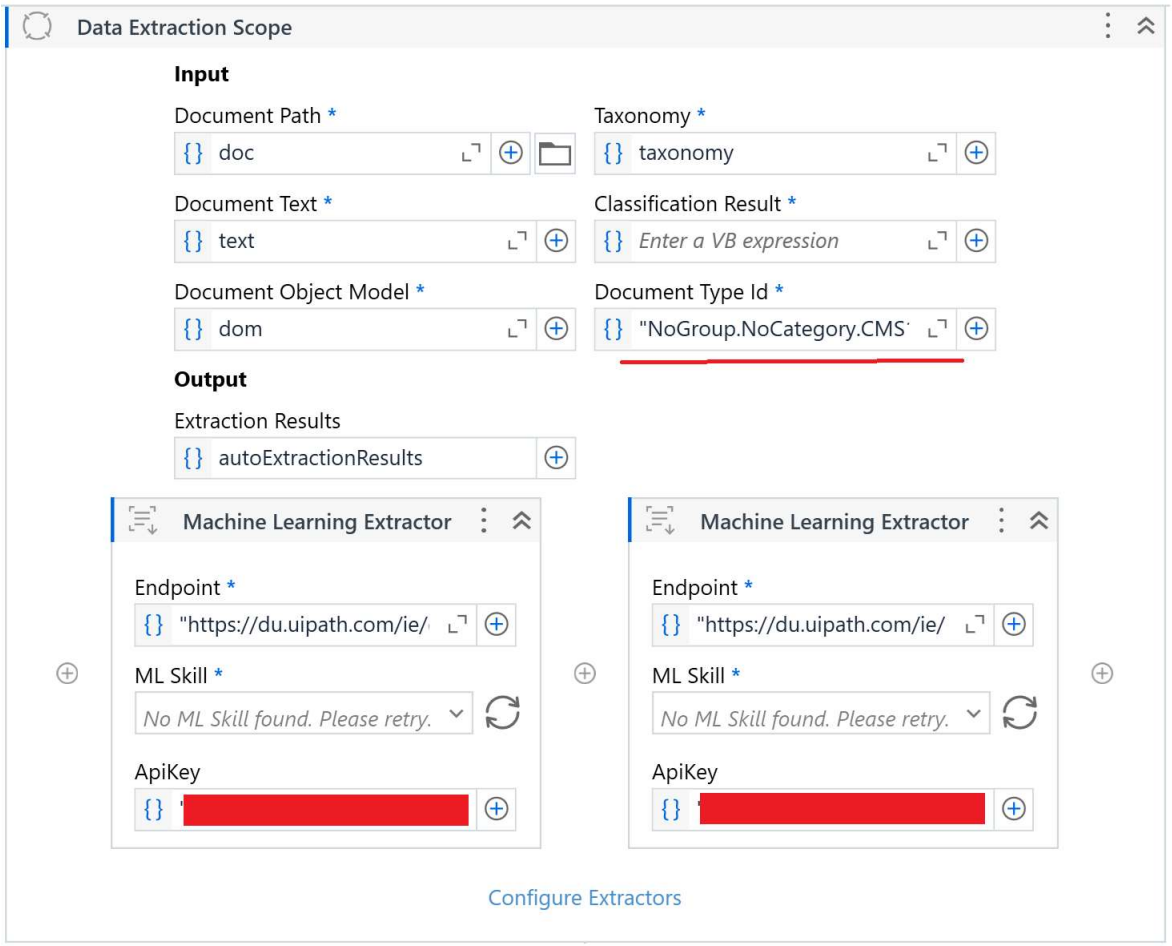


5) Make sure you are using the right Taxonomy and OCR. Also please provide your API key for UiPath OCR or chose your own OCR

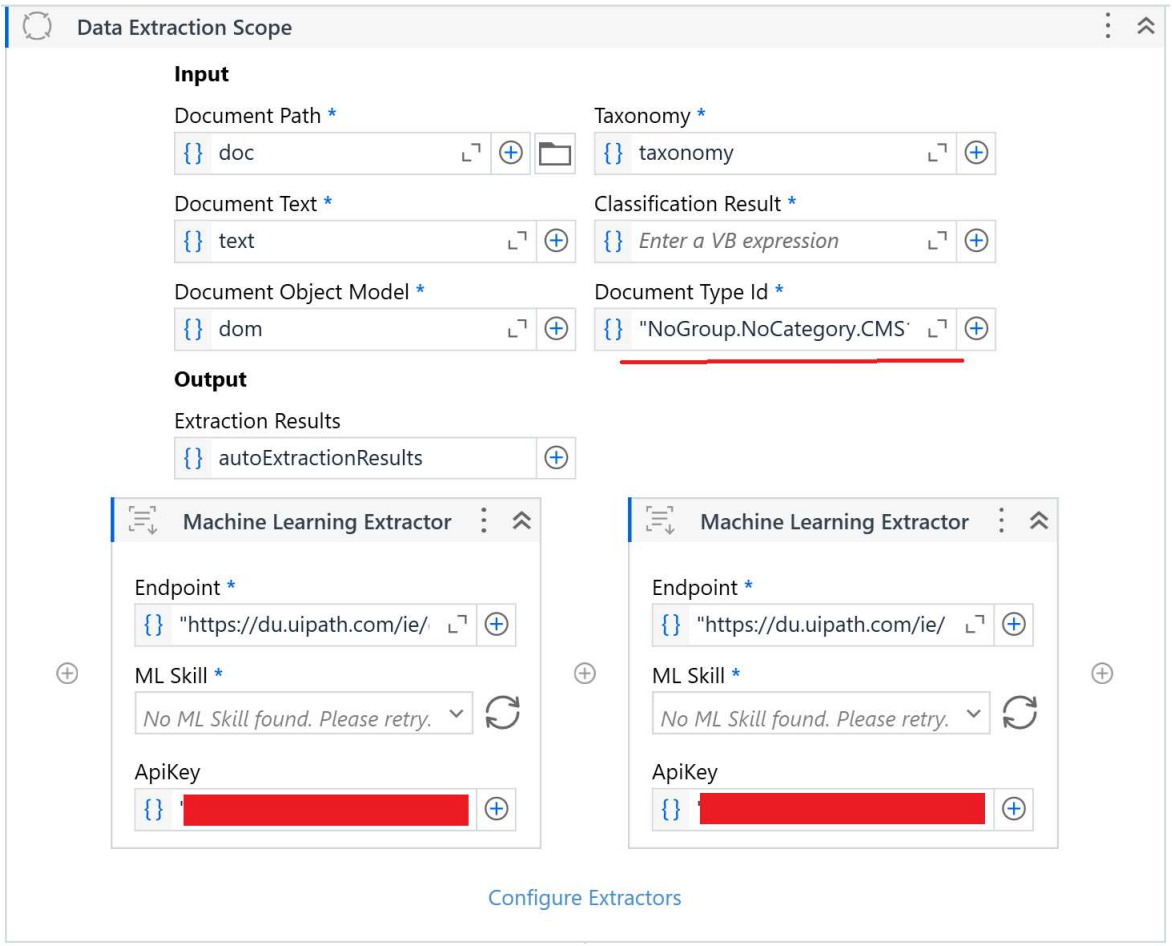




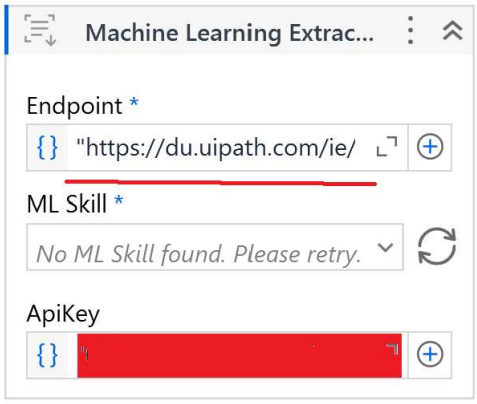
6) Please provide the API key below, you can use single classifier or other extractor based on your setup



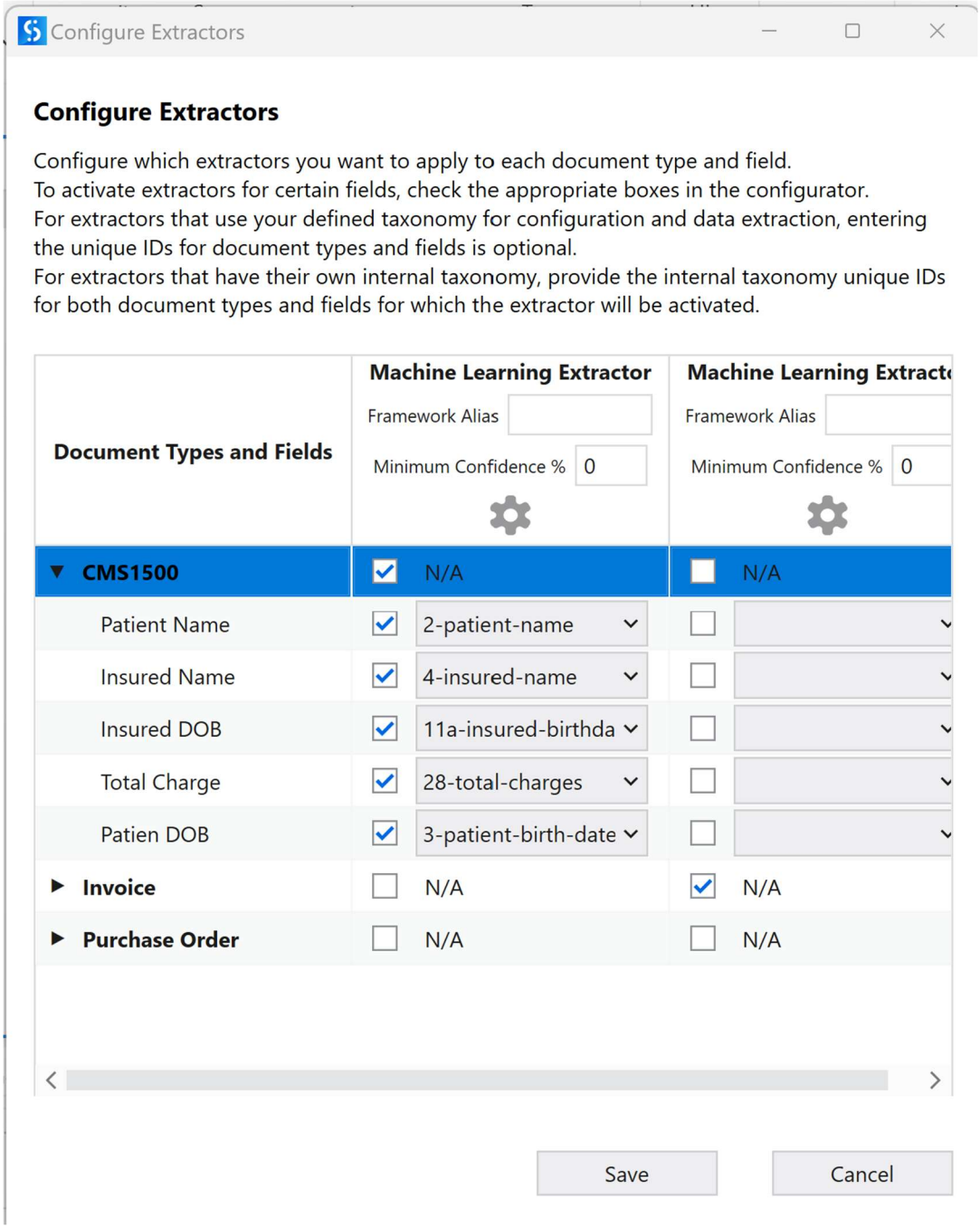
7) Please note we have hardcoded Doc Type ID directly from Taxonomy to be CMS 1500 , you can use a classifier activity if you have multiple document type



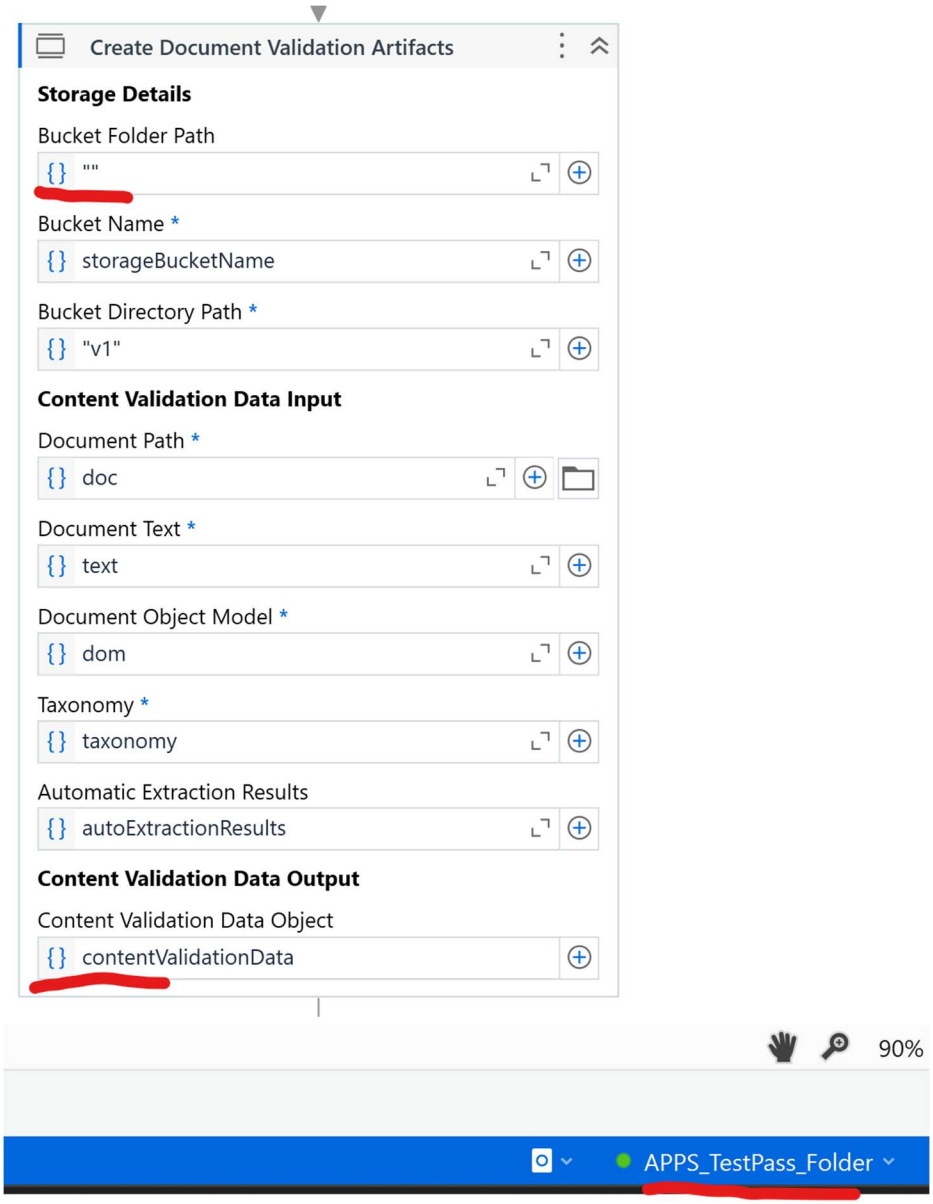
8) Next check the endpoint in the extractor. We have sourced the CMS1500 end point URL (https://du.uipath.com/ie/cms1500) from the below Public URL - https://docs.uipath.com/document-understanding/automation-cloud/latest/user-guide/public-endpoints . You can get the correct end point URL for other document type



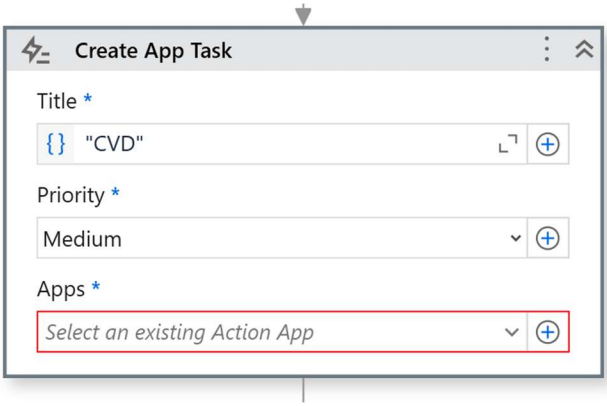
9) Make sure your extractor is configured correctly



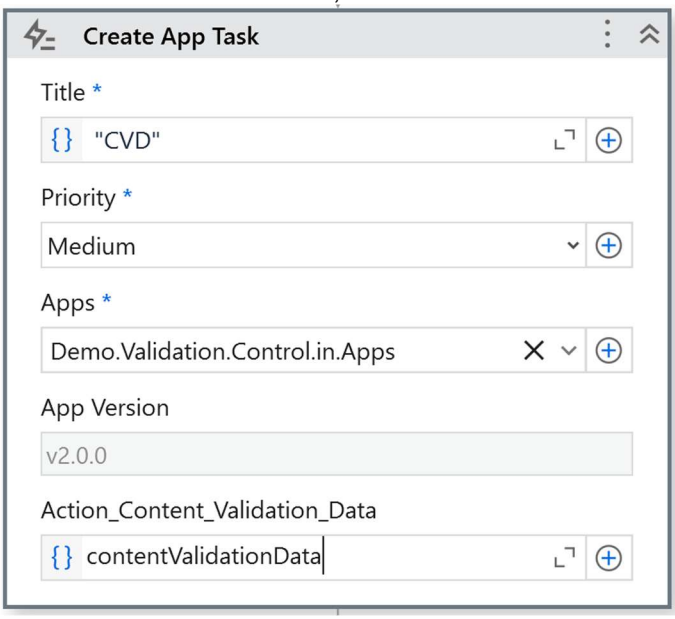
10) Ensure the Create Validation Artifacts activity is configured correctly. You should provide “” if the folder name is same as the one configured in Orchestrator and where Storage bucket is hosted, else please provide the folder path correctly. Also make sure you capture the output of this activity in the right variable as this will be passed to Create App Task as inputs for the app we created



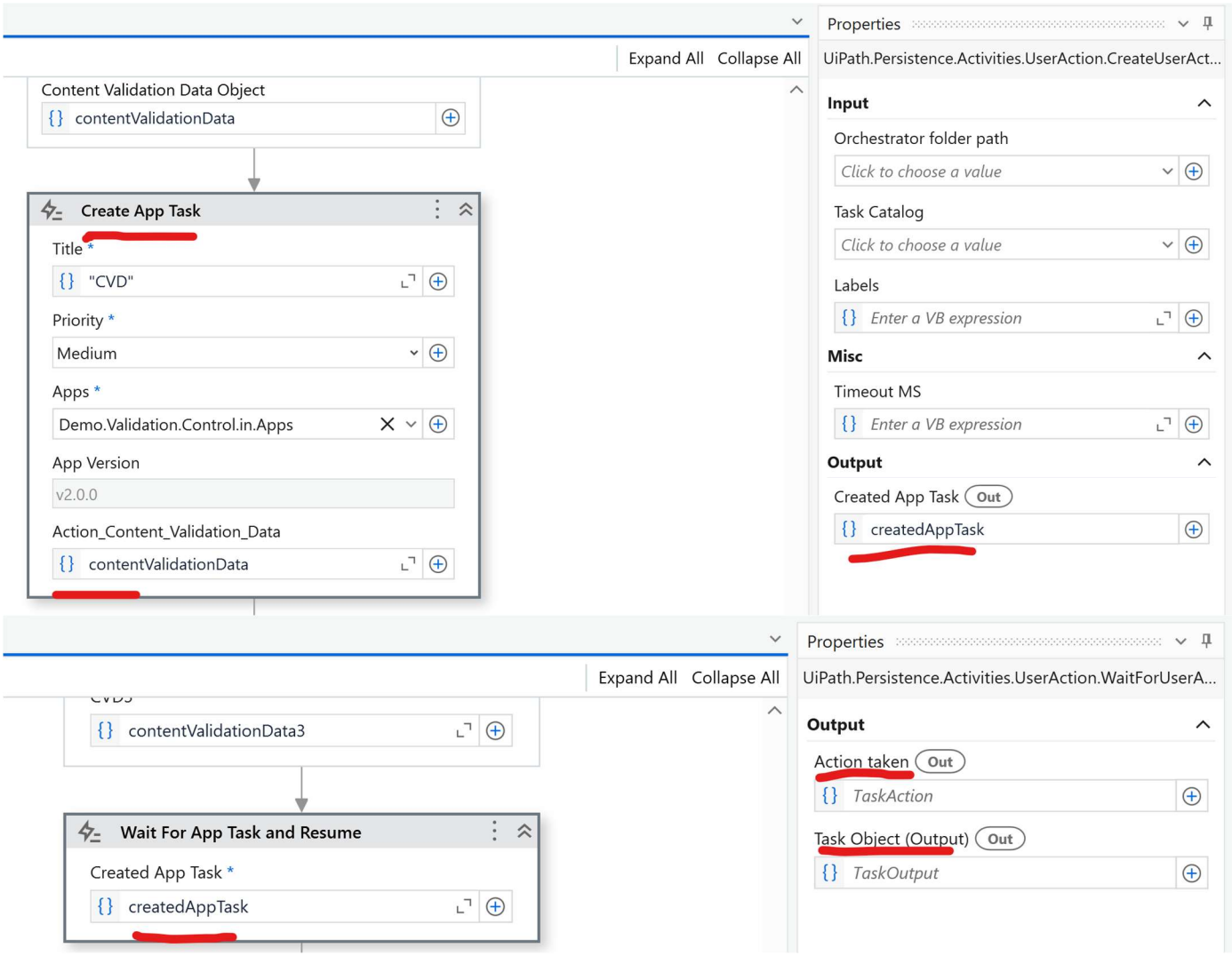
11) Next, we need to select the right published Action App from the list of Action App below. In case you see the Apps drop down not showing your App or wrong list of apps, please check the following
    a. the Persistence activities package version
    b. whether your app is deployed or not, if not deployed, please deploy it first
    c. folder in which your app is deployed is same as the one in Orchestrator, if not then it will show wrong list of Apps



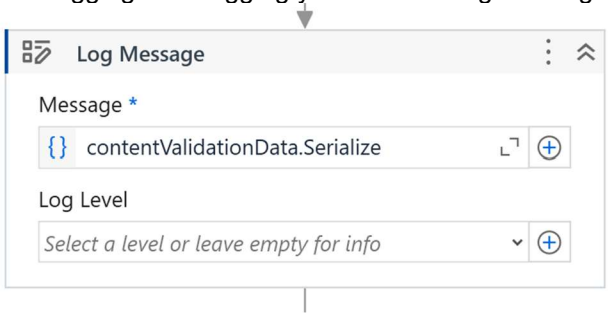
12) You can then provide the inputs for the Action app based on schema of the App. Here we are passing contentValidationData as sample property, make sure the app also has the same data type of Content Validation Data for it to work, else it will throw error of data type mismatch



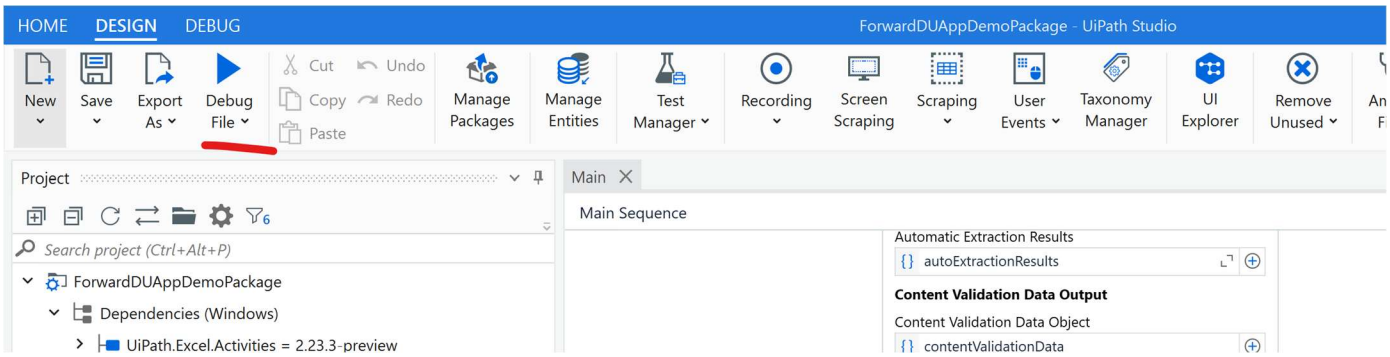
13) Also, you can click on Properties panel for this activity and add relevant Output property (“createdapptask “variable below) to understand whether user clicked on Approve or Reject and this can later be passed to Wait for App Task and Resume processing.
    a. Also, in Action Schema in App you can provide output properties, for this example we have only provided input properties, output properties is available for the process and is additional information you may wish to pass back to triggering process apart from Action Taken



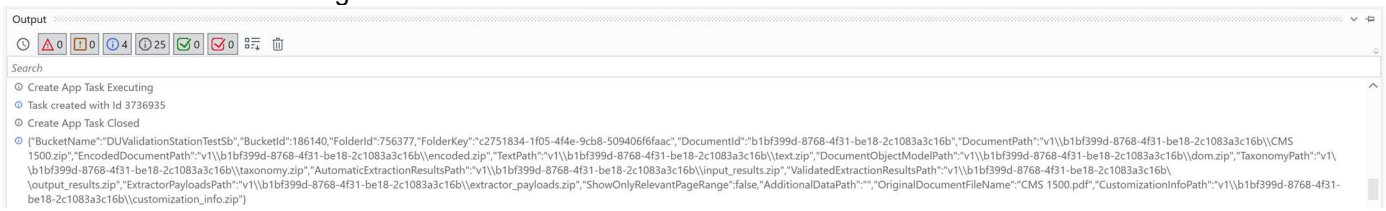
14) Whenever you update your app, it would be best to update the create app task as well, especially if you change number of parameters and then save your process, this is to avoid run time error, else it will pick older deployed version
15) For Debugging and Logging you can use Log Message activity as below

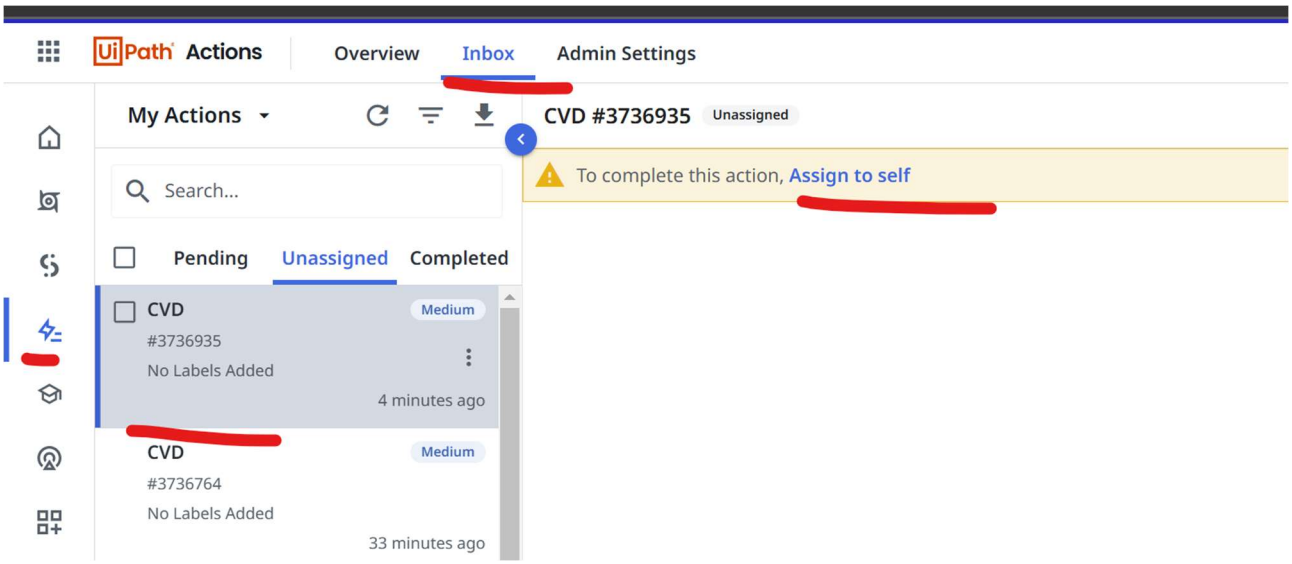


16) Next you can click on Debug File and wait for it to complete run , post which action center task should be created

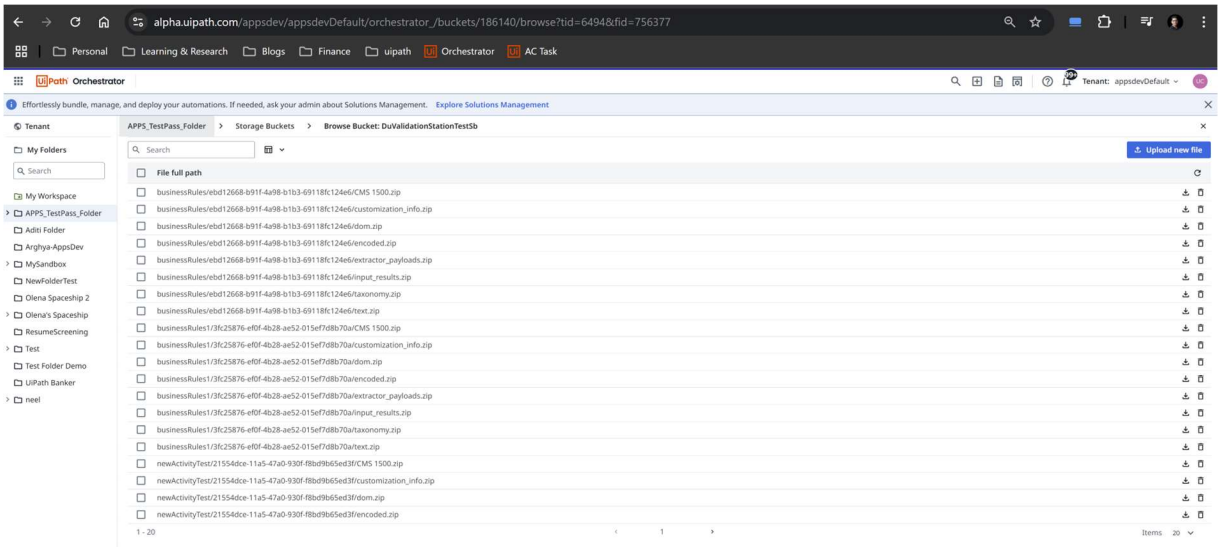


17) Next you can see the Output once everything is completed and then you can check in Action Center once task is created in unassigned section



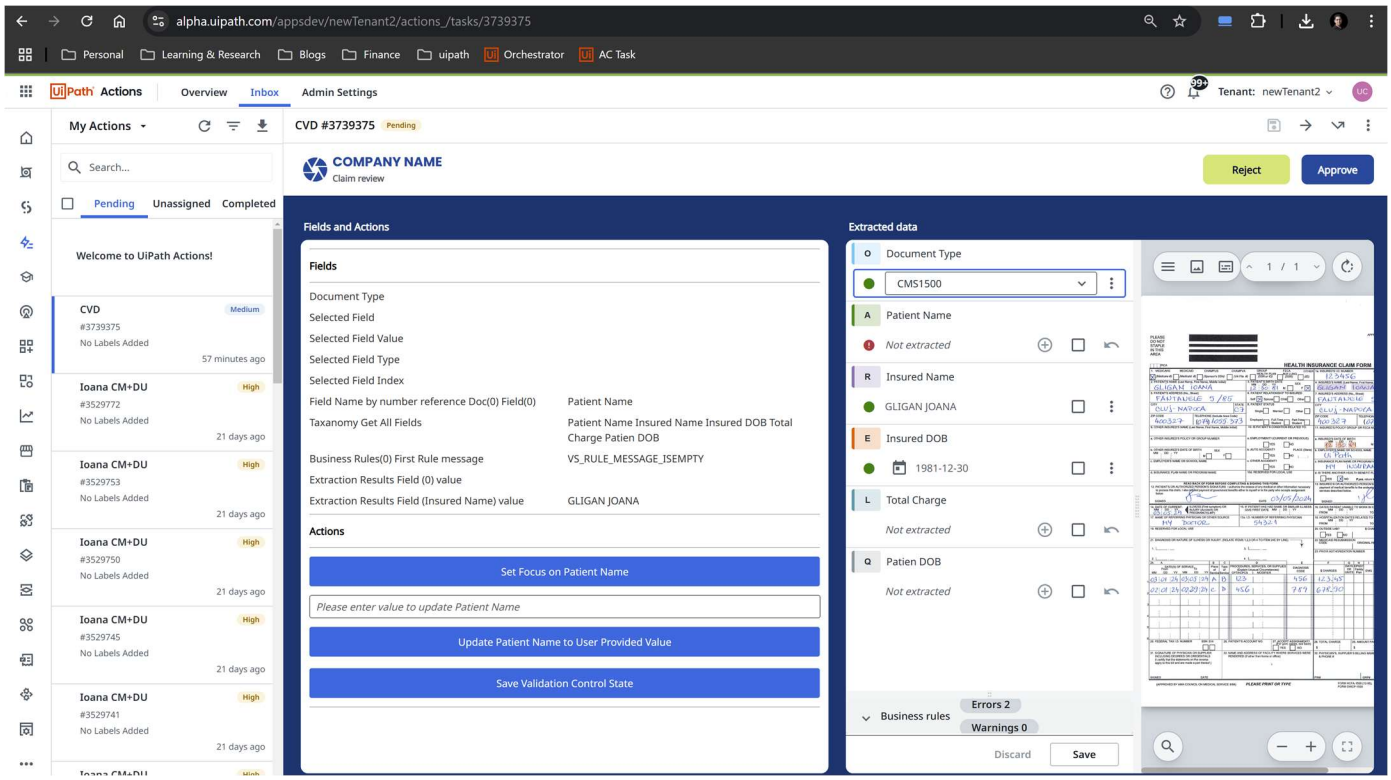


18) Also storage bucket would have now been populated with the results



Step by Step Walkthrough – Run Time

Testing the App
You can see the Action App being rendered in Action Center with VS control rendered and you can check all functionality working



For testing directly in Studio instead of Action Center, please refer debugging section

Best Practices

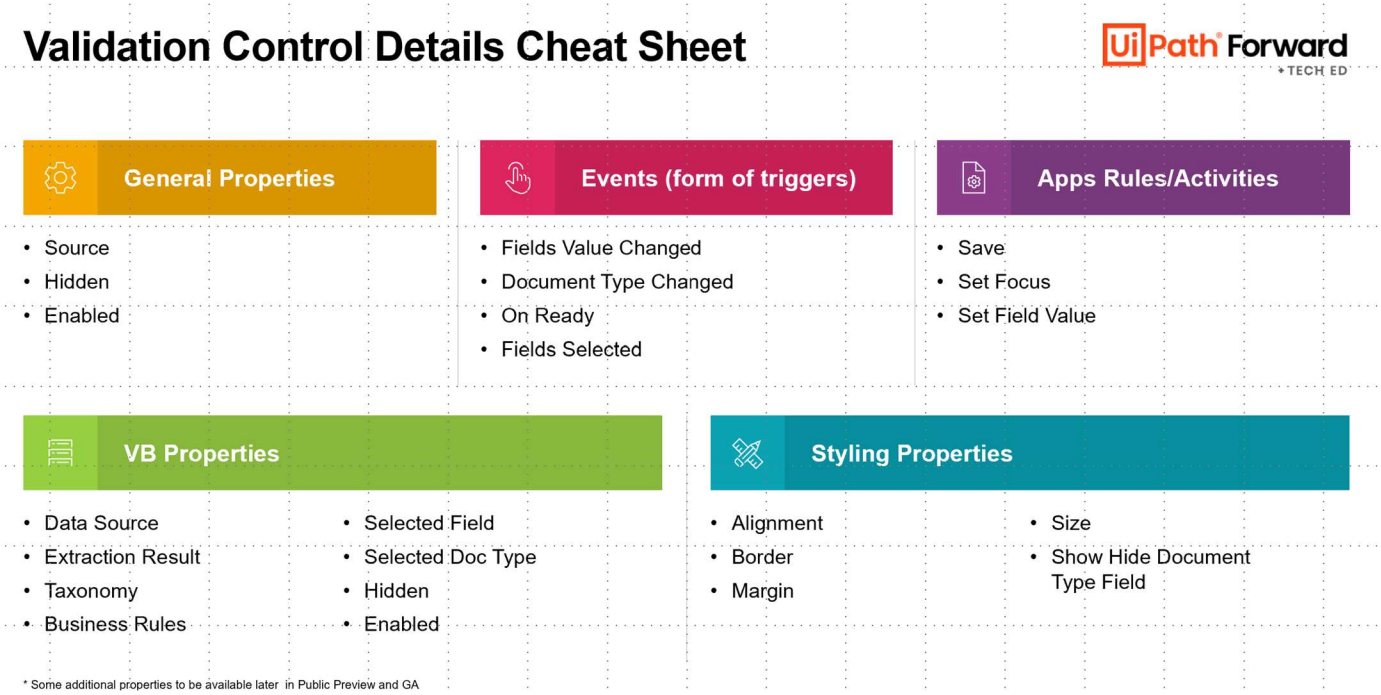
1) Better to upgrade the process associated with create app task activity in case you change the action schema of the action app to avoid any failure on process runs
2) To avoid any data mismatch, call any validation control's field fetch and rendering functions with OnReady event instead of Page OnLoad event
3) Please check for string length to be greater than 0 in case of some fetch operations return NULL; you can use if else with "" assignment for empty string in else condition
4) Please use For Each function to scan through the list of business rules or fields list wherever needed
5) Save function does check if all business rules are passing before saving, we will review this functionality later, so before save, please ensure all document rules in taxonomy are passing
6) If you don't see the Validation Station activities being listed – please follow **Studio Web – Web App Project Setup** section (this restriction is only till control goes GA)

Tips and Tricks

1) If you upgrade your published app, you need not run the Studio process again to update the action task, it will automatically pick up the upgraded process behind the action task
2) If you don't want multiple validation controls in your app, you can have single control and change its source at runtime by binding content validation data variable and changing it via set variable value activity
3) Web Apps have some separate syntax learning due to Standalone apps being merged with Studio Web, while working in Studio Web expression editor compared to App Studio expression editor; do refer the activities and property assignments in sample app to understand the syntax well. You can refer to the Apps Academy course and Web Apps documentation in below reference section to understand more. Web Apps Academy course to be provided at the time of Web Apps GA later in 2025

4) If you need to pass 2 or more contentvalidationdata objects to the action app, you can do 2 things in the app
    a. Separate validation controls binded to separate individual contentvalidationdata objects passed in action schema
    b. Single validation control tried to contentvalidationdata object variable and you change value of variable based on some logic and trigger (eg - button click 1 , you assign first contentvalidationdata object, button click 2 , you assign second content validation data object). You can pass contentvalidationdata arrays /list in action schema via create app task

Validation Control Cheat Sheet



References and Other Learning Material

Apps in Studio Web- https://docs.uipath.com/studio-web/automation-cloud/latest/user-guide/apps-in-studio-web

VB Apps Academy Course - https://academy.uipath.com/courses/uipath-apps (refer only Apps Standalone common functionality)

Document Understanding Course - https://academy.uipath.com/courses/uipath-document-understanding-overview

Action Center - https://academy.uipath.com/courses/uipath-action-center

Action Apps - https://docs.uipath.com/action-center/automation-cloud/latest/user-guide/action-definitions