**APWorks – Azure Migration**

**Technical Requirements**



Nexelus USA  
New York, NEW YORK (NY)  
646-558-1950

April 18, 2024

|  |  |  |  |
| --- | --- | --- | --- |
| **Revisions** | | | |
| **Date** | **Version Number** | **Revised By** | **Reason for Changes** |
| **4/18/2024** | **0.1** | **Tauseef** | **Initial Draft** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Introduction 1](#_Toc164355028)

[Table Changes 2](#_Toc164355029)

[Vendor and Invoice Mapping 2](#_Toc164355030)

[Workflow 3](#_Toc164355031)

[Process 3](#_Toc164355032)

[Step 1 – Upload and Scan 3](#_Toc164355033)

[Unmatched Invoices 4](#_Toc164355034)

[Unrecognized Invoices 4](#_Toc164355035)

[Azure Forms Keywords 6](#_Toc164355036)

[Standard fields 6](#_Toc164355037)

[Invoice Line Items (Items field) 6](#_Toc164355038)

[Additional Fields 7](#_Toc164355039)

[Differences in AWS and Azure Form Recognizer objects 8](#_Toc164355040)

[Key Value Pairs 8](#_Toc164355041)

[Field Values 8](#_Toc164355042)

[Field Coordinate 8](#_Toc164355043)

[Tables 8](#_Toc164355044)

[Invoice Recognition Workflow Changes: 8](#_Toc164355045)

# Introduction

APWorks currently utilizes Amazon Textract text detection and analysis API to recognize and identify specific AP Invoice elements to APWorks. APWorks background process (AP Invoice Scanner) scans preconfigured client directory that is setup in advance to collect initially all client AP invoices. The biggest hurdle to AWS Textract implementation is that it requires creation of models for every vendor and its invoice types such as Invoice and Credit Memo.

Microsoft has launched a new service Microsoft Azure Form Recognizer, which eliminates the need for invoice models for most invoices.

The scope of this document is to document required changes to implement Microsoft Azure Form Recognizer as substitute for AWS Textract. This is a different tech stack and requires fundamental changes in how invoices are processed and scanned in APWorks.

# Table Changes

One of the major changes is to decouple and loosely bind models from vendors. Models will be created without specifying Vendor Code. A separate table will be created to link a model with a vendor and client currency. The advantage is to reuse same model across

### Vendor and Invoice Mapping

Following new tables will be added:

**Database**: Company  
**Table:** apam\_document\_vendor\_mapping  
**Description**: [blubs].

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Len** | **Precision** | **NN** | **PK** |
| company\_code | Int |  |  | Y | Y |
| Vendor\_mapping\_id | Int |  |  |  |  |
| Model\_def\_id | int |  |  |  |  |
| Vendor\_Identifier | Varchar |  |  |  |  |
| Sample\_file\_location | Varchar |  |  |  |  |
| Vendor\_code | Varchar |  |  |  |  |
| Currency\_code | Varchar |  |  |  |  |
| Terms\_code | Varchar |  |  |  |  |
| Tax\_code | Varchar |  |  |  |  |
| Detail\_flag | Bit |  |  |  |  |
| Master\_document\_model\_id | Bit |  |  |  |  |
| Active\_flag | bit |  |  |  |  |

**Database**: Company  
**Table**: apam\_document\_vendor\_mapping\_detail  
**Description**: [blurbs].

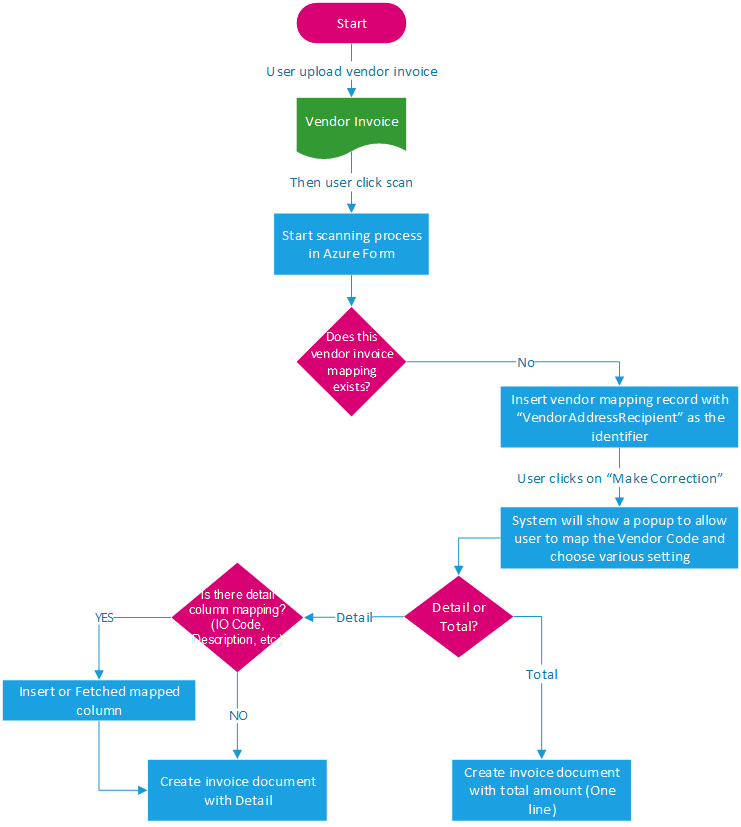
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Len** | **Precision** | **NN** | **PK** |
| company\_code | Int |  |  | Y |  |
| Vendor\_mapping\_id | int |  |  |  |  |
| Model\_def\_id | int |  |  |  |  |
| Po\_code\_map | Varchar |  |  |  |  |
| Description\_map | Varchar |  |  |  |  |
| Level2\_map | Varchar |  |  |  |  |
| Level3\_map | Varchar |  |  |  |  |
| Expense\_type\_map | Varchar |  |  |  |  |

**Database**: Company  
**Table**: APAM\_UNMAPPED\_INVOICES  
**Description**: Invoices with no vendor mapping will be saved in this table. Once vendor mapping is created, these invoices will be moved to their respective invoice tables based on their vendor mapping settings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Len** | **Precision** | **NN** | **PK** |
| DocumentID | AutoIncrement |  |  | Y | Y |
| company\_code | Int |  |  | Y |  |
| Model\_definition\_id | int |  |  |  |  |
| InvoiceID |  |  |  |  |  |
| Vendor\_Identifier | Varchar |  |  |  |  |
| File\_Path | Varchar |  |  |  |  |
| Scan\_Date | TimeStamp |  |  |  |  |
| Total\_Pages | int |  |  |  |  |
| Content | JSON |  |  |  |  |
|  |  |  |  |  |  |

# Workflow

The purpose of invoice matching process is to find appropriate Invoice Model with mapped ERP Vendor/Site and extract all necessary invoice data elements to be able to generate invoice record which can be further routed through approval process and finally posted to ERP System without errors and/or missing information

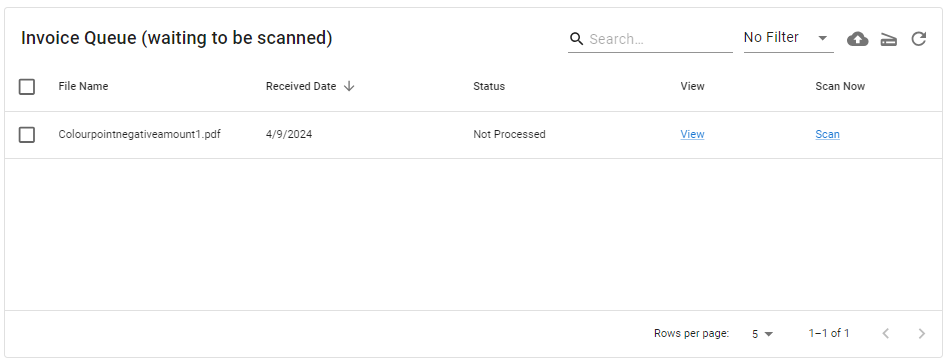


## Process

The user will upload invoices on dashboard in Invoice Queue session.

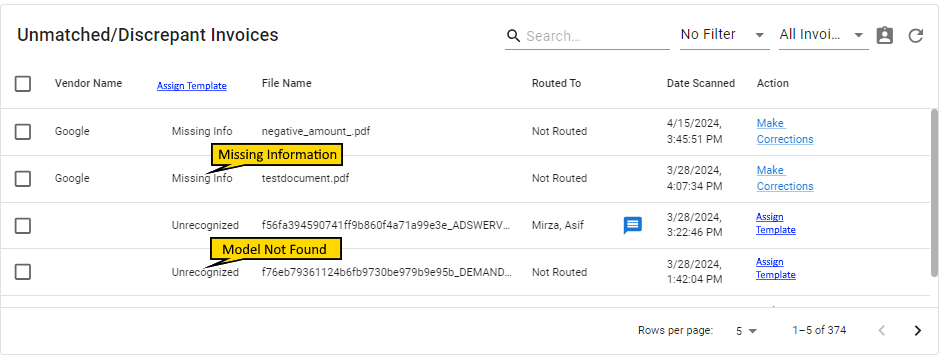
## Step 1 – Upload and Scan

The user will upload invoices in Invoice Queue section of Dashboard and click on scan button.



The system will process the selected invoices. If all information is provided, it will move invoices to pending Approval section of dashboard.

if there is some missing information or no ERP vendor is mapped against vendor identifier for this invoice, These invoices will also move to Unmatched/Discrepant invoice section.



### Unmatched Invoices

User will click on “Make Correction“ link on invoices with Missing information status and enter values not found automatically against invoice. No change in this workflow is required.

### Unrecognized Invoices

If no vendor is mapped against vendor identifier for an invoice, it will display open invoice model UI. The user will provide vendor mapping and default values to use for this vendor.



User will provide mapping for IO/PO information and click save. This will create the mapping and apply to all discrepant invoices with same Vendor Identifier.

Note: Since Azure Form Recognizer uses standard identifiers for fields such as Invoice Number, Invoice Date, etc., there is no need for labels anymore.

# Azure Forms Keywords

## Standard fields

Azure form recognizer returns Invoice elements using following keywords.

|  |  |
| --- | --- |
| Keyword | Description |
| InvoiceID | Invoice Number |
| InvoiceDate | Invoice Date |
| DueDate | Invoice Due Date |
| PaymentTerm | Payment Terms |
| VendorName | Vendor Name or Identifier |
| VendorAddressRecepient | Qualified Vendor Name |
| SubTotal | Invoice Value – Invoice Value includes sub properties for Value, Currency Code and Currency Symbol |
| TaxTotal | Total Tax Value |
| InvoiceTotal | Invoice Total Amount including Tax (If Tax is specified on Invoice) |
| TotalDiscount | Total discount on Invoice Amount |
| AmountDue | Net amount payable to vendor |
| PurchaseOrder | Purchase Order Number |
| ServiceStartDate | Service Terms Start Date |
| ServiceEndDate | Service Terms End Date |
| TaxItems | AN array that holds added tax information such as CGST, IGST, and SGST. This line item is currently only available for the Germany (de), Spain (es), Portugal (pt), and English Canada (en-CA) locales |
| Items | List of Invoice line details – (Provided in table below) |

### Invoice Line Items (Items field)

|  |  |
| --- | --- |
| Name | Description |
| Items | Full string text line of the line item |
| Amount | The amount of the line item |
| Description | The text description for the invoice line item |
| Quantity | The quantity for this invoice line item |
| OrderQuantity | The ordered quantity for this line item. May differ from the quantity shipped and invoiced |
| UnitPrice | The net or gross price (depending on the gross invoice setting of the invoice) of one unit of this item |
| ProductCode | Product code, product number, or SKU associated with the specific line item |
| Unit | The unit of the line item, e.g., kg, lb, etc. |
| Date | Date corresponding to each line item. Often, it's a date the line item was shipped |
| Tax | Tax associated with each line item. Possible values include tax amount and tax Y/N |
| TaxRate | Tax Rate associated with each line item. |

## Additional Fields

The following fields are specific to Nexelus/APWorks and are not returned by Azure Form Recognizer. These fields need to be mapped.

|  |  |
| --- | --- |
| Keyword | Description |
| Insertion Order |  |
| Level2 Key |  |
| Level 3 Key |  |
| Expense Type |  |
| ERP Tax Code |  |

# Differences in AWS and Azure Form Recognizer objects

Following initial differences are found in AWS Textract and Azure Form Recognizer objects.

## Key Value Pairs

AWS provides Key Value pairs based on text in Invoice Document. These values do not confirm to any standard values and will be different for different vendor invoices

Invoice Number key value pair may be (Invoice Number, 4562) or (Inv No., 4562) based on text in invoice document.

Microsoft Azure Form Recognizer returns values against standard identifiers, such as InvoiceID, InvoiceDate, etc. In this case Invoice number will always be returned against InvoiceID variable, regardless of text in invoice document.

## Field Values

AWS provides values as text and are not automatically translated into number, and dates. We need to pragmatically convert these text to proper values.

Microsoft Azure Form Recognizer automatically converts numbers and dates to proper number and date values. Invoice and Tax values additionally include currency code and currency Symbol values and amounts specified individually.

## Field Coordinate

AWS Textract provides coordinates, width, and height of each field as ratio of the overall document page width and height regardless of document type (PDF or image file).

Microsoft Azure Form Recognizer provides field coordinates and size in inches for PDF files. Field Coordinates and size are in pixels for image files.

## Tables

AWS Textract returns table objects with header and footer row specifiers.

Microsoft Azure Form Recognizer returns number of rows and columns in each table and all individual cells as an array.

## Invoice Recognition Workflow Changes:

AWS requires invoice documents to be placed on an AWS S3 bucket. It picks up the PDF file for recognition and places extracted information as JSON on a specified destination folder in S3 bucket. The application then picks up the JSON file from S3 bucket for further processing from this location.

Microsoft Form Recognizer requires an invoice to be uploaded for processing and returns extracted information directly to application. This greatly simplifies the invoice processing process. There will be a single Lambda function to scan PDF files, and move the scanned invoices to Discrepant or Pending Approval section

# Invoice Editing Changes

The following changes will be made to Invoice Editing.

* Merge Invoice Lines
* Assign Multiple PO to Invoice Line

## Merge Invoice Lines

Application will allow merging multiple invoice lines with same PO/IO number.



### Process:

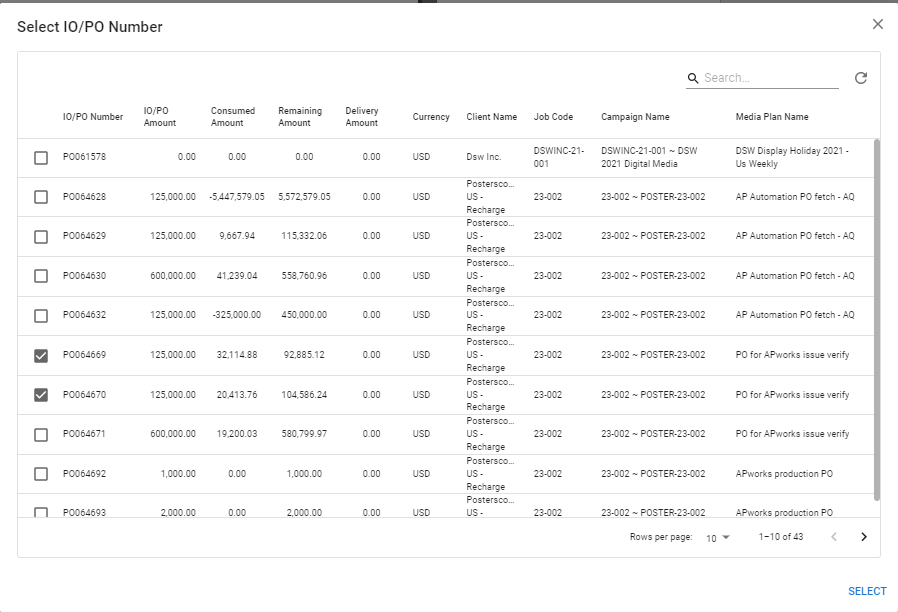
The user will select checkboxes against multiple lines and click merge button. If PO number is same against the selected lines, Invoice Net Amount and Tax Amount will sum up for these lines in first selected line and delete the remaining lines from invoice detail grid.

Note: The IO/PO number must be the same for all selected lines. If selected lines do not have same IO/PO number, the merge button will be disabled.

## Assign Multiple IO/PO to Same Invoice line

This feature allows assigning multiple IO/PO numbers to the same invoice line.

When user clicks on search button for IO/PO number, the popup Dialog will allow selecting more than one IO/PO number. This will be allowed only if multiple invoice lines are not selected. Otherwise only one IO will be allowed to be selected as system is currently working.



When a user selects a single IO/PO number, there will be no change in the current behavior of the application. However, if user selects multiple IO/PO numbers, system will verify if sum of remaining amount for selected IO/PO numbers is same as invoice line amount. If the amount is not the same, then a warning message will be displayed. However, if both values are same, then the system will create additional copies of invoice lines (Sum of selected PO lines minus one) and assign selected IO/PO numbers to selected invoice line and newly created lines.