 Microsoft Power Automate Desktop

Lab 04: Text Extraction from PDF Documents with OCR

Hands-on lab step-by-step

August 2024

Microsoft Power Automate Desktop – Advanced Workshop

Contents

[Microsoft Power Automate Desktop 1](#_Toc173486920)

[Goals for this lab 1](#_Toc173486921)

[Prerequisites 1](#_Toc173486922)

[Exercise: Extract purchased items from PDF invoices 2](#_Toc173486923)

[Terms of Use 12](#_Toc173486924)

# Microsoft Power Automate Desktop

This lab is subject to the Terms of Use found at the end of this document.

## Goals for this lab

|  |  |
| --- | --- |
| After this lab you will be able to:   * Extract text from PDF documents with OCR and specify text regions with Regular Expressions * Tabulate OCR-extracted text in an Excel worksheet | The time to complete  this lab is [60] minutes. |

## Prerequisites

The labs have been designed so if you have access to a Microsoft Power Automate Desktop trial, you can get started from most labs without having to complete the previous module to be able to move forward. However, for the best experience that shows the features and functionality that is possible within the product, it is recommended you have completed specific modules before starting some of the labs.

For Lab 04: Text Extraction from PDF Documents with OCR, you need:

* A computer with internet access.
* The application Power Automate Desktop installed in your computer. If you don’t have the application installed, please download it here: <https://go.microsoft.com/fwlink/?linkid=2102613>
* Be able to log into your corporate tenant.

## Exercise: Extract purchased items from PDF invoices

### Task 1: Download sample invoices from the repository

1. Navigate to <https://github.com/mcoloradodevs/PowerAutomateDesktopTraining/blob/main/Sample%20Invoices.zip> and download the zip folder named **‘Sample Invoices.zip’**
2. Once downloaded, unzip the file and store it in your Downloads folder or another location for easy access later in the exercise.

### Task 2: Log into Power Automate Desktop

1. Open the **Power Automate Desktop** app on your computer
2. Log into the application using your corporate account
3. If you don’t have a Power Automate Premium license, start a trial by clicking on the **Go Premium** button at the top right corner of the application

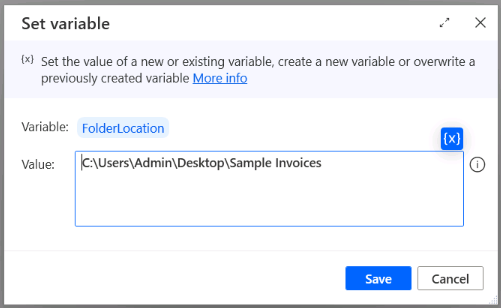


### Task 3: Read the PDF invoices with OCR

1. Make sure you create your automations in your own **Personal Development environment**. If you don’t have one, follow the steps here: [Get your developer environment - Power Apps | Microsoft Learn](https://learn.microsoft.com/en-us/power-apps/maker/maker-create-environment#create-your-own-developer-type-environment)

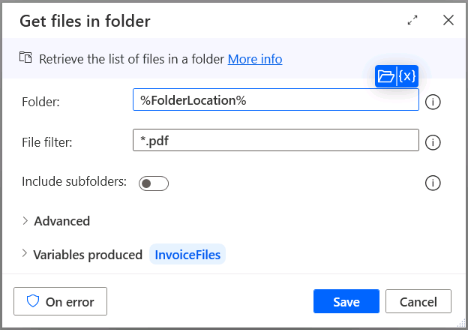
**Special note for Walmart 💡:** Power Platform administrators have blocked the option to create Personal Development Environments in the tenant. For this training, you can use the default environment titled **Walmart Store Inc.**, but in production scenarios, it is a best practice to develop the automations on a dedicated environment. Contact your team leader for more information about your dedicated environment.

1. Click on **+ New Flow** and create a flow named Lab 04: Text Extraction from PDF Documents with OCR Add your name at the end of the flow to recognize it easily.
2. Once on the flow designer, look for action **Set variable**
3. Create a new variable named FolderLocation and assign the value of the ‘Sample Invoices’ folder location. For example: C:\Users\Admin\Desktop\Sample Invoices

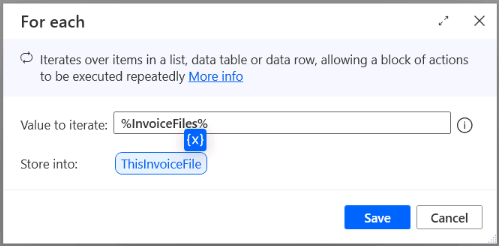


1. Add the action **Get files in folder** and configure its parameters as shown here:

* **Folder:** %FolderLocation%
* **File filter:** \*.pdf
* **Include subfolders:** No
* **Variables produced:** InvoiceFiles

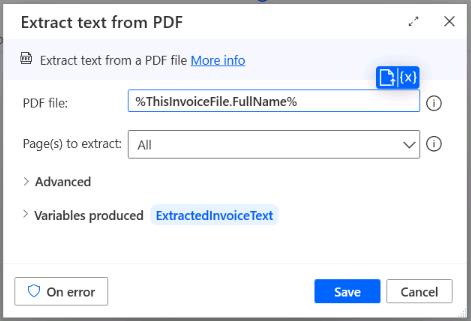


1. Add a **For each** loop to iterate your actions for every invoice file on the folder. Under **Value to iterate**, select the variable %InvoiceFiles%. Rename the **Store into** variable to ThisInvoiceFile



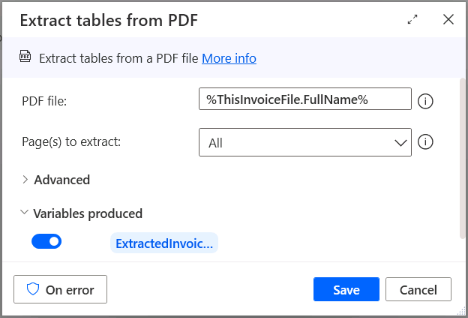
1. Drag the action **Extract text from PDF** and drop it into the **For each** loop. Configure it as shown:

* **PDF file:** %ThisInvoiceFile.FullName%
* **Page(s) to extract:** All
* **Variables produced:** ExtractedInvoiceText



1. Add the action **Extract tables from PDF** below the previous action and configure it as shown:

* **PDF file:** %ThisInvoiceFile.FullName%
* **Page(s) to extract:** All
* **Advanced > Merge tables that cross page margins:** Yes
* **Advanced > First line contains column names:** Yes
* **Variables produced:** ExtractedInvoiceTables



1. **Run** your flow and inspect the results of the two variables: ExtractedInvoiceText and ExtractedInvoiceTables. Notice that the variable ExtractedInvoiceTables contains 3 records, where the third one corresponds to the purchased products information. We will store this table on an independent data table later in the lab

A screenshot of a computer

Description automatically generated

### Task 4: Write Regular Expressions to identify fields in the OCR-extracted text

You will have noticed that the text you extracted from each invoice is not tagged, which means that we need to manipulate it to extract certain information, like the invoice date, ID, or customer name. To achieve this, you will write a regular expression.

1. Open your browser of choice (MS Edge, Chrome), and navigate to <https://copilot.microsoft.com/>

**Note:** You don’t need a Copilot for M365 Enterprise License for this step, since you will be interacting with the Copilot for Web, available for commercial users.

1. Write the following prompt:

*Compose a Regular Expression (RegEx) to extract the line of text after the word ‘DATE’. The text to be extracted is a date in the format Aug 02, 2024. Text to analyze:*

*INVOICE DATE Nov 9, 2019 INVOICE NO 1905 ADATUM CORPORATION 234 France Ave WA 76328 www.adatum.com sales@adatum.com INVOICE TO Fabrikam, Inc. 345 North St NY 98052 Customer Id: 5791 SALESPERSON PAYMENT TERMS Klarissa Wolf Due on Receipt QUANTITY DESCRIPTION UNIT PRICE LINE TOTAL 02 Type S1 filter $11.47 $22.94 02 Pool winterizing $53.99 $107.98 01 Pool connection hose $35.97 $35.97 Subtotal: $166.89 Sales Tax: $3.34 Total: $170.23*

1. Copilot will reply with a regular expression like this one. Remember that your answers may vary due to the nature of GPTs:

DATE\n([A-Za-z]{3}\d{1,2}, \d{4})

1. Write the following prompt. This will test the regular expression and help you validate it:

*Apply the Regular Expression* DATE\n([A-Za-z]{3}\d{1,2}, \d{4}) *to the following text:*

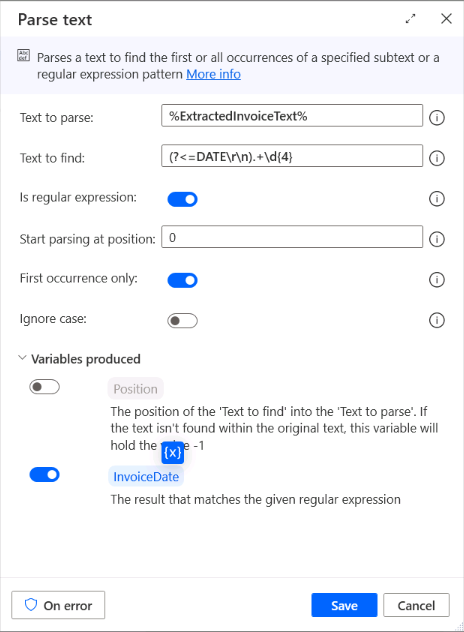
*INVOICE DATE Oct 23, 2016 INVOICE NO 1783 ADATUM CORPORATION 234 France Ave WA 76328 www.adatum.com sales@adatum.com INVOICE TO Fabrikam, Inc. 345 North St NY 98052 Customer Id: 5791 SALESPERSON PAYMENT TERMS Klarissa Wolf Due on Receipt QUANTITY DESCRIPTION UNIT PRICE LINE TOTAL 01 Solar heater mat $59.99 $59.99 03 Water clarifier $9.48 $28.44 Subtotal: $88.43 Sales Tax: $1.77 Total: $90.20*

1. You can use Copilot to write similar expressions to extract additional information from the invoice. For now, the expressions you can use in this lab are listed below. Try to use Copilot to get an explanation of each one of these expressions:

|  |  |
| --- | --- |
| To extract this field… | Use this regular expression |
| Invoice date | (?<=DATE\r\n).+\d{4} |
| Account name | (?<=INVOICE NO\r\n.\*\r\n).+ |
| Invoice number | (?<=INVOICE NO\r\n)\d{4} |

1. Back on the Power Automate designer console, add the action **Parse Text** and drag it below the action to Extract tables from PDF. Configure it as shown below:

* **Text to parse:** %ExtractedInvoiceText%
* **Text to find:** (?<=DATE\r\n).+\d{4}
* **Is regular expression:** Yes
* **Start parsing at position:** 0
* **First occurrence only:** Yes
* **Ignore case:** No
* **Variables produced:** Disable the variable **Position** and rename the variable **Match** to InvoiceDate



1. Copy and paste this action two times to extract the **Account name** and **Email** using the regular expressions from the table in step 5. Rename the produced variables to InvoiceAccountName and InvoiceNumber respectively.

A screenshot of a computer

Description automatically generated

1. **Run** your flow and inspect the values in the variables InvoiceDate, InvoiceAccountName and InvoiceNumber to validate the fields extraction.

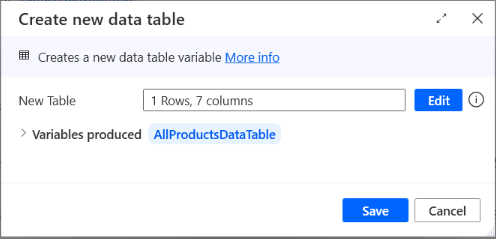
**A screenshot of a computer

Description automatically generated**

### Task 5. Create a data table with the extracted results

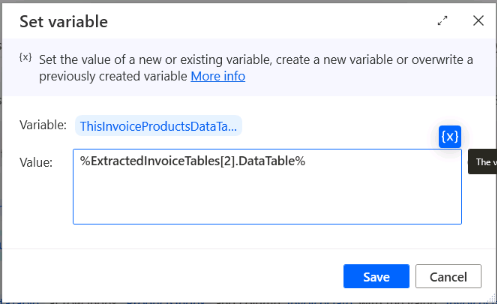
1. After step 2 on your flow, **Get files in folder**, add the action **Create new data table**. Create a table with 7 columns and name them as shown. Rename the data table to AllProductsDataTable:

* Quantity
* Description
* Unit price
* Line total
* Invoice date
* Account name
* Invoice number



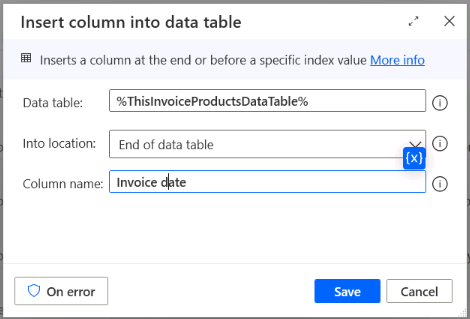
1. Add a **Set variable** action after the last **Parse text** action in your flow. Configure it as shown:

* **Variable:** ThisInvoiceProductsDataTable
* **Value:** %ExtractedInvoiceTables[2].DataTable%

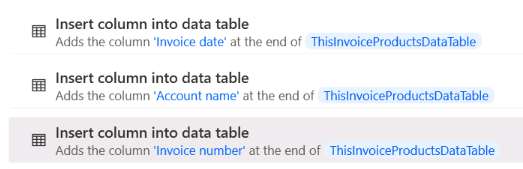


1. Add an action to **Insert column into data table** and configure it as shown here:

* **Data table:** %ThisInvoiceProductsDataTable%
* **Into location:** End of data table
* **Column name:** Invoice date

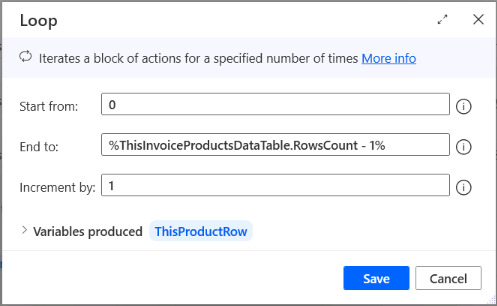


1. Repeat the last step two times to add the columns Account name and Invoice number to the table ThisInvoiceProductsDataTable



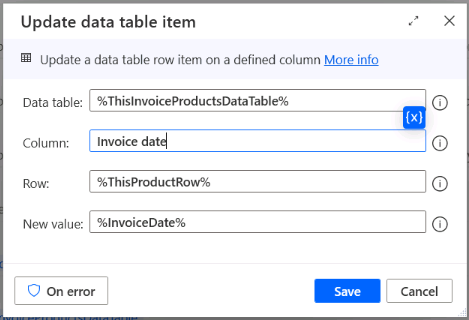
1. Add a **Loop** action right below and configure it as shown:

* **Start from:** 0
* **End to:** %ThisInvoiceProductsDataTable.RowsCount -1%
* **Increment by:** 1
* **Variables produced:** ThisProductRow



1. Drag the action to **Update data table item** within the Loop you just created and configure it as you see next:

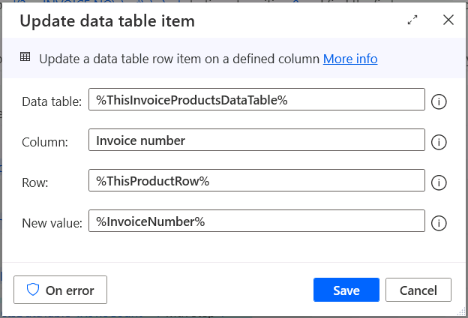
* **Data table:** %ThisInvoiceProductsDataTable%
* **Column:** Invoice date
* **Row:** %ThisProductRow%
* **New value:** %InvoiceDate%



1. Repeat the last action two more times to update the values in columns Account name and Invoice number with the values in variables %InvoiceAccountName% and %InvoiceNumber%, respectively.

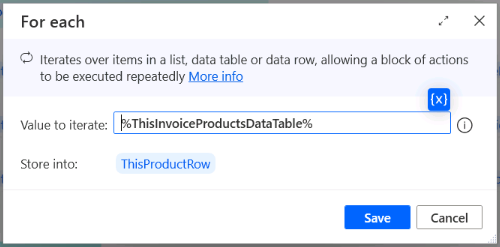
A screenshot of a computer

Description automatically generated



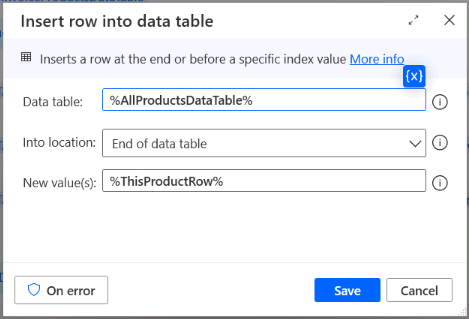
1. After the end of the **Loop** action, add a **For each** action and configure it as shown:

* **Value to iterate:** %ThisInvoiceProductsDataTable%
* **Store into:** ThisProductRow

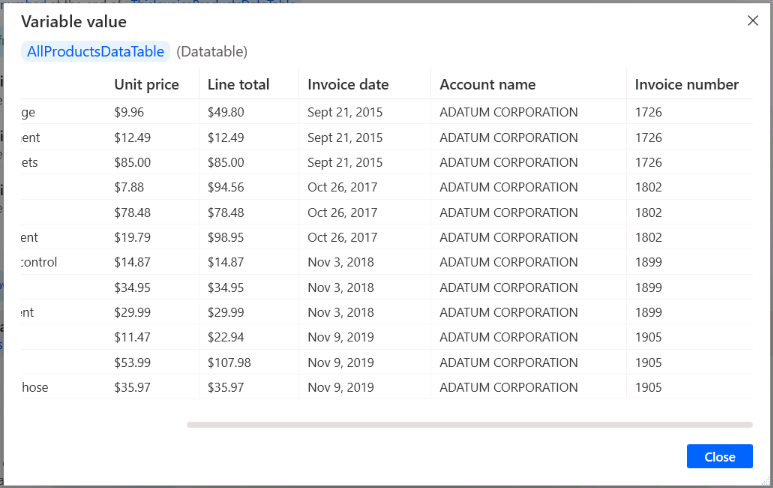


1. Add the action **Insert row into data table** within the **For each** cycle. Add each product row into the data table for all products by configuring the action as shown:

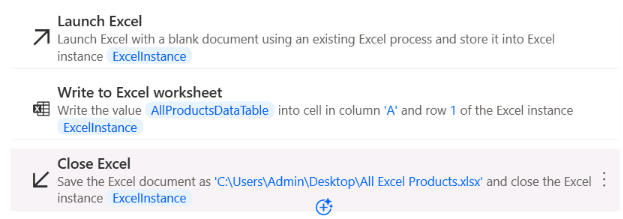
* **Data table:** %AllProductsDataTable%
* **Into location:** End of data table
* **New value(s):** %ThisProductRow%



1. **Run** your flow and inspect the results on the variable **AllProductsDataTable**, which should contain in total 12 rows, one for each of the invoiced products across all analyzed documents.



1. You can store the results of your flow into an Excel spreadsheet by following similar steps to the ones we saw in [Lab 02: Automate web page extraction](https://github.com/mcoloradodevs/PowerAutomateDesktopTraining/blob/main/Lab%2002%20-%20Automate%20web%20page%20extraction.docx). Here is a sample subflow of the actions you should consider adding to your automation:



## Terms of Use

By using this document, in whole or in part, you agree to the following terms:

### **Notice**

Information and views expressed in this document, including (without limitation) URL and other Internet Web site references, may change without notice. Examples depicted herein, if any, are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

### **Use Limitations**

Copying or reproduction, in whole or in part, of this document to any other server or location for further reproduction or redistribution is expressly prohibited. Microsoft provides you with this document for purposes of obtaining your suggestions, comments, input, ideas, or know-how, in any form, ("Feedback") and to provide you with a learning experience. You may use this document only to evaluate its content and provide feedback to Microsoft. You may not use this document for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this document or any portion thereof. You may copy and use this document for your internal, reference purposes only.

### **Feedback**

If you give Microsoft any Feedback about this document or the subject matter herein (including, without limitation, any technology, features, functionality, and/or concepts), you give to Microsoft, without charge, the right to use, share, and freely commercialize Feedback in any way and for any purpose. You also give third parties, without charge, the right to use, or interface with, any Microsoft products or services that include the Feedback. You represent and warrant that you own or otherwise control all rights to such Feedback and that no such Feedback is subject to any third-party rights.

### **DISCLAIMERS**

CERTAIN SOFTWARE, TECHNOLOGY, PRODUCTS, FEATURES, AND FUNCTIONALITY (COLLECTIVELY "CONCEPTS"),

INCLUDING POTENTIAL NEW CONCEPTS, REFERENCED IN THIS DOCUMENT ARE IN A SIMULATED ENVIRONMENT

WITHOUT COMPLEX SET-UP OR INSTALLATION AND ARE INTENDED FOR FEEDBACK AND TRAINING PURPOSES

ONLY. THE CONCEPTS REPRESENTED IN THIS DOCUMENT MAY NOT REPRESENT FULL FEATURE CONCEPTS AND MAY

NOT WORK THE WAY A FINAL VERSION MAY WORK. MICROSOFT ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH

CONCEPTS. YOUR EXPERIENCE WITH USING SUCH CONCEPTS IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

THIS DOCUMENT, AND THE CONCEPTS AND TRAINING PROVIDED HEREIN, IS PROVIDED “AS IS”, WITHOUT WARRANTY

OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING (WITHOUT LIMITATION) THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. MICROSOFT DOES NOT

MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, THE OUTPUT THAT DERIVES FROM USE OF THIS DOCUMENT OR THE CONCEPTS, OR THE SUITABILITY OF THE CONCEPTS OR INFORMATION CONTAINED IN THIS DOCUMENT FOR ANY PURPOSE.