5.2 - Use prebuilt Document Intelligence models

Overview

Learn what data you can analyze by choosing prebuilt Forms Analyzer models and how to deploy these models in a Document intelligence solution.

In this module, you'll learn to:

- Identify business problems that you can solve by using prebuilt models in Forms Analyzer.
- Analyze forms by using the General Document, Read, and Layout models.
- Analyze forms by using financial, ID, and tax prebuilt models.

Introduction

Many forms and documents that your business handles are common across disparate companies in different sectors. For example, most companies use **invoices and receipts.** Microsoft Azure AI Document Intelligence includes prebuilt models so you can handle common document types easily.

You work for a company that conducts polls for private companies and political parties. Participants submit their responses as paper forms or as online PDFs. You've decided to deploy Azure AI Document Intelligence to streamline data entry and you need to know if you can use the prebuilt models to generate meaningful data from your forms.

In this module, you'll learn about the capabilities of the prebuilt models in Azure AI Document Intelligence and how to use them.

At the end of this module, you'll be able to:

- Identify business problems that you can solve by using prebuilt models in Azure AI Document Intelligence.
- Analyze forms by using the General Document, Read, and Layout models.
- Analyze forms by using financial, ID, and tax prebuilt models.

Understand prebuilt models

Prebuilt models in Azure AI Document Intelligence enable you to extract data from common forms and documents without training your own models.

In your polling company, polling forms are unique to each survey project, but you also use invoices and receipts to record financial transactions and you have many unstructured documents. You want to know how much work is required to extract names, addresses, amounts, and other information from these documents.

Here, you learn how prebuilt models can help you analyze common document types.

What are prebuilt models?

The general approach used in AI solutions is to provide a large quantity of sample data and then train an optimized model by trying different data features, parameters, and statistical treatments. The combination that best predicts the values that interest you constitute the trained model, and you can use this model to predict values from new data.

Many of the forms that businesses use from day to day are of a few common types. For example, most businesses issue or receive invoices and receipts. Any business that has employees in the United States must use the W-2 tax declaration form.

Also you often have more general documents that you might want to extract data from. For these cases, Microsoft helps you by providing prebuilt models. **Prebuilt models are already trained on large numbers of their target form type.**

If you want to use Document Intelligence to extract data from one of these common forms or documents, you can choose to use a prebuilt model and you don't have to train your own. Because Microsoft trains these models on a large corpus of examples, you can expect them to provide accurate and reliable results when dealing with their intended forms.

Several of the prebuilt models are trained on specific form types:

- **Invoice model.** Extracts common fields and their values from invoices.
- **Receipt model.** Extracts common fields and their values from receipts.
- **US Tax model.** Unified US tax model that can extract from forms such as W-2, 1098, 1099, and 1040.
- **ID document model.** Extracts common fields and their values from US drivers' licenses, European Union IDs and drivers license, and international passports.
- **Business card model.** Extracts common fields and their values from business cards.
- **Health insurance card model.** Extracts common fields and their values from health insurance cards.
- **Marriage certificate.** Extracts information from marriage certificates.
- **Credit/Debit card model.** Extracts common information from bank cards.
- **Mortgage documents.** Extracts information from mortgage closing disclosure, Uniform Residential Loan Application (Form 1003), Appraisal (Form 1004), Validation of Employment (Form 1005), and Uniform Underwriting and Transmittal Summary (Form 1008).
- **Bank statement model.** Extracts account information including beginning and ending balances, transaction details from bank statements.
- **Pay Stub model.** Extracts wages, hours, deductions, net pay, and other common pay stub fields.
- **Check model.** Extracts payee, amount, date, and other relevant information from checks.

The other models are designed to extract values from documents with less specific structures. They are called the **document analysis models**:

- **Read model.** Extracts text and languages from documents.
- **General document model.** Extract text, keys, values, entities, and selection marks from documents.
- **Layout model.** Extracts text and structure information from documents.

Features of prebuilt models

The prebuilt models are designed to extract different types of data from the documents and forms users submit. To select the right model for your requirements, you must understand these features:

- **Text extraction.** All the prebuilt models extract lines of text and words from hand-written and printed text.
- **Key-value pairs.** Many models extract spans of text within a document that identify a label or key and its response or value as key-values pairs. For example, a typical key might be **Weight** and its value might be **31 kg**.
- **Entities.** Text that includes common, more complex data structures can be extracted as entities. Entity types include people, locations, and dates.
- **Selection marks.** Some models extract **spans of text that indicate a choice as selection marks.** These marks include radio buttons and check boxes.
- **Tables.** Many models can extract tables in scanned forms included the data contained in cells, the numbers of columns and rows, and column and row headings. **Tables with merged cells are supported.**
- **Fields.** Models trained for a specific form type identify the values of a fixed set of fields. For example, the Invoice model includes **CustomerName** and **InvoiceTotal** fields.

Also consider that prebuilt models are designed for and trained on generic document and form types. If you have an industry-specific or unique form type that you use often, you might be able to obtain more reliable and predictable results by using a custom model.

However, custom models take time to develop because you must invest the time and resources to train them on example forms before you can use it. The larger the number of example forms you provide for training, the better the model is at predicting form content accurately.

Input requirements

The prebuilt models are flexible but you can help them to return accurate and helpful results by **submitting one clear photo or high-quality scan for each document**.

You must also comply with these requirements when you submit a form for analysis:

- The file must be in JPEG, PNG, BMP, TIFF, or PDF format. Additionally, the **Read model can accept Microsoft Office files.**
- The file must be smaller than 500 MB for the standard tier, and 4 MB for the free tier.
- Images must have dimensions between 50 x 50 pixels and 10,000 x 10,000 pixels.
- PDF documents must have dimensions less than 17 x 17 inches or A3 paper size.
- PDF documents must not be protected with a password.

> Note: If you can, **submit text-embedded PDF files because they eliminate errors in character recognition.**

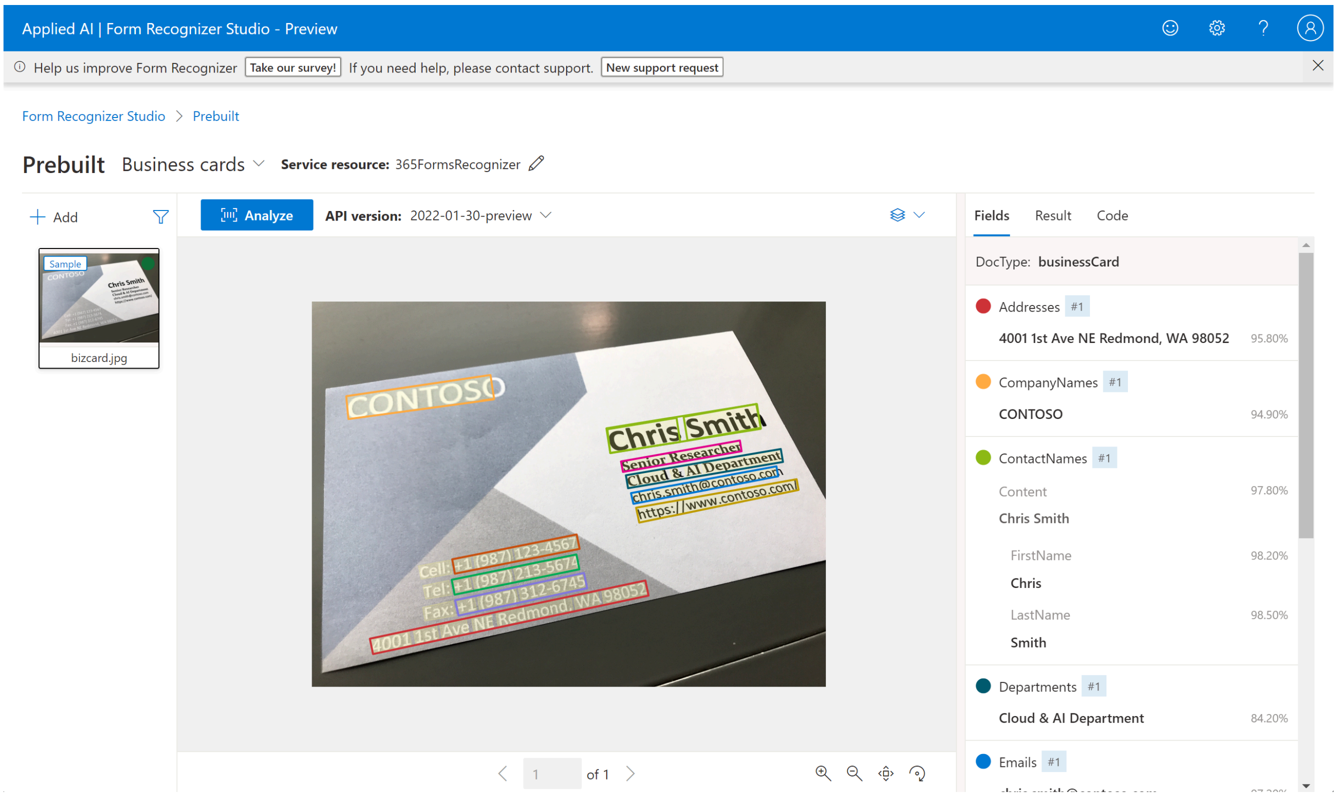
PDF and TIFF files can have any number of pages but, in the standard tier, only the first 2,000 pages are analyzed. In the free tier, only the first two pages are analyzed.

Try out prebuilt models with Azure AI Document Intelligence Studio

Azure AI Document Intelligence is designed as a web service you can call using code in your custom applications. However, it's often helpful to explore the models and how they behave with your forms visually.

You can perform such experiments by using Azure AI Document Intelligence Studio and use the experience to help design and write your code.

You can choose any of the prebuilt models in Azure AI Document Intelligence Studio. Microsoft provides some sample documents for use with each model or you can add your own documents and analyze them.



DocType: businessCard

Addresses #1
4001 1st Ave NE Redmond, WA 98052 95.80%

CompanyNames #1
CONTOSO 94.90%

ContactNames #1
Content 97.80%
Chris Smith
FirstName 98.20%
Chris
LastName 98.50%
Smith

Departments #1
Cloud & AI Department 84.20%

Emails #1

Calling prebuilt models by using APIs

Because Azure AI Document Intelligence implements RESTful web services, **you can use web service calls** from any language that supports them. However, when you use Microsoft's Azure AI Document Intelligence APIs, security and session management is simplified and you have to write less code.

APIs are available for:

- C# and other .NET languages.
- Java.
- Python.
- JavaScript.

Whenever you want to call Azure AI Document Intelligence, you must start by connecting and authenticating with the service in your Azure subscription. To make that connection, you need:

- **The service endpoint.** This value is the URL where the service is published.
- **The API key.** This value is a unique key that grants access.

You obtain both of these values from the Azure portal.

Because the service can take a few seconds to respond, it's best to use asynchronous calls to submit a form and then obtain results from the analysis:

```
poller = document_analysis_client.begin_analyze_document(
        "prebuilt-layout",
        AnalyzeDocumentRequest(url_source=docUrl)
        )
result: AnalyzeResult = poller.result()
```

The details you can extract from these results depend on the model you used.

Use the General Document, Read, and Layout models

If you want to extract text, languages, and other information from documents with *unpredictable structures*, you can use the read, general document, or layout models.

In your polling company, customers and partners often send specifications, tenders, statements of work, and other documents with unpredictable structures. You want to know if Azure AI Document Intelligence can analyze and extract values from these documents.

Here, you'll learn about the prebuilt models that Microsoft provides for general documents.

Using the read model

The Azure AI Document Intelligence read model extracts printed and handwritten text from documents and images. It's used to provide text extraction in all the other prebuilt models.

The read model can also detect the language that a line of text is written in and classify whether it's handwritten or printed text.

> Note: The **read model supports more languages for printed text than handwritten text.** Check the documentation to see the current list of supported languages.

For multi-page PDF or TIFF files, you can use the `pages` parameter in your request to fix a page range for the analysis.

The read model is ideal if you want to extract words and lines from documents with no fixed or predictable structure.

Using the general document model

The general document model **extends the functionality of the read model by adding the detection of key-value pairs, entities, selection marks, and tables.**

The model can extract these values from **structured, semi-structured, and unstructured documents.**

The general document model is the only prebuilt model to support entity extraction.

It can recognize entities such as people, organizations, and dates and it runs against the whole document, not just key-value pairs. This approach ensures that, when structural complexity has prevented the model extracting a key-value pair, an entity can be extracted instead.

Remember, however, that sometimes a single piece of text might return both a key-value pair and an entity.

The types of entities you can detect include:

- `Person` . The name of a person.
- `PersonType` . A job title or role.
- `Location` . Buildings, geographical features, geopolitical entities.
- `Organization` . Companies, government bodies, sports clubs, musical bands, and other groups.
- `Event` . Social gatherings, historical events, anniversaries.
- `Product` . Objects bought and sold.

- `Skill`. A capability belonging to a person.
- `Address`. Mailing address for a physical location.
- `Phone number`. Dialing codes and numbers for mobile phones and landlines.
- `Email`. Email addresses.
- `URL`. Webpage addresses.
- `IP Address`. Network addresses for computer hardware.
- `DateTime`. Calendar dates and times of day.
- `Quantity`. Numerical measurements with their units.

Using the layout model

As well as extracting text, the **layout model returns selection marks and tables from the input image or PDF file. It's a good model to use when you need rich information about the structure of a document.**

When you digitize a document, it can be at an odd angle. Tables can have complicated structures with or without headers, cells that span columns or rows, and incomplete columns or rows. **The layout model can handle all of these difficulties to extract the complete document structure.**

For example, each table cell is extracted with:

- Its content text.
- The size and position of its bounding box.
- If it's part of a header column.
- Indexes to indicate its row and column position in the table.

Selection marks are extracted with their bounding box, a confidence indicator, and whether they're selected or not.

Use financial, ID, and tax models

Azure AI Document Intelligence includes some prebuilt models that are trained on common form types. You can use these models to obtain the values of common fields from invoices, receipts, business cards, and more.

In your polling company, invoices and receipts are often submitted as photos or scans of the paper documents. Sometimes the scan is poor and the paper is creased or damaged. You want to know if Azure AI Document Intelligence can get this information into your databases more efficiently than manual data entry.

Here, you'll learn about the prebuilt models that handle financial, identity, and tax documents.

Using the invoice model

Your business both issues invoices and receives them from partner organization. There might be **many different formats on paper or in digitized forms and some will have been scanned poorly at odd angles or from creased paper.**

The invoice model in Azure AI Document Intelligence can handle these challenges and **uses the features of the read model** to extract text from invoice scans.

In addition, it extracts specific fields that are commonly used on invoices including:

- Customer name and reference ID
- Purchase order number
- Invoice and due dates
- Details about the vendor, such as name, tax ID, physical address.
- Similar details about the customer.
- Billing and shipping addresses.
- Amounts such as total tax, invoice total, and amount due.

Invoices also feature lines, usually in a table, each of which is one purchased item. For each line, the invoice model identifies details including:

- The description and product code of the product or service invoiced.
- Amounts such as the unit price, the quantity of items, the tax incurred, and the line total.

Using the receipt model

Receipts have similar fields and structures to invoices, but they record amounts paid instead of amounts charged. Azure AI Document Intelligence faces the same challenges of poor scanning or digitization but can reliably identify fields including:

- Merchant details such a name, phone number, and address.
- Amounts such as receipt total, tax, and tip.
- The date and time of the transaction.

As for invoices, receipts often include a table of items, each of which is a product or service purchased. For each of these lines, the model recognizes:

- The name of the item.
- The quantity of the item purchased.
- The unit price of the item.
- The total price for that quantity.

> Note: In Azure AI Document Intelligence v3.0 and later, the receipt model supports single-page hotel receipt processing. If a receipt is classified as a hotel receipt, the model extracts extra relevant fields such as arrival and departure dates.

Using the ID document model

The ID document model is trained to analyze two types of identity document:

- United States drivers licenses.
- International passports.

> Note: Only the **biographical pages of passports** can be analyzed. Visas and other travel documents are not supported.

The ID document model can extract fields including:

- First and last names.
- Personal information such as sex, date of birth, and nationality.

- The country and region where the document was issued.
- Unique numbers such as the document number and machine readable zone.
- Endorsements, restrictions, and vehicle classifications.

> Important: Since much of the data extracted by the ID document model is personal, it is of a **sensitive nature and covered by data protection laws in most jurisdictions.** Be sure that you have the permission of the individual to store their data and comply with all legal requirements in the way you handle this information.

Using the business card model

Business cards are a popular way to exchange contact information quickly and often include branding, unusual fonts, and graphic design elements. Fields that the business card model can extract include:

- First and last names.
- Postal addresses.
- Email and website addresses.
- Various telephone numbers.

Using other prebuilt models

Azure AI Document Intelligence offers several prebuilt models, with new models being released regularly. Before training a custom model, it's worth verifying if your use case can be analyzed accurately with one of these prebuilt models.

Using a prebuilt model will benefit from rigorous testing, updated model versions, and reduced cost compared to a custom model.

Exercise - Analyze a document using Azure AI Document Intelligence

In this exercise, you'll set up an Azure AI Document Intelligence resource in your Azure subscription. You'll use both the Azure AI Document Intelligence Studio and C# or Python to submit forms to that resource for analysis.

Create an Azure AI Document Intelligence resource

Before you can call the Azure AI Document Intelligence service, you must create a resource to host that service in Azure:

1. In a browser tab, open the Azure portal at https://portal.azure.com, signing in with the Microsoft account associated with your Azure subscription.
2. On the Azure portal home page, navigate to the top search box and type **Document Intelligence** and then press **Enter**.
3. On the **Document Intelligence** page, select **Create Document Intelligence**.
4. On the **Create Document Intelligence** page, use the following to configure your resource:
   - **Subscription**: Your Azure subscription.
   - **Resource group**: Select or create a resource group with a unique name such as *DocIntelligenceResources*.

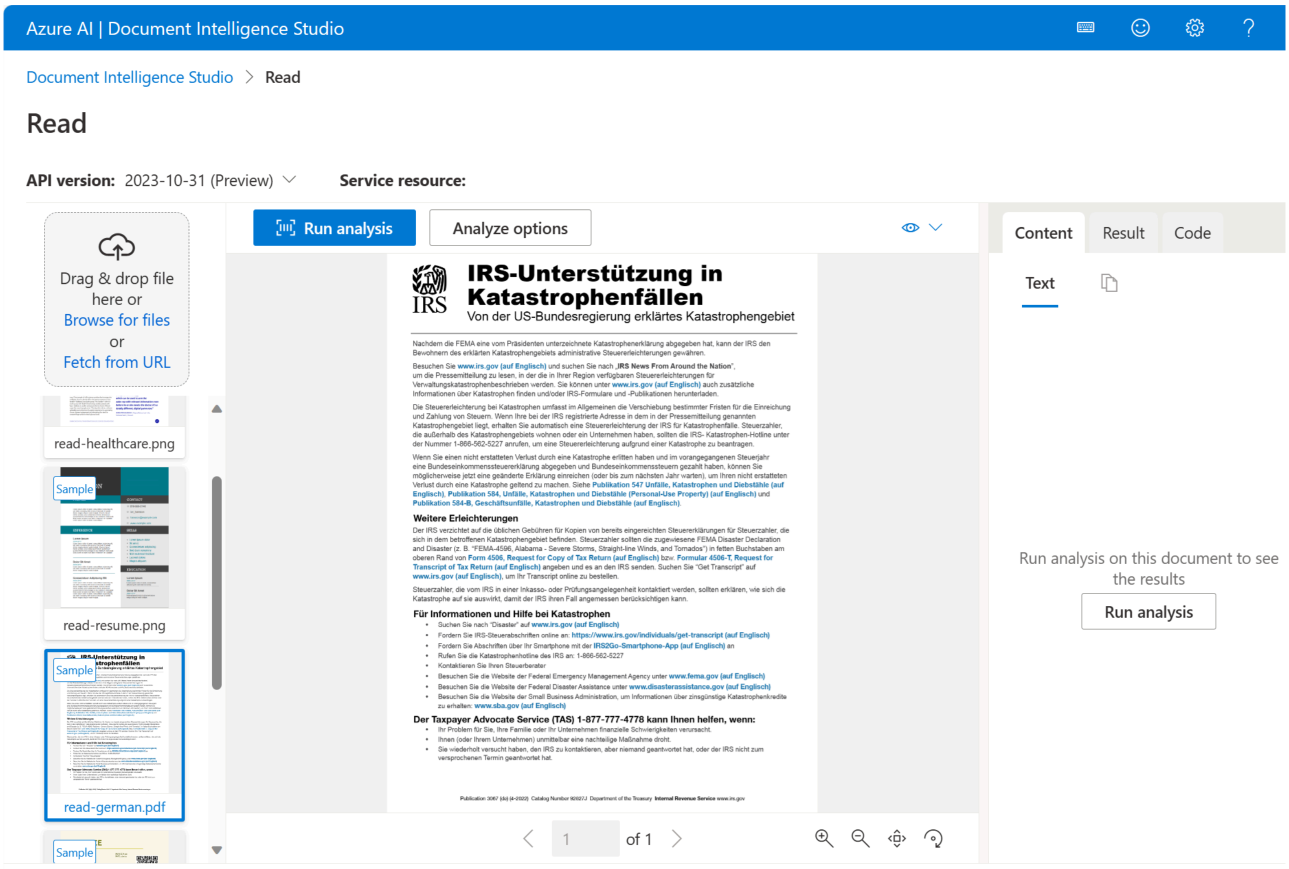
- **Region**: select a region near you.
- **Name**: Enter a globally unique name.
- **Pricing tier**: select **Free F0** (if you don't have a Free tier available, select **Standard S0**).

5. Then select **Review + create**, and **Create**. Wait while Azure creates the Azure AI Document Intelligence resource.
6. When the deployment is complete, select **Go to resource**. Keep this page open for the rest of this exercise.

Use the Read model

Let's start by using the **Azure AI Document Intelligence Studio** and the Read model to analyze a document with multiple languages. You'll connect Azure AI Document Intelligence Studio to the resource you just created to perform the analysis:

1. Open a new browser tab and go to the **Azure AI Document Intelligence Studio** at https://documentintelligence.ai.azure.com/studio.
2. Under **Document Analysis**, select the **Read** tile.
3. If you are asked to sign into your account, use your Azure credentials.
4. If you are asked which Azure AI Document Intelligence resource to use, select the subscription and resource name you used when you created the Azure AI Document Intelligence resource.
5. In the list of documents on the left, select **read-german.pdf**.



IRS-Unterstützung in Katastrophenfällen
Von der US-Bundesregierung erklärtes Katastrophengebiet

Nachdem die FEMA eine vom Präsidenten unterzeichnete Katastrophenerklärung abgegeben hat, kann der IRS den Bewohnern des erklärten Katastrophengebiets administrative Steuererleichterungen gewähren.

Besuchen Sie www.irs.gov (auf Englisch) und suchen Sie nach „**IRS News From Around the Nation**", um die Pressemitteilung zu lesen, in der die in Ihrer Region verfügbaren Steuererleichterungen für Verwaltungskatastrophenbeschrieben werden. Sie können unter www.irs.gov (auf Englisch) auch zusätzliche Informationen über Katastrophen finden und/oder IRS-Formulare und -Publikationen herunterladen.

Die Steuererleichterung bei Katastrophen umfasst im Allgemeinen die Verschiebung bestimmter Fristen für die Einreichung und Zahlung von Steuern. Wenn Ihre bei der IRS registrierte Adresse in dem in der Pressemitteilung genannten Katastrophengebiet liegt, erhalten Sie automatisch eine Steuererleichterung der IRS für Katastrophenfälle. Steuerzahler, die außerhalb des Katastrophengebiets wohnen oder ein Unternehmen haben, sollten die IRS- Katastrophen-Hotline unter der Nummer 1-866-562-5227 anrufen, um eine Steuererleichterung aufgrund einer Katastrophe zu beantragen.

Wenn Sie einen nicht erstatteten Verlust durch eine Katastrophe erlitten haben und im vorangegangenen Steuerjahr eine Bundeseinkommenssteuererklärung abgegeben und Bundeseinkommenssteuern gezahlt haben, können Sie möglicherweise jetzt eine geänderte Erklärung einreichen (oder bis zum nächsten Jahr warten), um Ihren nicht erstatteten Verlust durch eine Katastrophe geltend zu machen. Siehe **Publikation 547 Unfälle, Katastrophen und Diebstähle (auf Englisch)**, **Publikation 584, Unfälle, Katastrophen und Diebstähle (Personal-Use Property) (auf Englisch)** und **Publikation 584-B, Geschäftsunfälle, Katastrophen und Diebstähle (auf Englisch)**.

Weitere Erleichterungen

Der IRS verzichtet auf die üblichen Gebühren für Kopien von bereits eingereichten Steuererklärungen für Steuerzahler, die sich in dem betroffenen Katastrophengebiet befinden. Steuerzahler sollten die zugewiesene FEMA Disaster Declaration and Disaster (z. B. "FEMA-4596, Alabama - Severe Storms, Straight-line Winds, and Tornados") in fetten Buchstaben am oberen Rand von **Form 4506, Request for Copy of Tax Return (auf Englisch)** bzw. **Formular 4506-T, Request for Transcript of Tax Return (auf Englisch)** angeben und es an den IRS senden. Suchen Sie "Get Transcript" auf **www.irs.gov (auf Englisch)**, um Ihr Transcript online zu bestellen.

Steuerzahler, die vom IRS in einer Inkasso- oder Prüfungsangelegenheit kontaktiert werden, sollten erklären, wie sich die Katastrophe auf sie auswirkt, damit der IRS ihren Fall angemessen berücksichtigen kann.

Für Informationen und Hilfe bei Katastrophen

- Suchen Sie nach "Disaster" auf **www.irs.gov (auf Englisch)**
- Fordern Sie IRS-Steuerabschriften online an: **https://www.irs.gov/individuals/get-transcript (auf Englisch)**
- Fordern Sie Abschriften über Ihr Smartphone mit der **IRS2Go-Smartphone-App (auf Englisch)** an
- Rufen Sie die Katastrophenhotline des IRS an: 1-866-562-5227
- Kontaktieren Sie Ihren Steuerberater
- Besuchen Sie die Website der Federal Emergency Management Agency unter **www.fema.gov (auf Englisch)**
- Besuchen Sie die Website der Federal Disaster Assistance unter **www.disasterassistance.gov (auf Englisch)**
- Besuchen Sie die Website der Small Business Administration, um Informationen über zinsgünstige Katastrophenkredite zu erhalten: **www.sba.gov (auf Englisch)**

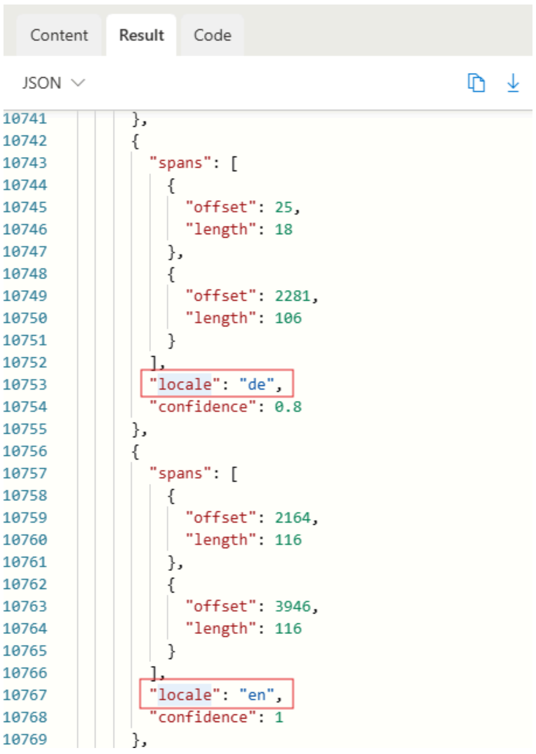
Der Taxpayer Advocate Service (TAS) 1-877-777-4778 kann Ihnen helfen, wenn:

- Ihr Problem für Sie, Ihre Familie oder Ihr Unternehmen finanzielle Schwierigkeiten verursacht.
- Ihnen (oder Ihrem Unternehmen) unmittelbar eine nachteilige Maßnahme droht.
- Sie wiederholt versucht haben, den IRS zu kontaktieren, aber niemand geantwortet hat, oder der IRS nicht zum versprochenen Termin geantwortet hat.

Publication 3067 (de) (4-2022) Catalog Number 92827J Department of the Treasury **Internal Revenue Service** www.irs.gov

6. At the top-left, select **Analyze options**, then enable the **Language** check-box (under **Optional detection**) in the **Analyze options** pane and click on **Save**.
7. At the top-left, select **Run Analysis**.
8. When the analysis is complete, the text extracted from the image is shown on the right in the **Content** tab. Review this text and compare it to the text in the original image for accuracy.
9. Select the **Result** tab. This tab displays the extracted JSON code.

10. Scroll to the bottom of the JSON code in the **Result** tab. Notice that the read model has detected the language of each span indicated by `locale`. Most spans are in German (language code `de`) but you can find other language codes in the spans (e.g. English - language code `en` - in one of the last span).



```
      },
      {
        "spans": [
          {
            "offset": 25,
            "length": 18
          },
          {
            "offset": 2281,
            "length": 106
          }
        ],
        "locale": "de",
        "confidence": 0.8
      },
      {
        "spans": [
          {
            "offset": 2164,
            "length": 116
          },
          {
            "offset": 3946,
            "length": 116
          }
        ],
        "locale": "en",
        "confidence": 1
      },
```

Prepare to develop an app in Visual Studio Code

Now let's explore the app that uses the Azure Document Intelligence service SDK. You'll develop your app using Visual Studio Code. The code files for your app have been provided in a GitHub repo.

> **Tip**: If you have already cloned the **mslearn-ai-document-intelligence** repo, open it in Visual Studio code. Otherwise, follow these steps to clone it to your development environment.

1. Start Visual Studio Code.
2. Open the palette (SHIFT+CTRL+P) and run a **Git: Clone** command to clone the `https://github.com/MicrosoftLearning/mslearn-ai-document-intelligence` repository to a local folder (it doesn't matter which folder).
3. When the repository has been cloned, open the folder in Visual Studio Code.

   > **Note**: If Visual Studio Code shows you a pop-up message to prompt you to trust the code you are opening, click on **Yes, I trust the authors** option in the pop-up.

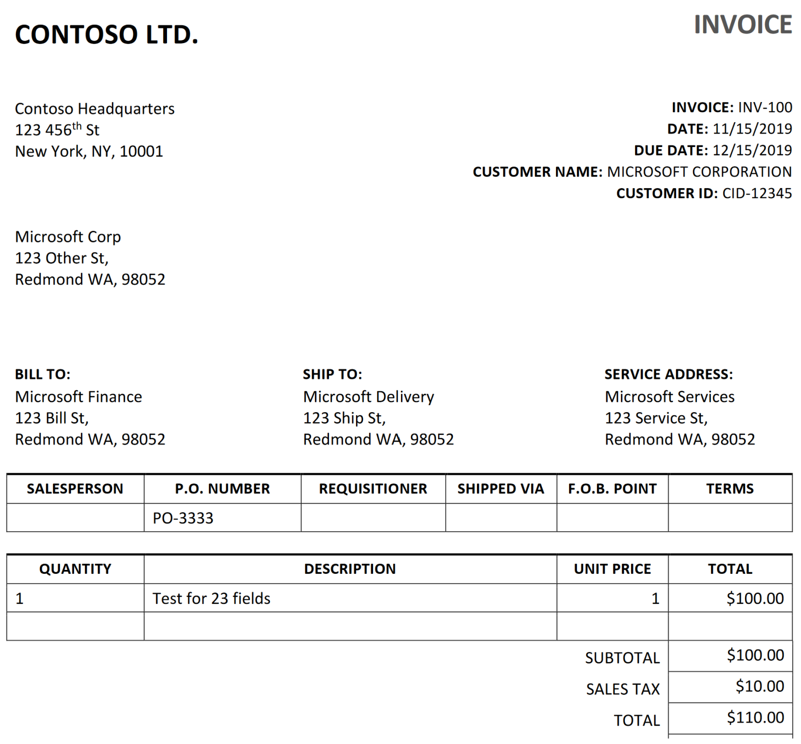
4. Wait while additional files are installed to support the C# code projects in the repo.

   > **Note**: If you are prompted to add required assets to build and debug, select **Not Now**. If you are prompted with the Message *Detected an Azure Function Project in folder*, you can safely close that message.

Configure your application

Applications for both C# and Python have been provided, as well as a sample pdf file you'll use to test Document Intelligence. Both apps feature the same functionality. First, you'll complete some key parts of the application to enable using your Azure Document Intelligence resource.

1. Examine the following invoice and note some of its fields and values. This is the invoice that your code will analyze.



CONTOSO LTD.

INVOICE

Contoso Headquarters
123 456th St
New York, NY, 10001

INVOICE: INV-100
DATE: 11/15/2019
DUE DATE: 12/15/2019
CUSTOMER NAME: MICROSOFT CORPORATION
CUSTOMER ID: CID-12345

Microsoft Corp
123 Other St,
Redmond WA, 98052

BILL TO:
Microsoft Finance
123 Bill St,
Redmond WA, 98052

SHIP TO:
Microsoft Delivery
123 Ship St,
Redmond WA, 98052

SERVICE ADDRESS:
Microsoft Services
123 Service St,
Redmond WA, 98052

SALESPERSON	P.O. NUMBER	REQUISITIONER	SHIPPED VIA	F.O.B. POINT	TERMS
	PO-3333				

QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL
1	Test for 23 fields	1	$100.00
		SUBTOTAL	$100.00
		SALES TAX	$10.00
		TOTAL	$110.00

2. In Visual Studio Code, in the **Explorer** pane, browse to the **Labfiles/01-prebuild-models** folder and expand the **CSharp** or **Python** folder depending on your language preference. Each folder contains the language-specific files for an app into which you're you're going to integrate Azure Document Intelligence functionality.
3. Right-click the **CSharp** or **Python** folder containing your code files and select **Open an integrated terminal**. Then install the **Azure Form Recognizer (the previous name for Document Intelligence)** SDK package by running the appropriate command for your language preference:

   C#:
   ```
   dotnet add package Azure.AI.FormRecognizer --version 4.1.0
   ```

   Python:
   ```
   pip install azure-ai-formrecognizer==3.3.3
   ```

Add code to use the Azure Document Intelligence service

Now you're ready to use the SDK to evaluate the pdf file.

1. Switch to the browser tab that displays the Azure AI Document Intelligence overview in the Azure portal. On the left pane, under *Resource Management*, select **Keys and Endpoint**. To the right of the **Endpoint** value, click the **Copy to clipboard** button.
2. In the **Explorer** pane, in the **CSharp** or **Python** folder, open the code file for your preferred language, and replace `<Endpoint URL>` with the string you just copied:

   C#: ***Program.cs***
   ```
   string endpoint = "<Endpoint URL>";
   ```

   Python: document-analysis.py
   ```
   endpoint = "<Endpoint URL>"
   ```

3. Switch to the browser tab that displays the Azure AI Document Intelligence **Keys and Endpoint** in the Azure portal. To the right of the **KEY 1** value, click the *Copy to clipboard** button.
4. In the code file in Visual Studio Code, locate this line and replace `<API Key>` with the string you just copied:

   C#
   `string apiKey = "<API Key>";`

   Python
   `key = "<API Key>"`
5. Locate the comment `Create the client`. Following that, on new lines, enter the following code:

   C#
   `var cred = new AzureKeyCredential(apiKey); var client = new DocumentAnalysisClient(new Uri(endpoint), cred);`

   Python

   ```
   document_analysis_client = DocumentAnalysisClient(endpoint=endpoint,
                                  credential=AzureKeyCredential(key)
                              )
   ```

6. Locate the comment `Analyze the invoice`. Following that, on new lines, enter the following code:

   C#
   `AnalyzeDocumentOperation operation = await client.AnalyzeDocumentFromUriAsync(WaitUntil.Completed, "prebuilt-invoice", fileUri);`

   Python

   ```
   poller = document_analysis_client.begin_analyze_document_from_url(
                                      fileModelId,
                                      fileUri,
                                      locale=fileLocale)
   ```

7. Locate the comment `Display invoice information to the user`. Following that, on news lines, enter the following code:

   C#
   `AnalyzeResult result = operation.Value; foreach (AnalyzedDocument invoice in result.Documents) { if (invoice.Fields.TryGetValue("VendorName", out DocumentField? vendorNameField)) { if (vendorNameField.FieldType == DocumentFieldType.String) { string vendorName = vendorNameField.Value.AsString(); Console.WriteLine($"Vendor Name: '{vendorName}', with confidence {vendorNameField.Confidence}."); } }`

   Python

   ```
   receipts = poller.result()

   for idx, receipt in enumerate(receipts.documents):
       vendor_name = receipt.fields.get("VendorName")
       if vendor_name:
   ```

```
        print(f"\nVendor Name: {vendor_name.value}, with confidence
                {vendor_name.confidence}.")
```

> You've added code to display the vendor name. The starter project also includes code to display the *customer name* and *invoice total*.

8. Save the changes to the code file.
9. In the interactive terminal pane, ensure the folder context is the folder for your preferred language. Then enter the following command to run the application.
10. ***For C# only***, to build your project, enter this command:

    C#:
    `dotnet build`
11. To run your code, enter this command:

    C#:
    `dotnet run`

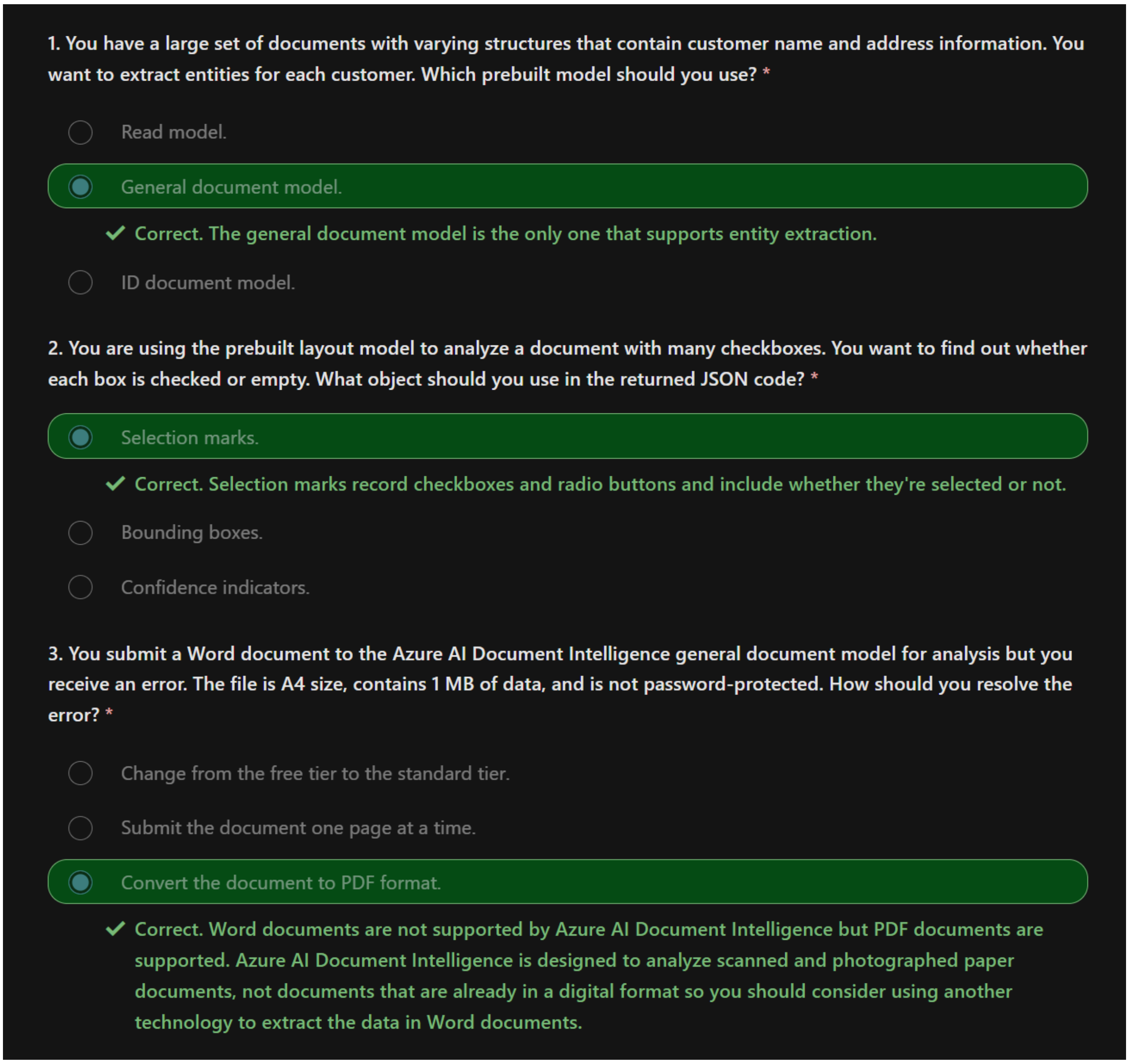
    Python:
    `python document-analysis.py`

The program displays the vendor name, customer name, and invoice total with confidence levels. Compare the values it reports with the sample invoice you opened at the start of this section.

Clean up

If you're done with your Azure resource, remember to delete the resource in the Azure portal to avoid further charges.

Knowledge Check



1. You have a large set of documents with varying structures that contain customer name and address information. You want to extract entities for each customer. Which prebuilt model should you use? *

- ○ Read model.
- [✓ General document model.]

  ✓ Correct. The general document model is the only one that supports entity extraction.
- ○ ID document model.

2. You are using the prebuilt layout model to analyze a document with many checkboxes. You want to find out whether each box is checked or empty. What object should you use in the returned JSON code? *

- [✓ Selection marks.]

  ✓ Correct. Selection marks record checkboxes and radio buttons and include whether they're selected or not.
- ○ Bounding boxes.
- ○ Confidence indicators.

3. You submit a Word document to the Azure AI Document Intelligence general document model for analysis but you receive an error. The file is A4 size, contains 1 MB of data, and is not password-protected. How should you resolve the error? *

- ○ Change from the free tier to the standard tier.
- ○ Submit the document one page at a time.
- [✓ Convert the document to PDF format.]

  ✓ Correct. Word documents are not supported by Azure AI Document Intelligence but PDF documents are supported. Azure AI Document Intelligence is designed to analyze scanned and photographed paper documents, not documents that are already in a digital format so you should consider using another technology to extract the data in Word documents.

Summary

There are many document types that are common to most business and Azure AI Document Intelligence includes prebuilt models to handle them.

If you have a collection of such forms that you want to analyze, you can extract data by using these prebuilt models and you don't have to train your own models. You can get up and running very quickly by submitting photos and scans to the most appropriate prebuilt model.

Now that you've completed this module, you can:

- Identify business problems that you can solve by using prebuilt models in Azure AI Document Intelligence.
- Analyze forms by using the General Document, Read, and Layout models.
- Analyze forms by using financial, ID, and tax prebuilt models.
- [What is Azure AI Document Intelligence?]
- [Azure AI Document Intelligence models]

- Language support for Azure AI Document Intelligence
- Azure AI Document Intelligence read model
- Azure AI Document Intelligence general document model
- Azure AI Document Intelligence layout model
- Azure AI Document Intelligence invoice model
- Azure AI Document Intelligence receipt model
- Azure AI Document Intelligence ID document model
- Azure AI Document Intelligence business card model
- Azure AI Document Intelligence W-2 model

✍️ Compiled by Kenneth Leung (2025)