**Plan an Azure AI Document Intelligence solution**

**Introduction**

Azure AI Document Intelligence uses Azure AI Services to analyze the content of scanned forms and convert them into data. It can recognize text values in both common forms and forms that are unique to your business.

In this module, we consider this scenario: 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 currently spend a lot of time and money entering these responses into databases. You want to assess Azure AI Document Intelligence to find out if you can use it to streamline this process.

In this module, you'll learn about the architecture of Azure AI Document Intelligence and how to design a solution that can ingest data from forms into a data store.

**Understand AI Document Intelligence**

Azure AI Document Intelligence is easy to use but, to create a reliable solution, you must understand its objects such as models, APIs, and tools.

In your polling company, you're assessing Azure AI Document Intelligence to see if it can streamline your data entry workflow. You need to understand what data Azure AI Document Intelligence can obtain from the different polling forms you use and how your development team will build a AI data entry system and integrate it with your mobile and desktop apps and databases.

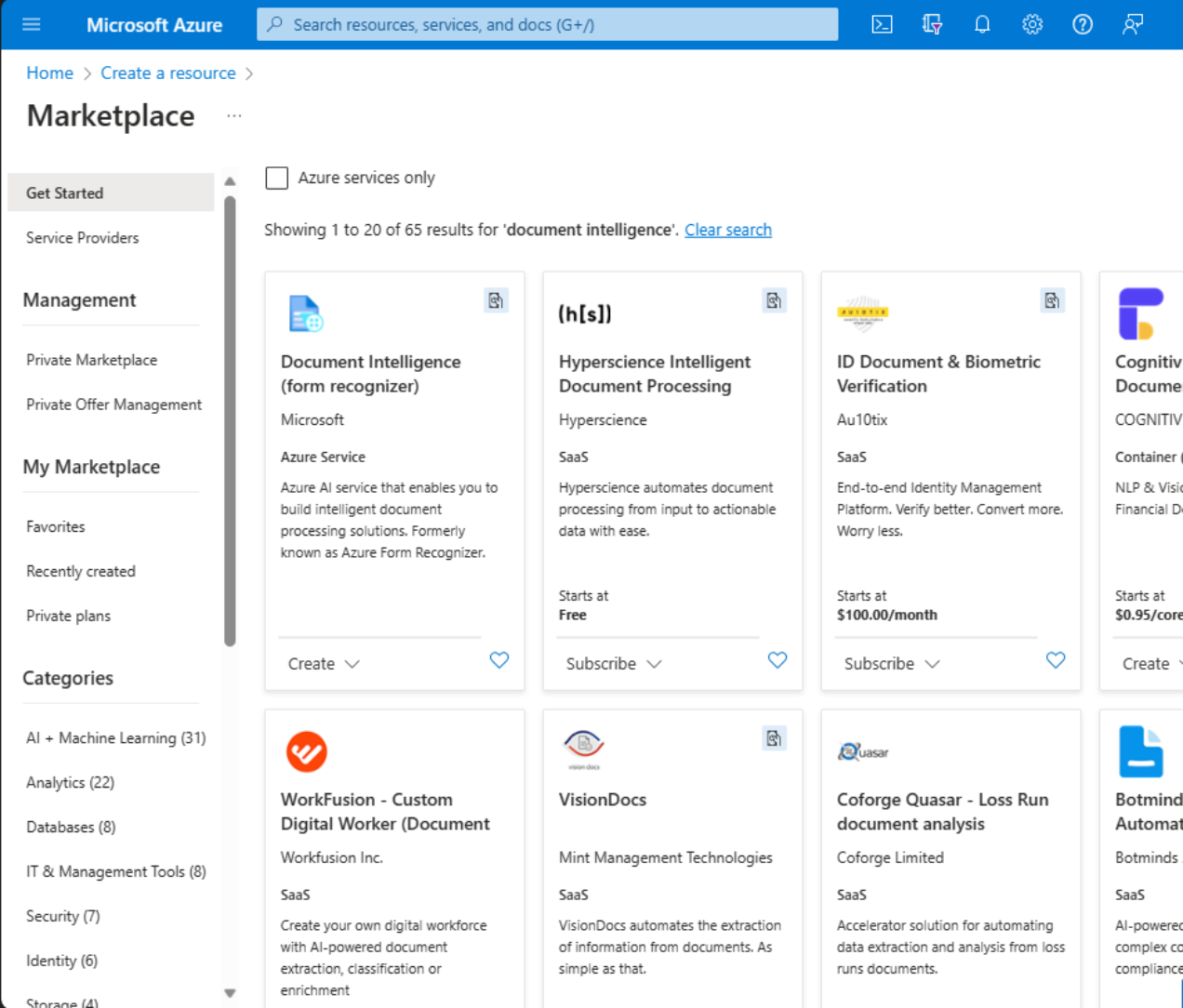
In this unit, you'll learn what Azure AI Document Intelligence does and how developers can configure it to support their forms and documents.

**What is Azure AI Document Intelligence?**

Until recently, getting data from a completed form into a database or any other data store required someone to enter it manually. Manual data entry is a slow and intensive task and can be expensive, especially if you have thousands of forms to enter. Operators often make reading or typing errors that reduce the accuracy of your data.

Manual data entry was the only option because it was difficult for computers to recognize printed or hand-written text. Now, AI has become commonplace and has enabled computers to recognize patterns, such as letter shapes in a piece of text, with a high degree of accuracy. We can use AI as an alternative to manual data entry with lower costs and fewer errors in the extracted data.

Azure AI Document Intelligence is an Azure service that you can use to analyze forms completed by your customers, partners, employers, or others and extract the data that they contain.



**Responsible use of AI**

AI technologies are powerful and have the potential to make great impacts on people's lives. To ensure that such impacts are positive, Microsoft uses the following principles when it designs and builds AI solutions. You should consider these principles whenever you make use of AI:

* **Fairness.** All AI systems should treat people fairly, regardless of race, belief, gender, sexuality, or other factors.
* **Reliability and safety.** All AI systems should give reliable answers with quantifiable confidence levels.
* **Privacy and security.** All AI systems should secure and protect sensitive data and operate within applicable data protection laws.
* **Inclusiveness.** All AI systems should be available to all users, regardless of their abilities.
* **Transparency.** All AI systems should operate understandably and openly.
* **Accountability.** All AI systems should be run by people who are accountable for the actions of those systems.

To conform to these principles, first you should take time to understand the AI system you're using and comprehend what it can do. For example, when using Document Intelligence, make sure you test your solution thoroughly with the forms you want it to read to ensure it extracts the data you expect. Ensure you collect only the data you need in forms and have the user's consent to store and analyze their information. Obtain legal advice on your solution, especially if the data it handles is personal or sensitive. Continue to use human agents to check on the deployed system and ensure your administrators can intervene in the solution to prevent harm. Continuously check the security of the system and its resilience against malicious attack and accidental data loss.

**Using models with Azure AI Document Intelligence**

Use a model to inform Azure AI Document Intelligence about the type of data you expect to be in the documents you're analyzing. If your forms have a common structure or layout, you can increase the accuracy of the results and control the structure of the output data by using the most appropriate model. Azure AI Document Intelligence outputs data in JSON format, which is widely compatible with many databases, other storage locations, and programming languages.

Azure AI Document Intelligence includes several prebuilt models for common types of forms and documents. If your forms are of one of these types, you can extract information from them without training your own custom models. It's very quick to create and deploy an Azure AI Document Intelligence solution when you use prebuilt models.

In Azure AI Document Intelligence, three of the prebuilt models are for general document analysis:

* Read
* General document
* Layout

The other prebuilt models expect a common type of form or document:

* Invoice
* Receipt
* W-2 US tax declaration
* ID Document
* Business card
* Health insurance card

**Important**

This list shows the documented prebuilt models avaiable at the time of writing. More prebuilt models are in development will be deployed soon. Check the Azure AI Document Intelligence documentation for the latest models.

If you have an unusual or unique type of form, you can use the above general document analysis prebuilt models to extract information from them. However, if you want to extract more specific information than the prebuilt models support, you can create a **custom model** and train it by using examples of completed forms.

You can also associate multiple custom models, trained on different types of document, into a single model, known as a **composed model**. With a composed model, users can submit forms of different types to a single service, which identifies them and selects the most appropriate custom model to use in their analysis.

**Azure AI Document Intelligence and Azure AI Vision**

As an Azure AI Service, Azure AI Document Intelligence is a high-level AI service that enables developers to access data in forms quickly. It's built on the lower level Azure AI Services, including Azure AI Vision.

If you use Azure AI Vision with its Optical Character Recognition (OCR) feature, you can submit photographed or scanned documents and extract their words and text in JSON format. This functionality is similar to Azure AI Document Intelligence and can make it difficult to choose from these services.

If you want to extract simple words and text from a picture of a form or document, without contextual information, Azure AI Vision OCR is an appropriate service to consider. You might want to use this service if you already have your own analysis code, for example. However, Azure AI Document Intelligence includes a more sophisticated analysis of documents. For example, it can identify key/value pairs, tables, and context-specific fields. If you want to deploy a complete document analysis solution that enables users to both extract and understand text, consider Azure AI Document Intelligence.

**Azure AI Document Intelligence tools**

If you want to try many features of Azure AI Document Intelligence without writing any code, you can use [**Azure AI Document Intelligence Studio**](https://formrecognizer.appliedai.azure.com/). This provides a visual tool for exploring and understanding the capabilities of Azure AI Document Intelligence and its support for your forms.

For example, you can use Azure AI Document Intelligence Studio to try analyzing your sales invoices and to explore the data produced by the **Invoice** prebuilt model. Then you could decide whether the prebuilt model extracts the values you need or whether to create your own custom model for a more unusual type of invoice.

To integrate Azure AI Document Intelligence into your own applications you'll need to write code. For example, you could enable users of your sales mobile app to scan receipts with their device's camera and call Azure AI Document Intelligence to obtain prices, costs, and custom details. The app could store this information in your customer relationship management database.

Azure AI Document Intelligence includes Application Programming Interfaces (APIs) for each of the model types you've seen. The following languages are supported:

* C#/.NET
* Java
* Python
* JavaScript

If you prefer to use another language, you can call Azure AI Document Intelligence by using its RESTful web service.

**Plan Azure AI Document Intelligence resources**

To build an Azure AI Document Intelligence solution, you must create and configure the necessary resources in your Azure subscription.

In your polling company, you're assessing Azure AI Document Intelligence to see if it can streamline your data entry workflow. You've decided to deploy an Azure AI Document Intelligence solution that will analyze data from your polling forms and you must plan the deployment to support your requirements. You want to know what resources to create in your Azure subscription.

In this unit, you'll learn how to choose and create Azure AI Document Intelligence resources.

**Azure AI Document Intelligence resources**

Azure AI Document Intelligence is an Azure service and conforms to Azure's resource management model. To create an Azure AI Document Intelligence solution, you start by adding a resource to your Azure subscription. When you create an Azure AI Document Intelligence resource, you can choose from **Free (F0)** or **Standard (S0)** tiers.

**Important**

If you're using the **Standard** tier, and find your requests are being throttled, you can submit an Azure Support Request to have the default limits increased. The **Free** tier is not available if you are using a multi-service resource.

**Create an Azure AI Document Intelligence resource**

To create an Azure AI Document Intelligence resource in Azure and obtain connection details, complete these steps:

1. In the [Azure portal](https://portal.azure.com/#home), select **Create a resource**.
2. In the **Search services and marketplace** box, type **Document Intelligence** and then press **Enter**.
3. In the **Document intelligence** page, select **Create**.
4. In the **Create Document intelligence** page, under **Project Details**, select your **Subscription** and either select an existing **Resource group** or create a new one.
5. Under **Instance details**, select a **Region** near your users.
6. In the **Name** textbox, type a unique name for the resource.
7. Select a **Pricing tier** and then select **Review + create**.
8. If the validation tests pass, select **Create**. Azure deploys the new Azure AI Document Intelligence resource.

**Connect to Azure AI Document Intelligence**

When you write an application that uses Azure AI Document Intelligence, you need two pieces of information to connect to the resource:

* **Endpoint.** This is the URL where the resource can be contacted.
* **Access key.** This is unique string that Azure uses to authenticate the call to Azure AI Document Intelligence.

To obtain these details:

1. In the [Azure portal](https://portal.azure.com/#home), navigate to the Azure AI Document Intelligence resource.
2. Under **Resource Management**, select **Keys and Endpoint**.
3. Copy either **KEY 1** or **KEY 2** and the **Endpoint** values and store them for use in your application code.

The following code shows how to use these connection details to connect your application to Azure AI Document Intelligence. If running this code, you'll need the libraries included in the top of the snippet.

In this example, a sample document at a specified URL is submitted for analysis to the general document model. Replace <endpoint> and <access-key> with the connection details you obtained from the Azure portal:

from azure.core.credentials import AzureKeyCredential

from azure.ai.documentintelligence import DocumentIntelligenceClient

from azure.ai.documentintelligence.models import AnalyzeResult

endpoint = "<your-endpoint>"

key = "<your-key>"

docUrl = "<url-of-document-to-analyze>"

document\_analysis\_client = DocumentIntelligenceClient(endpoint=endpoint,

credential=AzureKeyCredential(key))

poller = document\_analysis\_client.begin\_analyze\_document(

"prebuilt-layout", AnalyzeDocumentRequest(url\_source=docUrl

))

result: AnalyzeResult = poller.result()

**Choose a model type**

Completed100 XP

* 7 minutes

Azure AI Document Intelligence uses models to describe the documents and forms to expect. You can either use one of the prebuilt models, if you have a common type of document, or create and train your own models.

In your polling company, you use many different forms for surveys for different clients. They have some fields in common, such as "Respondent Name" and "Contact Telephone" but other fields are unique to each client company or party. You want to choose which models to use in your Document Intelligence solution and plan how to create them.

In this unit, you'll learn about the prebuilt models available in Azure AI Document Intelligence and when to create your own custom and composed models.

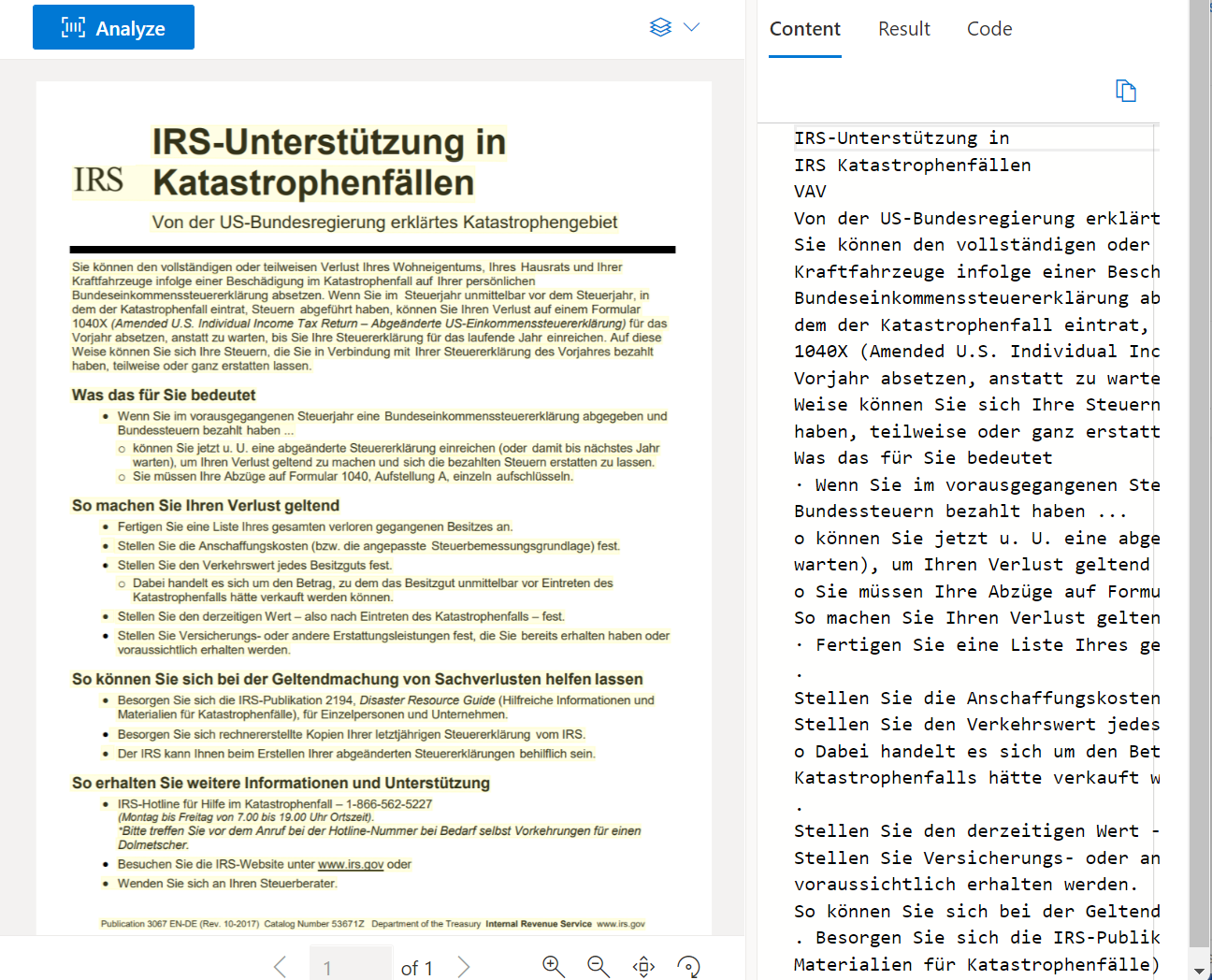
**Prebuilt models**

Document types such as invoices and receipts vary in different businesses and industry but have similar structures and key-value pairs. For example, a "Total cost" value is likely to appear on almost all invoices although it might be called "Total", "Sum", or some other name. Microsoft has provided a set of prebuilt models with Azure AI Document Intelligence to handle the most common types of documents. You don't have to train these models and you can create solutions using them very quickly.

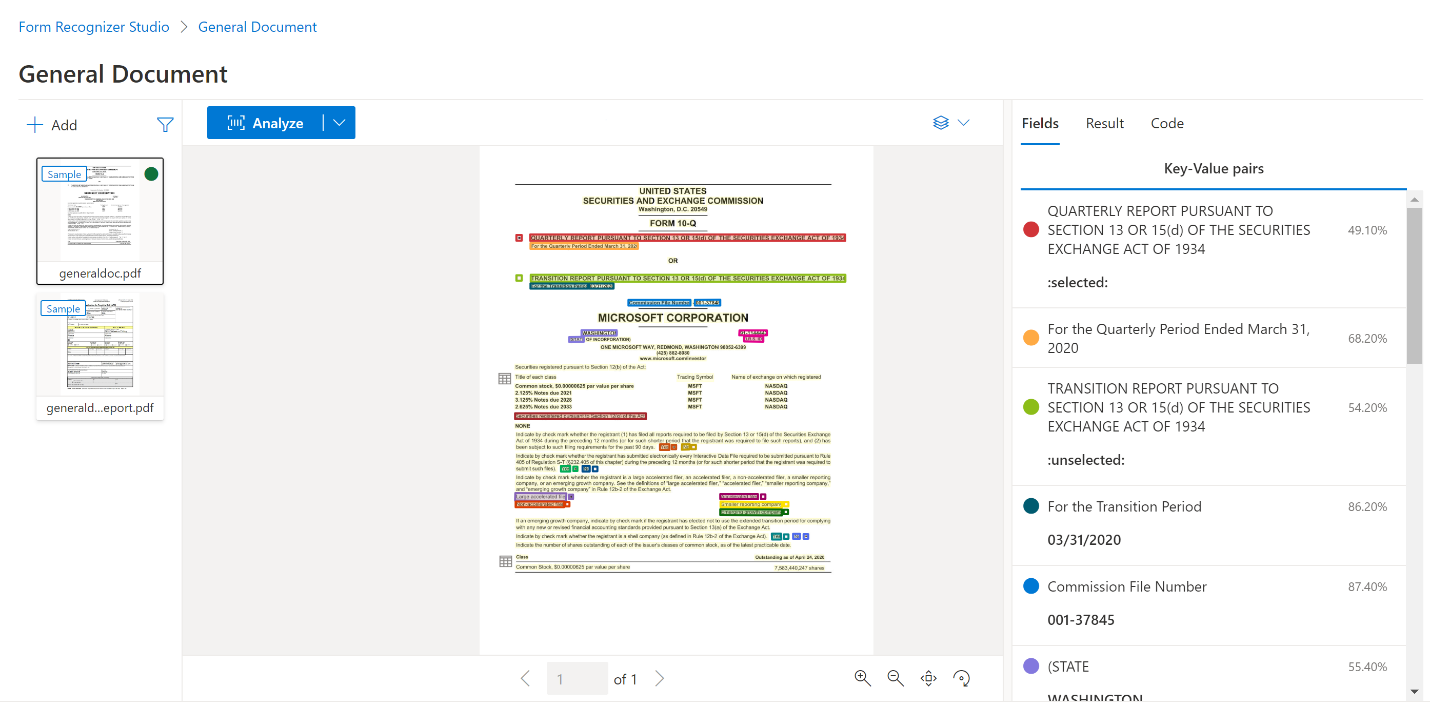
**General document analysis models**

Three of the prebuilt models are designed to handle general documents and extract words, lines, structure and other information such as the language the document is written in:

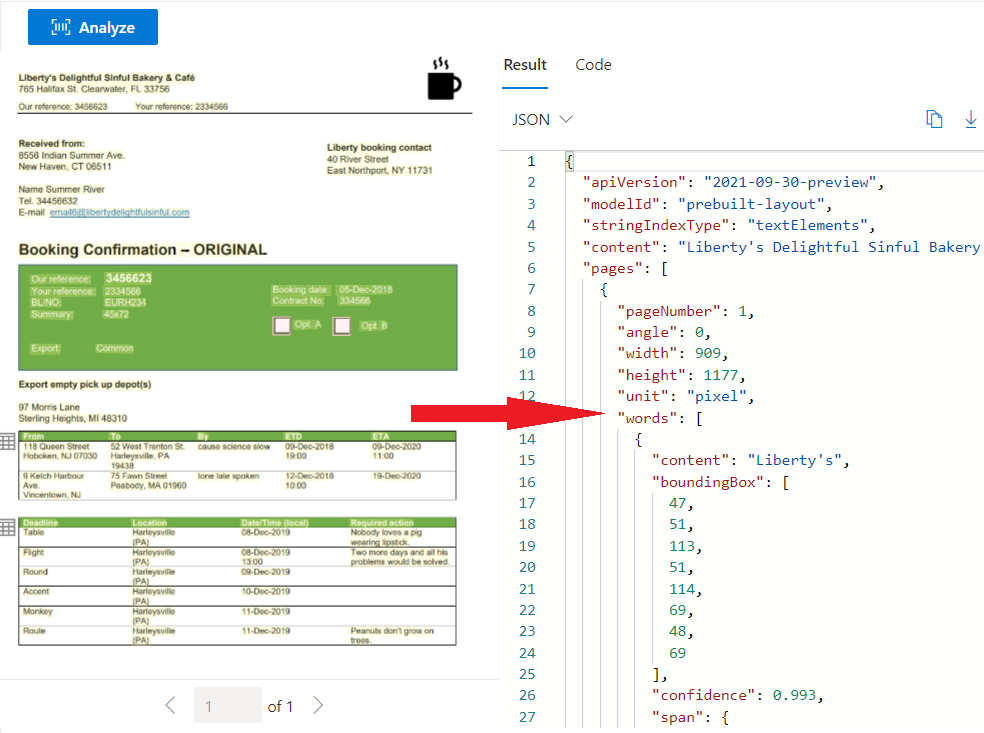
* **Read.** Use this model to extract words and lines from both printed and hand-written documents. It also detects the language used in the document.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-read-model.png#lightbox)

* **General document.** Use this model to extract key-value pairs and tables in your documents.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-general-document-model.png#lightbox)

* **Layout.** Use this model to extract text, tables, and structure information from forms. It can also recognize selection marks such as check boxes and radio buttons.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-layout-model.png#lightbox)

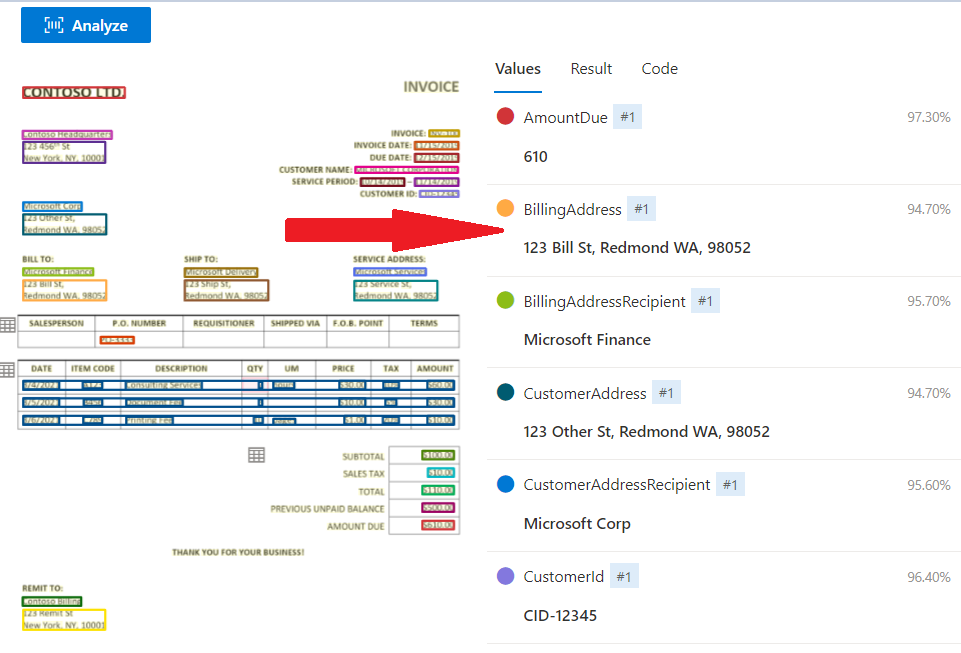
**Note**

The model screenshots above show Document Intelligence models extracting data in Azure AI Document Intelligence Studio.

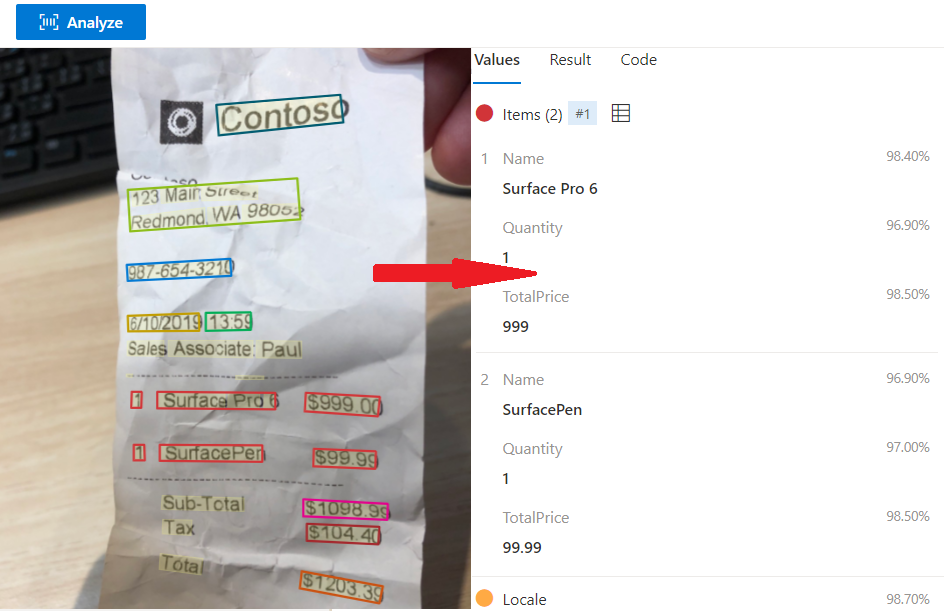
**Specific document type models**

The other prebuilt models are each designed to handle, and trained on, a specific and commonly used type of document. Some examples include:

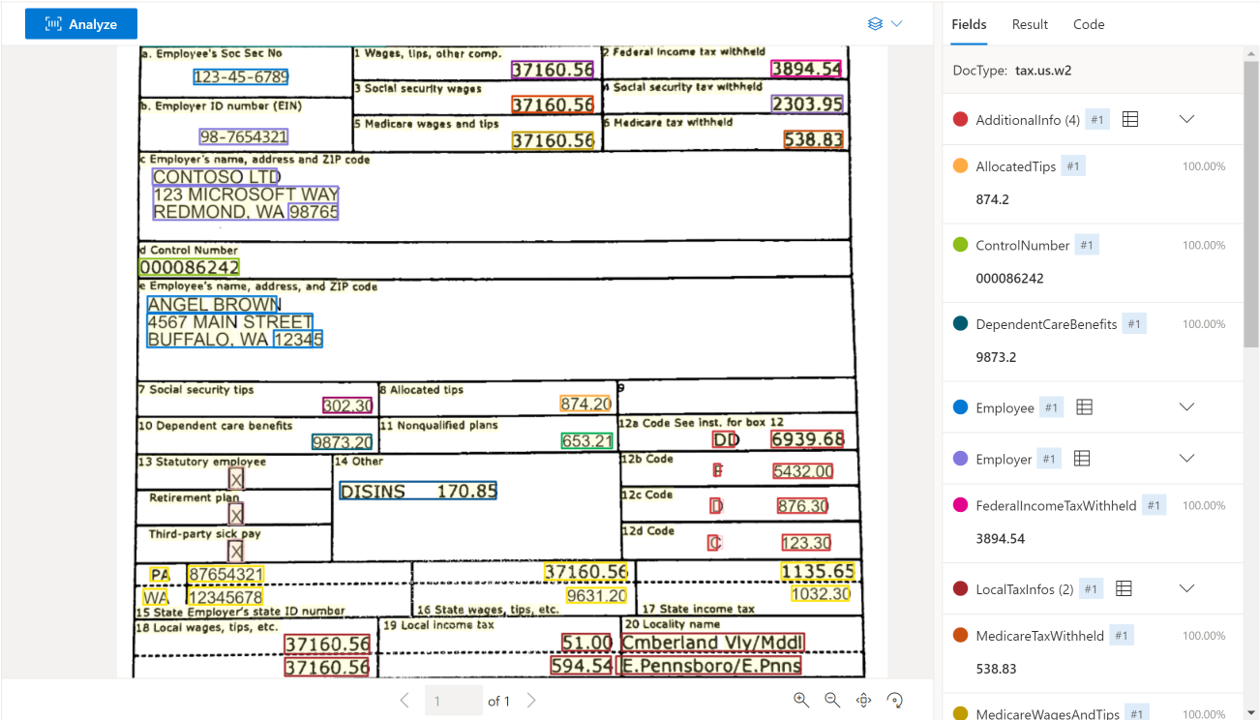
* **Invoice.** Use this model to extract key information from sales invoices in English and Spanish.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-invoice-model.png#lightbox)

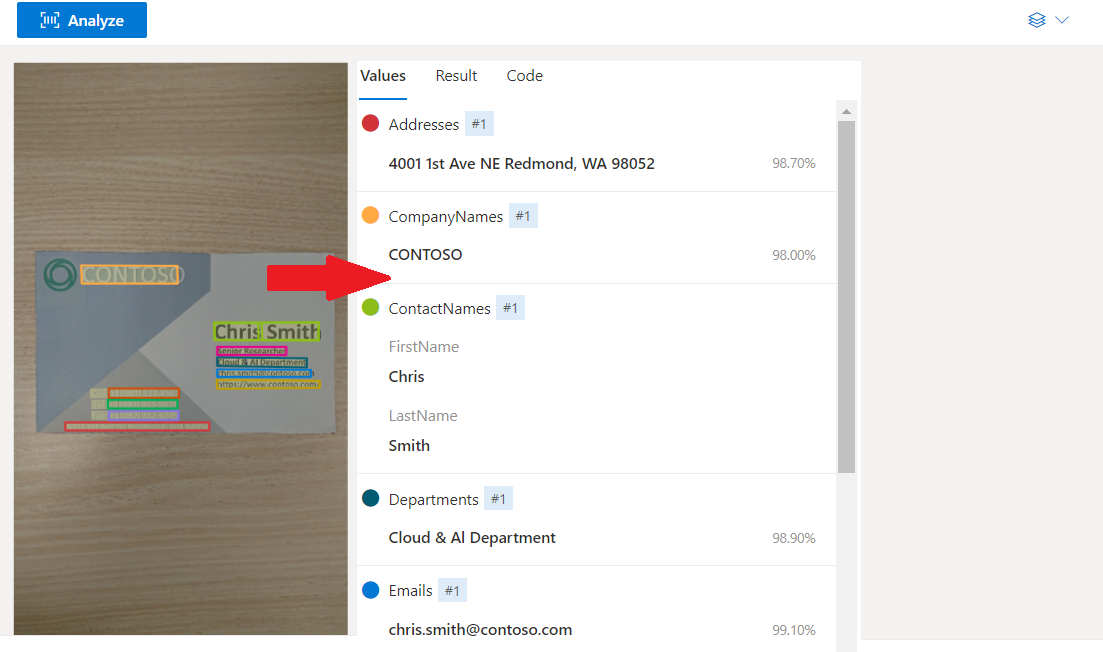
* **Receipt.** Use this model to extract data from printed and handwritten receipts.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-receipt-model.png#lightbox)

* **W-2.** Use this model to extract data from United States government's W-2 tax declaration form.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-w2-model.png#lightbox)

* **ID document.** Use this model to extract data from United States driver's licenses and international passports.
* **Business card.** Use this model to extract names and contact details from business cards.

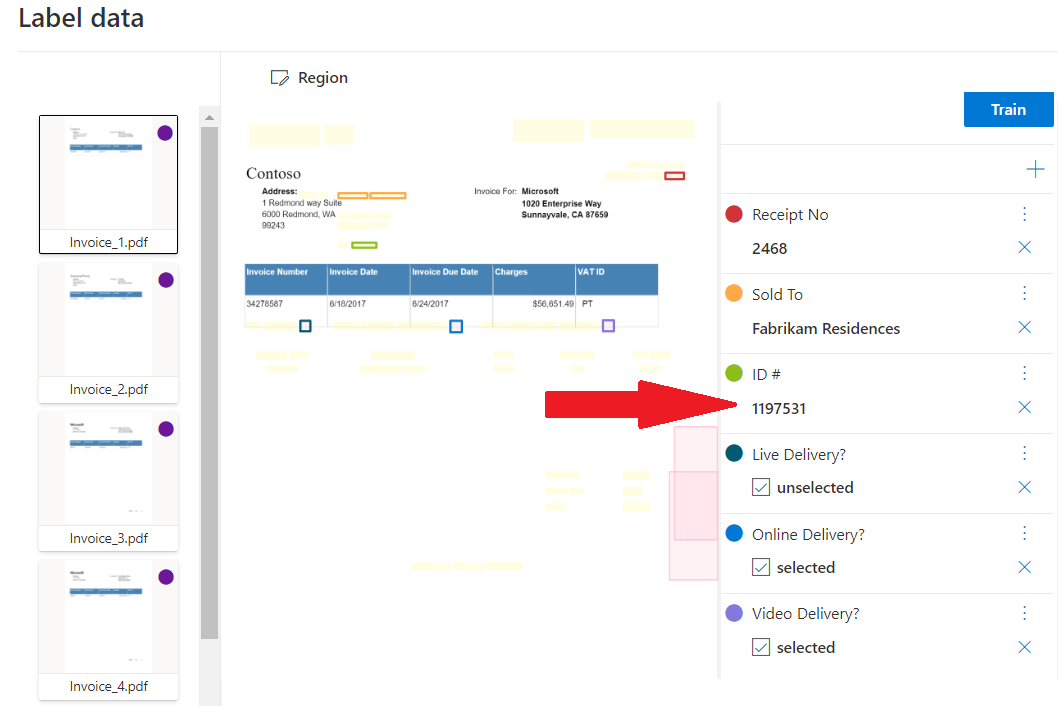
[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-business-card-model.png#lightbox)

**Custom models**

If the prebuilt models don't suit your purposes, you can create a custom model and train it to analyze the specific type of document users will send to your Azure AI Document Intelligence service. The general document analyzer prebuilt models can extract rich information from these forms and you might be able to use them if your requirements are to obtain general data. However, by using a custom model, trained on forms with similar structures and key-value pairs, you will obtain more predictable and standardized results from your unusual form types.

To train a custom model, you must supply at least five examples of the completed form but the more examples you supply, the greater the confidence levels Azure AI Document Intelligence will return when it analyzes input. The more varied your documents are in terms of structure and terminology, the greater the number of example documents you will need to supply to train a reliable model. You can either supply a labeled dataset to describe the expected data or allow the model to identify key-value pairs and table data based on what it finds in the example forms. Also, make sure your training forms include examples that span the full range of possible input. For example, if you are expecting both hand-written and printed entries, include them both in your training.

Once you have trained a custom model in this way, Azure AI Document Intelligence can accurately and predictably identify information in your unique forms.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/plan-form-recognizer-solution/media/04-train-custom-model.png#lightbox)

There are two kinds of custom model:

* **Custom template models.** A custom template model is most appropriate when the forms you want to analyze have a consistent visual template. If you remove all the user-entered data from the forms and find that the blank forms are identical, use a custom template model. Custom template models support 9 different languages for handwritten text and a wide range of languages for printed text. If you have a few different variations of the form templates, train a model for each of the variations and then compose the models together into a single model. The service will invoke the model best suited to analyze the document.
* **Custom neural models.** A custom neural model can work across the spectrum of structured to unstructured documents. Documents like contracts with no defined structure or highly structured forms can be analyzed with a neural model. Neural models work on English with the highest accuracy and a marginal drop in accuracy for other languages like German, French, Italian, Spanish, and Dutch. Try using the custom neural model first if your scenario is addressed by the model.

**Composed models**

A composed model is one that consists of multiple custom models. Typical scenarios where composed models help are when you don't know the submitted document type and want to classify and then analyze it. They are also useful if you have multiple variations of a form, each with a trained individual model. When a user submits a form to the composed model, Document Intelligence automatically classifies it to determine which of the custom models should be used in its analysis. In this approach, a user doesn't have to know what kind of document it is before submission. That can be helpful when you're using lots of similar forms or when you want to publish a single endpoint for all your form types.

**Important**

The results from a composed model include the docType property, which indicates the custom model that was chosen to analyze each form.

If you're using the Standard pricing tier, you can add up to 100 custom models into a single composed model. If you're using the Free pricing tier, you can only add up to 5 custom models.

**Check your knowledge**

Top of Form

**1. You want to create an Azure AI Document Intelligence model where the documents are in one of three formats: wills, probate declarations, and affidavits. Each has their own specific layout. What type of model should you use that will understand the format of the three document categories?**

A Read model.

A Layout model.

A Composed model.

**Correct. A Composed model consists of multiple custom models. Each submitted form is categorized as one of the custom form types and analyzed using the corresponding custom model.**

**2. You have developed a custom model that analyzes health assessment forms returned by patients to a medical practice. You've observed too much inaccuracy in the values that the model extracts for each field. What should you do to address this problem?**

Retrain the model with a larger number of example forms.

**Correct. The larger the number of example forms you use to train a model, the more accurate it will be and the higher the confidence levels will be.**

Change from a custom model to the general document model.

Change from the free tier to the standard tier.

**3. You want to call your Azure AI Document Intelligence solution from a mobile app by using an API. Which of the following programming languages is natively supported as an Azure AI Document Intelligence SDK?**

Python

**Correct. Microsoft publishes a Python API you can use to call Azure AI Document Intelligence services.**

Go

R

**4. Which of the following is an Azure AI Document Intelligence prebuilt model?**

Employment record

Resume

Receipt

**Correct. The receipt model can identify commonly used fields and their values in scanned or photographed receipt documents.**

**Use prebuilt Document intelligence 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.

**Understand prebuilt models**

Bottom of FormPrebuilt 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:

* **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**

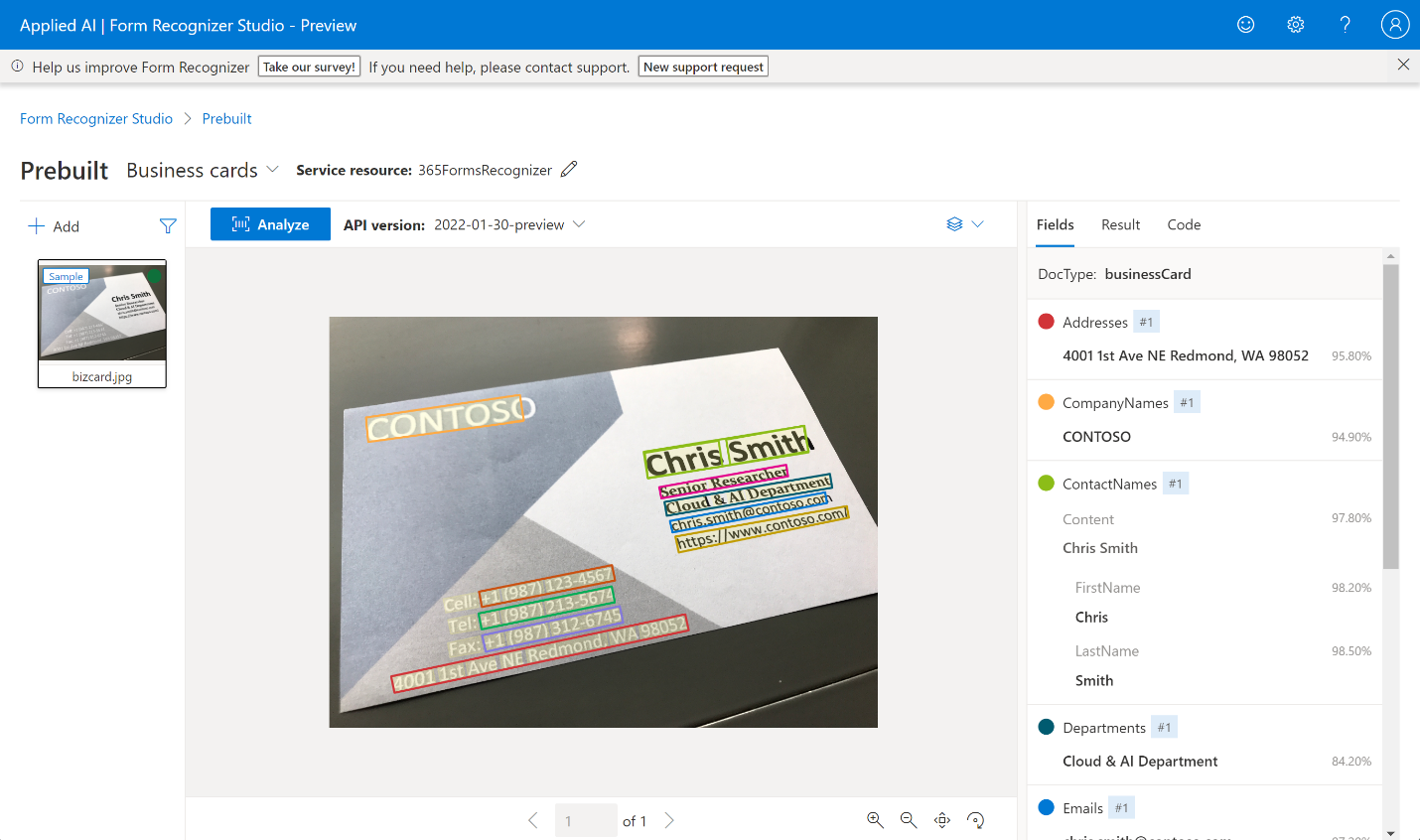
If you can, submit text-embedded PDF files because they eliminate errors in character recognition.

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](https://formrecognizer.appliedai.azure.com/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.

[[](https://learn.microsoft.com/en-us/training/wwl-data-ai/use-prebuilt-form-recognizer-models/media/2-studio-business-card-example.png#lightbox)](https://learn.microsoft.com/en-us/training/wwl-data-ai/use-prebuilt-form-recognizer-models/media/2-studio-business-card-example.png#lightbox)

**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.

Azure AI Document Intelligence is 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**

Completed100 XP

* 3 minutes

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**](https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/language-support-ocr#read-model) 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.

**Check your knowledge**

Top of Form

**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.

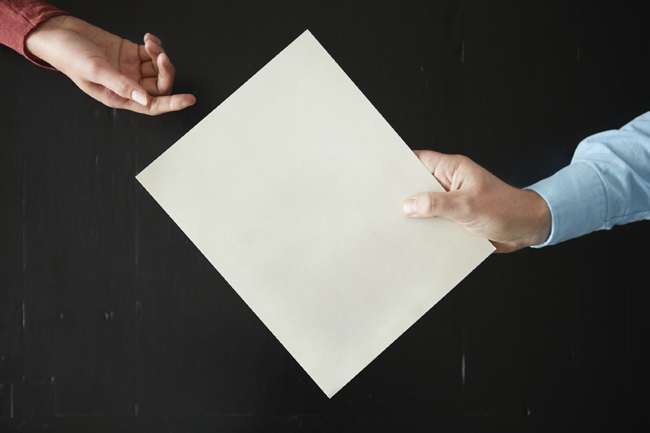
Convert the document to PDF format.

**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.**

**Extract data from forms with Azure Document intelligence**

**Introduction**

Forms are used to communicate information in every industry, every day. Many people still manually extract data from forms to exchange information.



Consider some of the instances when a person needs to process form data:

* When filing claims
* When enrolling new patients in an online management system
* When entering data from receipts to an expense report
* When reviewing an operations report for anomalies
* When selecting data from a report to give to a stakeholder

Without AI services, people need to manually sort through form documents to identify important information and then manually reenter data to record it. Some may also need to complete these tasks in real-time with a customer.

Azure Document Intelligence services provide the building blocks for automation by using intelligent services to extract data at scale and with accuracy. *Azure Document Intelligence* is a Vision API that extracts key-value pairs and table data from form documents.

**Uses of the Azure Document Intelligence service include**:

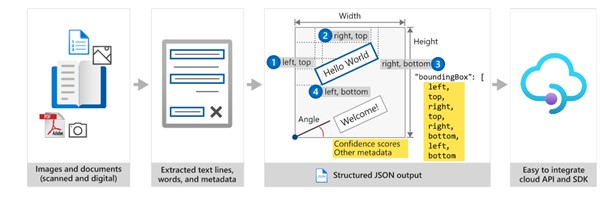
* Process automation
* Knowledge mining
* Industry-specific applications

Bottom of Form

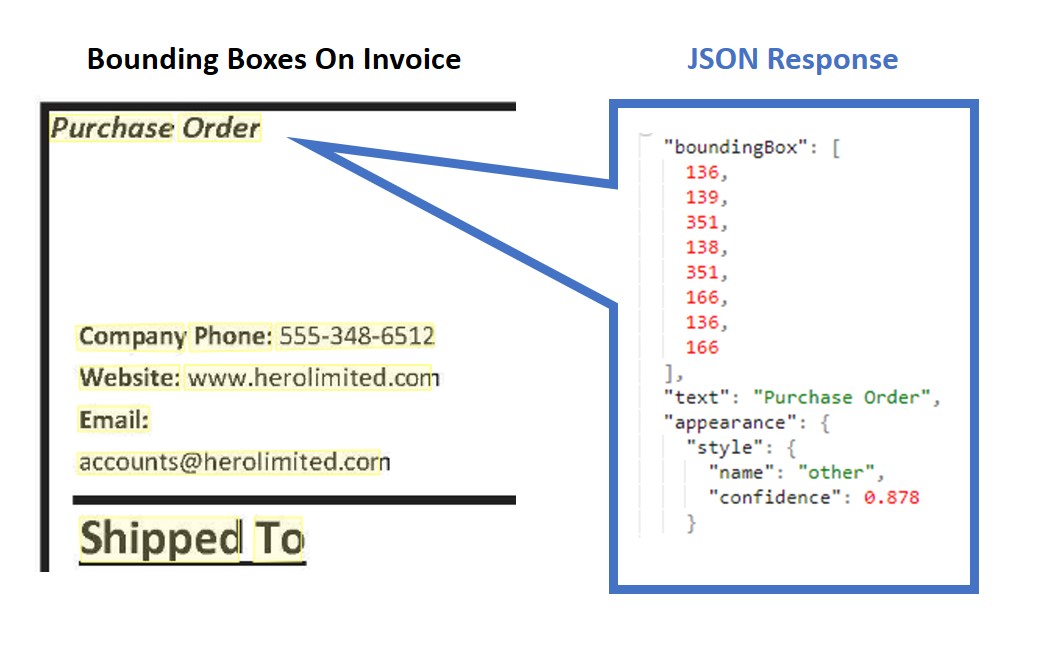
**What is Azure Document Intelligence?**

Azure Document Intelligence is one of many Azure AI Services, cloud-based artificial intelligence (AI) services with REST APIs and client library SDKs that can be used to build intelligence into your applications.

Azure Document Intelligence uses Optical Character Recognition (OCR) capabilities and deep learning models to extract text, key-value pairs, selection marks, and tables from documents.



OCR captures document structure by creating bounding boxes around detected objects in an image. The locations of the bounding boxes are recorded as coordinates in relation to the rest of the page. Azure Document Intelligence services return bounding box data and other information in a structured form with the relationships from the original file.



To build a high-accuracy model from scratch, people need to build deep learning models, use a large amount of compute resources, and face long model training times. These factors could make a project infeasible. Azure Document Intelligence provides underlying models that have been trained on thousands of form examples. The underlying models enable you to do high-accuracy data extraction from your forms with little to no model training.

**Azure Document Intelligence service components**

Azure Document Intelligence is composed of the following services:

* **Document analysis models**: which take an input of JPEG, PNG, PDF, and TIFF files and return a JSON file with the location of text in bounding boxes, text content, tables, selection marks (also known as checkboxes or radio buttons), and document structure.
* **Prebuilt models**: which detect and extract information from document images and return the extracted data in a structured JSON output. Azure Document Intelligence currently supports prebuilt models for several forms, including:
  + W-2 forms
  + Invoices
  + Receipts
  + ID documents
  + Business cards
* **Custom models**: custom models extract data from forms specific to your business. Custom models can be trained through the [Azure Document Intelligence Studio](https://formrecognizer.appliedai.azure.com/studio).

**Note**

Some Azure Document Intelligence features are in preview, as of the time this content was authored, and as a result, features and usage details might change. You should refer to the [**official page**](https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/overview) for up-to-date information.

**Access services with the client library SDKs or REST API**

You can access Azure Document Intelligence services by using a REST API, client library SDKs, and through the Azure Document Intelligence Studio to integrate the services into your workflow or application.

**Get started with Azure Document Intelligence**

To start a project with Azure Document Intelligence services, you need to prepare the following:

* An Azure resource subscription
* A selection of form files for data extraction

**Subscribe to a resource**

You can access Azure Document Intelligence services via:

* An **Azure AI Service resource**: a multi-service subscription key (used across multiple Azure AI Services)

**OR**

* An **Azure Document Intelligence resource**: a single-service subscription key (used only with a specific Azure AI Service)

**Note**

Create an Azure AI Services resource if you plan to access multiple Azure AI services under a single endpoint/key. For Azure Document Intelligence access only, create an Azure Document Intelligence resource. Please note that you'll need a single-service resource if you intend to use Microsoft Entra authentication.

You can subscribe to a service in the Azure portal or with the Azure Command Line Interface (CLI). You can learn more about the CLI commands [here](https://learn.microsoft.com/en-us/cli/azure/cognitiveservices/account#commands).

**Understand Azure Document Intelligence file input requirements**

Azure Document Intelligence works on input documents that meet these requirements:

* Format must be JPG, PNG, BMP, PDF (text or scanned), or TIFF.
* The file size must be less than 500 MB for paid (S0) tier and 4 MB for free (F0) tier.
* Image dimensions must be between 50 x 50 pixels and 10000 x 10000 pixels.
* The total size of the training data set must be 500 pages or less.

More input requirements can be found in the [documentation](https://learn.microsoft.com/en-us/azure/cognitive-services/form-recognizer/overview) for specific models.

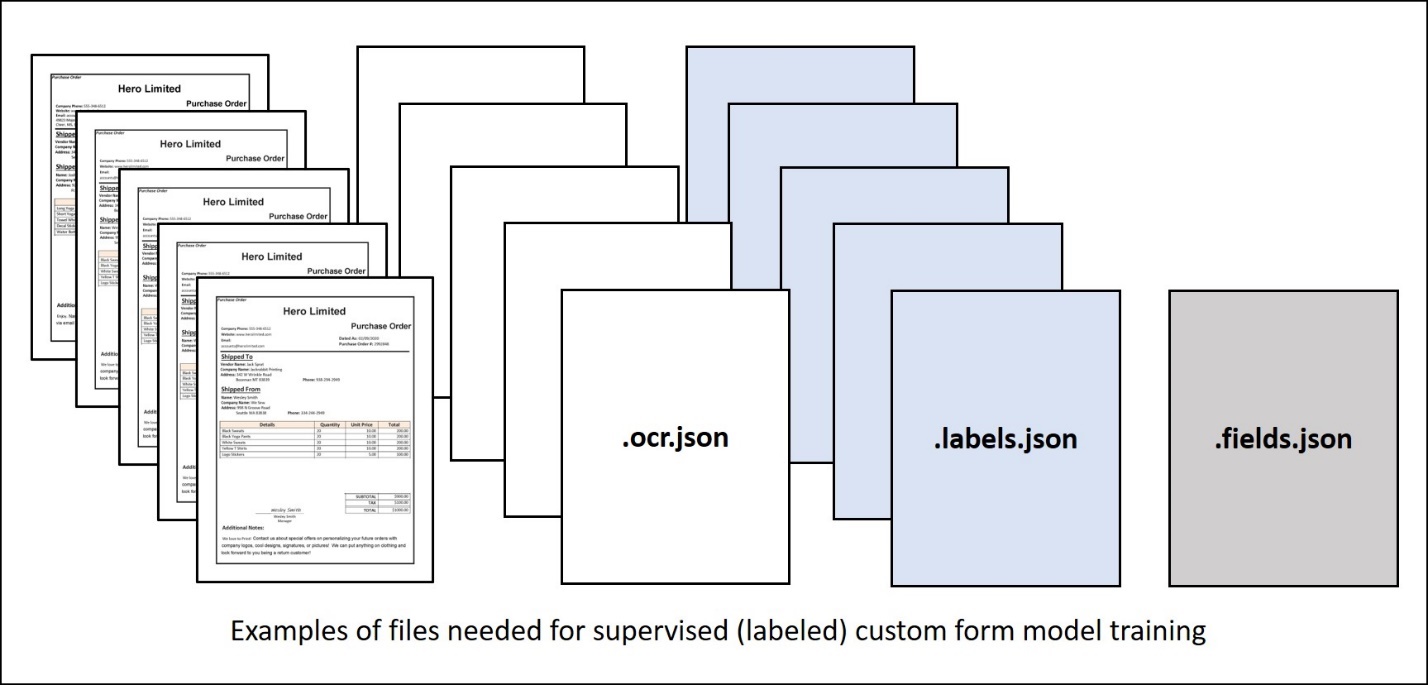
**Decide what component of Azure Document Intelligence to use**

After you have collected your files, decide what you need to accomplish.

* To use OCR capabilities to capture document analysis, use the [Layout model](https://learn.microsoft.com/en-us/azure/applied-ai-services/form-recognizer/concept-model-overview#layout), [Read model](https://learn.microsoft.com/en-us/azure/applied-ai-services/form-recognizer/concept-model-overview#read-preview), or [General Document model](https://learn.microsoft.com/en-us/azure/applied-ai-services/form-recognizer/concept-model-overview#general-document-preview).
* To create an application that extracts data from W-2s, Invoices, Receipts, ID documents, Health insurance, vaccination, and business cards, use a prebuilt model. These models do not need to be trained. Azure Document Intelligence services analyze the documents and return a JSON output.
* To create an application to extract data from your industry-specific forms, create a custom model. This model needs to be trained on sample documents. After training, the custom model can analyze new documents and return a JSON output.

**Train custom models**

Azure's Azure Document Intelligence service supports supervised machine learning. You can train custom models and create composite models with form documents *and* JSON documents that contain labeled fields.



To train a custom model:

1. Store sample forms in an Azure blob container, along with JSON files containing layout and label field information.
   * You can generate an **ocr.json** file for each sample form using the Azure Document Intelligence's **Analyze document** function. Additionally, you need a single **fields.json** file describing the fields you want to extract, and a **labels.json** file for each sample form mapping the fields to their location in that form.
2. Generate a shared access security (SAS) URL for the container.
3. Use the **Build model** REST API function (or equivalent SDK method).
4. Use the **Get model** REST API function (or equivalent SDK method) to get the trained **model ID**.

**OR**

1. Use the Azure Document Intelligence Studio to label and train. There are two types of underlying models for custom forms *custom template models* or *custom neural models*.
   * **Custom template models** accurately extract labeled key-value pairs, selection marks, tables, regions, and signatures from documents. Training only takes a few minutes, and more than 100 languages are supported.
   * **Custom neural models** are deep learned models that combine layout and language features to accurately extract labeled fields from documents.This model is best for semi-structured or unstructured documents.

**Use Azure Document Intelligence models**

**Using the API**

To extract form data using a custom model, use the **analyze document** function of either a supported SDK, or the REST API, while supplying model ID (generated during model training). This function starts the form analysis. which you can then request the result to get the analysis.

Example code to call your model:

endpoint = "YOUR\_DOC\_INTELLIGENCE\_ENDPOINT"

key = "YOUR\_DOC\_INTELLIGENCE\_KEY"

model\_id = "YOUR\_CUSTOM\_BUILT\_MODEL\_ID"

formUrl = "YOUR\_DOCUMENT"

document\_analysis\_client = DocumentAnalysisClient(

endpoint=endpoint, credential=AzureKeyCredential(key)

)

# Make sure your document's type is included in the list of document types the custom model can analyze

task = document\_analysis\_client.begin\_analyze\_document\_from\_url(model\_id, formUrl)

result = task.result()

A successful JSON response contains **analyzeResult** that contains the content extracted and an array of pages containing information about the document content.

Example **analyze document** JSON response:

{

"status": "succeeded",

"createdDateTime": "2023-10-18T23:39:50Z",

"lastUpdatedDateTime": "2023-10-18T23:39:54Z",

"analyzeResult": {

"apiVersion": "2022-08-31",

"modelId": "DocIntelModel",

"stringIndexType": "utf16CodeUnit",

"content": "Purchase Order\nHero Limited\nCompany Phone: 555-348-6512 Website: www.herolimited.com Email: accounts@herolimited.com\nPurchase Order\nDated As: 12/20/2020 Purchase Order #: 948284\nShipped To Vendor Name: Balozi Khamisi Company Name: Higgly Wiggly Books Address: 938 NE Burner Road Boulder City, CO 92848 Phone: 938-294-2949\nShipped From Name: Kidane Tsehaye Company Name: Jupiter Book Supply Address: 383 N Kinnick Road Seattle, WA 38383\nPhone: 932-299-0292\nDetails\nQuantity\nUnit Price\nTotal\nBindings\n20\n1.00\n20.00\nCovers Small\n20\n1.00\n20.00\nFeather Bookmark\n20\n5.00\n100.00\nCopper Swirl Marker\n20\n5.00\n100.00\nSUBTOTAL\n$140.00\nTAX\n$4.00\nTOTAL\n$144.00\nKidane Tsehaye\nManager\nKidane Tsehaye\nAdditional Notes: Do not Jostle Box. Unpack carefully. Enjoy. Jupiter Book Supply will refund you 50% per book if returned within 60 days of reading and offer you 25% off you next total purchase.",

"pages": [

{

"pageNumber": 1,

"angle": 0,

"width": 1159,

"height": 1486,

"unit": "pixel",

"words": [

{

"content": "Purchase",

"polygon": [

89,

90,

174,

91,

174,

112,

88,

112

],

"confidence": 0.996,

"span": {

"offset": 0,

"length": 8

}

},

{

"content": "Order",

"polygon": [

178,

91,

237,

91,

236,

113,

178,

112

],

"confidence": 0.997,

"span": {

"offset": 9,

"length": 5

}

},

...

**Understanding confidence scores**

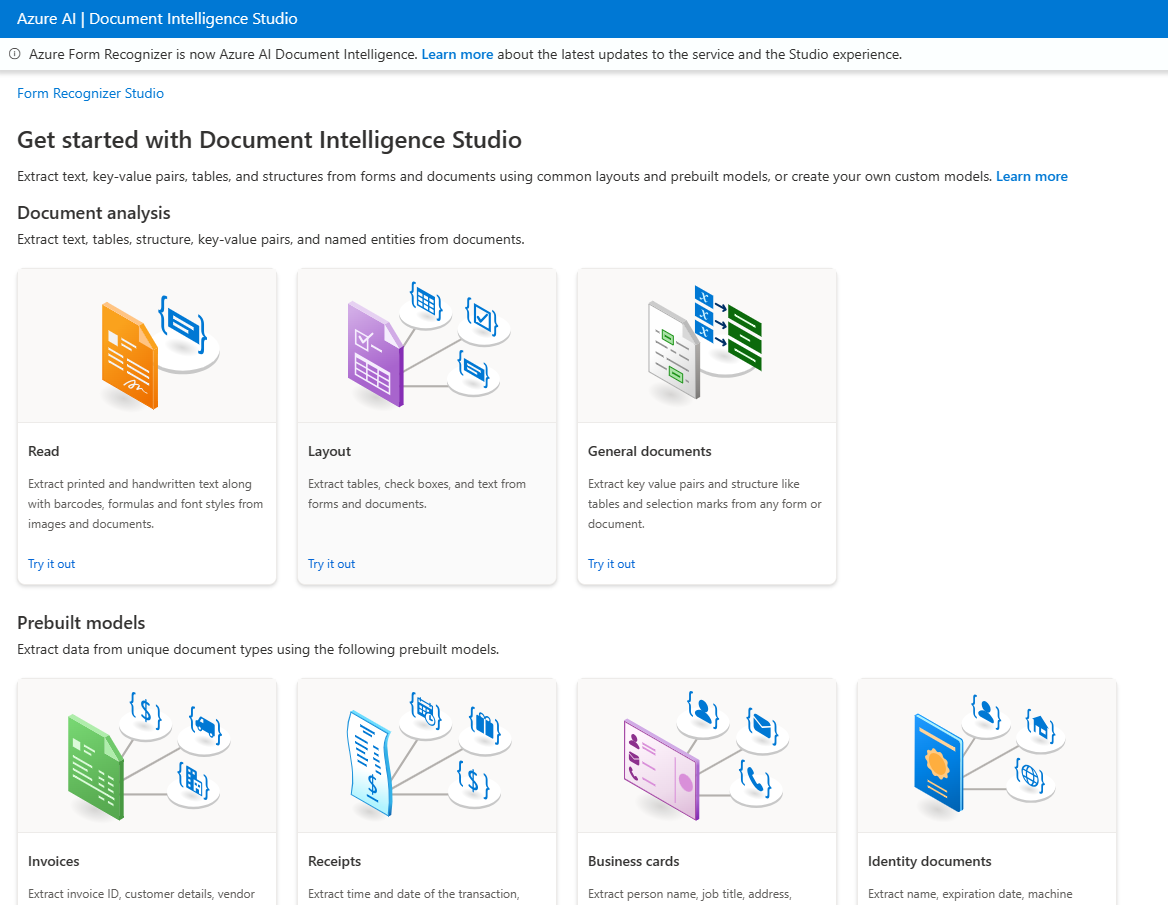
If the confidence values of the **analyzeResult** are low, try to improve the quality of your input documents.

You want to make sure that the form you're analyzing has a similar appearance to forms in the training set if the confidence values are low. If the form appearance varies, consider training more than one model, with each model focused on one form format.

Depending on the use case, you might find that a confidence score of 80% or higher is acceptable for a low-risk application. For more sensitive cases, like reading medical records or billing statements, a score of 100% is recommended.

**Use the Azure Document Intelligence Studio**

In addition to SDKs and the REST API, Azure Document Intelligence services can be accessed through a user interface called the Azure Document Intelligence Studio (preview), an online tool for visually exploring, understanding, and integrating features from the Azure Document Intelligence service. The Studio can be used to analyze form layouts, extract data from prebuilt models, and train custom models.



The Azure Document Intelligence Studio currently supports the following projects:

* **Document analysis models**
  + Read: Extract printed and handwritten text lines, words, locations, and detected languages from documents and images.
  + Layout: Extract text, tables, selection marks, and structure information from documents (PDF and TIFF) and images (JPG, PNG, and BMP).
  + General Documents: Extract key-value pairs, selection marks, and entities from documents.
* **Prebuilt models**
* **Custom models**

**Build Document analysis model projects**

To extract text, tables, structure, key-value pairs, and named entities with document analysis models:

* Create an Azure Document Intelligence or Azure AI Services resource
* Select either "Read", "Layout", or "General Documents" under the Document analysis models category
* Analyze your document. You'll need your Azure Document Intelligence or Azure AI service endpoint and key.

**Build prebuilt model projects**

To extract data from common forms with prebuilt models:

* Create an Azure Document Intelligence or Azure AI Services resource
* Select one of the "prebuilt models" including W-2s, Invoices, Receipts, ID documents, Health insurance, vaccination, and business cards.
* Analyze your document. You'll need your Azure Document Intelligence or Azure AI service endpoint and key.

**Build custom model projects**

You can use Azure Document Intelligence Studio's custom service for the entire process of training and testing custom models.

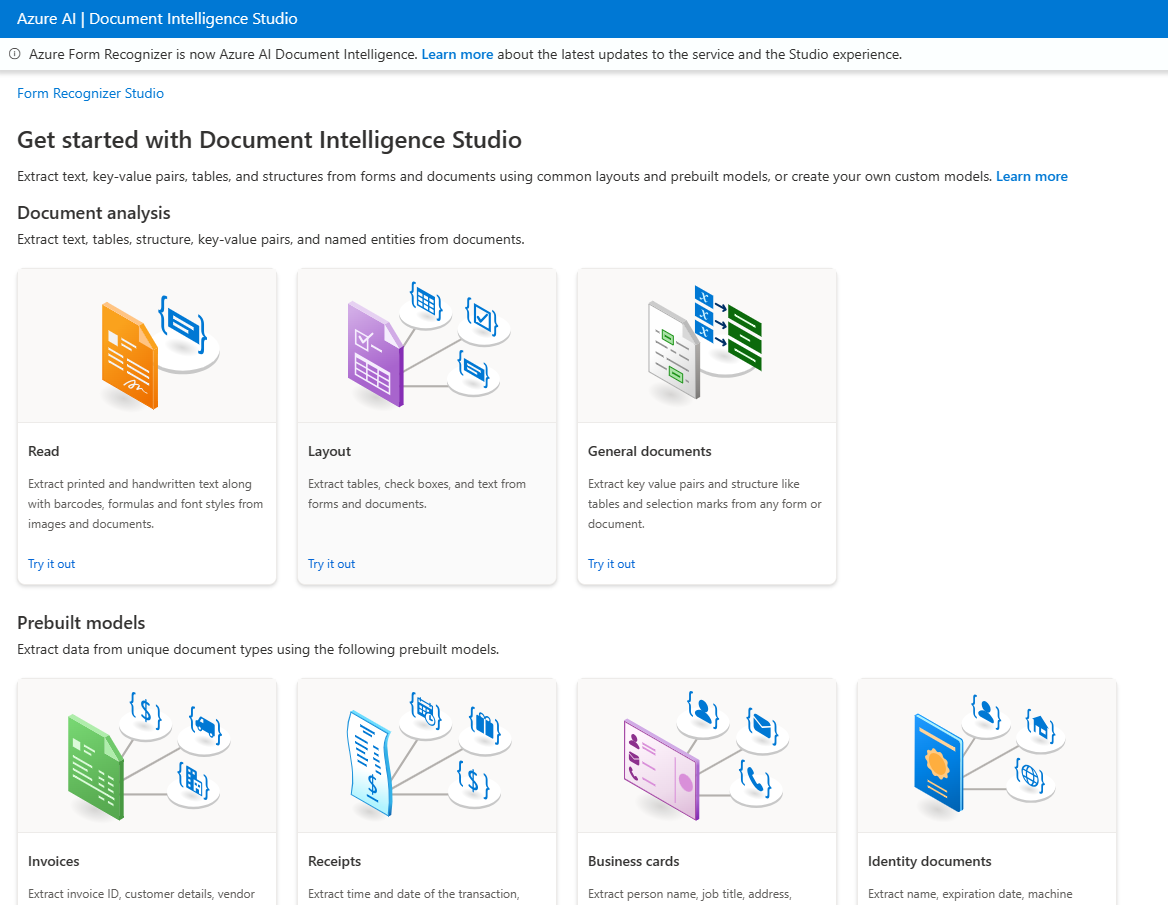
When you use Azure Document Intelligence Studio to build custom models, the **ocr.json** files, **labels.json** files, and **fields.json** file needed for training are automatically created and stored in your storage account.

To train a custom model and use it to extract data with custom models:

* Create an Azure Document Intelligence or Azure AI Services resource
* Collect at least 5-6 sample forms for training and upload them to your storage account container.
* Configure cross-domain resource sharing (CORS). CORS enables Azure Document Intelligence Studio to store labeled files in your storage container.
* Create a custom model project in Azure Document Intelligence Studio. You'll need to provide configurations linking your storage container and Azure Document Intelligence or Azure AI Service resource to the project.
* Use Azure Document Intelligence Studio to apply labels to text.
* Train your model. Once the model is trained, you'll receive a Model ID and Average Accuracy for tags.
* Test your model by analyzing a new form that wasn't used in training.

**Use the Azure Document Intelligence Studio**

In addition to SDKs and the REST API, Azure Document Intelligence services can be accessed through a user interface called the Azure Document Intelligence Studio (preview), an online tool for visually exploring, understanding, and integrating features from the Azure Document Intelligence service. The Studio can be used to analyze form layouts, extract data from prebuilt models, and train custom models.



The Azure Document Intelligence Studio currently supports the following projects:

* **Document analysis models**
  + Read: Extract printed and handwritten text lines, words, locations, and detected languages from documents and images.
  + Layout: Extract text, tables, selection marks, and structure information from documents (PDF and TIFF) and images (JPG, PNG, and BMP).
  + General Documents: Extract key-value pairs, selection marks, and entities from documents.
* **Prebuilt models**
* **Custom models**

**Build Document analysis model projects**

To extract text, tables, structure, key-value pairs, and named entities with document analysis models:

* Create an Azure Document Intelligence or Azure AI Services resource
* Select either "Read", "Layout", or "General Documents" under the Document analysis models category
* Analyze your document. You'll need your Azure Document Intelligence or Azure AI service endpoint and key.

**Build prebuilt model projects**

To extract data from common forms with prebuilt models:

* Create an Azure Document Intelligence or Azure AI Services resource
* Select one of the "prebuilt models" including W-2s, Invoices, Receipts, ID documents, Health insurance, vaccination, and business cards.
* Analyze your document. You'll need your Azure Document Intelligence or Azure AI service endpoint and key.

**Build custom model projects**

You can use Azure Document Intelligence Studio's custom service for the entire process of training and testing custom models.

When you use Azure Document Intelligence Studio to build custom models, the **ocr.json** files, **labels.json** files, and **fields.json** file needed for training are automatically created and stored in your storage account.

To train a custom model and use it to extract data with custom models:

* Create an Azure Document Intelligence or Azure AI Services resource
* Collect at least 5-6 sample forms for training and upload them to your storage account container.
* Configure cross-domain resource sharing (CORS). CORS enables Azure Document Intelligence Studio to store labeled files in your storage container.
* Create a custom model project in Azure Document Intelligence Studio. You'll need to provide configurations linking your storage container and Azure Document Intelligence or Azure AI Service resource to the project.
* Use Azure Document Intelligence Studio to apply labels to text.
* Train your model. Once the model is trained, you'll receive a Model ID and Average Accuracy for tags.
* Test your model by analyzing a new form that wasn't used in training.

**Knowledge check**

Top of Form

**1. A person plans to use an Azure Document Intelligence prebuilt invoice model. To extract document data using the model and REST API language, what are two calls they need to make to the API?**

Train Model and Get Model Labels

Analyze Invoice and Get Analyze Invoice Result

**Correct: The Analyze Invoice function starts the form analysis and returns a result ID, which they can pass in a subsequent call to the Get Analyze Invoice Result function to retrieve the results.**

Create Azure Document Intelligence and Get Analyze Invoice Result

**2. A person needs to build an application that submits expense claims and extracts the merchant, date, and total from scanned receipts. What's the best way to build the application?**

Use the Read API of the Computer Vision service.

Use Azure Document Intelligence's prebuilt receipts model

**Correct: Use the Azure Document Intelligence's prebuilt receipts model. It can intelligently extract the required fields even if the scanned receipts have different names in them.**

Use Azure Document Intelligence's Layout service

**3. A person is building a custom model with Azure Document Intelligence services. What is required to train a model?**

Along with the form to analyze, JSON files need to be provided.

**Correct: The labels needed in training are referenced in the ocr.json files, labels.json files, and single fields.json file.**

Training must be done through language-specific SDKs.

Nothing else is required.

**Create a composed Document intelligence model**

**Introduction**

If you've created and trained custom models in Azure AI Document Intelligence, you can combine them into a single composite model and publish that as a single service. Composite models help when there are multiple versions of a form in use or when users find it difficult to keep track of the correct model for each form. They can also assist when you want customers to upload different types of documents to a single location for analysis and you don't know which type was uploaded.

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. For each poll you conduct, respondents may complete up to three different form types but you want to analyze them with a single service. You want to create an AI service that can recognize these three form types and apply a different analysis to each. The three form types each have a different set of fields that you want to extract.

**Understand composed models**

Composed models in Azure AI Document Intelligence enable users to submit a form when they don't know which is the best model to use.

In your polling company, you often change versions of the forms you use to collect data from respondents. When your users submit these forms for analysis, they sometimes choose the wrong custom model. You'd like to find a way for them to submit forms without having to specify the model version.

Here, you'll learn how composed models can help to send a form to the correct custom model automatically.

**What are composed models?**

When you have forms with unusual or unique formats, you can create and train your own custom models in Azure AI Document Intelligence. A custom model can provide field extraction for the data that is unique to your form and generate data targeted to your unique business application.

You can create custom models of two types:

* **Custom template models.** Use custom template models when your forms have a consistent visual template. The formatting and layout should be consistent across all completed examples of the form.
* **Custom neural models.** Use custom neural models when your forms are less consistent, semi-structured or unstructured.

You can create hundreds of custom models in a single Azure AI Document Intelligence resource. When you send a form to be analyzed, you have to specify the identity of the model you want to use in the request:

// Create an Azure AI Document Intelligence client

string endpoint = "<endpoint>";

string apiKey = "<apiKey>";

var credential = new AzureKeyCredential(apiKey);

var client = new DocumentAnalysisClient(new Uri(endpoint), credential);

// Specify the model to use

string modelId = "<modelId>";

// Specify the file to analyze

Uri fileUri = new Uri("<fileUri>");

// Call the model

AnalyzeDocumentOperation operation = await client.StartAnalyzeDocumentFromUriAsync(modelId, fileUri);Bottom of Form

If you have a lot of custom models, it can be difficult to keep track of them and specify the right model in the request. You might also have used lots of slightly different forms to collect data.

For example, suppose you've been taking a weekly survey of the electorate for the last year. In that time, you've revised your form twice with a new layout and you've trained a separate custom template model for each version. The new forms haven't always been distributed to all your surveyors quickly so there is a mix of form versions in each weekly survey.

In situations like this, a composed model can be helpful. A composed model consists of multiple custom models. When you submit a form for analysis, Azure AI Document Intelligence categorizes it and selects the best custom model to use for the analysis. This categorization means you don't have to keep track of the correct custom model yourself and specify it in the request.

**Using Composed models**

Once you've created a set of custom models, you must assemble them into a composed model. You can do this in a Graphical User Interface (GUI) by using Azure AI Document Intelligence Studio, or by using the StartCreateComposedModelAsync() method in custom code.

Submit a form for analysis in the same way you do for your individual custom models. Remember to specify the model ID of the composed model.

In the results from the composed model, you can determine which custom model has been used for the analysis by checking the docType field.

The number of custom models you can create in an Azure AI Document Intelligence resource depends on the type of custom forms you use and your tier:

| **Type of model** | **Maximum number in Free (F0) tier** | **Maximum number in Standard (S0) tier** |
| --- | --- | --- |
| Custom Template | 500 | 5000 |
| Custom Neural | 100 | 500 |
| Composed | 5 | 200 |

The maximum number of custom models that can be added to a single composed model is 100.

**Custom model compatibility**

There are some restrictions on the models that can be added to the same composed models:

* Custom template models are composable with other custom template models across 3.0 and 2.1 API versions.
* Custom neural models are composable with other custom neural models.
* Custom neural models can't be composed with custom template models

**Assemble composed models**

You can create a composed model by assembling custom models in Azure AI Document Intelligence or in your own code.

In your polling company, you want to create a composed model that will categorize and correctly analyze all the versions of your main political polling form. You need to know how to compose models.

Here, you'll learn how to create composed models.

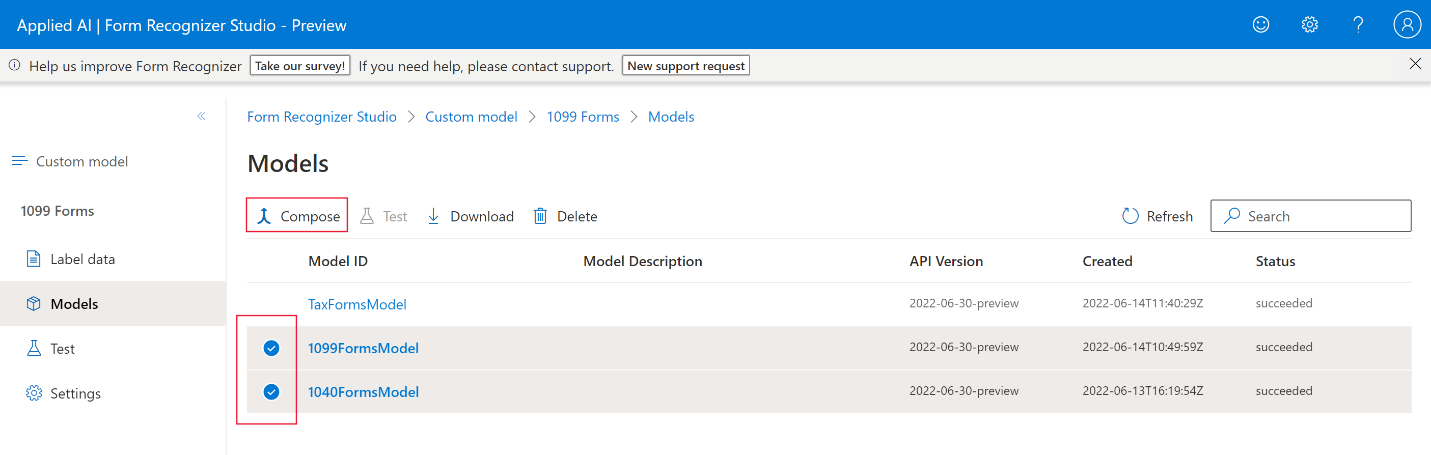
**Create a composed model in Document Intelligence Studio**

Before you start creating a composed model, you'll need:

* An Azure AI Document Intelligence resource in your Azure subscription.
* A set of custom models, trained and labeled, that you want to add to the composed model.

If you prefer to use a Graphical User Interface (GUI), you can create a composed model in the Azure AI Document Intelligence Studio:

1. In [Azure AI Document Intelligence Studio](https://formrecognizer.appliedai.azure.com/studio), on the home page, select **Custom model**.
2. Under **My Projects** select one of the custom models and then in the left navigation, select **Models**.
3. In the **Models** list, select all the models that you want to include in the new composed model, and then select **Compose**.

[](https://learn.microsoft.com/en-us/training/wwl-data-ai/create-composed-form-recognizer-model/media/3-create-composed-model.png#lightbox)

1. In the **Compose a new model** dialog, enter a **Model ID** and a **Description** for the composed model and then select **Compose**.

**Create a composed model in code**

If you're using one of the Azure AI Document Intelligence SDKs to create a composed model by executing code, you have to start by creating an instance of the DocumentModelAdministrationClient object, and connecting it to Azure AI Document Intelligence with its endpoint and API key:

C#

string endpoint = "<endpoint>";

string apiKey = "<apiKey>";

var credential = new AzureKeyCredential(apiKey);

var client = new DocumentModelAdministrationClient(new Uri(endpoint), credential);

To create the composed model, assemble the model IDs of all the custom models in a List, and then pass that list to the StartCreateComposedModelAsync() method:

C#

List<string> modelIds = new List<string>()

{

firstCustomModel.ModelId,

secondCustomModel.ModelId,

thirdCustomModel.ModelId,

};

BuildModelOperation operation = await client.StartCreateComposedModelAsync(modelIds, modelDescription: "Composed model example");

Response<DocumentModel> operationResponse = await operation.WaitForCompletionAsync();

Once the composed model has been created, you can send a form to it for analysis using the same code you would use to send a form to any other custom model. Remember to specify the model ID of the composed model in your call.

In the results, use the docType property to determine the constituent custom model that was used to analyze each document.

**Check your knowledge**

Top of Form

**1. You have a composed model that consists of three custom models. You're writing code that sends forms to the composed model and you need to check which of the custom models was used to analyze each form. Which property should you use from the returned JSON?**

modelId.

status.

docType.

**Correct. The docType property includes the model ID of the custom model that was used to analyze the document.**

**2. You're trying to create a composed model but you're receiving an error. Which of the following should you check?**

That the custom models were trained with labels.

**Correct. Only custom models that have been trained with labeled example forms can be added to a composed model.**

That the custom models all have the same model ID.

That the custom models all have the same list of fields.

Bottom of Form