

## 5.4 - Create a composed Document Intelligence model

- [Overview](#)
  - [Introduction](#)
  - [Understand composed models](#)
    - [What are composed models?](#)
    - [Using Composed models](#)
    - [Custom model compatibility](#)
  - [Assemble composed models](#)
    - [Create a composed model in Document Intelligence Studio](#)
    - [Create a composed model in code](#)
  - [Exercise - Create a composed model](#)
    - [Run Cloud Shell](#)
    - [Set up resources](#)
    - [Create the 1040 Forms custom model](#)
    - [Label the 1040 Forms custom model](#)
    - [Train the 1040 Forms custom model](#)
    - [Create the 1099 Forms custom model](#)
    - [Label the 1099 Forms custom model](#)
    - [Train the 1099 Forms custom model](#)
    - [Create and assemble a composed model](#)
    - [Use the composed model](#)
    - [Clean up the exercise resources](#)
  - [Knowledge Check](#)
  - [Summary](#)
- 

### Overview

Learn how to assemble custom models into composed solutions that can analyze different types of your own documents.

In this module, you'll learn to:

- Describe business problems that you would use custom models and composed models to solve.
  - Train a custom model to obtain data from forms with unusual structures.
  - Create a composed model that can analyze forms in multiple formats.
- 

### 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.

Here, you'll learn how to use composite models to handle multiple form types and analyze them with different custom models in a single service.

---

## 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);
```

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**
- [Composed custom models](#)
- [Code sample - Model compose](#)

## 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.

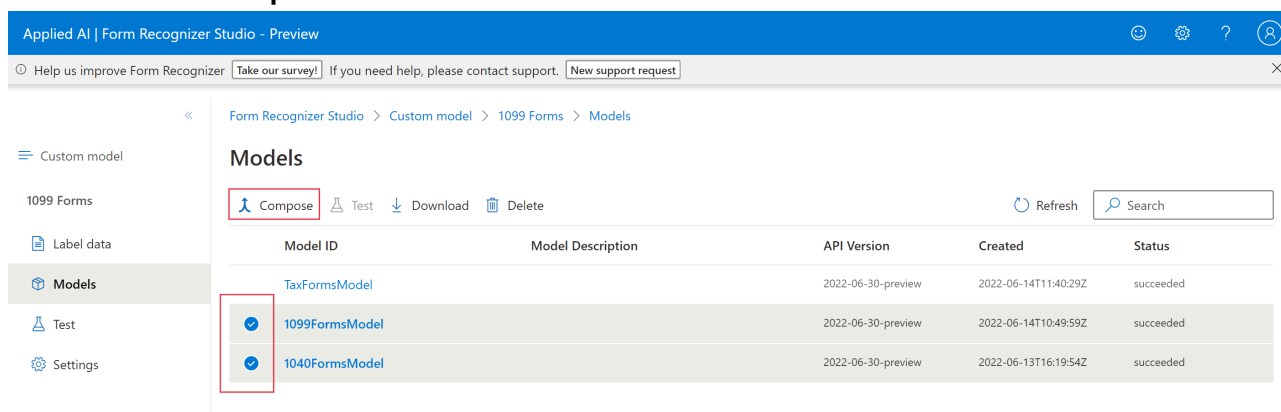
## Create a composed model in Document Intelligence Studio

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](#), 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**.
4. 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:

```
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:

```
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.

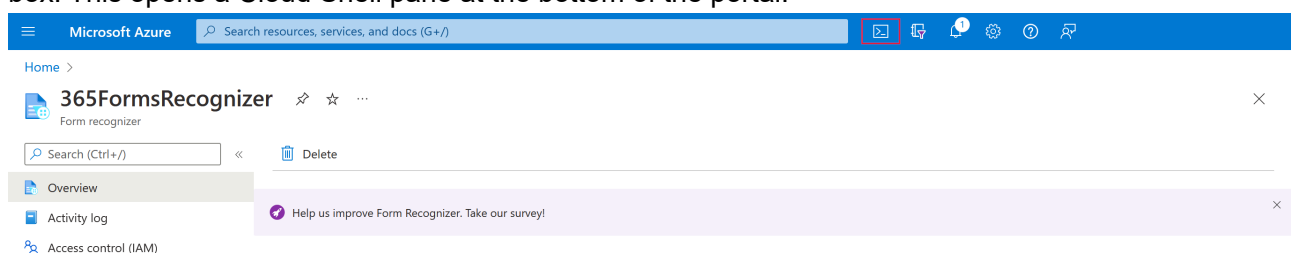
## Exercise - Create a composed model

In this exercise, you'll create and train two custom models that analyze different tax forms. Then, you'll create a composed model that includes both of these custom models. You'll test the model by submitting a form and you'll check that it recognizes the document type and labeled fields correctly.

### Run Cloud Shell

To start the exercise, let's connect to Cloud Shell, which you'll use to run the setup script:

1. In the [Azure portal](#), select the **[>\_]** (Cloud Shell) button at the top of the page to the right of the search box. This opens a Cloud Shell pane at the bottom of the portal.



2. The first time you open the Cloud Shell, you might be prompted to choose the type of shell you want to use (*Bash* or *PowerShell*). Select **Bash**. If you don't see this option, skip the step.
3. If you're prompted to create storage for your Cloud Shell, ensure your subscription is specified and select **Create storage**. Then wait a minute or so for the storage to be created.

4. Make sure the type of shell indicated on the top left of the Cloud Shell pane is switched to *Bash*. If it is *PowerShell*, switch to *Bash* by using the drop-down menu.
5. Wait for Bash to start.

## Set up resources

We'll use a script to create the Azure AI Document Intelligence resource, a storage account with sample forms, and a resource group:

1. In the Cloud Shell, to clone the code repository, enter this command:

```
rm -r doc-intelligence -f
git clone https://github.com/MicrosoftLearning/mslearn-ai-document-intelligence doc-intelligence
```

2. Change the **03-composed-model** directory and then execute the setup script:

```
cd doc-intelligence/Labfiles/03-composed-model/
bash setup.sh
```

## Create the 1040 Forms custom model

To create a composed model, we must first create two or more custom models. To create the first custom model:

1. In a new browser tab, start the [Azure AI Document Intelligence Studio](#).
2. Scroll down, and then under **Custom model**, select **Custom model**.
3. If you're asked to sign into your account, use your Azure credentials.
4. If you're 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. Under **My Projects**, select **+ Create a project**.
6. In the **Project name** textbox, type **1040 Forms**, and then select **Continue**.
7. On the **Configure service resource** page, in the **Subscription** drop-down list, select your Azure subscription.
8. In the **Resource group** drop-down list, select **DocumentIntelligenceResources**.
9. In the **Azure AI Document Intelligence or Azure AI Service Resource** drop-down list, select **DocumentIntelligence**.
10. In the **API version** drop-down list, ensure that **2022-06-30-preview** is selected and then select **Continue**.

- ✓ Enter project details
- **Configure service resource**
- Connect training data source
- Review and create

### Configure service resource

To create a project in Form Recognizer Studio, you'll need an Azure subscription containing a service resource for usage and billing.

If you instead have an API endpoint and key, you can use a temporary session to use Form Recognizer Studio. [Learn how to retrieve them](#)

#### Access By

- ☒ Resource ☐ API endpoint and key

#### Subscription \*

Visual Studio Enterprise Subscription

#### Resource group \*

FormsRecognizerResources

[Create new](#)

#### Form Recognizer or Cognitive Service Resource \*

FormsRecognizer

☐ Create new resource

#### API version \*

Back

Continue

Cancel

11. On the **Configure training data source** page, in the **Subscription** drop-down list, select your Azure subscription.
12. In the **Resource group** drop-down list, select **DocumentIntelligenceResources**.
13. In the **Storage account** drop-down list, select the only storage account listed.
14. In the **Blob container** drop-down list, select **1040examples**, and then select **Continue**.

- ✓ Enter project details
- ✓ Configure service resource
- **Connect training data source**
- Review and create

### Connect training data source

Link the Azure Blob Storage account and the folder that contains your training data. [Learn more](#)

#### Subscription \*

Visual Studio Enterprise Subscription

#### Resource group \*

FormsRecognizerResources

#### Storage account \*

formsrecstorage

#### Blob container \*

1040examples

#### Folder path

Enter folder path

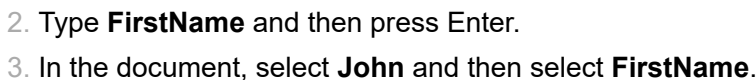
Back

Continue

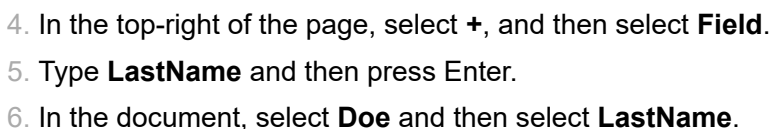
Cancel

## Label the 1040 Forms custom model

1. In the **Label data** page, in the top-right of the page, select **+**, and then select **Field**.



## Label data





7. In the top-right of the page, select **+**, and then select **Field**.
8. Type **City** and then press Enter.
9. In the document, select **Los Angeles** and then select **City**.
10. In the top-right of the page, select **+**, and then select **Field**.
11. Type **State** and then press Enter.
12. In the document, select **CA** and then select **State**.
13. Repeat the labeling process for the remaining forms in the list on the left. Label the same four fields: *FirstName*, *LastName*, *City*, and *State*.

Note: For the purposes of this exercise, we're using only five example forms and labeling only four fields. In your real-world models, you should use as many samples as possible to maximize the accuracy and confidence of your predictions. You should also label all the available fields in the forms, rather than just four fields.

## Train the 1040 Forms custom model

Now that the sample forms are labeled, we can train the first custom model:

1. In the Azure AI Document Intelligence Studio, select **Train**.
2. In the **Train a new model** dialog, in the **Model ID** textbox, type **1040FormsModel**.
3. In the **Build mode** drop-down list, select **Template**, and then select **Train**.
4. In the **Training in progress** dialog, select **Go to Models**.

## Create the 1099 Forms custom model

Now, you must create a second model, which you'll train on example 1099 tax forms:

1. In Azure AI Document Intelligence Studio, select **Custom model**.
2. Under **My Projects**, select **+ Create a project**.
3. In the **Project name** textbox, type **1099 Forms**, and then select **Continue**.
4. On the **Configure service resource** page, in the **Subscription** drop-down list, select your Azure subscription.
5. In the **Resource group** drop-down list, select **DocumentIntelligenceResources**.
6. In the **Azure AI Document Intelligence or Azure AI Service Resource** drop-down list, select **DocumentIntelligence**.
7. In the **API version** drop-down list, ensure that **2022-06-30-preview** is selected and then select **Continue**.

**Configure service resource**

To create a project in Form Recognizer Studio, you'll need an Azure subscription containing a service resource for usage and billing.

If you instead have an API endpoint and key, you can use a temporary session to use Form Recognizer Studio. [Learn how to retrieve them](#)

Access By

☒ Resource ☐ API endpoint and key

Subscription \*

Visual Studio Enterprise Subscription

Resource group \*

FormsRecognizerResources

[Create new](#)

Form Recognizer or Cognitive Service Resource \*

FormsRecognizer

☐ Create new resource

API version \*

Back Continue Cancel

8. On the **Configure training data source** page, in the **Subscription** drop-down list, select your Azure subscription.
9. In the **Resource group** drop-down list, select **DocumentIntelligenceResources**.
10. In the **Storage account** drop-down list, select the only storage account listed.
11. In the **Blob container** drop-down list, select **1099examples**, and then select **Continue**.
12. In the **Review and create** page, select **Create project**.

## Label the 1099 Forms custom model

Now, label the example forms with some fields:

1. In the **Label data** page, in the top-right of the page, select **+**, and then select **Field**.
2. Type **FirstName** and then press Enter.
3. In the document, select **John** and then select **FirstName**.
4. In the top-right of the page, select **+**, and then select **Field**.
5. Type **LastName** and then press Enter.
6. In the document, select **Doe** and then select **LastName**.
7. In the top-right of the page, select **+**, and then select **Field**.
8. Type **City** and then press Enter.
9. In the document, select **New Haven** and then select **City**.
10. In the top-right of the page, select **+**, and then select **Field**.
11. Type **State** and then press Enter.
12. In the document, select **CT** and then select **State**.
13. Repeat the labeling process for the remaining forms in the list on the left. Label the same four fields: *FirstName*, *LastName*, *City*, and *State*.

## Train the 1099 Forms custom model

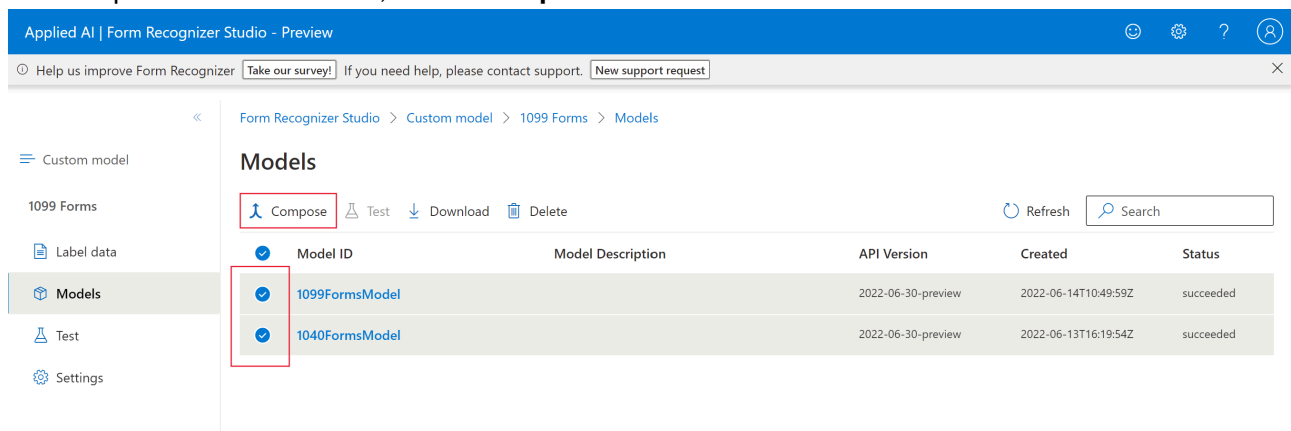
You can now train the second custom model:

1. In the Azure AI Document Intelligence Studio, select **Train**.
2. In the **Train a new model** dialog, in the **Model ID** textbox, type **1099FormsModel**.
3. In the **Build mode** drop-down list, select **Template**, and then select **Train**.
4. In the **Training in progress** dialog, select **Go to Models**.
5. The training process can take a few minutes. Refresh the browser occasionally until both models display the **succeeded** status.

## Create and assemble a composed model

The two custom models, which analyze 1040 and 1099 tax forms, are now complete. You can proceed to create the composed model:

1. In the Azure AI Document Intelligence Models page, select both **1040FormsModel** and **1099FormsModel**.
2. At the top of the list of models, select **Compose**.



3. In the **Compose a new model** dialog, in the **Model ID** textbox, type **TaxFormsModel** and then select **Compose**. Azure AI Document Intelligence creates the composed model and displays it in the list of custom models.

## Use the composed model

Now that the composed model is complete, let's test it with an example form:

1. In the [Azure portal](#), select **All resources** and then select the **formsrecstorage <xxxxxx>** storage - account, where **<xxxxxx>** is a random number
2. Under **Data storage** select **Containers** and then select **TestDoc**.
3. To the right of **f1040\_7.pdf**, select ... and then select **Download**.
4. Save the PDF document to your local computer and make a note of the saved location.
5. In the Azure AI Document Intelligence Studio, select **TaxFormsModel**, and then select **Test**.
6. Select **+ Add** and then browse to the location where you saved the PDF document.
7. Select **f1040\_7.pdf**, and then select **Open**.
8. Select **Analyze**. Azure AI Document Intelligence analyses the form by using the composed model.

## Knowledge Check

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.

---

## Summary

If you've created multiple custom models in Azure AI Document Intelligence, it can be difficult to determine the best model to use when you submit a form or document for analysis. Azure AI Document Intelligence can help with this if you create a **composed model** from two or more of the custom models. When you submit a form to a composed model, Forms Analyzer automatically classifies it and determines the best custom model to use for the analysis.

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

- Describe business problems that you would use composed models to solve.
- Create a composed model that can analyze forms in multiple formats.

[Compose custom models](#)

[Build your training dataset for a custom model](#)

---

👉 Compiled by [Kenneth Leung](#) (2025)