Azure Document Intelligence AI Action

Simple rule configuration · https://extensionkit.unitfly.com/documentation/document-ai-azure-document-intelligence-ai-action/

Azure Document Intelligence AI Action is a specific implementation of Document AI that uses Microsoft Azure's Document Intelligence service (formerly Form Recognizer) for document processing. This action type provides specialized prebuilt models optimized for common document types such as invoices, receipts, contracts, and layouts.

This page describes the following configuration options:

- Common fields for both action types (Extract and Classify)
- Options available when Action type is set to Extract
- Options available when Action type is set to Classify

Common fields

Action type

The **Action type** determines the operation that will be performed on the document using Azure Document Intelligence models. Choose from the following options:

- **Extract** (default): Choose this option to extract structured data from documents using Azure's prebuilt models.

- **Classify**: Choose this option to automatically classify documents based on their content.

Save response to file(s)

Available for both action types.

Save the AI service response as a file in M-Files.

Within the section, configure the following fields:

- **Label:** Unique set file action name. When 'Save full response to file' is disabled, this is also the AI suggestion name used to find the file content. Multiple file entries can be defined — each is processed independently. Entries with 'Save full response to file' enabled save the full AI response as JSON. Entries with it disabled match the AI suggestion (Query field) whose name equals this label and save its binary content.
- **Save full response to file:** Enable to save the complete AI service response as a JSON file. When disabled, the action searches for a named AI suggestion and saves its binary content.
- **Filename expression:** Filename expression returning filename and file extension. The file extension is required (e.g. 'response.json') — the file will not be saved if the extension is missing. When left empty and UpdateFile is set to true, the first file on the object will be updated.
- **Update file content:** Enable to update existing file content.

Each configured file entry is processed independently.

Pages to process

Available for both action types.

Tells the AI model which pages of the document to process. Instead of sending the entire document to the AI model, you can limit processing to specific pages or ranges. Use a comma-separated list of page numbers and/or page ranges (e.g., '1,3-5,7'). Leave empty to process all pages.

Examples:

- `1` — Process only the first page.
- `1,3,5` — Process pages 1, 3, and 5.
- `1-5` — Process pages 1 through 5.
- `1,3-5,7` — Process page 1, pages 3 through 5, and page 7.

If Action type is Extract

Select **Extract** as the Action type to process documents using Azure Document Intelligence prebuilt extraction models.

The available settings are:

- Extraction model
- Set properties (model-specific)
- Create object(s) (model-specific)
- Save response to file(s)
- Pages to process

The settings for updating properties and creating objects change according to the selected model.

Extraction model

Displayed when Action type is Extract.

The **Extraction model** specifies which Azure Document Intelligence model will process the document.

Default value: `prebuilt-read`

The following models are available:

Model	Description	Typical use
`prebuilt-read`	Extracts text from documents and images using OCR.	General text extraction, multi-language documents, scanned documents, and images with text
`prebuilt-layout`	Extracts text, tables, selection marks, and document structure.	Forms with complex layouts, documents with tables, structured documents requiring layout preservation, documents with checkboxes or selection marks
`prebuilt-invoice`	Extracts predefined invoice fields and related information.	Vendor invoices, purchase invoices, service invoices, commercial documents
`prebuilt-receipt`	Extracts predefined receipt fields.	Purchase receipts, sales receipts, expense receipts, point-of-sale documents
`prebuilt-contract`	Extracts predefined contract fields and related information.	Legal contracts, service agreements, business contracts, terms and conditions documents

Each model has its own configuration options, described below.

Set file(s) from prebuilt-read

This setting allows you to set files from prebuilt-read model responses.

Within the section, configure the following fields:

- **Property name:** Select the output provided by the prebuilt-read model from the available dropdown.
- **Save full response to file:** If enabled, saves the entire AI service response as a JSON file instead of extracting a specific named suggestion. If disabled, the AI suggestion whose name matches the Label is found and its binary content is saved to the file.

- **Filename expression:** Specify the filename and extension for the output file, for example `extracted-text.txt`. A file extension is required.
- **Update file content:** Enable this option to update existing file content instead of creating a separate file.

If Filename expression is empty and Update file content is enabled, the first file on the object is updated.

Set properties prebuilt-layout

Map information returned by the layout model to M-Files properties. Within the section, configure the following fields:

- **Set property from response:** Choose whether to set the property from the model response or as a static property from existing objects in the vault.
- **Model property source:** Determines which part of the model output to use.
  - `FromInputTerms`: Uses query fields defined for the information you want to find in the document.
  - `FromOutputTerms`: Uses key-value pairs identified in the model output.
- **Query field:** Specify the term or field to search for when Model property source is **FromInputTerms**.
- **Key value pair:** Select the key-value pair to use when Model property source is **FromOutputTerms**.
- **Result M-Files property type**: Specify what the result represents: simple value, object lookup or value list lookup.
- **Allow to overwrite existing value:** Defines whether rule is allowed to overwrite an existing property value on the object. If enabled, rule will update the property value even when a value already exists. If disabled, rule will not modify the property when it already contains a value.
- **Result property:** Select the M-Files property that will receive the value.

Each mapping also supports the common property mapping settings, including lookup handling and Smart lookup.

Create object(s) from prebuilt-layout

Create new M-Files objects from extracted data. For example, information extracted from a table can be used to create individual records in M-Files.

Within the section, configure the following fields:

- **Label:** A unique name identifying the object creation action in logs and the configuration UI.
- **Object cardinality:** Choose whether to create one object or multiple objects from an extracted array.
- **Object type:** The M-Files object type to create.
- **Object class:** The class assigned to the new objects. If left empty, the default class of the selected object type is used.
- **Set properties on new object:** Map model fields to the properties of a single newly created object.
- **Array query field:** When creating objects from an array, specify the field containing the array values. This is the query field name (without []) that holds the array of items in the model response. Each element of this array will produce one new M-Files object. For example, enter 'LineItems' if the model returns a 'LineItems[]' array.
- **Set file(s):** Configure files to attach to the created object.
- **Duplicates detection**: When enabled, you can configure **Duplicates detection rules** to check whether objects with matching property values already exist before creating new ones.

Set properties prebuilt-invoice

Populate M-Files properties with values extracted from an invoice. Within the section, configure the following fields:

- **Set property from response:** Choose whether to set the property from the model response or as a static property from existing objects in the vault.
- **Property name:** Select the field to extract from the list of supported prebuilt-invoice fields (e.g. CustomerName, InvoiceId, DueDate, etc.).
- **Allow to overwrite existing value:** Defines whether rule is allowed to overwrite an existing property value on the object. If enabled, rule will update the property value even when a value already exists. If disabled, rule will not modify the property when it already contains a value.
- **Result property:** Select the M-Files property where the extracted value will be stored.

Additional property mapping settings allow you to control lookup resolution, formatting, and how existing property values are handled.

Create object(s) from prebuilt-invoice

Create related M-Files objects from extracted invoice data. For example, each invoice line item can be created as a separate object containing its description, quantity, price, and amount.

In addition to the common object creation settings (listed in the Create object(s) from prebuilt-layout section), the invoice model supports the following **Array property key** options:

- **Items**: Create objects from extracted invoice line items.
- **PaymentDetails**: Create objects from payment details returned by the model.
- **TaxDetails**: Create objects from extracted tax details.
- **PaidInFourInstallements**: Create objects from installment payment information returned by the model.

Depending on the selected array, configure the corresponding property mappings:

- **Set items properties**: Set properties and files on newly created items object(s).
- **Set payment details properties**: Set properties and files on newly created payment details object(s).

- **Set tax details properties**: Set properties and files on newly created tax details object(s).
- **Set paid installements properties**: Set properties and files on newly created paid in four installments object(s).

Each property mapping uses a **Property name** selected from the corresponding array's available fields and a target M-Files property.

The resulting objects can store individual details from the invoice without requiring all information to be combined into a single metadata field.

Set properties prebuilt-receipt

Map extracted receipt information to M-Files properties. Within the section, configure the following fields:

- **Set property from response:** Choose whether to set the property from the model response or as a static property from existing objects in the vault.
- **Property name:** Select the field to extract from the list of supported prebuilt-receipt fields (e.g. MerchantName, Total.Amount, ReceiptType, etc.).
- **Result M-Files property type**: Specify what the result represents: simple value, object lookup or value list lookup.
- **Allow to overwrite existing value:** Defines whether rule is allowed to overwrite an existing property value on the object. If enabled, rule will update the property value even when a value already exists. If disabled, rule will not modify the property when it already contains a value.
- **Result property:** Select the M-Files property where the extracted value will be stored.

The common property mapping settings are also available for configuring lookups and handling extracted values.

Create object(s) from prebuilt-receipt

Create M-Files objects from extracted receipt information. Within the section, configure the following fields:

- **Label:** A unique name identifying the object creation action in logs and the configuration UI.
- **Object cardinality:** Choose whether to create one object or multiple objects from an extracted array.
- **Object type:** The M-Files object type to create.
- **Object class:** The class assigned to the new objects. If left empty, the default class of the selected object type is used.
- **Set properties on new object:** Map model fields to the properties of a single newly created object.
- **Array query field:** When creating objects from an array, specify the field containing the array values. This is the query field name (without []) that holds the array of items in the model response. Each element of this array will produce one new M-Files object. For example, enter 'LineItems' if the model returns a 'LineItems[]' array.
- **Set file(s):** Configure files to attach to the created object.
- **Duplicates detection**: When enabled, you can configure **Duplicates detection rules** to check whether objects with matching property values already exist before creating new ones.

Set properties prebuilt-contract

Map extracted contract fields to M-Files properties. Within the section, configure the following fields:

- **Set property from response:** Choose whether to set the property from the model response or as a static property from existing objects in the vault.
- **Property name:** Select the field to extract from the list of supported prebuilt-contract fields (e.g. ContractId, Parties.Name, EffectiveDate, etc.).
- **Result M-Files property type**: Specify what the result represents: simple value, object lookup or value list lookup.

- **Allow to overwrite existing value:** Defines whether rule is allowed to overwrite an existing property value on the object. If enabled, rule will update the property value even when a value already exists. If disabled, rule will not modify the property when it already contains a value.
- **Result property:** Select the M-Files property where the extracted value will be stored.

The common property mapping settings are also available.

Create object(s) from prebuilt-contract

Create M-Files objects from extracted contract information. For example, the model can return information about several parties in a contract, which can then be mapped to separate M-Files objects.

In addition to the common object creation settings (listed in the Create object(s) from prebuilt-layout section), the contract model supports two **Set property array key** options:

- **Parties:** Create objects from information about the parties identified in the contract.
- **Jurisdictions:** Create objects from jurisdiction information returned by the model.

Depending on the selected array, configure:

- **Set array properties (Parties):** Map fields from each extracted party to the properties of a newly created object.
- **Set array properties (Jurisdictions):** Map fields from each extracted jurisdiction to the properties of a newly created object.

Select the appropriate Property name from the available array fields and specify the corresponding M-Files property.

If Action type is Classify

Select **Classify** as the Action type to classify documents using a trained Azure Document Intelligence custom classifier.

Unlike the prebuilt extraction models, a custom classifier must be trained on the document types you want to recognize.

Classification results can be used to set an M-Files class or populate another M-Files property.

The following settings are available:

- Classifier model
- Confidence threshold
- Map mode
- Document type mappings or property mapping settings, depending on the selected Map mode
- Save response to file(s)
- Pages to process

Classifier model

Displayed when Action type is Classify.

Select the trained classifier model to use for document classification.

The available options are populated from custom classifiers configured in your Azure Document Intelligence resource.

The classifier analyzes document content and returns a document type label with a confidence score.

Note: A custom classifier must be trained and available in the connected Azure Document Intelligence resource before it can be selected.

Confidence threshold

Displayed when Action type is Classify.

Determines the minimum confidence score required to apply the classification result.

Default value: `0.8`

The threshold uses values between `0.0` and `1.0`.

For example, with the default threshold of `0.8`, a classification result with a confidence score below 0.8 will not be applied.

Increase the threshold to require higher confidence before updating the document classification.

Map mode

Determines where the classification result is stored in M-Files. Choose one of the following options:

- **Class (default):** Sets the M-Files class based on the document type returned by the classifier. When selected, the **Document type mappings** section becomes available.
- **Property:** Writes the classification result to a selected M-Files property. When selected, the **Target property** settings become available.

Document type mappings

Displayed when Map mode is set to Class.

Maps classifier labels to M-Files classes. Within the section, configure the following fields:

- **Document type:** Select the label returned by the trained classifier.

- **M-Files class:** Select the corresponding M-Files class.

For example, a classifier label representing an invoice can be mapped to the Invoice class in M-Files.

If the classifier returns a document type without an explicit mapping, the system attempts to resolve the M-Files class by name.

Target property

Displayed when Map mode is set to Property.

Select the M-Files property where the result will be stored.

Target property is value list based

Displayed when Map mode is set to Property.

Enable this when the target property is a value list lookup (additional **Value list item mappings** become available). This reveals a value list picker and label-to-item mappings. Leave off for text or multiline text target properties.

Value list item mappings

*Displayed when **Target property is value list based** is enabled.*

Maps classifier labels to value list items. Within the section, configure the following fields:

- **Document type:** Select the document type label returned by the classifier.
- **Value:** Select the corresponding value list item.

If the classifier returns a label without an explicit mapping, the system attempts to resolve the value list item by name.