USER GUIDE

Artificial Intelligence

2025.2 - June 2025

Table of Contents

Artificial Intelligence

Artificial Intelligence (AI) offers a wide range of benefits across various industries and enhances efficiency by automating routine tasks, allowing businesses to streamline operations, aids in creating content, etc. Overall, AI aids companies by making processes faster, smarter, and more efficient.

STEP enables customers to integrate and benefit from AI through Azure OpenAI or Azure Vision. Refer to the following topics on how to integrate with AI:

- Azure OpenAI Integration

- Azure Vision Integration

Also, in the Matching, Linking, and Merging documentation:

- Machine Learning Matcher employs a pretrained machine learning model to match individual party data elements. Information on this solution can be found in the Matcher: Machine Learning Matcher section.

- Machine Learning Match Recommendations uses a matching agent that works alongside the daily activities of the data steward by observing their duplicate review decisions during clerical review sessions and continuously refines its learning. It eases the workload by providing recommendations for merging or rejecting based on the data steward's previous decisions. Information on this solution can be found in the Machine Learning Match Recommendations section.

Customers are encouraged to talk to their Customer Success Manager or Account Managers about other ways they can incorporate AI capabilities into their solution.

Azure OpenAI Integration

Azure OpenAI enables organizations to stay current with AI technology, and is intended to be used as a catalyst to help customers curate the Azure OpenAI integration to their specific needs. A start-up package has been created that provides examples of how this integration could be used to optimize business processes, such as the creation of generative AI-powered product descriptions, or the ability to utilize automated translations. This integration enables organizations to work more efficiently, broaden their delivery base, and improve market responsiveness.

> **Important:** The use of Azure OpenAI integration requires customers to independently obtain an API key through Azure.

Getting Started with OpenAI

Your Azure administrator will need to deploy an Azure OpenAI service and provide you with the endpoint URL and the API key associated with the deployed resource. The URL will be used in the REST Gateway Integration Endpoint, and your API key will be used in the Gateway Integration Authentication function. The format is as follows:

```
https://{your-resource-name}.openai.azure.com/openai/deployments/{deployment-id}/chat/completions
```

Additionally, you will need an API version of the Chat Completions Model for the Gateway Integration Authentication function. Search the web for an Azure Chat Completions API version. It is recommended not to use preview versions. For example the API version could be the following:

```
2024-02-01
```

After you have acquired an API key through Azure, and you have installed the necessary license and components on your system, the AI Setup group displays in the System Setup tab.
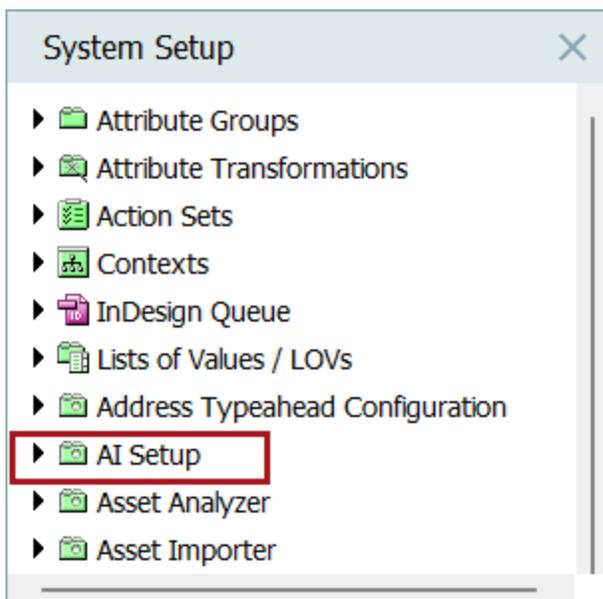
Follow the steps below to get familiar with the starter package provided for OpenAI:

1. Expand the AI Setup folder to display the folders below.

2. Expand the Text Generation folder and its subsequent folders. Each of these subfolders together contain, business rules, a business functions, a Generation Gateway Endpoint, and an Event Processor.
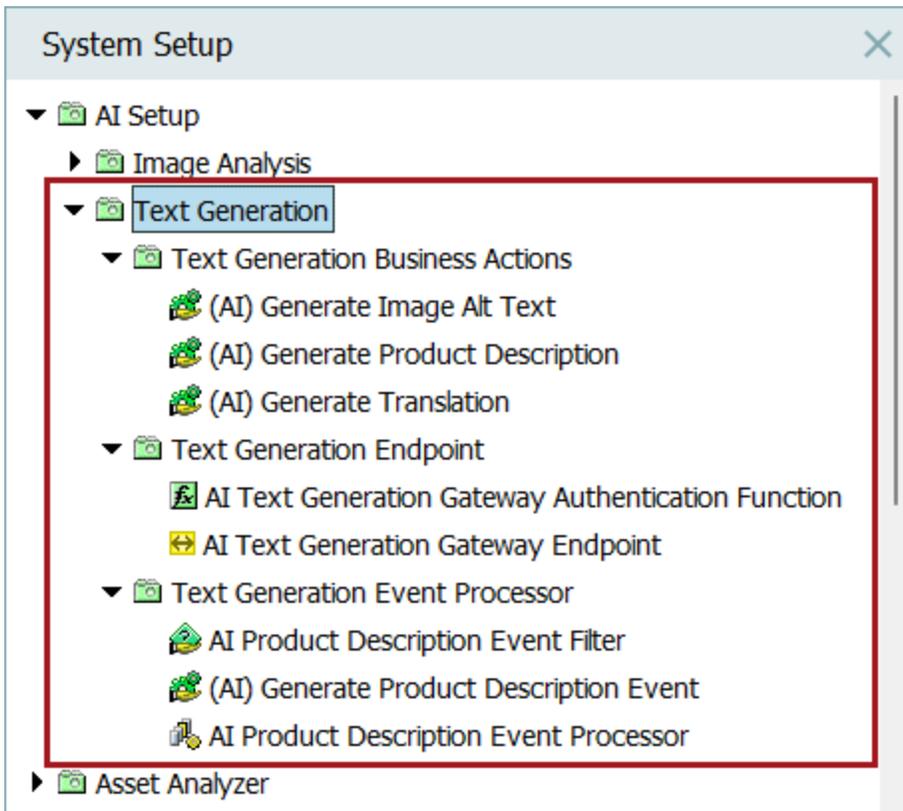
3. Starting with the Text Generation Endpoint folder, configure a REST **AI Text Generation Gateway Endpoint** as needed. For more information on how to configure the AI Text Generation Gateway Endpoint, refer to the Configuring a Gateway Integration Endpoint topic in the Data Exchange documentation.

   For this example, the following REST gateway integration endpoint was configured:

AI Text Generation Gateway Endpoint
Gateway Integration Endpoint Type

Gateway Integration Endpoint   **Configuration**   Statistics   Error Log   Status   Log

⌄ Gateway Configuration

	Gateway Plugin Type: REST	
⋮	Server URL	https://sys-rd-pmdm-azure-ai-sweden.openai.azure.c...
⋮	Default content type	
⋮	Statistic groups	[]
⋮	SSL trust store location	
⋮	MTLS Authentication KeyStore	
⋮	Proxy Configuration	
⋮	Username	
⋮	Password	
⋮	Use preemptive authentication	false
⋮	Auth Header value Function	AI Text Generation Gateway Authentication Function

Edit

⌄ Gateway Connectivity
Last successful connectivity check:

[Check Connectivity]

4. Configure the **AI Text Generation Gateway Authentication Function**.

   - Update the secret bind, in this example the variable is called 'apiKey,' with your Azure OpenAI API key

   - Ensure that the return type is set to `Map<String, String>`

> **Important:** You need to put **Parameter:** in front of the key *API-version*.

In the example below, the following JavaScript Function was created:

Important: Should additional Azure OpenAI regions need to be configured, each one requires its own gateway integration endpoint and corresponding authentication header value function. All endpoints that use the same 'your-resource-name' at the start of the URL can use the same authentication function.

Business Rule Examples

Users can create business rules to suit their needs when working with OpenAI. For more information on how to create business rules, refer to the Business Rules topic in the Business Rules documentation.

Three example business rules have been provided to create starting points. Refer to the online help version of this topic for the complete JavaScript example code. The binds will have to be modified, and attributes will need to be selected from your data model.

- **Generate Image Alt Text** - This example will analyze images and create alternative text suitable for a shopping website.

> **Note:** When selecting the attribute for the Attribute bind intended to hold alternate text, in this example it is called 'aiAltText,' ensure that the attribute has a large enough character limit to accommodate the expected result from Azure OpenAI.

- **Generate Product Descriptions** - This example creates product descriptions for an online clothing store.

- **Generate Translation** - This example translates English to German.

Event Processor Example

Event Processors allow for the collection of events to be logically processed, and actions automatically performed, based upon each event processor's specific configuration. For example, organizations can use event processors to process events via business rules asynchronous in the background, if desired, when working with Azure OpenAI. For more information on how to configure and use event processors, refer to the Event Processors topic in the System Setup documentation. An example event processor has been provided to create a starting point.

- This example business rule ensures that the event gets picked up by the Azure OpenAI event processor. Refer to the online help version of this topic for the complete JavaScript example code.

Edit Operation ✕

Execute JavaScript ▾

Binds:

∨ **Binds**

Variable name ⋮	Binds to ⋮	Parameter ⋮
eventQueue	Event Queue	AI Product Description Event Processor (AI.TextGeneration.ProdDescEventProcessor)
derivedEvent	Derived Event Type	AI.TextGeneration.GenerateProdDesc (AI.TextGeneration.GenerateProdDesc)
currentObject	Current Object	

Messages:

∨ **Messages**

Variable name ⋮	Message ⋮	Translations ⋮

JavaScript:

```
1  eventQueue.queueDerivedEvent(derivedEvent, currentObject);
```

[Save] [Test JavaScript] [Cancel]

Azure Vision Integration

Azure Vision integration is used to extract detailed information from images, including keywords, captions, and text identification. Organizations can utilize this information in order to optimize search engine results, identify similar products based on linked images, enrich product descriptions and attributions based on image content, and provide richer product information to customers.

> **Important:** The use of Azure Vision integration requires customers to independently obtain an API key through Azure.

Getting Started with Azure Vision

Your Azure administrator will need to deploy a Computer Vision instance and provide you with the endpoint URL and the API key associated with the URL. The URL will be used in the REST Gateway Integration Endpoint, and the API key will be used in the Gateway Integration Authentication function.

Additionally, you will need an API version of the Computer Vision instance for the Gateway Integration Authentication function. Search the web for an Azure Computer Vision API version. It is recommended not to use preview versions. For example the API version could be the following

```
2024-02-01
```

After you have acquired an API key through Azure, and you have installed the necessary license and components on your STEP system, the AI Setup group displays in the System Setup tab.
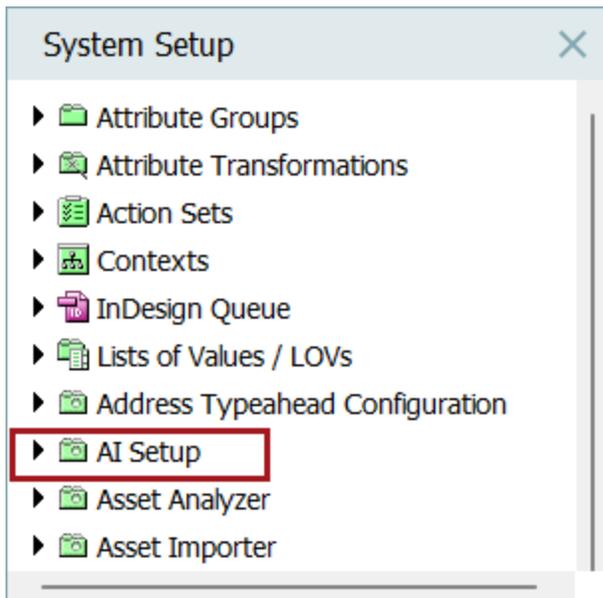
Follow the steps below to get familiar with the starter package provided for Azure Vision:

1. Expand the AI Setup folder to display the folders below.

2. Expand the Image Analysis folder and its subsequent folders. Each of these subfolders together contain, business rules, a Generation Gateway Endpoint, an authentication function for the generation gateway endpoint, and an Event Processor.
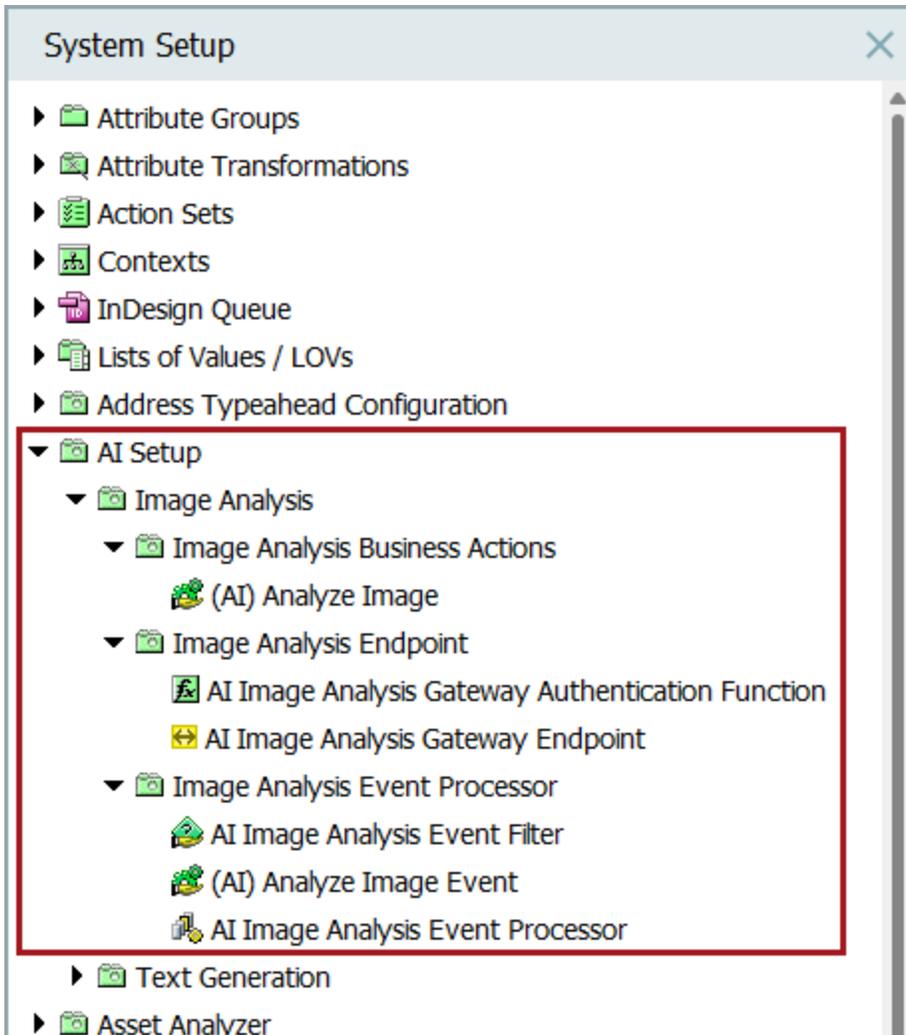
3. Starting with the Image Analysis Endpoint folder, configure a REST **AI Image Analysis Gateway Endpoint** as needed. For more information on how to configure Generation Gateway Endpoints, refer to the Configuring a Gateway Integration Endpoint topic in the Data Exchange documentation.

   For this example, the following **REST Generation Gateway Endpoint** was configured:

AI Image Analysis Gateway Endpoint
Gateway Integration Endpoint Type

Gateway Integration Endpoint   Configuration   Statistics   Error Log   Status   Log

⌄ Gateway Configuration

	Gateway Plugin Type: REST	
⋮	Server URL	https://sys-rd-pmdm-computer-vision.cognitiveservices.azure.com/
⋮	Default content type	
⋮	Statistic groups	[]
⋮	SSL trust store location	
⋮	MTLS Authentication KeyStore	
⋮	Proxy Configuration	
⋮	Username	
⋮	Password	
⋮	Use preemptive authentication	false
⋮	Auth Header value Function	AI Image Analysis Gateway Authentication Function

Edit

⌄ Gateway Connectivity
Last successful connectivity check:

[Check Connectivity]

Important: Should additional Azure Vision regions need to be configured, each one requires its own Generation Gateway Integration Endpoint and corresponding Generation Gateway Authentication Function.

4. Configure the **AI Image Analysis Gateway Authentication Function**.

   - Update the secret bind, in this example the variable is called 'apiKey,' with your Azure Vision API key

   - Ensure that the return type is set to `Map<String, String>`

     Important: You need to put **Parameter:** in front of the key *API-version*.

Refer to the online help version of this topic for the complete JavaScript example code. In the example below, the following JavaScript Function was created:

5. To make this example work, lastly, configure the **Image Analysis Business Action**. A bind must be made to the relevant gateway integration endpoint and attributes to produce the relevant tags, caption, description, and detected text. Refer to the online help version of this topic for the complete JavaScript example code.

The produced values will be stored in their matching attributes. For example, detected text will be stored within the Attribute bind that has the 'detectedTextAttribute' variable name.

Event Processor

Event Processors allow for the collection of events to be logically processed, and actions automatically performed, based upon each event processor's specific configuration. For example, organizations can use event processors to process events via business rules asynchronous in the background, if desired, when working with Azure Vision. For more information on how to configure and use event processors, refer to the Event Processors topic in the System Setup documentation. An example event processor has been provided to create a starting point.

- This example business rule ensures that the event gets picked up by the Azure Vision event processor. Refer to the online help version of this topic for the complete JavaScript example code.

Edit Operation ☐   ✕

Execute JavaScript ▼

Binds:

∨ **Binds**

Variable name	Binds to	Parameter
eventQueue	Event Queue	AI Image Analysis Event Processor (AI.ImageAnalysis.EventProcessor)
derivedEvent	Derived Event Type	AI.ImageAnalysis.AnalyzeImage (AI.ImageAnalysis.AnalyzeImage)
currentObject	Current Object	

Messages:

∨ **Messages**

Variable name	Message	Translations

JavaScript:

```
1  eventQueue.queueDerivedEvent(derivedEvent, currentObject);
```

Save   Test JavaScript   Cancel