



CDW Documentation

Azure Speech & Translator with Key Vault Integration

Azure Speech & Translator with Key Vault Integration

[Overview](#)

This setup securely connects to Azure **Speech-to-Text** and **Translator** services using secrets stored in **Azure Key Vault**, accessed via ``DefaultAzureCredential``.

—

[Azure Key Vault Setup](#)

1. **Create a Key Vault** in Azure Portal. 2. Add the following **secrets**:

Secret Name	Example Value
_____	_____
<code>`speech-key`</code>	(Your Azure Speech API key)
<code>`speech-region`</code>	<code>`eastus`</code>
<code>`translator-key`</code>	(Your Translator API key)
<code>`translator-region`</code>	<code>`global`</code>

3. Assign the executing identity (e.g. user or Managed Identity):

- Role: **Key Vault Secrets User**
- Scope: Your Key Vault resource

—

[Local Setup \(Developer Machine\)](#)

1. Install tools:

```
```bash brew install azure-cli pip install azure-identity azure-keyvault-secrets azure-cognitiveservices-speech requests ```
```

### 2. Authenticate:

```
```bash az login ```
```

This enables ``DefaultAzureCredential`` to work locally.

—

[Full Python Script Using Azure Key Vault](#)

```
```python
import azure.cognitiveservices.speech as speechsdk
import requests
from azure.identity import DefaultAzureCredential
from azure.keyvault.secrets import SecretClient

Load secrets from Azure Key Vault
VAULT_NAME = "your-keyvault-name" # Replace with your Key Vault name
KV_URI = f"https://{VAULT_NAME}.vault.azure.net"
```

```
credential = DefaultAzureCredential()
secret_client = SecretClient(vault_url=KV_URI, credential=credential)

Fetch secrets
speech_key = secret_client.get_secret("speech-key").value
speech_region = secret_client.get_secret("speech-region").value
translator_key = secret_client.get_secret("translator-key").value
translator_region = secret_client.get_secret("translator-region").value

□ Language settings
SPEECH_LANGUAGE = "en-US"
TARGET_LANGUAGE = "ta" # Tamil

□ Speech Recognition
speech_config = speechsdk.SpeechConfig(subscription=speech_key,
region=speech_region)
speech_config.speech_recognition_language = SPEECH_LANGUAGE
recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config)

print("□ Say something...")
result = recognizer.recognize_once()

if result.reason == speechsdk.ResultReason.RecognizedSpeech:
 text = result.text
 print("□ Recognized:", text)

 # □ Translate
 endpoint =
f"https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to
={TARGET_LANGUAGE}"
 headers = {
 "Ocp-Apim-Subscription-Key": translator_key,
 "Ocp-Apim-Subscription-Region": translator_region,
 "Content-Type": "application/json"
 }
 body = [{"text": text}]
 response = requests.post(endpoint, headers=headers, json=body)

 if response.status_code == 200:
 translated = response.json()[0]["translations"][0]["text"]
 print(f"□ Translated ({TARGET_LANGUAGE}): {translated}")
 else:
 print("□ Translation failed:", response.status_code)
 print("□", response.text)

elif result.reason == speechsdk.ResultReason.NoMatch:
 print("△□ Speech not recognized.")
elif result.reason == speechsdk.ResultReason.Canceled:
 print("□ Error:", result.cancellation_details.reason)
...

```