**Translate speech-to-speech by using the Azure AI Speech service**

Azure AI Speech includes a speech translation API that you can use to translate spoken language. For example, suppose you want to develop a translator application that people can use when traveling in places where they don’t speak the local language. They would be able to say phrases such as “Where is the station?” or “I need to find a pharmacy” in their own language, and have it translate them to the local language.

<https://github.com/Azure-Samples/cognitive-services-speech-sdk/blob/master/samples/csharp/sharedcontent/console/translation_samples.cs#L472>

**Create a speech translation configuration**

To call the Speech service by using the Speech SDK, you need to create a [SpeechTranslationConfig](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.speechtranslationconfig) instance. This class includes information about your subscription, like your key and associated region, endpoint, host, or authorization token.

A screenshot of a computer program

Description automatically generated

**Change the source language**

One common task of speech translation is specifying the input (or source) language. The following example shows how you would change the input language to Italian. In your code, interact with the SpeechTranslationConfig instance by assigning it to the [SpeechRecognitionLanguage](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.speechconfig.speechrecognitionlanguage) property:

A screenshot of a computer program

Description automatically generated

**Add a translation language**

Another common task of speech translation is to specify target translation languages. At least one is required, but multiples are supported. The following code snippet sets both French and German as translation language targets:

A screen shot of a computer program

Description automatically generated

With every call to [AddTargetLanguage](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.speechtranslationconfig.addtargetlanguage), a new target translation language is specified. In other words, when speech is recognized from the source language, each target translation is available as part of the resulting translation operation.

**Initialize a translation recognizer**

After you created a [SpeechTranslationConfig](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.speechtranslationconfig) instance, the next step is to initialize [TranslationRecognizer](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.translation.translationrecognizer). When you initialize **TranslationRecognizer**, you need to pass it your **speechTranslationConfig** instance. The configuration object provides the credentials that the Speech service requires to validate your request.

If you're recognizing speech by using your device's default microphone, here's what the TranslationRecognizer instance should look like:



If you want to specify the audio input device, then you need to create an [AudioConfig](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.audio.audioconfig) class instance and provide the **audioConfig** parameter when initializing **TranslationRecognizer**.

First, reference the AudioConfig object as follows:

A screenshot of a computer program

Description automatically generated

If you want to provide an audio file instead of using a microphone, you still need to provide an audioConfig parameter. However, when you create an AudioConfig class instance, instead of calling FromDefaultMicrophoneInput, you call FromWavFileInput and pass the filename parameter:

A screenshot of a computer program

Description automatically generated

**Translate speech**

To translate speech, the Speech SDK relies on a microphone or an audio file input. Speech recognition occurs before speech translation. After all objects are initialized, call the recognize-once function and get the result:

A screenshot of a computer program

Description automatically generated

**Event based translation**

The TranslationRecognizer object exposes a Recognizing event. The event fires several times and provides a mechanism to retrieve the intermediate translation results.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Synthesize translations**

After a successful speech recognition and translation, the result contains all the translations in a dictionary. The [Translations](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.translation.translationrecognitionresult.translations) dictionary key is the target translation language, and the value is the translated text. Recognized speech can be translated and then synthesized in a different language (speech-to-speech).

**Event-based synthesis**

The TranslationRecognizer object exposes a Synthesizing event. The event fires several times and provides a mechanism to retrieve the synthesized audio from the translation recognition result. If you're translating to multiple languages, see [Manual synthesis](https://learn.microsoft.com/en-us/azure/ai-services/speech-service/how-to-translate-speech?tabs=terminal&pivots=programming-language-csharp#manual-synthesis).

Specify the synthesis voice by assigning a [VoiceName](https://learn.microsoft.com/en-us/dotnet/api/microsoft.cognitiveservices.speech.speechtranslationconfig.voicename) instance, and provide an event handler for the Synthesizing event to get the audio. The following example saves the translated audio as a .wav file.

**Important**

The event-based synthesis works only with a single translation. *Do not* add multiple target translation languages. Additionally, the VoiceName value should be the same language as the target translation language. For example, "de" could map to "de-DE-Hedda".

A screenshot of a computer

Description automatically generated

**Manual synthesis**

You can use the Translations dictionary to synthesize audio from the translation text. Iterate through each translation and synthesize it. When you're creating a **SpeechSynthesizer** instance, the **SpeechConfig** object needs to have its SpeechSynthesisVoiceName property set to the desired voice.

The following example translates to five languages. Each translation is then synthesized to an audio file in the corresponding neural language.

A screenshot of a computer

Description automatically generated

