AVFAudio / ··· / Audio Units / Creating a custom speech synthesizer

Sample Code

Creating a custom speech synthesizer

Use your custom voices to synthesize speech by building a speech synthesis provider.

Download

iOS 17.5+ | iPadOS 17.5+ | macOS 14.5+ | Xcode 15.4+

Overview

A speech synthesis provider allows you to bring your custom voices to iOS and macOS for system use with text-to-speech features like VoiceOver. A speech synthesizer receives text and information about speech properties, and provides an audio representation of the speech. To generate audio, you create an audio unit extension.

The sample app shows you how to provide a list of voices to the system, and how to create a basic speech synthesizer to represent the voices you specify. It explores how to create an audio unit that's responsible for handling text-to-speech requests to synthesize speech by using Speech Synthesis Markup Language (SSML). The sample inspects a request's SSML for two strings — *hello* and *goodbye* — and plays the associated audio file.

Configure the sample code project

To run this sample app, you'll need the following:

- A Mac with macOS 13 or later, or an iPhone with iOS 16 or later
- Xcode 14.1 beta or later

To create the audio files that the extension uses, perform the following steps:

1. Open Terminal.

2. Run the command `say -v Zarvox hello -o hello.aiff`.

3. Run the command `say -v Zarvox goodbye -o goodbye.aiff`.

4. In the project navigator, expand the `CustomSpeechSynthesizerExampleExtension` group.

5. Drag and drop the audio files you generate into the Audio group.

After you run the project, provide a name for your voice and choose Add Voice. The system takes up to 30 seconds to refresh the list of available voices. Verify that the voice is available in macOS by opening System Settings and choosing Accessibility > Spoken Content. Select the system voice drop-down menu to view the voices you add.

In iOS, open the Settings app and choose Accessibility > Spoken Content > Voices.

Create an app to provide voices

The sample creates a host app to customize the list of voices available to the speech synthesis extension, using an App Group to share information between the host app and the audio unit extension. For more information about App Groups, see `Configuring App Groups`.

```
let groupDefaults = UserDefaults(suiteName: "group.com.example.apple.samplecode.Cust
```

When the sample changes the list of available voices, it informs the system that they're available for use by calling `updateSpeechVoices()`.

```
private func saveVoicesToGroupDefaults() {

    // Update the list of voices in the shared group defaults.
    groupDefaults?.set(voices, forKey: "voices")

    // Inform the system that the available voices changed.
    AVSpeechSynthesisProviderVoice.updateSpeechVoices()

}
```

Get the list of available voices

The extension is an `AVSpeechSynthesisProviderAudioUnit` that's responsbile for handling speech synthesis tasks. The audio unit provides a list of available voices to the system, inspects a request's SSML, and provides audio buffers to the system. To provide a list of voices, the sample retrieves the list of voices the host app provides and initializes an `AVSpeechSynthesisProviderVoice` for each one.

```
public override var speechVoices: [AVSpeechSynthesisProviderVoice] {
    get {
        let voices: [String] = (groupDefaults?.value(forKey: "voices") as? [String]
        return voices.map { voice in
            return AVSpeechSynthesisProviderVoice(name: voice,
                                                  identifier: "com.identifier.\(voic
                                                  primaryLanguages: ["en-US"],
                                                  supportedLanguages: ["en-US"])
        }
    }
    set { }
}
```

Handle the speech request

When there’s text available to synthesize, the system calls `synthesizeSpeechRequest(_ speechRequest: AVSpeechSynthesisProviderRequest)` with an `AVSpeechSynthesisProviderRequest`. The request contains the SSML representation that describes the text to synthesize and the corresponding attributes for customizing pitch, rate, intonation, and more.

```
public override func synthesizeSpeechRequest(_ speechRequest: AVSpeechSynthesisProvi
    request = speechRequest
    currentBuffer = getAudioBufferForSSML(speechRequest.ssmlRepresentation)
    framePosition = 0
}
```

The sample calls a helper method with the SSML to retrieve a buffer that contains the audio to play back in a render block. The method `cancelSpeechRequest()` discards the current speech request.

Provide audio buffers to the system

The system calls an audio unit’s rendering block to get a list of audio buffers for playback. The block receives the number of frames the system requests. The sample copies the current buffer’s frames into the target buffer for rendering. When the sample exhausts the available audio buffer, it sets the render action to `offlineUnitRenderAction_Complete`.

```
// Iterate through the requested number of frames.
for frame in 0..<frameCount {
    // Copy the source frames into the target buffer.
    frames[Int(frame)] = sourceFrames[Int(self.framePosition)]
    self.framePosition += 1
    // Complete the request if the frame position exceeds the available buffer.
    if self.framePosition >= self.currentBuffer!.frameLength {
        actionFlags.pointee = .offlineUnitRenderAction_Complete
        break
    }
}
```

See Also

Speech synthesis

`class AVSpeechSynthesisProviderAudioUnit`

An object that generates speech from text.