

Multimedia: Extras

Mobile Application Development in iOS

School of EECS

Washington State University

Instructor: Larry Holder

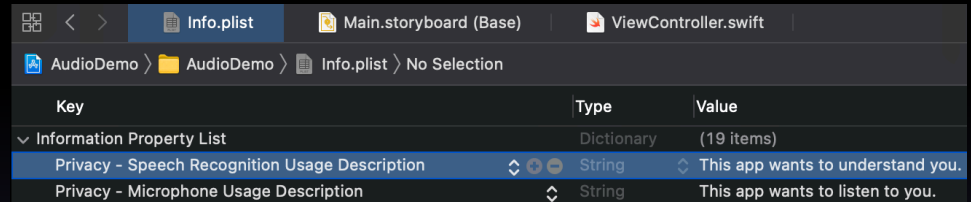
Outline

- Speech recognition and synthesis
- PhotoKit



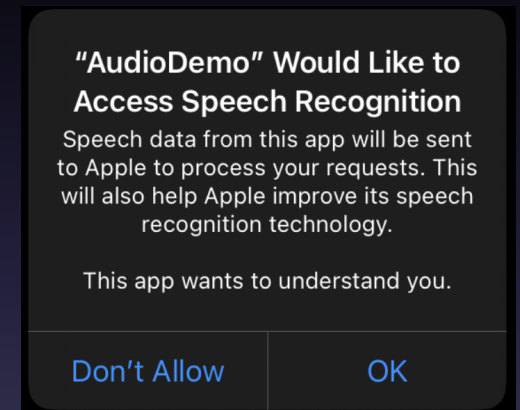
Speech Recognition

- Speech framework
- SFSpeechRecognizer
 - requestAuthorization()
 - authorizationStatus()
- SFSpeechRecognizerDelegate
 - speechRecognizer(availabilityDidChange)



The screenshot shows the Xcode interface with the Info.plist file open. The 'Privacy - Speech Recognition Usage Description' key is selected, showing its value as 'This app wants to understand you.' The table below represents the data shown in the screenshot.

Key	Type	Value
Information Property List	Dictionary	(19 items)
Privacy - Speech Recognition Usage Description	String	This app wants to understand you.
Privacy - Microphone Usage Description	String	This app wants to listen to you.



Speech Recognition: Setup

```
import Speech

class ViewController: UIViewController, SFSpeechRecognizerDelegate {

    var speechRecognizer = SFSpeechRecognizer()

    func authorizeSpeech() {
        speechRecognizer?.delegate = self
        SFSpeechRecognizer.requestAuthorization{ authStatus in
            if authStatus == .authorized {
                print("Speech recognition authorized")
            } else {
                print("Speech recognition not authorized")
            }
        }
    }
}
```

Speech Recognition: Delegate

```
// Delegate method

func speechRecognizer(_ speechRecognizer: SFSpeechRecognizer,
    availabilityDidChange available: Bool) {
    if available {
        print("Speech recognition available")
    } else {
        print("Speech recognition not available")
    }
}
```

Speech Recognition

- Create request
 - `SFSpeechURLRecognitionRequest(audioURL)`
- Execute recognition task
 - `SFSpeechRecognizer.recognitionTask(with: request, resultHandler: speechRecognitionHandler)`
- Handler receives `SFSpeechRecognitionResult`
 - `result.bestTranscription.formattedString`

Speech Recognition

```
func recognizeSpeech () {
    let status = SFSpeechRecognizer.authorizationStatus()
    if (status == .authorized) && speechRecognizer!.isAvailable {
        let request = SFSpeechURLRecognitionRequest(url: audioFileURL)
        request.shouldReportPartialResults = false
        speechRecognizer?.recognitionTask(with: request,
                                           resultHandler: speechRecognitionHandler)
    }
}

func speechRecognitionHandler (result: SFSpeechRecognitionResult?,
                              error: Error?) {
    if let res = result {
        self.textView.text = res.bestTranscription.formattedString
    } else {
        print("speech recognition error")
    }
}
```

Speech Synthesis

- Create `AVSpeechSynthesizer()`
- Create `AVSpeechUtterance(stringToBeSpoken)`
 - Optionally set `utterance.voice`
- Execute `AVSpeechSynthesizer.speak(utterance)`
- Voices (`AVSpeechSynthesisVoice`)
- `AVSpeechSynthesisVoice.speechVoices()`
 - Name: "Maged", "Zuzana", "Sara", "Anna", ...
 - Identifier: "com.apple.ttsbundle.Maged-compact"

Speech Synthesis

```
import AVFoundation

var speechSynthesizer = AVSpeechSynthesizer()

func synthesizeSpeech() {
    let utterance = AVSpeechUtterance(string: self.spokenText)
    // utterance.voice = AVSpeechSynthesisVoice(identifier: "...")
    speechSynthesizer.speak(utterance)
}
```

PhotoKit (iOS 14+)

- PhotoKit
- **PHPickerViewController** (new in iOS 14)
 - Replaced **UIImagePickerController**
 - Select a picture or video from library
 - Returns **PHAsset** (image, live photo, video)



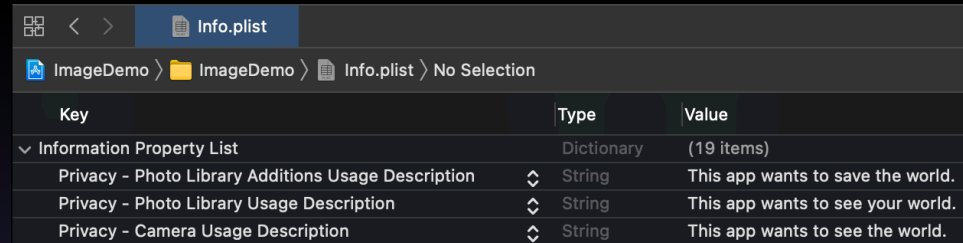
Privacy and Authorization

- Add privacy properties for access to camera and photo library

- Limited library ([Select Photos](#))

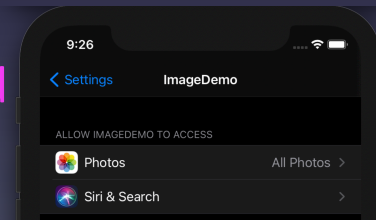
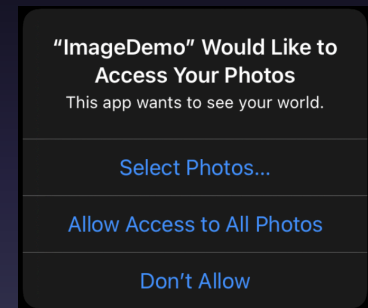
- Check/request authorization

- `PHPhotoLibrary.authorizationStatus`(for `accessLevel: PHAccessLevel`) -> `PHAuthorizationStatus`
- `PHPhotoLibrary.requestAuthorization`(for `accessLevel: PHAccessLevel`, `handler: (PHAuthorizationStatus) -> Void`)
- `.notDetermined`, `.restricted`, `.denied`, `.authorized`, `.limited`



The screenshot shows the Xcode interface with the Info.plist file open. The table below represents the content of the 'Information Property List' section.

Key	Type	Value
Information Property List	Dictionary	(19 items)
Privacy - Photo Library Additions Usage Description	String	This app wants to save the world.
Privacy - Photo Library Usage Description	String	This app wants to see your world.
Privacy - Camera Usage Description	String	This app wants to see the world.



Privacy and Authorization

```
import Photos

func checkAuthorization () {
    let authStatus = PHPhotoLibrary.authorizationStatus(for: .readWrite)
    if authStatus == .authorized {
        print("access allowed")
    } else {
        PHPhotoLibrary.requestAuthorization(for: .readWrite,
            handler: {status in
                if status == .authorized {
                    print("access allowed")
                } else {
                    print("access denied")
                }
            })
    }
}
```

PHPickerViewController

- PHPickerViewController(configuration: PHPickerConfiguration)
 - Set delegate
- PHPickerConfiguration
 - Set filter (.images, .livePhotos, .videos)
 - Set selectionLimit: Int

PHPickerViewControllerDelegate

- `func picker(_ picker: PHPickerViewController, didFinishPicking results: [PHPickerResult])`
 - `PHPickerResult.itemProvider`
 - `canLoadObject(ofClass: <Type>.self)`
 - `loadObject(ofClass: <Type>.self, completionHandler: (data, error) -> Void)`
 - Cast data as <Type>
 - If updating view, then use `DispatchQueue.main.async {}`

Select Image from Photo Library

```
import PhotosUI

class ViewController: UIViewController, PHPickerViewControllerDelegate {

    func selectImageFromLibrary () {
        var configuration = PHPickerConfiguration()
        configuration.filter = .images
        configuration.selectionLimit = 1
        let picker = PHPickerViewController(configuration: configuration)
        picker.delegate = self
        present(picker, animated: true, completion: nil)
    }
}
```

Select Image from Photo Library

```
// PHPickerViewControllerDelegate method

func picker(_ picker: PHPickerViewController,
            didFinishPicking results: [PHPickerResult]) {
    picker.dismiss(animated: true, completion: nil)
    for result in results {
        if result.itemProvider.canLoadObject(ofClass: UIImage.self) {
            result.itemProvider.loadObject(ofClass: UIImage.self,
                                           completionHandler: {(imageData, error) in
                                               if let image = imageData as? UIImage {
                                                   DispatchQueue.main.async {
                                                       self.imageView.image = image
                                                   }
                                               }
                                           })
        }
    }
}
```


Resources

- Speech recognition

developer.apple.com/documentation/speech

- Speech synthesis

– developer.apple.com/documentation/avfaudio/avspeechsynthesizer

- PhotoKit

– developer.apple.com/documentation/photokit