95. Recording Audio on iOS 10 with AVAudioRecorder

오디오 재생 외에도 iOS AV Foundation Framework는 AVAudioRecorder 클래스를 사용하여 iOS에서 사운드를 녹음하는 기능을 제공합니다. 이 장에서는 AVAudioRecorder 클래스를 사용하여 오디오를 녹음하는 방법을 보여주는 자습서를 단계별로 설명합니다.

AVAudioRecorder 튜토리얼 개요

이 장의 목적은 오디오를 녹음하고 재생하는 iOS 10 응용 프로그램을 만드는 것입니다. AVAudioRecorder 클래스의 인스턴스를 만들고 오디오를 포함하는 파일 및 오디오의 품질과 형식을 지정하는 설정 범위로 구성하여 AVAudioRecorder 클래스를 구성합니다. 녹음 된 오디오 파일의 재생은 AVAudioPlayer 클래스를 사용하여 수행됩니다. 자세한 내용은 AVAudioPlayer를 사용하여 iOS 10에서 오디오 재생 장에 설명되어 있습니다.

오디오 녹음 및 재생은 사용자 인터페이스에서 액션 메서드에 연결된 버튼으로 제어되며,이 메서드는 AVAudioRecorder 및 AVAudioPlayer 객체의 인스턴스 메서드를 각각 적절하게 호출합니다.

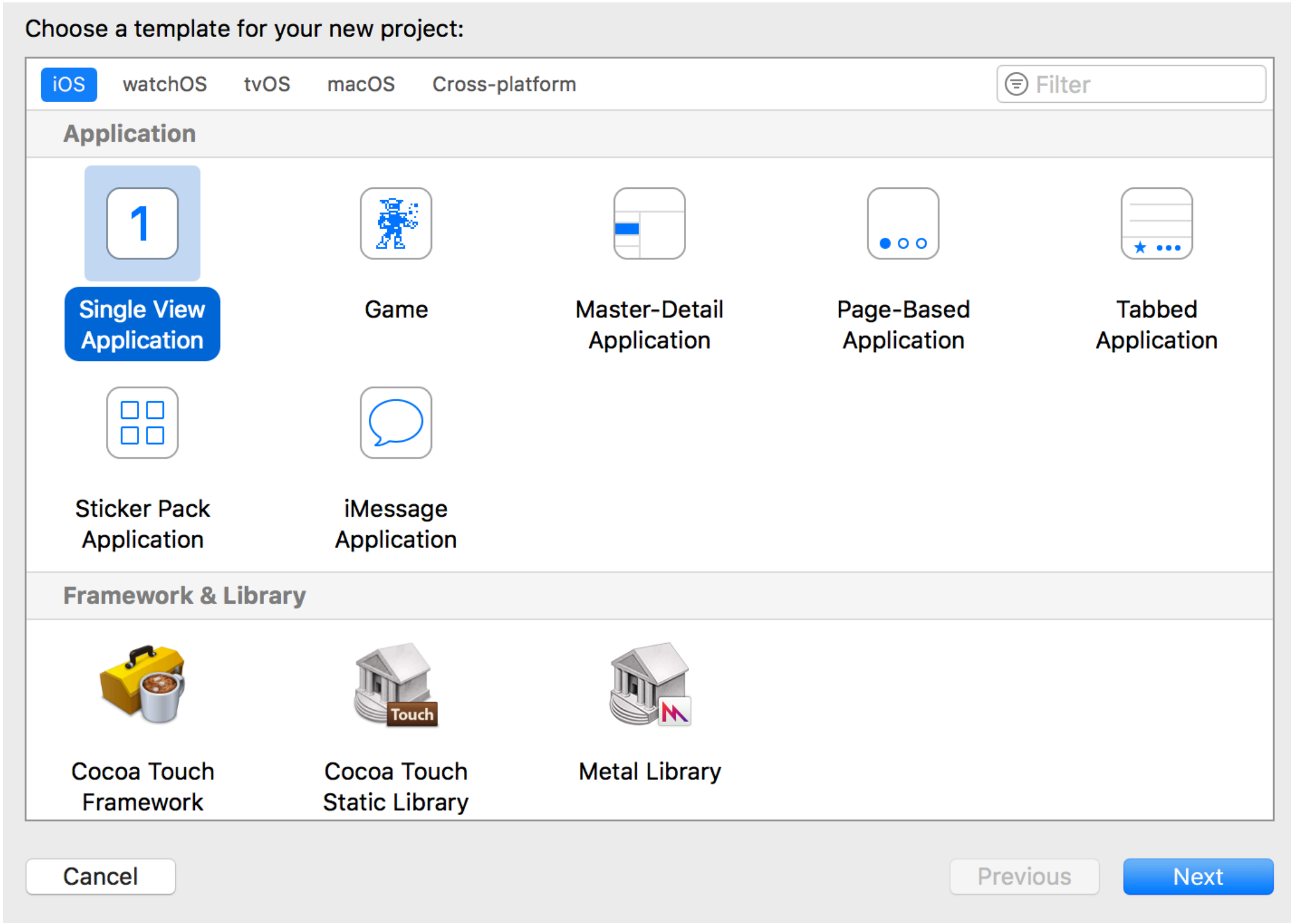
예제 애플리케이션의보기 컨트롤러는 재생 및 녹음과 관련된 이벤트 알림을 받기 위해 AVAudioRecorderDelegate 및 AVAudioPlayerDelegate 프로토콜과 여러 가지 상응하는 대리자 메서드도 구현합니다.

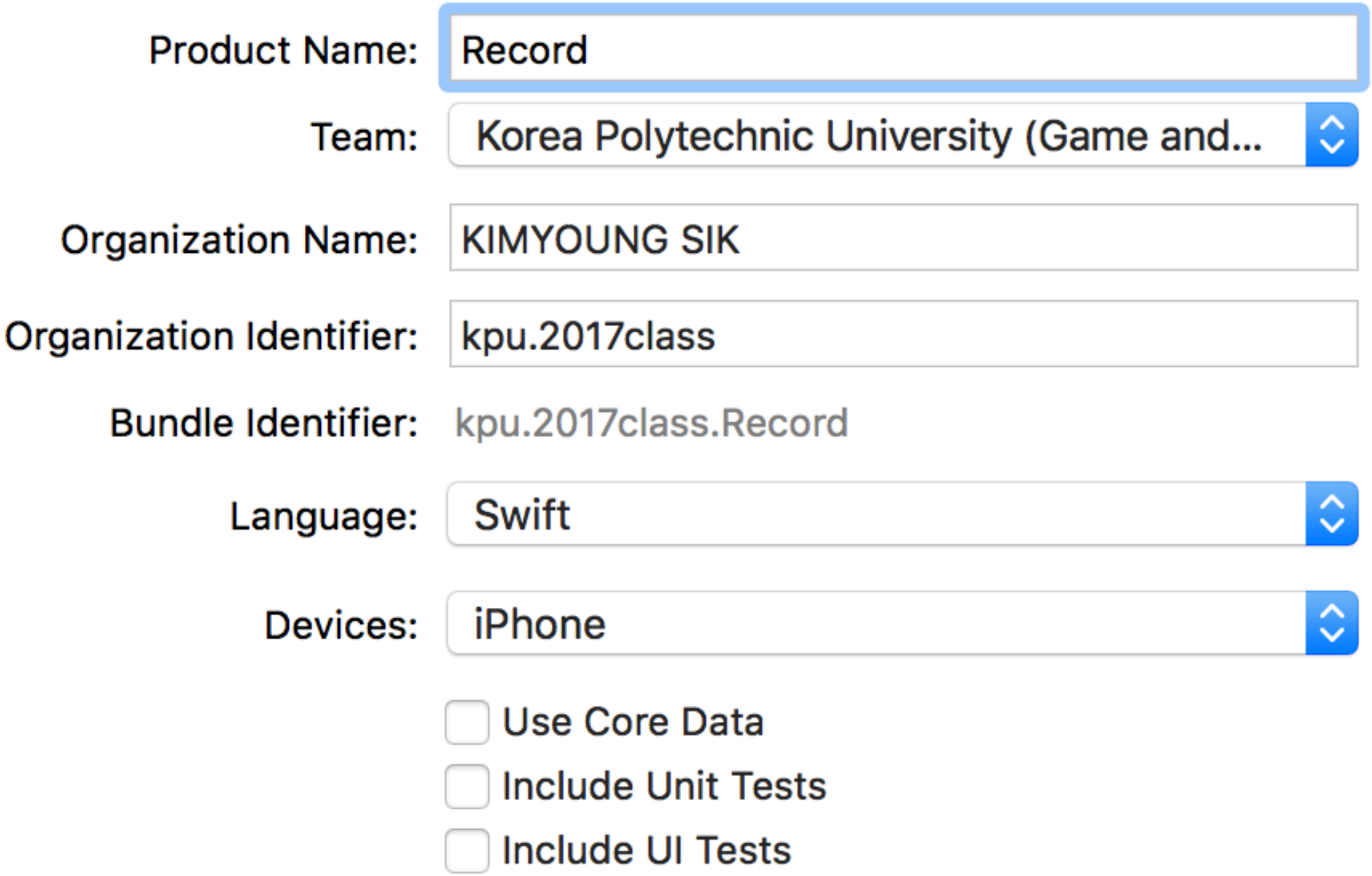
레코더 프로젝트 만들기

Xcode를 시작하고 Swift 프로그래밍 언어를 사용하여 Record라는 새로운 Universal 단일보기 기반 응용 프로그램을 만듭니다.

Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.


Choose a template for your new project:
iOS
watchOS
tvOS
macOS
Cross-platform
Filter
Application
Single View Application
Game
Master-Detail Application
Page-Based Application
Tabbed Application
Sticker Pack Application
iMessage Application
Framework & Library
Cocoa Touch Framework
Cocoa Touch Static Library
Touch
Metal Library
Cancel
Previous
Next


Product Name:
Record
Team:
Korea Polytechnic University (Game and...
Organization Name:
KIMYOUNG SIK
Organization Identifier:
kpu.2017class
Bundle Identifier:
kpu.2017class.Record
Language:
Swift
Devices:
iPhone
Use Core Data
Include Unit Tests
Include UI Tests

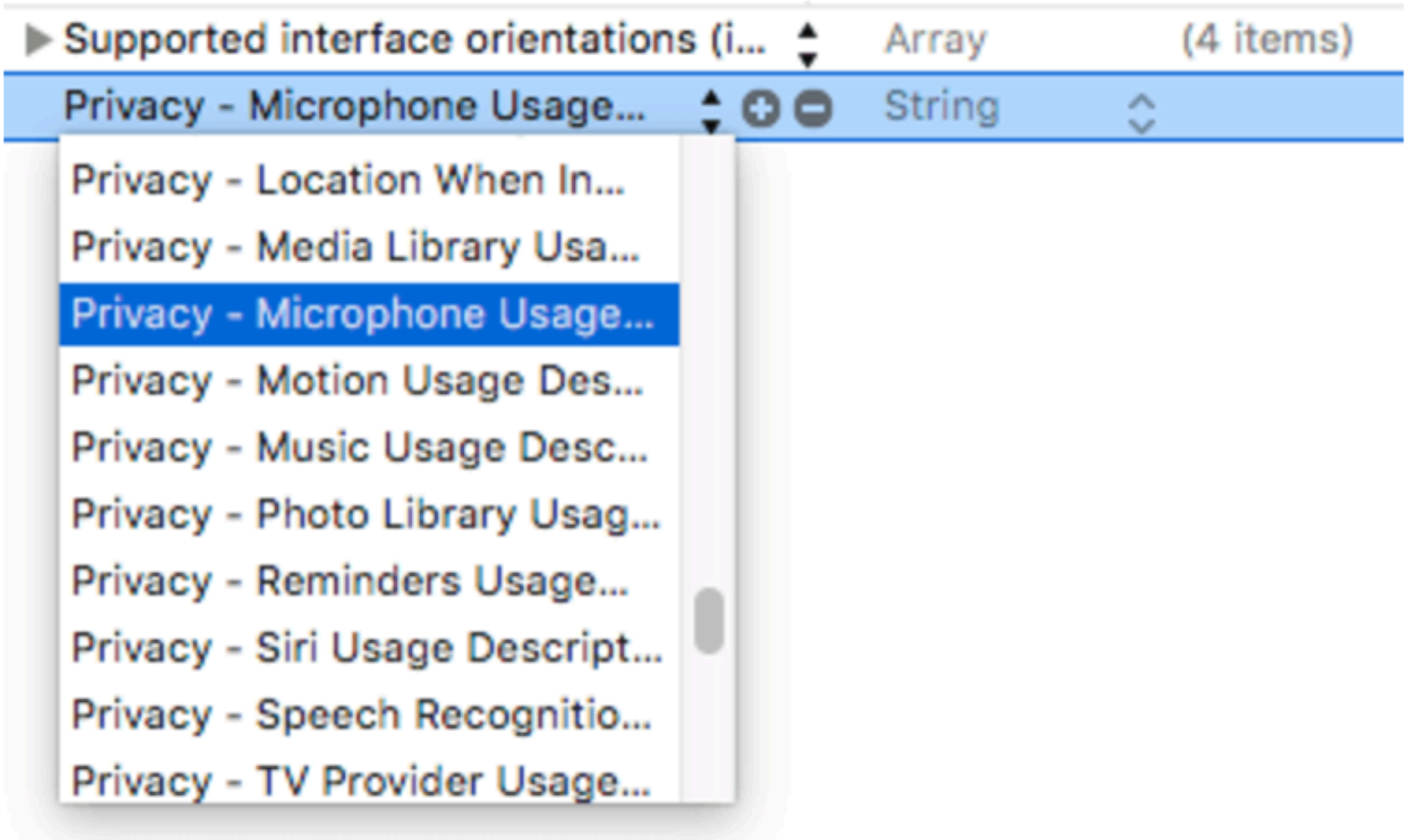
마이크 사용 설명 구성

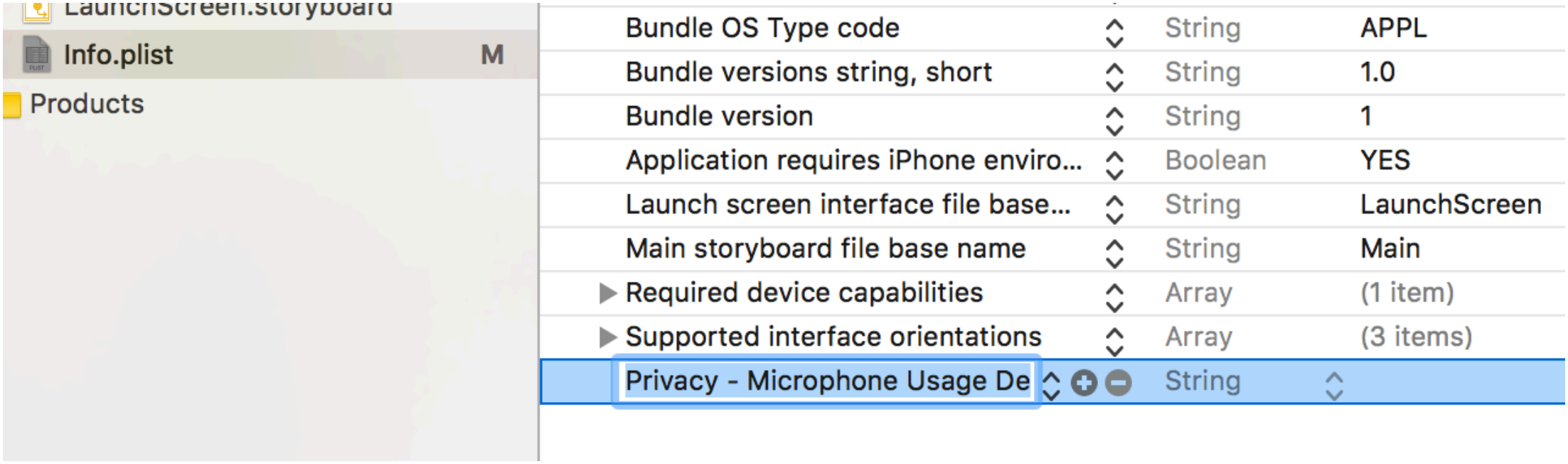
iOS 10 앱 내에서 마이크에 액세스하는 것은 잠재적으로 사용자의 개인 정보를 위험에 빠뜨리는 것으로 간주됩니다. 앱이 마이크에 액세스하려고하면 운영 체제는 앱에 대한 승인을 요청하는 경고 대화 상자를 사용자에게 표시합니다. 이 대화 상자의 내용에는 마이크 사용에 대한 정당성을 제공하는 앱의 메시지가 포함되어 있습니다. 이 텍스트 메시지는 NSMicrophoneUsageDescription 키를 사용하여 Info.plist 파일 내에서 지정해야합니다. 이 키가 없으면 런타임에 앱이 다운됩니다.

마이크 사용 설명 구성

iOS 10 앱 내에서 마이크에 액세스하는 것은 잠재적으로 사용자의 개인 정보를 위험에 빠뜨리는 것으로 간주됩니다. 앱이 마이크에 액세스하려고하면 운영 체제는 앱에 대한 승인을 요청하는 경고 대화 상자를 사용자에게 표시합니다. 이 대화 상자의 내용에는 마이크 사용에 대한 정당성을 제공하는 앱의 메시지가 포함되어 있습니다. 이 텍스트 메시지는 NSMicrophoneUsageDescription 키를 사용하여 Info.plist 파일 내에서 지정해야합니다. 이 키가 없으면 런타임에 앱이 다운됩니다.

이 값을 추가하려면 프로젝트 탐색기 패널에서 Info.plist 파일을 선택하여 속성 목록 편집기에로드하십시오. 목록에서 마지막 항목을 선택하고 + 버튼을 클릭하여 새 항목을 추가하십시오. 새 항목의 키 필드를 클릭하고 그림 96-1에서 설명한대로 Privacy - Microphone Usage Description 메뉴 옵션을 찾아 선택합니다.

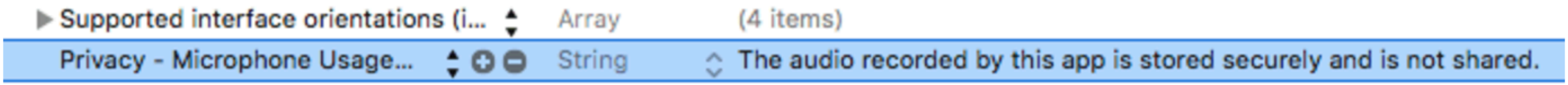


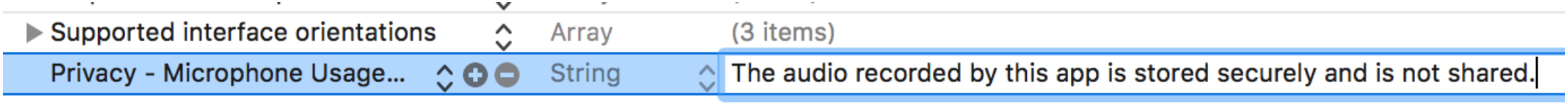

LaunchScreen.storyboard
Info.plist
M
Products
Bundle OS Type code
String
APPL
Bundle versions string, short
String
1.0
Bundle version
String
1
Application requires iPhone enviro...
Boolean
YES
Launch screen interface file base...
String
LaunchScreen
Main storyboard file base name
String
Main
Required device capabilities
Array
(1 item)
Supported interface orientations
Array
(3 items)
Privacy - Microphone Usage De
String

키가 지정되면 해당 값 필드를 다음과 같이 설정합니다.

> 이 앱으로 녹음 된 오디오는 안전하게 저장되며 공유되지 않습니다.

사용 키가 구성된 후에는 속성 목록 편집기 내의 항목이 그림 96-2에 표시된 항목과 일치해야합니다.




Supported interface orientations
Array
(3 items)
Privacy - Microphone Usage...
String
The audio recorded by this app is stored securely and is not shared.

이후에 앱을 처음 실행하면 사용법 메시지가 포함 된 대화 상자가 나타납니다. 사용자가 확인 버튼을 탭한 경우에만 마이크 액세스 권한이 사용자에게 부여됩니다.

사용자 인터페이스 디자인

Main.storyboard 파일을 선택하고로드 된 후에는 오브젝트 라이브러리 창 (보기 -> 유틸리티 -> 오브젝트 라이브러리 표시)에서 Button 오브젝트를 끌어서보기 창에 놓습니다. 보기에 배치되면 그림 96-3과 같이 사용자 인터페이스가 나타나도록 각 단추의 텍스트를 수정하십시오.

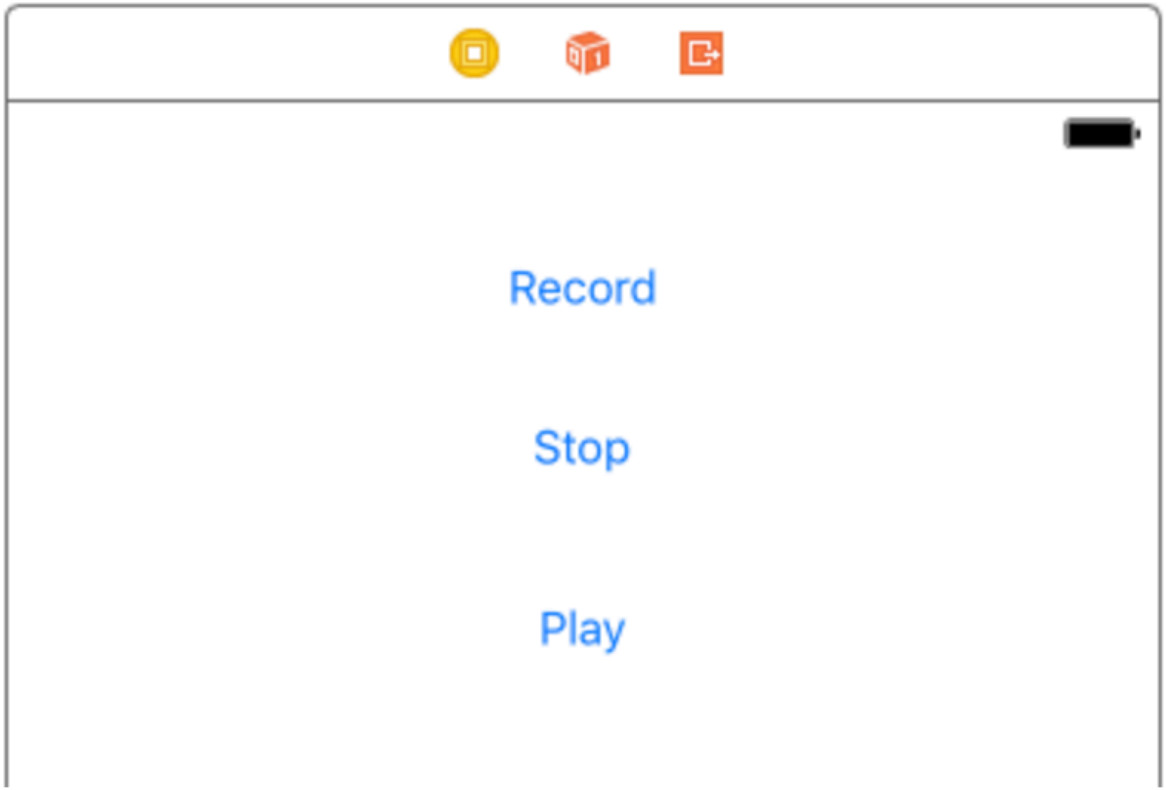
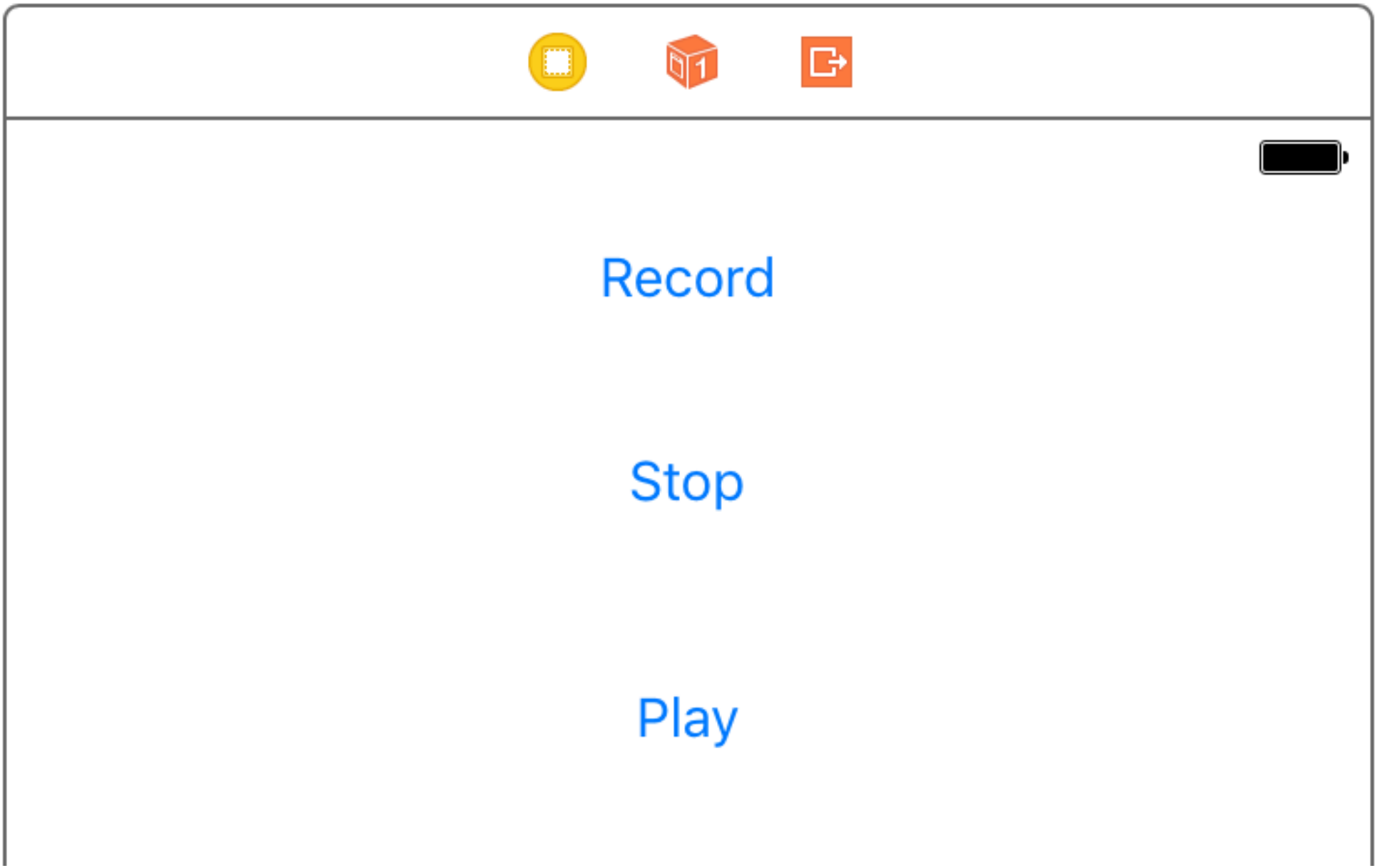
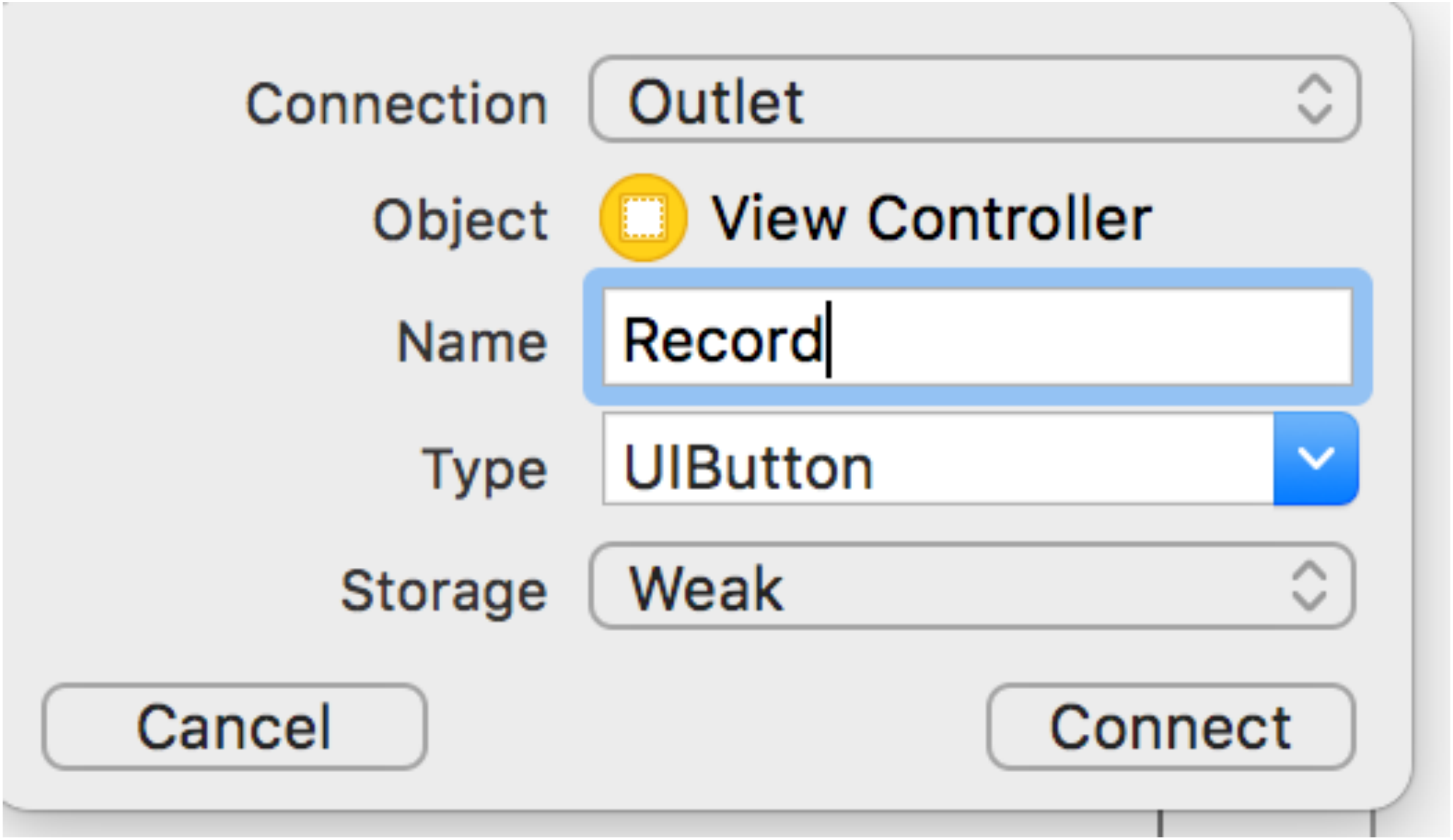


그림 96-3

Button - Intercepts touch events and sends an action message to a target object when it's tapped.


Record
Stop
Play

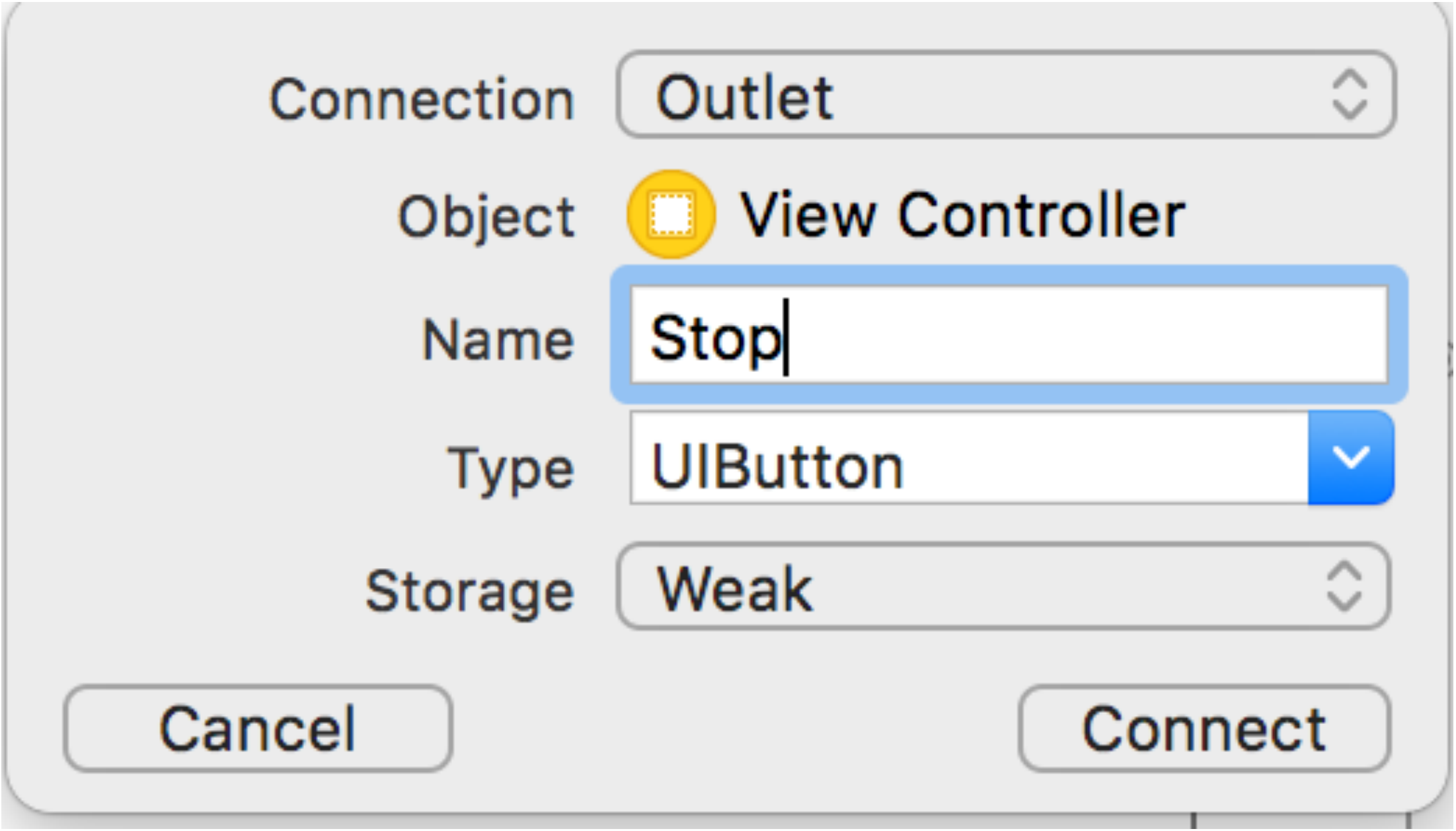
뷰 캔버스에서 "Record"버튼 객체를 선택하고 Assistant Editor 패널을 표시하고 편집기에 ViewController.swift 파일의 내용이 표시되는지 확인하십시오. Record 버튼 객체를 Ctrl- 클릭하고 Assistant Editor에서 클래스 선언 라인 바로 아래로 드래그합니다. 회선을 해제하고 recordButton이라는 콘센트 연결을 설정하십시오. PlayButton 및 stopButton이라는 "재생"및 "중지"버튼에 대한 콘센트 연결을 설정하려면이 단계를 반복하십시오.

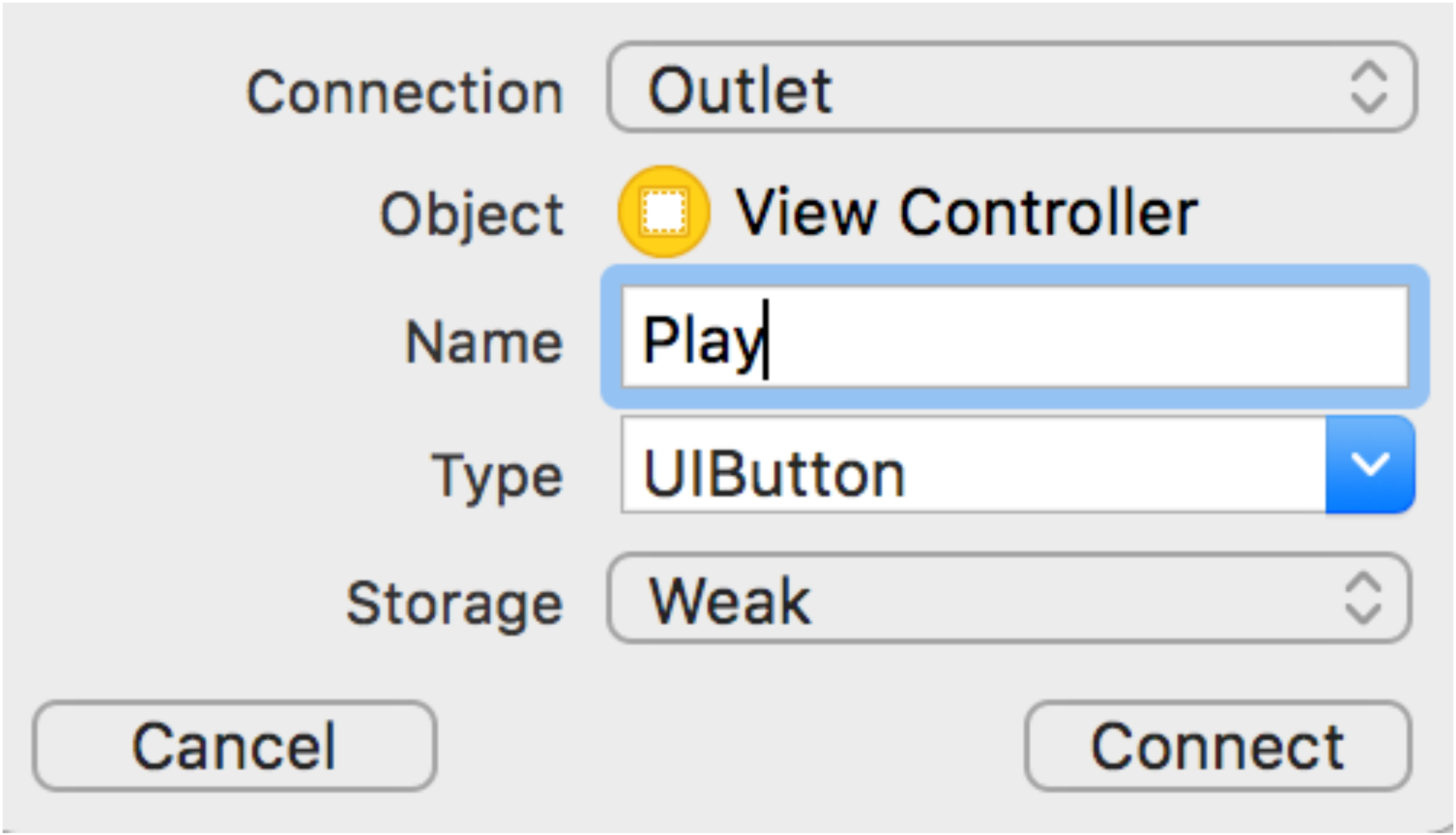

Connection
Outlet
Object
View Controller
Name
Record
Type
UIButton
Storage
Weak
Cancel
Connect

Stop

Play

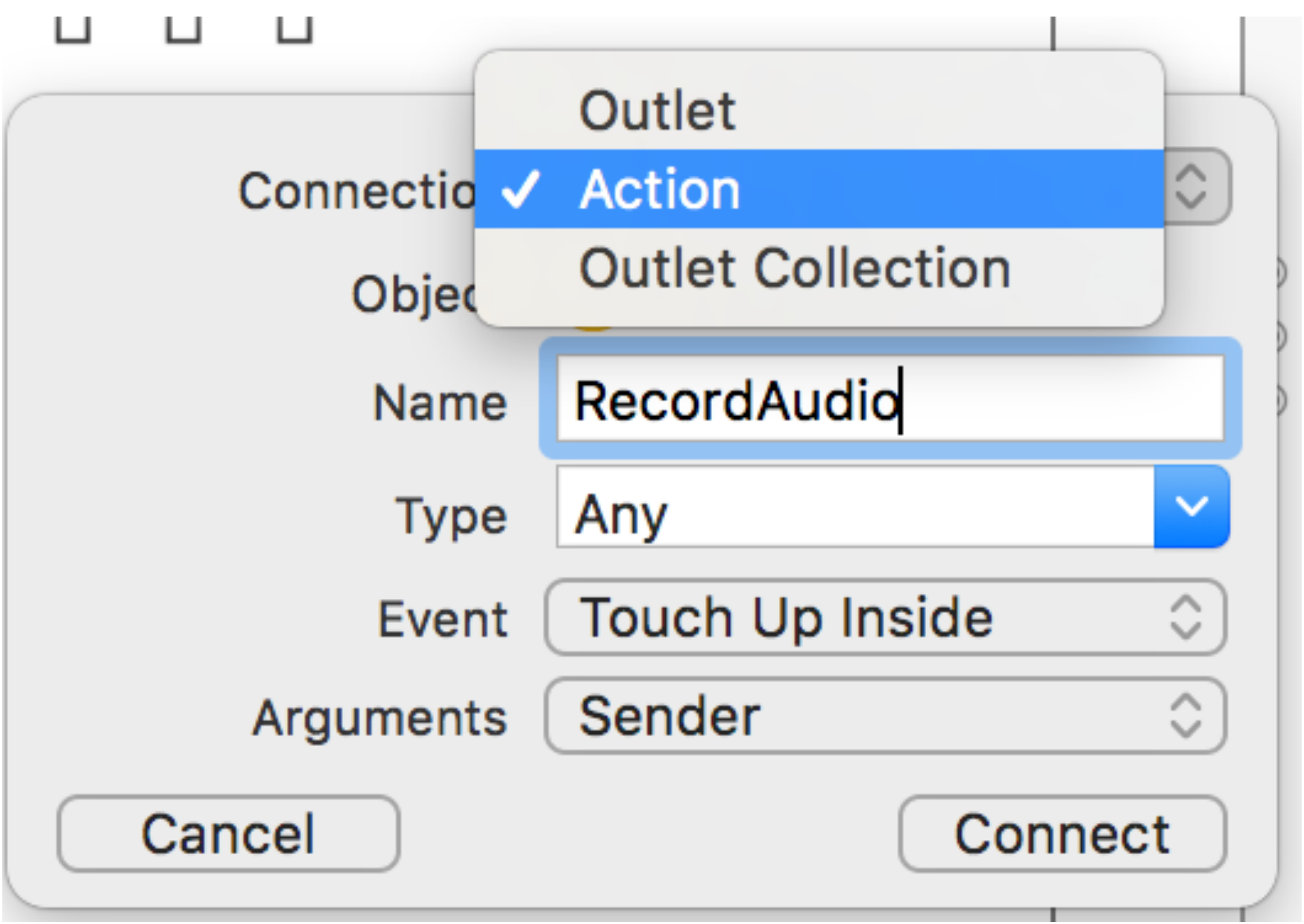
```
//  Created by KIMYOUNG SIK on 2017. 1. 25..
//  Copyright © 2017년 KIMYOUNG SIK. All righ
//

import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var Record: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loa
            nib.
    }
```

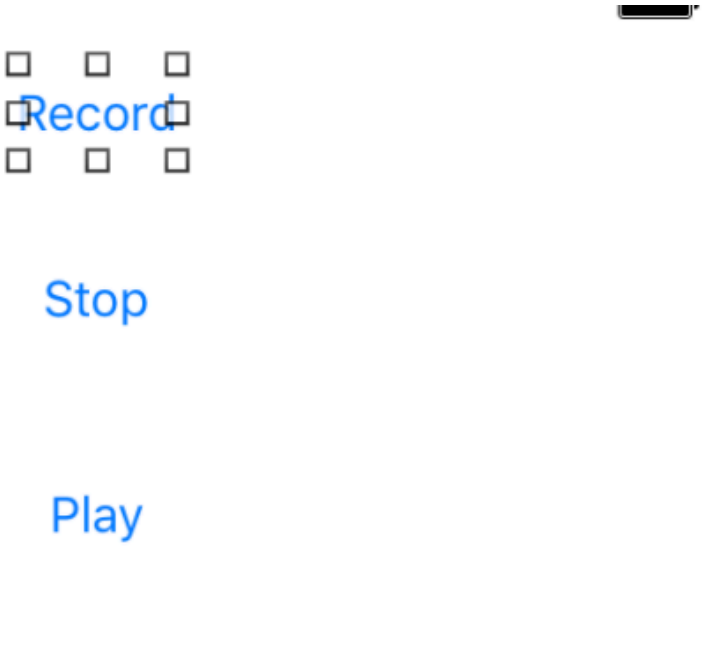

Connection
Outlet
Object
View Controller
Name
Stop
Type
UIButton
Storage
Weak
Cancel
Connect


Connection
Outlet
Object
View Controller
Name
Play
Type
UIButton
Storage
Weak
Cancel
Connect

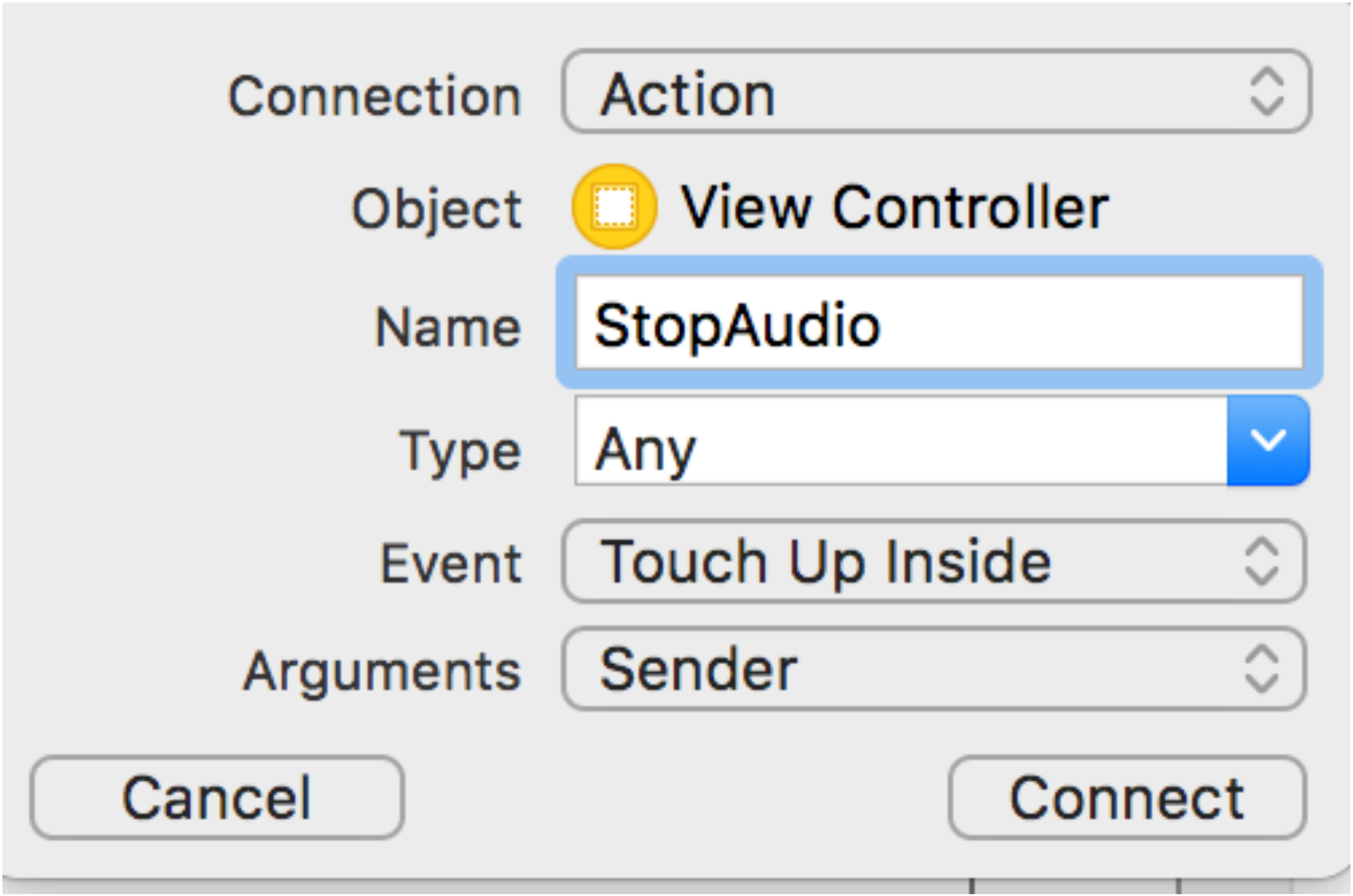
```
class ViewController: UIViewController {
    @IBOutlet weak var Record: UIButton!
    @IBOutlet weak var Stop: UIButton!
    @IBOutlet weak var Play: UIButton!
```

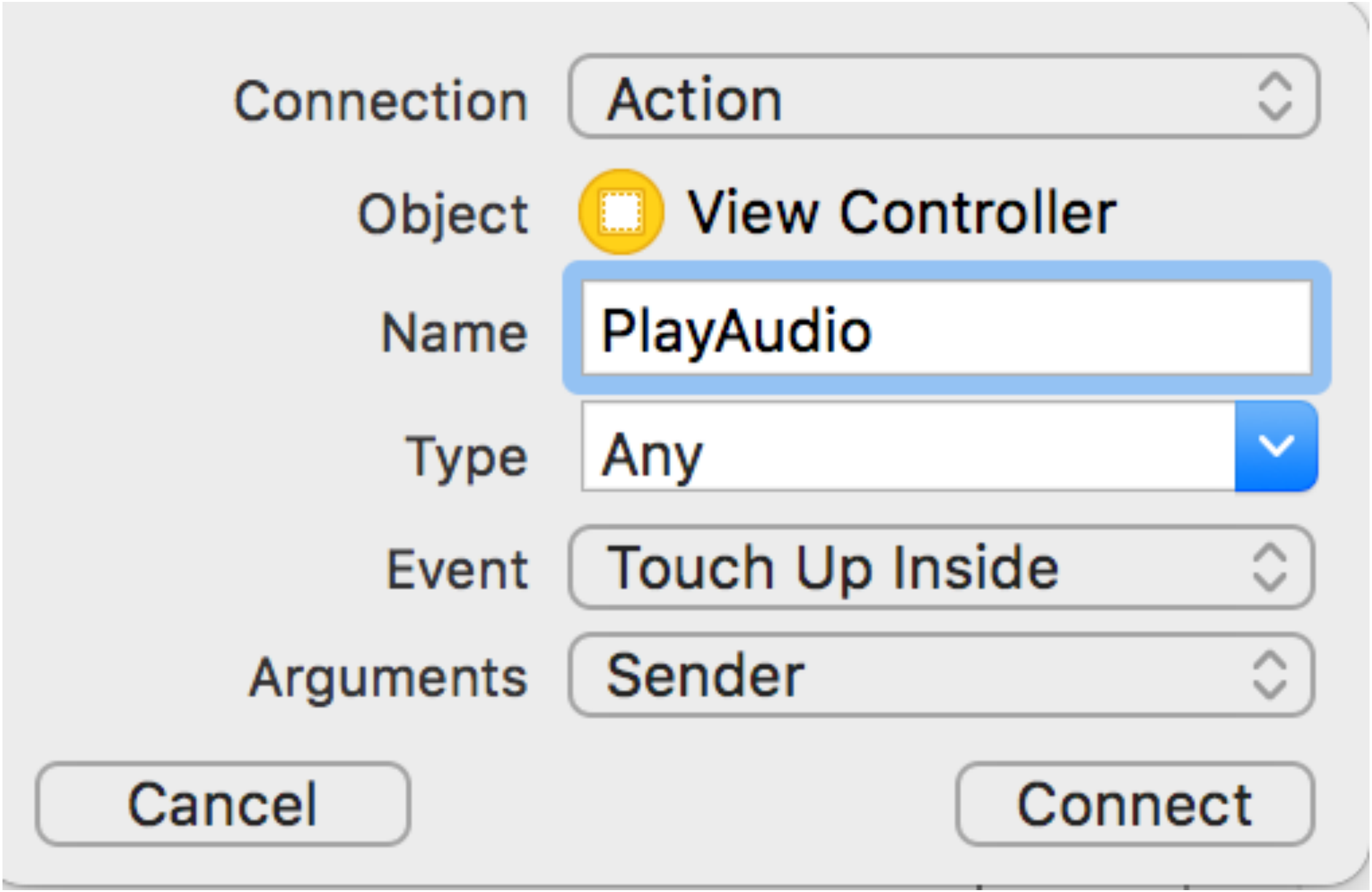
Assistant Editor를 계속 사용하면서 세 버튼에서 recordAudio, playAudio 및 stopAudio라는 메서드로 Action 연결을 설정합니다.


Outlet
Action
Outlet Collection
Connectio
Objec
Name
RecordAudio
Type
Any
Event
Touch Up Inside
Arguments
Sender
Cancel
Connect




```
//  Created by KIMYOUNG SIK on 2017. 1. 25..
//  Copyright © 2017년 KIMYOUNG SIK. All rights res
//

import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var Record: UIButton!
    @IBOutlet weak var Stop: UIButton!
    @IBOutlet weak var Play: UIButton!

    @IBAction func RecordAudio(_ sender: Any) {
    }
    override func viewDidLoad() {
```


Connection
Action
Object
View Controller
Name
StopAudio
Type
Any
Event
Touch Up Inside
Arguments
Sender
Cancel
Connect


Connection
Action
Object
View Controller
Name
PlayAudio
Type
Any
Event
Touch Up Inside
Arguments
Sender
Cancel
Connect

```swift
import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var Record: UIButton!
    @IBOutlet weak var Stop: UIButton!
    @IBOutlet weak var Play: UIButton!

    @IBAction func RecordAudio(_ sender: Any) {
    }
    @IBAction func StopAudio(_ sender: Any) {
    }
    @IBAction func PlayAudio(_ sender: Any) {
    }
    override func viewDidLoad() {
```

어시스턴트 편집기 패널을 닫고 ViewController.swift 파일을 선택하여 AVFoundation 프레임 워크를 가져오고 일부 위임 프로토콜을 준수하도록 선언하고 AVAudioRecorder 및 AVAudioPlayer 인스턴스에 대한 참조를 저장하는 속성을 추가합니다.

```
import UIKit
import AVFoundation
class ViewController: UIViewController, AVAudioPlayerDelegate, AVAudioRecorderDelegate {
    var audioPlayer: AVAudioPlayer?
    var audioRecorder: AVAudioRecorder?
```

AVAudioRecorder 인스턴스 만들기

흔히 그렇듯이 AVAudioRecorder 인스턴스를 초기화하는 좋은 위치는 ViewController.swift 파일에있는 뷰 컨트롤러의 viewDidLoad 메서드입니다. 프로젝트 탐색기에서 파일을 선택하고이 방법을 찾은 다음 수정하여 다음과 같이 읽습니다.

```swift
override func viewDidLoad() {
    super.viewDidLoad()
    //응용 프로그램을 처음 시작하면 AVAudioRecorder 클래스의 인스턴스를 만들어야합니다.
    //녹음 된 오디오가 저장 될 파일의 URL로 초기화됩니다.
    //또한 비트 레이트, 샘플 속도 및 오디오 품질과 같은 레코딩 설정을 나타내는 Dictionary 객체가
    //초기화 메소드의 인수로 전달됩니다.
    //사용 가능한 설정에 대한 자세한 설명은 해당 Apple iOS 참조 자료에서 찾을 수 있습니다.

    //아직 오디오가 녹음되지 않았기 때문에 재생 및 정지 버튼을 비활성화합니다.
    Play.isEnabled = false
    Stop.isEnabled = false

    //그런 다음 응용 프로그램의 문서 디렉토리를 식별하고 sound.caf라는 위치에있는 파일의 URL을 구성합니다.
    let fileMgr = FileManager.default
    let dirPaths = fileMgr.urls(for: .documentDirectory,
                                in: .userDomainMask)
    let soundFileURL = dirPaths[0].appendingPathComponent("sound.caf")

    //그런 다음 오디오 세션과 AVAudioRecorder 클래스의 인스턴스가 만들어지기 전에
    //녹음 품질 설정을 포함하는 사전 개체가 만들어집니다.
    let recordSettings =
        [AVEncoderAudioQualityKey: AVAudioQuality.min.rawValue,
         AVEncoderBitRateKey: 16,
         AVNumberOfChannelsKey: 2,
         AVSampleRateKey: 44100.0] as [String : Any]

    let audioSession = AVAudioSession.sharedInstance()

    do {
        try audioSession.setCategory(
            AVAudioSessionCategoryPlayAndRecord)
    } catch let error as NSError {
        print("audioSession error: \(error.localizedDescription)")
    }
    //오류가 없다고 가정하면 audioRecorder 인스턴스는 사용자가 요청할 때 녹음을 시작할 준비가되어 있습니다.
    do {
        try audioRecorder = AVAudioRecorder(url: soundFileURL,
                                            settings: recordSettings as [String : AnyObject])
        audioRecorder?.prepareToRecord()
    } catch let error as NSError {
        print("audioSession error: \(error.localizedDescription)")
    }
}
```

액션 메소드 구현하기

다음 단계는 3 개의 버튼 객체에 연결된 액션 메소드를 구현

위의 각 메소드는 사용자 인터페이스에서 적절한 버튼을 활성화 및 비활성화하고 AVAudioRecorder 및 AVAudioPlayer 객체 인스턴스와 상호 작용하여 오디오를 녹음 또는 재생하는 데 필요한 단계를 수행합니다.

```swift
@IBAction func RecordAudio(_ sender: AnyObject) {
    if audioRecorder?.isRecording == false {
        Play.isEnabled = false
        Stop.isEnabled = true
        audioRecorder?.record()
    }
}

@IBAction func StopAudio(_ sender: AnyObject) {
    Stop.isEnabled = false
    Play.isEnabled = true
    Record.isEnabled = true

    if audioRecorder?.isRecording == true {
        audioRecorder?.stop()
    } else {
        audioPlayer?.stop()
    }
}

@IBAction func PlayAudio(_ sender: AnyObject) {
    if audioRecorder?.isRecording == false {
        Stop.isEnabled = true
        Record.isEnabled = false

        do {
            try audioPlayer = AVAudioPlayer(contentsOf:
                (audioRecorder?.url)!)
            audioPlayer!.delegate = self
            audioPlayer!.prepareToPlay()
            audioPlayer!.play()
        } catch let error as NSError {
            print("audioPlayer error: \(error.localizedDescription)")
        }
    }
}
```

대리자 메서드 구현

성공 또는 녹음 또는 재생에 대한 알림을 받으려면 일부 대리자 방법을 구현해야합니다. 이 튜토리얼에서는 오류가 발생했음을 나타내는 메소드와 재생이 완료된 메소드를 구현해야한다. 다시 한번 ViewController.swift 파일을 편집하고 다음과 같이이 메소드를 추가하십시오.

```swift
//재생 완료 메시지
func audioPlayerDidFinishPlaying(_ player: AVAudioPlayer, successfully flag: Bool) {
    Record.isEnabled = true
    Stop.isEnabled = false
}
//오류 메시지
func audioPlayerDecodeErrorDidOccur(_ player: AVAudioPlayer, error: Error?) {
    print("Audio Play Decode Error")
}

func audioRecorderDidFinishRecording(_ recorder: AVAudioRecorder, successfully flag: Bool) {
}

func audioRecorderEncodeErrorDidOccur(_ recorder: AVAudioRecorder, error: Error?) {
    print("Audio Record Encode Error")
}
```

응용 프로그램 테스트

연결 장치 또는 시뮬레이터 세션에 응용 프로그램을 설치하고 기본 도구 모음에서 실행 버튼을 클릭하여 응용 프로그램을 빌드하고 실행하도록 Xcode를 구성합니다. 기기에로드되면 운영체제는 앱이 마이크에 액세스하도록 허용 할 수있는 권한을 찾습니다. 액세스를 허용하고 녹음 단추를 눌러 소리를 녹음하십시오. 녹음이 완료되면 중지 버튼을 터치하고 재생 버튼을 사용하여 오디오를 재생합니다.