Configuring Audio Settings for iOS and tvOS

Media playback apps for iOS and tvOS require you to perform some configuration of your audio session to enable certain behaviors and capabilities. This chapter discusses how to perform this configuration to ensure that you provide the best possible playback experience to your users.

Configuring Your iOS and tvOS Audio Session

An audio session acts as an intermediary between your app and the operating system—and in turn, the underlying audio hardware. You use it to communicate to the operating system the *nature* of your app's audio without detailing the specific behavior or required interactions with the audio hardware. This delegates the management of those details to the audio session, which ensures that the operating system can best manage the user's audio experience.

All iOS and tvOS apps have a default audio session that comes preconfigured as follows:

- Audio playback is supported, but audio recording is disallowed (audio recording applies only to iOS).
- In iOS, setting the Ring/Silent switch to silent mode silences any audio being played by the app.
- In iOS, when the device is locked, the app's audio is silenced.
- When the app plays audio, any other background audio is silenced.

You get a lot of useful behavior from the default audio session, but it isn't the behavior you need when building a media playback app. To change this behavior, you configure your app's audio session *category*.

An audio session category defines the general audio behavior your app requires. There are seven possible categories you can use (see Audio Session Categories and Modes), but the one needed by most playback apps is called `AVAudioSessionCategoryPlayback`. This category indicates that audio playback is a central feature of your app. When you specify this category, your app's audio continues with the Ring/Silent switch set to silent mode (iOS only). With this category, your app can also play background audio if you're using the Audio, AirPlay, and Picture in Picture background mode. For more information, see Enabling Background Audio.

You use an `AVAudioSession` object to configure your app's audio session. This is a singleton object used to set the audio session category as well as perform other configuration settings. You can interact with the audio session throughout your app's life cycle, but it's often useful to perform this configuration at app launch as shown in the following example:

```
func application(_ application: UIApplication,
                 didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey : Any]?) -> Bool {
    let audioSession = AVAudioSession.sharedInstance()
    do {
        try audioSession.setCategory(AVAudioSessionCategoryPlayback)
    } catch {
        print("Setting category to AVAudioSessionCategoryPlayback failed.")
    }
    // Other project setup
    return true
}
}
```

This category is used when you activate the audio session using the `setActive:error:` or `setActive:withOptions:error:` method.

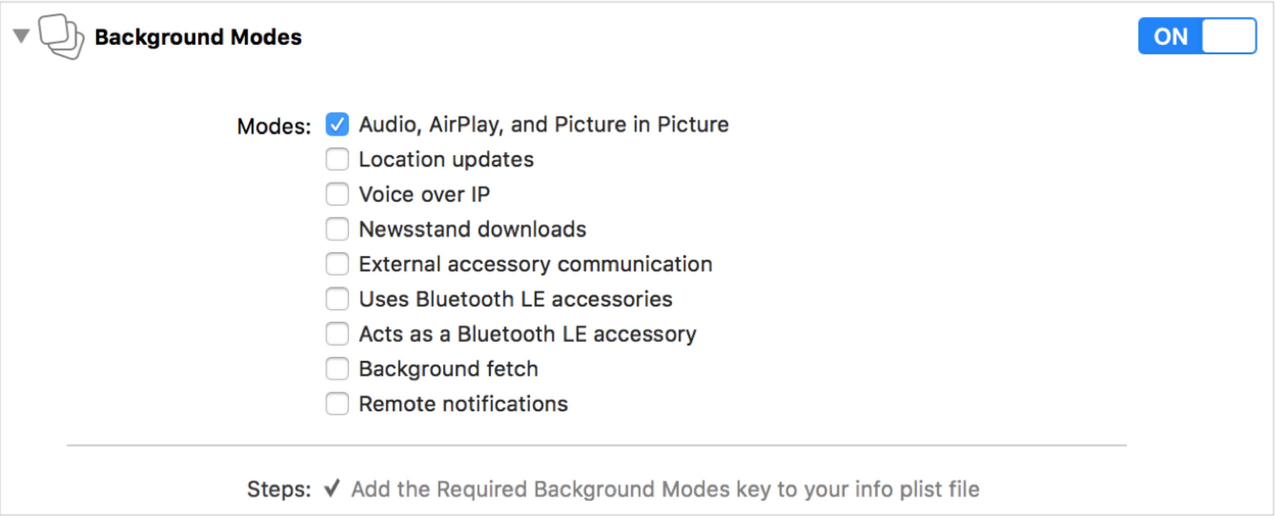
> **Note:** You can activate the audio session at any time after setting its category, but it's generally preferable to defer this call until your app begins audio playback. This ensures that you won't prematurely interrupt any other background audio that may be in progress.

Setting the category is the minimal interaction you'll need to have with `AVAudioSession`, but there are many other configuration options and features you can use as well. See *Audio Session Programming Guide* to learn more about using audio sessions.

Enabling Background Audio

iOS and tvOS apps require you to enable certain capabilities for some background operations. A common capability required by playback apps is to play background audio. With this capability enabled, your app's audio can continue when users switch to another app or when they lock their iOS devices. This capability is also required for enabling advanced playback features like AirPlay streaming and Picture in Picture playback in iOS.

The easiest way to configure these capabilities is by using Xcode. Select your app's target in Xcode and select the Capabilities tab. Under the Capabilities tab, set the Background Modes switch to ON and select the "Audio, AirPlay, and Picture in Picture" option under the list of available modes.



Background Modes ON

Modes:
☑ Audio, AirPlay, and Picture in Picture
☐ Location updates
☐ Voice over IP
☐ Newsstand downloads
☐ External accessory communication
☐ Uses Bluetooth LE accessories
☐ Acts as a Bluetooth LE accessory
☐ Background fetch
☐ Remote notifications

Steps: ✓ Add the Required Background Modes key to your info plist file

With this mode enabled and your audio session configured, your app is ready to play background audio.