Running **Radio** in the Background (iOS/Android)

This is a short manual how to run **Radio** in the background on an iOS or Android device.

Visit us at the Unity AssetStore

Note:

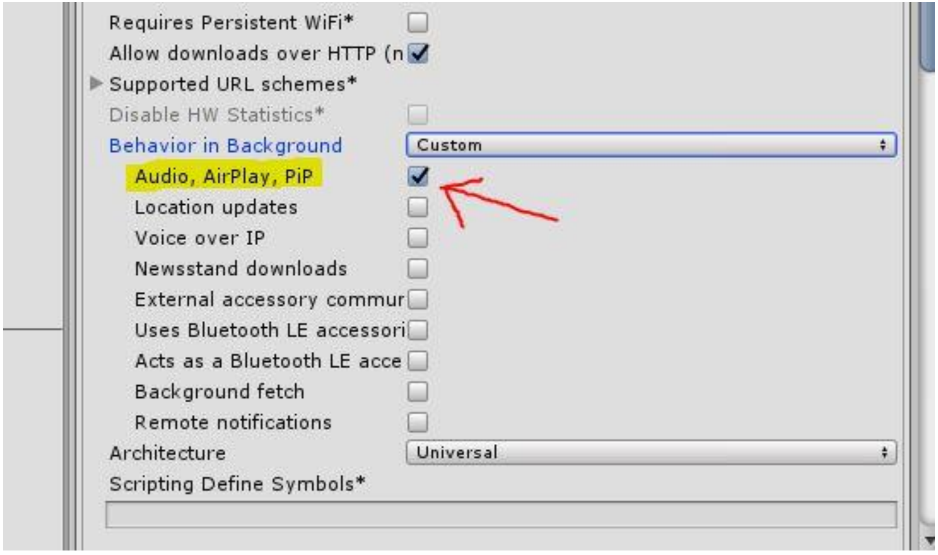
In this tutorial we used
- **Unity 5.6.1**
- **Xcode 8** (tested with **iOS 10.3**)
- **Android Studio 2.1.3** (tested with **Android 5.1**)

iOS

1. Go to the **PlayerSettings**
   File -> Build Settings -> PlayerSettings
2. Change the "Behaviour in Background" from ***Suspend*** to ***Custom***

   Tab: Other Settings

3. Set the "Audio, AirPlay, PIP" top active *(from Unity 5.6.0 on)*



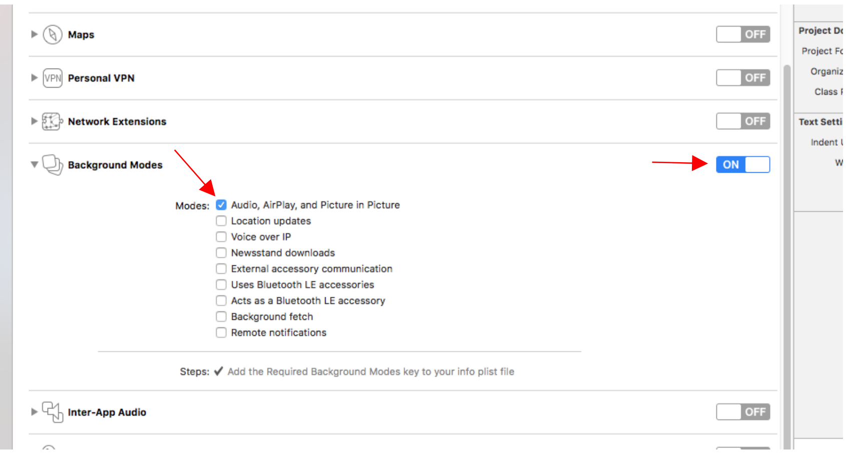
Requires Persistent WiFi* □
Allow downloads over HTTP (n ☑
▶ Supported URL schemes*
Disable HW Statistics* □
Behavior in Background: Custom
- Audio, AirPlay, PiP ☑
- Location updates □
- Voice over IP □
- Newsstand downloads □
- External accessory commur □
- Uses Bluetooth LE accessori □
- Acts as a Bluetooth LE acce □
- Background fetch □
- Remote notifications □

Architecture: Universal
Scripting Define Symbols*

4. Build the project for **iOS**

Xcode Settings

1. Open the Xcode project
2. Click on the project (with the blue icon to the left) and open the **Capabilities** tab
3. Scroll to the **Background Modes** and check if the "Audio, AirPlay, and Picture in Picture" is **active**



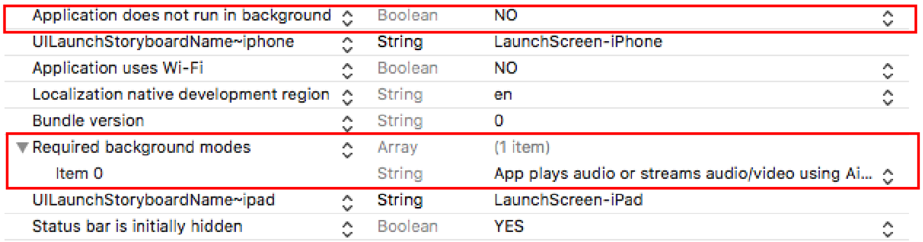
▶ Maps — OFF
▶ Personal VPN — OFF
▶ Network Extensions — OFF
▼ Background Modes — ON

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

▶ Inter-App Audio — OFF

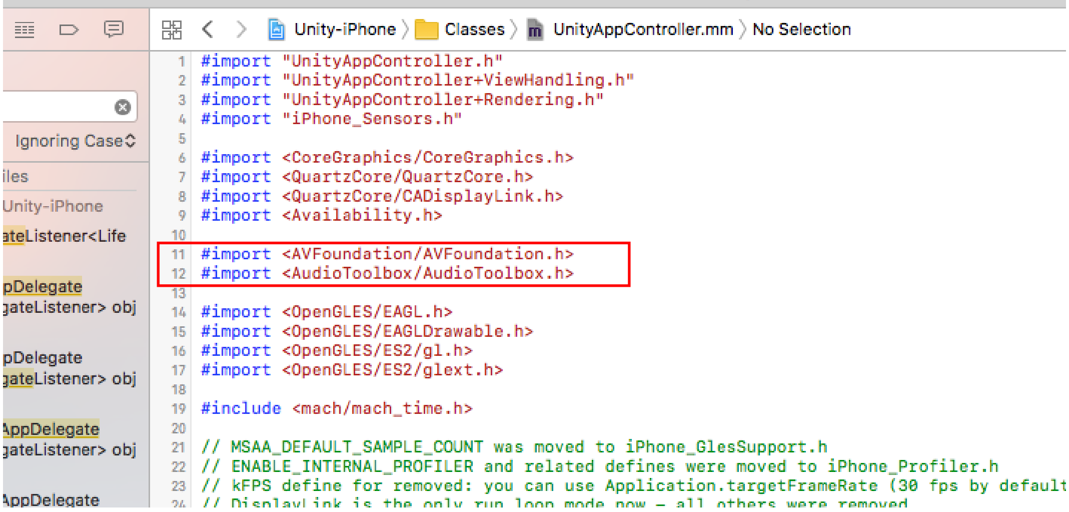
4. Open the ***info.plist*** file and check the following values



Application does not run in background	Boolean	NO
UILaunchStoryboardName~iphone	String	LaunchScreen-iPhone
Application uses Wi-Fi	Boolean	NO
Localization native development region	String	en
Bundle version	String	0
▼ Required background modes	Array	(1 item)
Item 0	String	App plays audio or streams audio/video using Ai...
UILaunchStoryboardName~ipad	String	LaunchScreen-iPad
Status bar is initially hidden	Boolean	YES

5. Search for “AppDelegate” and open the ***UnityAppController.mm*** file
6. Add the **import** lines for **AVFoundation** and **AudioToolbox**

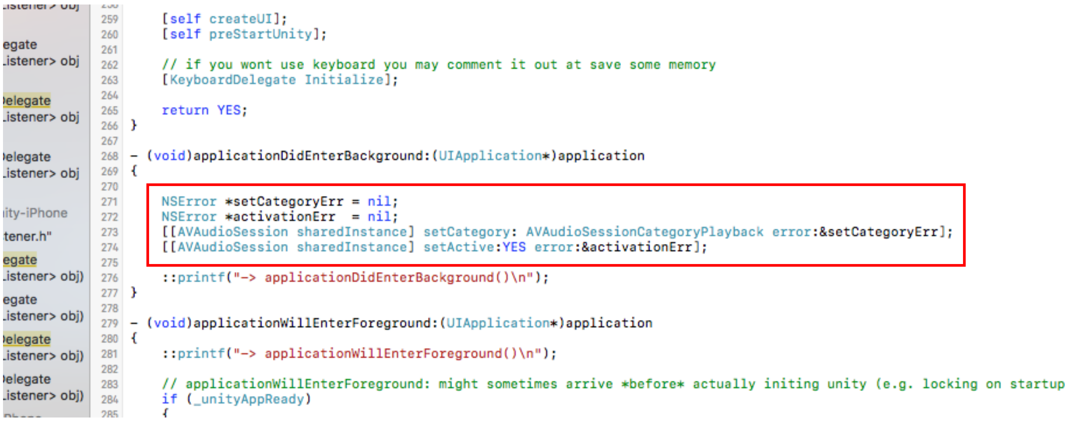
```
#import <AVFoundation/AVFoundation.h>
#import <AudioToolbox/AudioToolbox.h>
```



```
#import "UnityAppController.h"
#import "UnityAppController+ViewHandling.h"
#import "UnityAppController+Rendering.h"
#import "iPhone_Sensors.h"

#import <CoreGraphics/CoreGraphics.h>
#import <QuartzCore/QuartzCore.h>
#import <QuartzCore/CADisplayLink.h>
#import <Availability.h>

#import <AVFoundation/AVFoundation.h>
#import <AudioToolbox/AudioToolbox.h>

#import <OpenGLES/EAGL.h>
#import <OpenGLES/EAGLDrawable.h>
#import <OpenGLES/ES2/gl.h>
#import <OpenGLES/ES2/glext.h>

#include <mach/mach_time.h>

// MSAA_DEFAULT_SAMPLE_COUNT was moved to iPhone_GlesSupport.h
// ENABLE_INTERNAL_PROFILER and related defines were moved to iPhone_Profiler.h
// kFPS define for removed: you can use Application.targetFrameRate (30 fps by default
// DisplayLink is the only run loop mode now - all others were removed
```

7. Next, search in the same file for “**applicationDidEnterBackground**” and add these lines

```
NSError *setCategoryErr = nil;
NSError *activationErr  = nil;
[[AVAudioSession sharedInstance] setCategory: AVAudioSessionCategoryPlayback error:&setCategoryErr];
[[AVAudioSession sharedInstance] setActive:YES error:&activationErr];
```

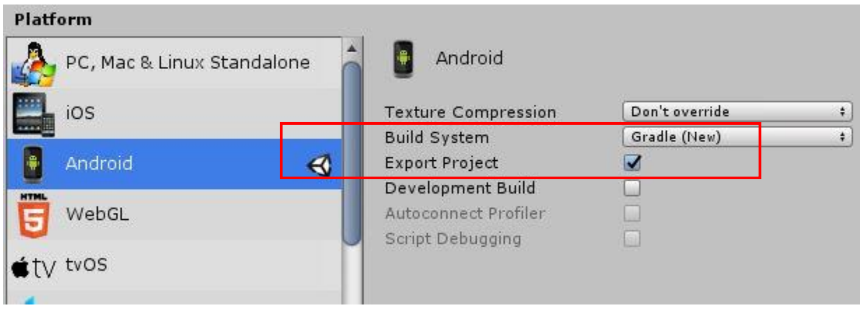


```
    [self createUI];
    [self preStartUnity];

    // if you wont use keyboard you may comment it out at save some memory
    [KeyboardDelegate Initialize];

    return YES;
}

- (void)applicationDidEnterBackground:(UIApplication*)application
{

    NSError *setCategoryErr = nil;
    NSError *activationErr  = nil;
    [[AVAudioSession sharedInstance] setCategory: AVAudioSessionCategoryPlayback error:&setCategoryErr];
    [[AVAudioSession sharedInstance] setActive:YES error:&activationErr];

    ::printf("-> applicationDidEnterBackground()\n");
}

- (void)applicationWillEnterForeground:(UIApplication*)application
{
    ::printf("-> applicationWillEnterForeground()\n");

    // applicationWillEnterForeground: might sometimes arrive *before* actually initing unity (e.g. locking on startup
    if (_unityAppReady)
    {
```

Android

Unity Settings

1. Open the **Build Settings**
    a. **File -> Build Settings**
2. For *Build System*, choose **Gradle (New)**
3. Set export to **active**



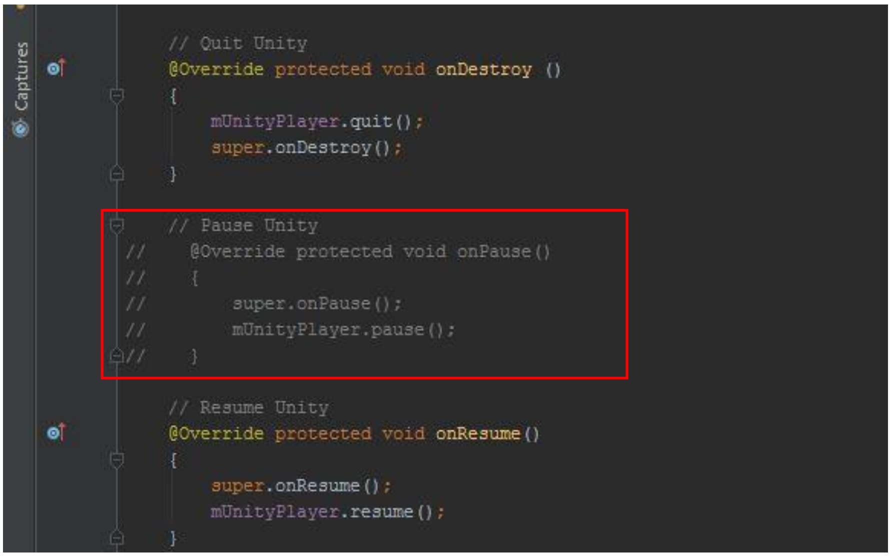
Texture Compression: Don't override
Build System: Gradle (New)
Export Project ☑
Development Build □
Autoconnect Profiler □
Script Debugging □

    a.

4. Click **Export** and choose a folder

Android Studio Settings

1. Open **Android Studio**
2. Open the project folder
3. Open the ***UnityPlayerActivity**.java* file
   Radio -> src -> main -> java -> com -> crosstales -> Radio
4. Search for "onPause"
5. Comment out the function



```
    // Quit Unity
    @Override protected void onDestroy ()
    {
        mUnityPlayer.quit();
        super.onDestroy();
    }

    // Pause Unity
//    @Override protected void onPause()
//    {
//        super.onPause();
//        mUnityPlayer.pause();
//    }

    // Resume Unity
    @Override protected void onResume()
    {
        super.onResume();
        mUnityPlayer.resume();
    }
```

Done :-)