

<http://www.idevgames.com/forum/showthread.php?p=141850>

```
BOOL      gOtherAudiolsPlaying = YES;

void CheckIfOtherAudiolsPlaying(void)
{
    UInt32  propertySize, audiolsAlreadyPlaying;
    AudioSessionInitialize(NULL, NULL, NULL, NULL);
    // do not open the track if the audio hardware is already in use (could be the iPod app playing
    music)
    propertySize = sizeof(UInt32);
    AudioSessionGetProperty(kAudioSessionProperty_OtherAudiolsPlaying,          &propertySize,
    &audiolsAlreadyPlaying);
    if (audiolsAlreadyPlaying != 0)
    {
        gOtherAudiolsPlaying = YES;
        //NSLog(@"other audio is playing");

        UInt32  sessionCategory = kAudioSessionCategory_AmbientSound;
        AudioSessionSetProperty(kAudioSessionProperty_AudioCategory,          sizeof(sessionCategory),
    &sessionCategory);
        AudioSessionSetActive(YES);
    }
    else
    {
        //NSLog(@"no other audio is playing ...");

        gOtherAudiolsPlaying = NO;

        // since no other audio is *supposedly* playing, then we will make darn sure by changing the audio
        session category temporarily
        // to kick any system remnants out of hardware (iTunes (or the iPod App, or whatever you wanna call
        it) sticks around)
        UInt32  sessionCategory = kAudioSessionCategory_MediaPlayback;
        AudioSessionSetProperty(kAudioSessionProperty_AudioCategory,          sizeof(sessionCategory),
    &sessionCategory);
        AudioSessionSetActive(YES);

        // now change back to ambient session category so our app honors the "silent switch"
        sessionCategory = kAudioSessionCategory_AmbientSound;
        AudioSessionSetProperty(kAudioSessionProperty_AudioCategory,          sizeof(sessionCategory),
    &sessionCategory);
    }
}
```