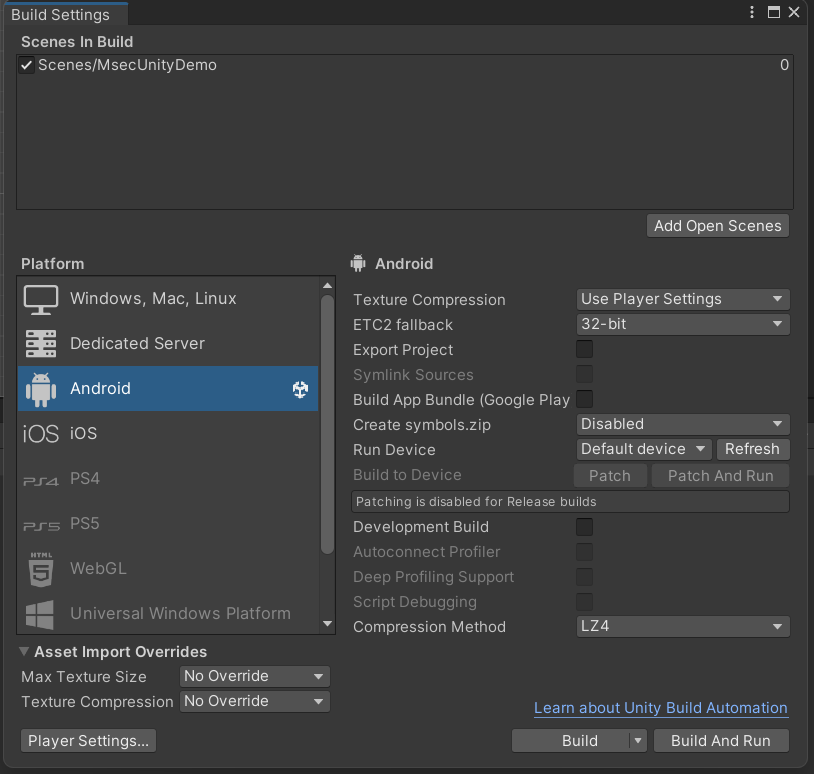
**GUIDE FOR RUNNING & BUILDING MSECUNITYDEMO**

### **Prepare the Unity Project:**

### To download the project, use the command:

### git clone <https://github.com/msecsys-dev/msecunitydemo.git>

### Ensure that your project is complete and thoroughly tested.



### 

### **For Android**

### Set Up and Configure Android Build:

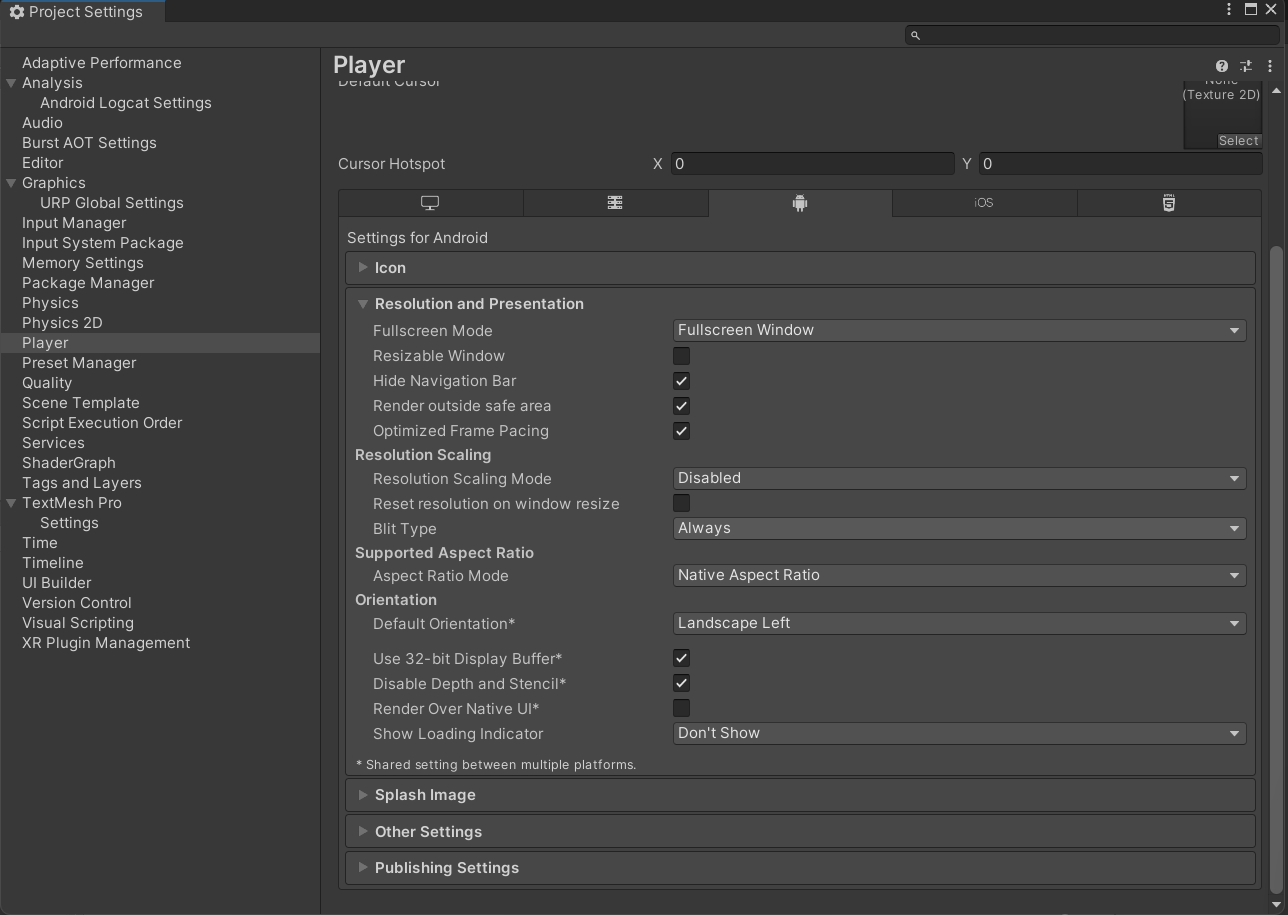
### In Unity, go to File > Build Settings.

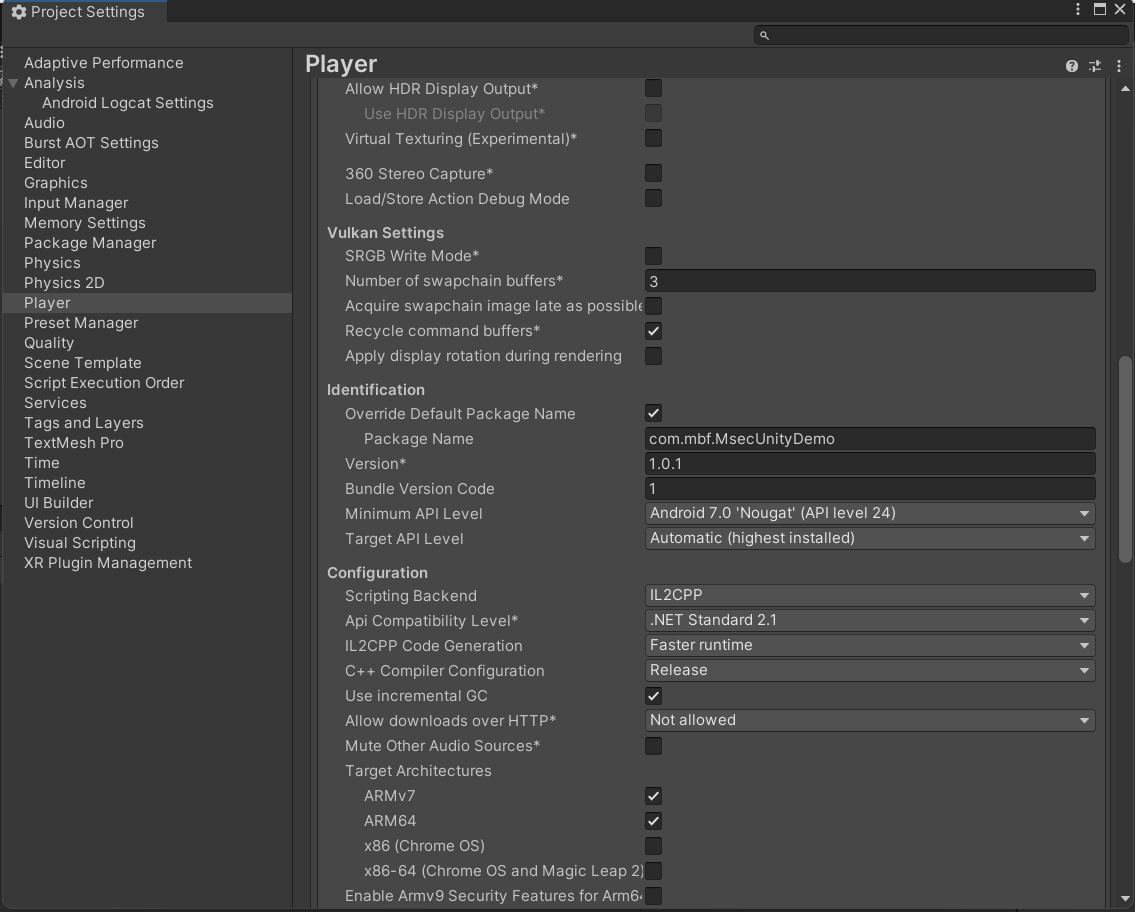
### Select Android and click Switch Platform if it hasn’t been done yet.

### Under Scenes In Build, add the following scene:

Scenes/MsecUnityDemo

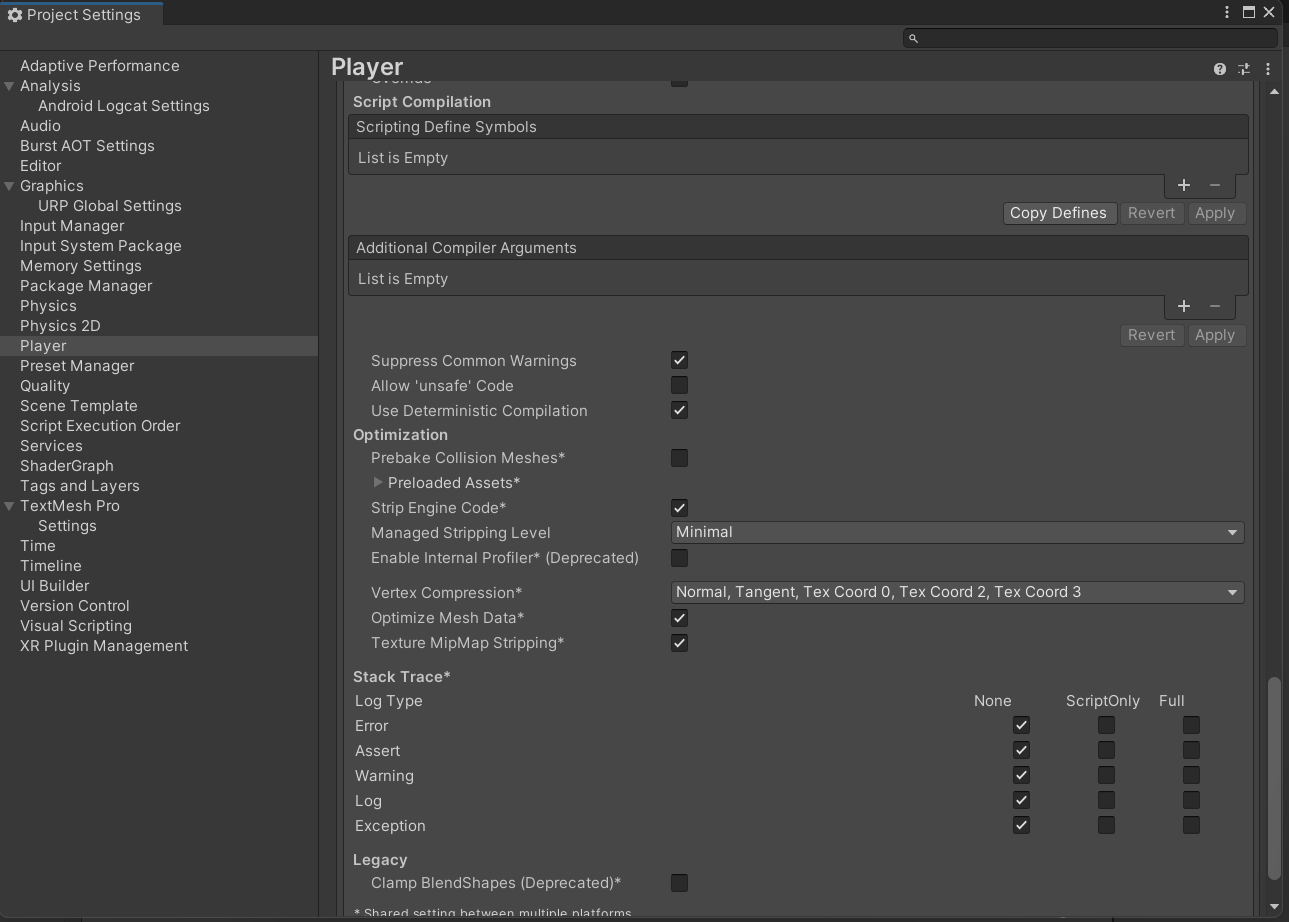
* Adjust the following settings according to your requirements in Player Settings and Resolution and Presentation:

****

****

#### Resolution and Presentation:

* Identification:
  + Package Name: Set the package name for your app in the format, e.g., com.mbf.MsecUnityDemo.
  + Version: Set the app version.
  + Bundle Version Code: This is the internal version number of the app. It must be incremented each time the app is updated.
  + Minimum API Level: Select the minimum API level that your app supports (usually Android 7.0 'Nougat' API level 24 or higher).
  + Target API Level: Typically, this is set to Automatic (highest installed) to ensure support for the latest Android version.
* Configuration:
  + Scripting Backend: Choose the scripting backend, usually IL2CPP for best performance.
  + API Compatibility Level: Generally set to .NET Standard 2.1.

****

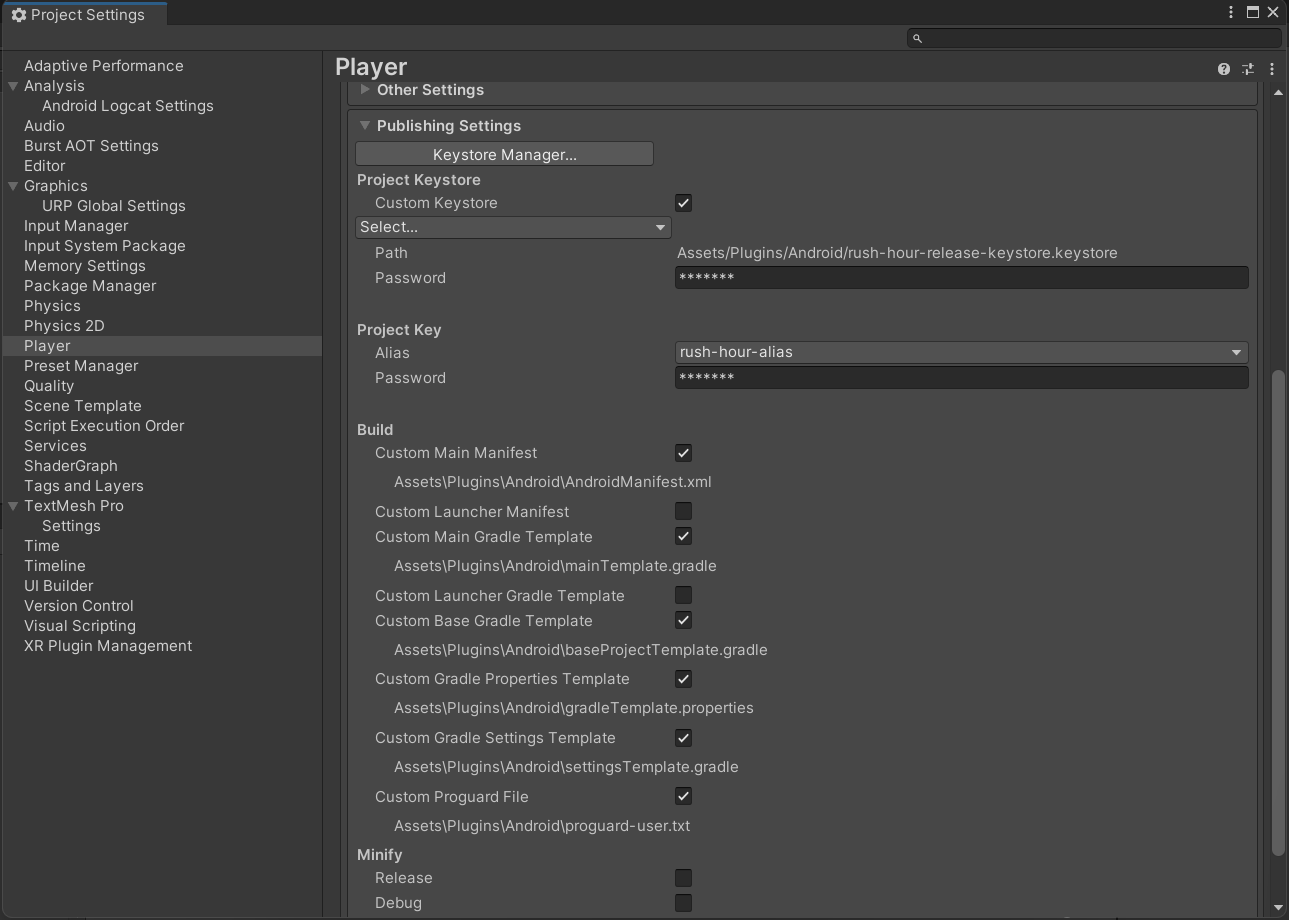
* **Optimization:**
  + Preload Shaders: If you have essential shaders, you can preload them to enhance performance.
  + Strip Engine Code: Remove unnecessary code to reduce app size.
  + Managed Stripping Level: Choose the stripping level, typically Medium or High.

### Create or Import a Keystore:

* Before publishing, you need a keystore to sign your app.
  + Go to Player Settings > Publishing Settings.
  + Create a new keystore or import an existing one.
  + Sign your app with this keystore.
  + Project Keystore:
    - Select Custom Keystore
    - Path:

Assets/Plugins/Android/msecunitykeystore.keystore

* + - Password: mbf2024
  + Project Key:
    - Alias: Select msec
    - Password: mbf2024
* In the Build settings, check the following:
  + Custom Main Manifest
  + Custom Main Gradle Template
  + Custom Base Gradle Template
  + Custom Gradle Properties Template
  + Custom Gradle Settings Template
  + Custom Proguard File



* Return to Build Settings and proceed with the build process.

**3. For iOS**

* Prepare the Unity Project for iOS:
  + Open Unity and navigate to File > Build Settings.
  + Select iOS as the target platform and click Switch Platform if not already selected.
  + Add necessary scenes under Scenes In Build (e.g., Scenes/MsecUnityDemo).

### Configure iOS Project Settings in Unity:

### Open Player Settings and go to iOS Settings.

### Set up the following configurations:

dentification:

* + - Bundle Identifier: Set the unique bundle identifier, e.g., com.mbf.MsecUnityDemo.
    - Version: Define the app version (e.g., 1.0).
    - Build Number: Increment the build number for each new release.
  + Configuration:
    - Scripting Backend: Use IL2CPP for better performance.
    - API Compatibility Level: Set to .NET Standard 2.1.
    - Architecture: Choose ARM64 for 64-bit iOS devices.
  + Optimization:
    - Strip Engine Code: Enable to reduce app size.
    - Managed Stripping Level: Set to Medium or High for further optimization.
* Other Settings:
  + Set the Target minimum iOS version to at least iOS 13.0.
  + Set the Graphics API to Metal for optimal iOS performance.

### Export the Project to Xcode:

### **In Unity, go to File > Build Settings.**

### **Click Build and select a folder to save the Xcode project.**

### **Unity will export your project to this folder. Open the generated .xcodeproj file in Xcode.**

### Set Up CocoaPods:

### Open Terminal and navigate to the directory where your Xcode project was exported by Unity.

### Run the following command to initialize CocoaPods: **pod init**

### in Terminal, run the following command to install the specified dependencies: pod install

CocoaPods will create a Unity-iPhone.xcworkspace file. Always open the project using Unity-iPhone.xcworkspace instead of Unity-iPhone.xcodeproj to ensure that CocoaPods dependencies are properly included.

### Configure the Xcode Project:

### Open Unity-iPhone.xcworkspace in Xcode.

### Ensure that all dependencies are properly linked by checking Build Phases > Link Binary With Libraries.

### Go to Signing & Capabilities and:

* + - Select your Apple Development Team.
    - Ensure the correct provisioning profile is selected for both Debug and Release builds.
* Enable any necessary capabilities (e.g., Push Notifications, In-App Purchases) based on the app's requirements.

### Build and Run on Device:

### Connect your iOS device to your computer.

### In Xcode, select your device as the target.

### Clean the build folder by clicking Product > Clean Build Folder.

### Build the project by selecting Product > Build.

### Once built, select Product > Run to install and run the app on your connected device.