

构建后处理与脚本编译符号 (Build PostProcessing & Scripting Define Symbols)

后处理：iOS

在构建后处理过程中：

- 添加框架：
 - `CoreMIDI.framework`
 - `CoreAudioKit.framework`
- 修改 `Info.plist`：
 - 添加 `NSBluetoothAlwaysUsageDescription`

后处理：Android

在构建后处理过程中：

- 修改 `AndroidManifest.xml` 并添加以下权限：
 - `android.permission.BLUETOOTH`
 - `android.permission.BLUETOOTH_ADMIN`
 - `android.permission.ACCESS_FINE_LOCATION`
 - `android.permission.BLUETOOTH_SCAN`
 - `android.permission.BLUETOOTH_CONNECT`
 - `android.permission.BLUETOOTH_ADVERTISE`
- 添加所需功能：
 - `android.hardware.bluetooth_le`
 - `android.hardware.usb.host`

Meta Quest (Oculus Quest): USB MIDI 设备检测

如果你想在 Meta Quest 设备上使用 USB MIDI，必须在后处理期间启用 USB intent filter（意图过滤器）。

在 `PostProcessBuild.cs` 中，取消注释添加 Oculus USB 意图过滤器的行：

```
// androidManifest.AddUsbIntentFilterForOculusDevices();
```

Android: 用于 BLE MIDI 的 CompanionDeviceManager

你可以使用 Android 的伴侣设备配对（Companion Device Pairing）来进行 BLE MIDI 设备连接。

启用方法：

- 添加脚本编译符号：`FEATURE_ANDROID_COMPANION_DEVICE`
- Unity 路径：`Project Settings > Player > Other Settings > Script Compilation > Scripting Define Symbols`


注意：

- Meta Quest 设备可以使用此功能来查找/连接蓝牙 MIDI 设备。
- 根据 Android 版本或行为的不同，此功能可能需要请求位置权限。

Nearby Connections MIDI (Google Nearby)

添加依赖包

在 Unity Package Manager 中：

- 点击 
- 选择 **Add package from git URL...**
- 输入以下之一：
 - `git+https://github.com/kshoji/Nearby-Connections-for-Unity`
 - (SSH 方式) `ssh://git@github.com/kshoji/Nearby-Connections-for-Unity.git`

如果已经安装，请更新到最新版本。

启用脚本编译符号

添加：

- `ENABLE_NEARBY_CONNECTIONS`

Android 项目设置

将 Target API level 设置为 33 或更高：

- `Project Settings > Player > Identification > Target API Level`

用法概览

广播 (Advertising)：

- `MidiManager.Instance.StartNearbyAdvertising()`
- `MidiManager.Instance.StopNearbyAdvertising()`

发现 (Discovering)：

- `MidiManager.Instance.StartNearbyDiscovering()`
- `MidiManager.Instance.StopNearbyDiscovering()`

连接后，发送/接收 MIDI 数据的方式与普通 MIDI 相同。

Maestro / MPTK 集成 (可选)

该插件包含一个针对 **Maestro / MidiPlayerTK (MPTK)** 的可选集成层。

要启用它，请添加脚本编译符号：

- `FEATURE_USE_MPTK`

Unity 路径：

- `Project Settings > Player > Other Settings > Script Compilation > Scripting Define Symbols`

启用后的功能：

- 基于 MPTK 的**虚拟 MIDI 输出**设备（将 `MidiManager` 发送的消息路由到 MPTK 合成器）。
- 一个适配器，可以将 MPTK 播放器作为**虚拟 MIDI 输入源**（将事件注入到 `MidiManager`）。

注意：

- 仅当项目中存在 MPTK 资产时才启用此符号；否则，由于缺少 MPTK 类型，编译将失败。
- 另请参阅：[Maestro / MPTK 集成](#)