Build PostProcessing & Scripting Define Symbols

PostProcessing: iOS

During build postprocess:

- Adds frameworks:
  - `CoreMIDI.framework`
  - `CoreAudioKit.framework`
- Adjusts `Info.plist`:
  - adds `NSBluetoothAlwaysUsageDescription`

PostProcessing: Android

During build postprocess:

- Adjusts `AndroidManifest.xml` with permissions:
  - `android.permission.BLUETOOTH`
  - `android.permission.BLUETOOTH_ADMIN`
  - `android.permission.ACCESS_FINE_LOCATION`
  - `android.permission.BLUETOOTH_SCAN`
  - `android.permission.BLUETOOTH_CONNECT`
  - `android.permission.BLUETOOTH_ADVERTISE`
- Adds required features:
  - `android.hardware.bluetooth_le`
  - `android.hardware.usb.host`

Meta Quest (Oculus Quest): USB MIDI device detection

If you want USB MIDI on Meta Quest devices, you must enable the USB intent filter during postprocess.

In `PostProcessBuild.cs`, uncomment the line that adds the Oculus USB intent filter:

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

Android: CompanionDeviceManager for BLE MIDI

You can use Android's Companion Device Pairing for BLE MIDI device connection.

To enable:

- Add scripting define symbol: `FEATURE_ANDROID_COMPANION_DEVICE`
- Unity path: `Project Settings > Player > Other Settings > Script Compilation > Scripting Define Symbols`

Notes:

- Meta Quest devices can use this feature to find/connect Bluetooth MIDI devices.
- This feature may require requesting location permission depending on Android version/behavior.

Nearby Connections MIDI (Google Nearby)

Add dependency package

In Unity Package Manager:

- click `+`
- choose **Add package from git URL...**
- enter one of:
  - `git+https://github.com/kshoji/Nearby-Connections-for-Unity`
  - (SSH alternative) `ssh://git@github.com/kshoji/Nearby-Connections-for-Unity.git`

If already installed, update to latest.

Enable scripting define symbol

Add:

- `ENABLE_NEARBY_CONNECTIONS`

Android project setting

Set Target API level to 33 or higher:

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

Usage overview

Advertising:

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

Discovering:

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

Sending/receiving MIDI data is the same as normal MIDI once connected.

Maestro / MPTK integration (optional)

The plugin includes an optional integration layer for **Maestro / MidiPlayerTK (MPTK)**.

To enable it, add the scripting define symbol:

- `FEATURE_USE_MPTK`

Unity path:

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

What it enables:

- An MPTK-backed **virtual MIDI output** device (route `MidiManager` sends into an MPTK synth).
- An adapter that can treat MPTK players as a **virtual MIDI input** source (injecting events into `MidiManager`).

Notes:

- Only enable this symbol when the MPTK asset is present in the project; otherwise compilation will fail due to missing MPTK types.
- See also: Maestro / MPTK Integration