ビルド後の処理 (PostProcessing) とスクリプト定義シンボル

PostProcessing: iOS

ビルド後の処理中に行われる内容:

- 以下のフレームワークを追加:
  - `CoreMIDI.framework`
  - `CoreAudioKit.framework`
- `Info.plist` を調整:
  - `NSBluetoothAlwaysUsageDescription` を追加

PostProcessing: Android

ビルド後の処理中に行われる内容:

- `AndroidManifest.xml` に以下の権限を追加:
  - `android.permission.BLUETOOTH`
  - `android.permission.BLUETOOTH_ADMIN`
  - `android.permission.ACCESS_FINE_LOCATION`
  - `android.permission.BLUETOOTH_SCAN`
  - `android.permission.BLUETOOTH_CONNECT`
  - `android.permission.BLUETOOTH_ADVERTISE`
- 必要な機能 (features) を追加:
  - `android.hardware.bluetooth_le`
  - `android.hardware.usb.host`

Meta Quest (Oculus Quest): USB MIDI デバイスの検出

Meta Quest デバイスで USB MIDI を使用する場合は、ビルド後の処理で USB インテントフィルタを有効にする必要があります。

`PostProcessBuild.cs` 内の、Oculus 用 USB インテントフィルタを追加する行のコメントアウトを解除してください:

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

Android: BLE MIDI 用の CompanionDeviceManager

BLE MIDI デバイスの接続に Android の Companion Device Pairing を使用できます。

有効にする方法:

- スクリプト定義シンボルを追加: `FEATURE_ANDROID_COMPANION_DEVICE`
- Unity でのパス: `Project Settings > Player > Other Settings > Script Compilation > Scripting Define Symbols`

注意点:

- Meta Quest デバイスで Bluetooth MIDI デバイスを検索・接続するためにこの機能を使用できます。
- Android のバージョンや挙動によっては、位置情報の許可リクエストが必要になる場合があります。

Nearby Connections MIDI (Google Nearby)

依存パッケージの追加

Unity Package Manager で:

- `+` をクリック
- **Add package from git URL...** を選択
- 以下のいずれかを入力:
  - `git+https://github.com/kshoji/Nearby-Connections-for-Unity`
  - (SSH の場合) `ssh://git@github.com/kshoji/Nearby-Connections-for-Unity.git`

既にインストール済みの場合は、最新にアップデートしてください。

スクリプト定義シンボルの有効化

以下を追加:

- `ENABLE_NEARBY_CONNECTIONS`

Android プロジェクト設定

Target API level を 33 以上に設定してください:

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

使用方法の概要

アドバタイズ (公開):

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

ディスカバリ (検索):

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

接続後は、通常の MIDI データと同じ方法で送受信できます。

Maestro / MPTK 統合 (オプション)

このプラグインには、**Maestro / MidiPlayerTK (MPTK)** 用のオプションの統合レイヤーが含まれています。

有効にするには、以下のスクリプト定義シンボルを追加してください:

- `FEATURE_USE_MPTK`

Unity でのパス:

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

このシンボルによって有効になる機能:

- MPTK をバックエンドとする **仮想 MIDI 出力** デバイス (`MidiManager` の送信を MPTK シンセにルーティング)。
- MPTK プレイヤーを **仮想 MIDI 入力** ソースとして扱うアダプター (`MidiManager` にイベントを注入)。

注意点:

- プロジェクトに MPTK アセットが存在する場合のみ、このシンボルを有効にしてください。存在しない場合、MPTK の型が見つからずコンパイルエラーになります。
- 詳細は Maestro / MPTK 統合 を参照してください。