

# Maestro / MPTK 統合 (仮想デバイス + アダプター)

---

このページでは、オプションの **Maestro / MidiPlayerTK (MPTK)** 統合レイヤーについて説明します。

この統合により、主に 2 つのワークフローが提供されます:

## 1. 仮想 MIDI 出力デバイス (シンク) としての MPTK

**MidiManager** のイベントを MPTK シンセにルーティングします（リアルタイム演奏）。

## 2. 仮想 MIDI 入力デバイス (ソース) としての MPTK

MPTK のコールバック (**MidiFilePlayer** や **MidiStreamPlayer** からのもの) を受け取り、あたかもデバイスから届いたかのように **MidiManager** に MIDI イベントを **注入** します。

用語解説: このプラグインにおける「仮想デバイス (virtual device)」とは、プラットフォーム標準の MIDI バックエンドを介さずに、**MidiManager** のデバイスリストやイベントパイプラインに参加するソフトウェア上のエンドポイントを指します。

## 有効化 / 無効化 (コンパイル時)

この統合機能は、以下のスクリプト定義シンボルによって制御されます:

- `FEATURE_USE_MPTK`

この定義が **存在しない** 場合（または MPTK がインストールされていない場合）、統合スクリプトは空のスクリプトとしてコンパイルされるため、プロジェクト全体のビルドは維持されます。

### 有効にする方法

Unity で以下の設定を行います:

- **Project Settings** → **Player** → **Other Settings** → **Script Compilation** → **Scripting Define Symbols**
- `FEATURE_USE_MPTK` を追加します。

# 含まれているもの

## 仮想デバイスの登録と入力注入

**MidiManager** には、ソフトウェアエンドポイント用のヘルパー API が用意されています:

- 仮想デバイス ID の登録/解除（入力用、出力用、またはその両方として）。
- MIDI イベントを内部パイプラインに直接注入（「このデバイスが Note On を送信したことにする」など）。

デバイスをシミュレートしたり、他のシステムと橋渡ししたり、ハンドラのユニットテストを行いたい場合に使用します。

## MPTK 仮想 MIDI デバイス (シンク)

**MptkMidiDevice** は MIDI イベントハンドラインターフェースを実装しており、イベントを内部の MPTK **MidiStreamPlayer** に転送します。

以下のような場合に使用します:

- **MidiManager** → MPTK シンセでの再生。
- 実際のデバイスと並んで表示される「ソフトウェア出力デバイス ID」として利用。

## MPTK → MidiManager アダプター (ソース)

**MptkMidiManagerInputAdapter** は MPTK のコールバックイベント（MPTK プレイヤーから公開されるもの）を購読し、対応する MIDI 1.0 形式のイベントを **MidiManager** に注入できます。

以下のような場合に使用します:

- MPTK の再生やリアルタイム生成内容によって、既存の **MidiManager** ハンドラを動作させる。
- MPTK を仮想的な 入力 デバイスのように振る舞わせる。

## 便利なユーティリティ

**MptkUtility** は一般的なセットアップをラップしています:

- MPTK グローバル設定の存在確認。
- MPTK プレイヤーの作成。
- 仮想デバイスの作成と登録。
- アダプターの作成。
- MIDI イベントを仮想入力として注入するためのヘルパーセット。

## サンプルコード (即座に実行可能)

以下のサンプルが `Assets/MIDI/Samples/DocumentationExamples/` に用意されています:

- **MIDI 1.0 → MPTK (仮想出力シンク)**

`Assets/MIDI/Samples/DocumentationExamples/MptkVirtualOutputSinkExample.cs`

- **MPTK → MIDI 1.0 (仮想入力として MidiManager に注入)**

`Assets/MIDI/Samples/DocumentationExamples/MptkToMidiManagerInputExample.cs`

## サンプルシーン (統合済み)

同梱のサンプルシーンにも、オプションで MPTK 統合が含まれています:

- MIDI 1.0 サンプルシーンスクリプト:

[Assets/MIDI/Samples/Scripts/MidiSampleScene.cs](#)

(MPTK バックエンドの仮想出力デバイスを使用するトグルを追加します)

- MIDI 2.0 サンプルシーンスクリプト:

[Assets/MIDI/Samples/Scripts/Midi2SampleScene.cs](#)

(MIDI 2.0 の送信内容を MPTK MIDI 1.0 仮想シンクに ミラーリング するトグルを追加します)

## ID、グループ、ルーティングの選択

### 推奨される慣例:

- 仮想デバイスであることが明確なプレフィックスを使用してください（例: `mptk:internal`, `virtual:sequencer`, `test:device`）。
- 意図的に複数のグループをモデル化する場合を除き、`group = 0` を使用してください。

仮想デバイスはデバイスセット内に表示されるため、ハードウェアデバイスと同じようにユーザーが UI から選択できる仕組みを構築できます。

## トラブルシューティング

「エディタではコンパイルできるが、CI や他のマシンで失敗する」

- `FEATURE_USE_MPTK` は、実際に MPTK アセットが存在する場合にのみ有効にしてください。

「イベントが受信されない」

- 注入されたイベントが入力デバイスからのものとして認識されるよう、仮想デバイスが **入力 (input)** として登録されているか確認してください。
- ハンドラが `MidiManager` に登録されているか確認してください。
- `deviceId` と `group` が期待通りであることを確認してください。

「MPTK シンクから音が鳴らない」

- 仮想デバイスが **出力 (output)** として登録されているか確認してください。
- シンクの `deviceId` が、送信先のデバイス ID と一致しているか確認してください。
- プロジェクト内の MPTK グローバル設定やリソース (SoundFont/構成) が有効であることを確認してください。

## 関連ドキュメント

- [MIDI 1.0 \(MidiManager\)](#)
- [サンプル](#)