

audio**kinetic**

# Cubeインテグ レーション



# Cubeインテグレーション

Wwise

Copyright © 2025 Audiokinetic Inc. All rights reserved.

This document (whether in written, graphic or video form) is supplied as a guide for the Wwise® product. This documentation is the property of Audiokinetic Inc. ("Audiokinetic"), and protected by Canadian copyright law and in other jurisdictions by virtue of international copyright treaties.

This documentation may be duplicated, reproduced, stored or transmitted, exclusively for your internal, non-commercial purposes, but you may not alter the content of any portion of the documentation. Any copy of the documentation shall retain all copyright and other proprietary notices contained therein.

The content of the Cube Integration documentation is furnished for information purposes only, and its content is subject to change without notice. Reasonable care has been taken in preparing the information contained in Cube Integration, however, we disclaim all representations, warranties and conditions, whether express, implied or arising out of usage of trade or course of dealing, concerning this documentation and assume no responsibility or liability for any losses or damages of any kind arising out of the use of this guide or of any error or inaccuracy it may contain, even if we have been advised of the possibility of such loss or damage.

Wwise®, Audiokinetic®, Actor-Mixer®, SoundFrame® and SoundSeed® are registered trademarks, and Master-Mixer™, SoundCaster™ and Randomizer™ are trademarks, of Audiokinetic. Other trademarks, trade names or company names referenced herein may be the property of their respective owners.

---

# Table of Contents

1. Wwise Cubeインテグレーションについて .....	1
2. Cube Demoのインストールと実行 .....	2
3. Cubeへのサウンドエンジンの統合を理解する .....	7
4. Wwiseプロジェクトを理解する .....	9
5. サウンドデザインを変更する .....	12
6. ヘルプを利用する .....	14

---

# Chapter 1. Wwise Cubeインテグレーションについて

Audiokineticは、オープンソースのファーストパーソンゲームCubeから暴力を排除したバージョンに、Wwiseサウンドエンジンを統合しました。このバージョンにはWwise Adventure GameサンプルのWwise Wwizardが登場します。

Cube Integrationは、Wwiseサウンドエンジンをゲームに統合する方法について理解を深め、Wwiseを試せる実用的なゲームをサウンドデザイナーに提供するために作られました。ゲーム例としてCubeが選ばれたのは、構造がシンプルで、必要に応じて比較的簡単に変更できるためです。

このドキュメントは、以下の主要セクションに分かれています:

- **Chapter 2, Cube Demoのインストールと実行** #Mac®およびWindowsにCube Demoをインストールし、実行する方法を、Audiokinetic Launcherを使用する場合と使用しない場合で紹介します。
- **Chapter 3, Cubeへのサウンドエンジンの統合を理解する** #どのようにしてWwiseサウンドエンジンをCubeに統合したかを紹介します。
- **Chapter 4, Wwiseプロジェクトを理解する** #Wwiseプロジェクトにおける設計上の決定について説明します。
- **Chapter 5, サウンドデザインを変更する** #サウンドプロパティをリアルタイムで微調整し、新しく生成したSoundBankを使用してそれらの変更を統合する方法を紹介します。

---

## Chapter 2. Cube Demoのインストールと実行

Cube Demoパッケージには以下のコンポーネントが含まれます:

- AudiokineticのCubeゲームのデバッグ、プロファイル、リリースの各バージョン。
- ゲームの変更とサウンドの追加に使用されたWwiseプロジェクト。
- Cubeゲームエンジンソースコードとドキュメント。



### Note

コンテンツプロバイダーから提供された素材は元々すべて高品質（48 kHz、16または24ビット）でした。ダウンロード時間とディスクスペースを節約するため、一部のサウンドアセットのサイズを縮小しました。

Cube Demoのインストールと実行の詳細については、以下のセクションをご参照ください。

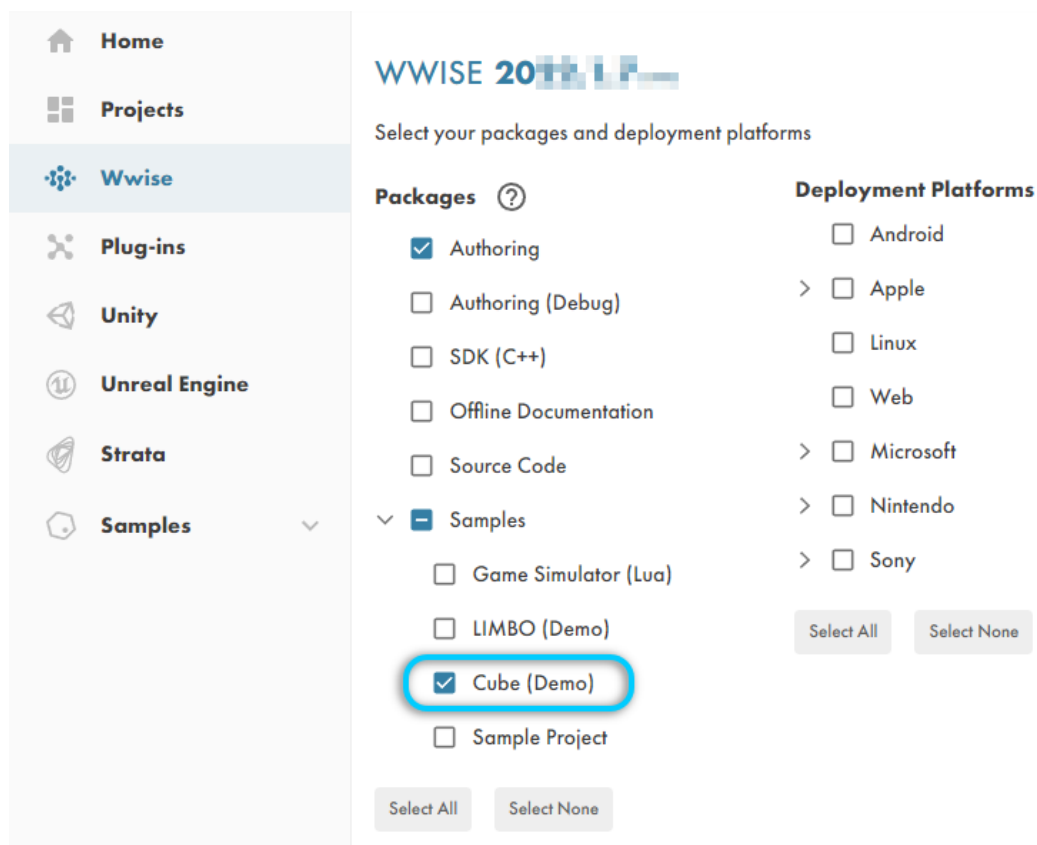
## Cube Demoをインストールする

Cube DemoをAudiokinetic Launcherを使用してインストールします。

### Cube Demoをインストールするには:

- [Wwiseのインストール](#)時（または既存のインストールの変更時）に、**Samples**メニューを展開し、**Cube (Demo)**を選択します。

既存のWwiseインストールを変更するには、LauncherのWwiseページで、インストールの右にあるレンチのボタンを選択して**Modify**をクリックします。



## LauncherからCube Demoを実行する

Cube Demoをインストールすると、LauncherのSamples > Wwise Samplesページに表示されます。

### Cube Demoを実行するには:

1. Samples > Wwise Samplesページで、Cube DemoをインストールしたWwiseバージョンを選択します。
2. **Run Cube**をクリックするとゲームが開始されます。



## Launcherを使用せずにCube Demoを実行する

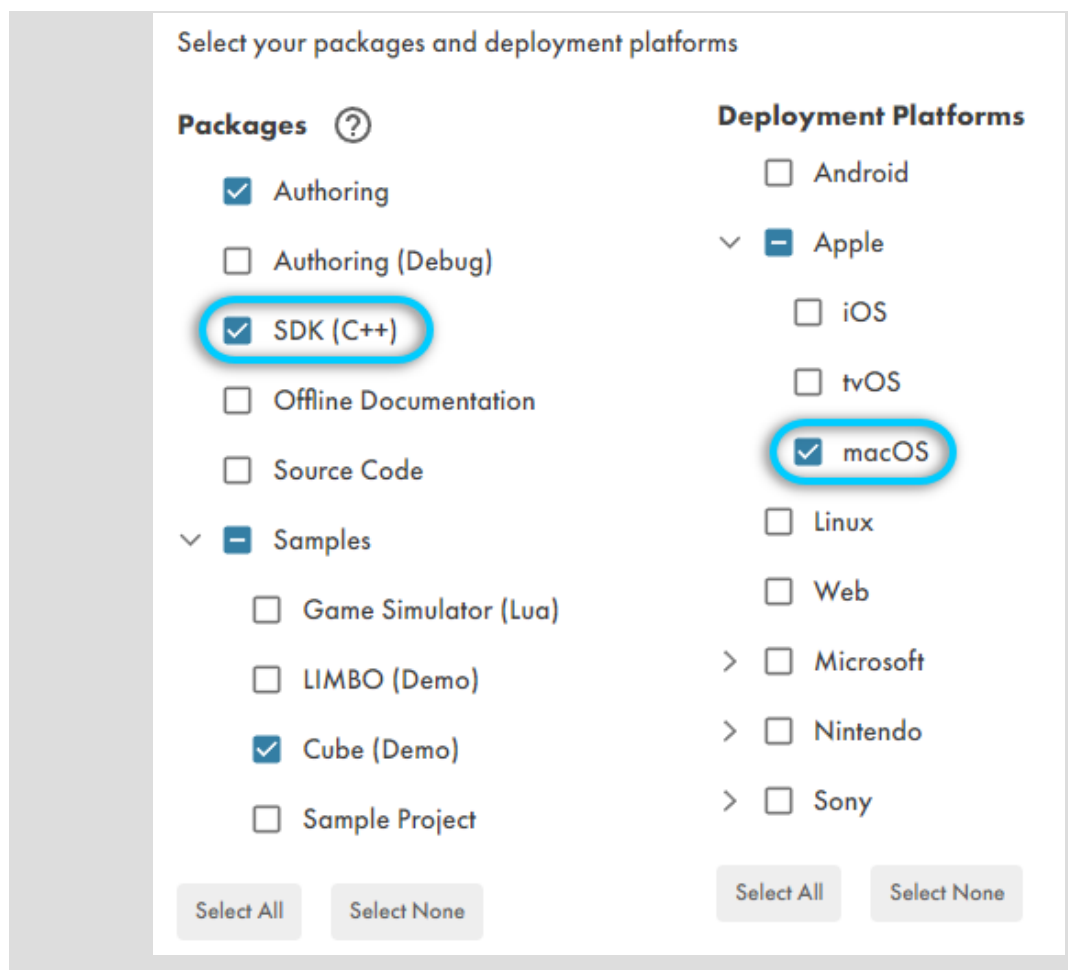
さまざまな理由で、Cube DemoをLauncherを使用せずに実行したい場合もあります。代替の方法はいくつかあり、MacとWindowsのどちらで実行するかに応じて多少の違いがあります。

### Mac OS Xの場合



#### Note

始める前に、Mac用のWwise SDKをインストールしておく必要があるため、Wwiseインストール時にSDK (C++)とmacOSを選択します。これらのオプションを既存のWwiseインストールに追加するには、LauncherのWwiseページで、インストールの右にあるレンチのボタンを選択してModifyをクリックします。



1. 方法1: シェルスクリプトを使用する。
  - a. 圧縮されていないCube DemoフォルダとインストールされたWwise SDK Macの配布が、同じ親フォルダを共有していることを確認します。例えばWwise SDKフォルダが`${HOME}/Wwise/wwise_${version_tag}/SDK`であれば、Cube Demoのルートフォルダは`${HOME}/Wwise/wwise_${version_tag}/CubeDemo`とする必要があります。
  - b. 実行可能なシェルスクリプトは`$Applications/Audiokinetic/<Wwise_version>/Cube/Mac/<mode>/bin/RunCubeDemo.sh`にあり、「<mode>」は該当するDebug、Profile、Releaseのいずれかのモードフォルダに対応します。
  - c. .shファイルがTerminalで開くように設定されている場合はFinderでダブルクリックします。またはTerminalをシェルスクリプトのディレクトリで開き、`RunCubeDemo.sh`と入力して**Enter**を押します。
2. 方法2: Xcode下で実行する。
  - a. Xcodeがインストールされていることを確認します。XcodeがApplicationsフォルダにない場合はMac App Storeからダウンロードしてインストールします。



- b. 展開したCube Demoフォルダが、インストールされたWwise SDK macOS ディストリビューションと同じ親フォルダを共有していることを確認します。例えばWwise SDKフォルダが`${HOME}/Wwise/wwise_${version_tag}/SDK`であれば、Cube Demoのルートフォルダは`${HOME}/Wwise/wwise_${version_tag}/CubeDemo`とする必要があります。
- c. Xcodeで以下のXcodeプロジェクトを開きます: `/path/to/CubeDemo/cube_source/src/Mac/CubeMac.xcodeproj`。
- d. プロジェクトをビルドして実行します。

## Windowsの場合

1. 方法1: BATスクリプトを使用する。
  - a. Audiokinetic Launcherを使用してCube Demoをインストールした際、BATファイルが"`<WwiseRoot>\Cube\cube`"フォルダ内に作成されました。このフォルダをFile Explorerで開きます。
  - b. 実行するCube Demoバージョン（デバッグ、プロファイル、リリースのいずれかのモード）のBATファイルをダブルクリックします。またはCommand Promptウィンドウを開き、BATファイルへのパスを入力してEnterを押します。
2. 方法2: Command Promptから実行ファイルを実行する。
  - a. 実行ファイル「`cube.exe`」に移動し、実行します。「`<WwiseRoot>\Cube\Win32\<mode>\bin`」下であり、「`<mode>`」は該当するDebug、Profile、Releaseのいずれかのモードフォルダに対応します。

---

## Chapter 3. Cubeへのサウンドエンジンの統合を理解する

CubeへのWwiseサウンドエンジンの統合は以下のタスクを伴います:

- Wwise用にCubeを準備する
- ゲームオブジェクトを登録する
- オーディオを統合する
- 新しいスクリプトコマンドを追加する

### Wwise用にCubeを準備する

Cubeは、元々Wouter van Oortmerssen氏によって開発されたオープンソースのシングルまたはマルチプレイヤー対応ファーストパーソンシューティングゲームで、Cubeのウェブサイト (<http://cube.sourceforge.net/>) からすぐに入手できます。Wwiseの統合に適応させるため、Cubeウェブサイトからダウンロードしたバージョンに以下の変更を加えました:

- 既存のサウンドコードを書き直し、Wwiseサウンドエンジンを使用するようにしました。
- オーディオコンテンツをWwiseプロジェクトに移動しました。
- ほかのライブラリ (SDL、SDL\_image、libpng) への依存を排除しました。
- オーディオ (足音など) を追加しました。
- 上記の変更に対応させるため、一部のゲーム設定ファイルとコンテンツファイルを変更しました。

### ゲームオブジェクトを登録する

2種類のゲームオブジェクトをサウンドエンジンに登録しました:

- baseent構造体へのポインタ。これらは、ゲームワールドを移動するエンティティ (プレイヤー、モンスター、発射物) です。
- ダミーのゲームオブジェクトで、64から127の番号が付いています。これらは、移動するエンティティに関連付けられていない3Dサウンド (生成されるアイテムなど) に使用します。

### オーディオを統合する

主要なオーディオインテグレーションコードは、<WwiseRoot>\Cube\cube\_source\srcフォルダにある以下のファイル内にあります:

- Sound.cpp#サウンドエンジンにアクセスするすべてのコードが含まれます。
- Protos.hのSoundのセクション#宣言されたパブリック関数が含まれます。

統合処理で影響を受けたのは以下のサウンドタイプです:

- モンスター再生サウンド:

- 各キャラクターに関連する苦痛イベント、死イベント、足音イベントがあります。（`monster.cpp`の`monstertypes[]`をご参照ください。）
- 足音イベントは`monsterfootstep()`関数からポストされます。この関数はキャラクターの下にある素材に応じて「Material」Switchも設定し、足音のSwitch Containerを動作させます。
- アイテムには関連する入手イベントがあります。（`entities.cpp`の`itemstats[]`をご参照ください。）
- **魔法サウンド:**
  - 各魔法タイプに関連するイベントがあります。（`weapon.cpp`の`guns[]`をご参照ください。）
  - 発射物にはスプラッシュイベント（`S_FEXPLODE`、`S_RLHIT`）があります。
  - クワッドダメージが有効な時に氷のジェムを発射するとイベントの`S_ITEMPUP`がポストされます。
- **ネットワークサウンド:**
  - 一部のイベントはサーバに送りネットワークで再生する必要があります。これは`snd_clientevent()`で対応します。
- **その他のサウンド:**
  - ゲームプレイ中に発生するその他のサウンドイベントを見つけるには、コードベースで`snd_event()`コールを検索します。

## 新しいスクリプトコマンドを追加する

以下のサウンド関連のコマンドを新しく作成したので、サウンド管理に活用してください。これらは`.cfg`スクリプトファイルで使用できます:

- `akevent <string>`: ローカルプレイヤーゲームオブジェクト（`player1`）に対してイベントをポストします（イベント名を使用）。
- `soundvol <int>`: サウンドボリュームを設定します（0-255）。
- `musicvol <int>`: ミュージックボリュームを設定します（0-255）。
- `voicevol <int>`: ボイスボリュームを設定します（0-255）。
- `texturematerial <int> <string1> <string2>`: 本質的に「texture」と同じコマンドですが、関連する足音用の素材のパラメータが追加されています。

---

## Chapter 4. Wwiseプロジェクトを理解する

WwiseサウンドエンジンをCubeに統合しやすくするため、Wwiseプロジェクトを作成しました。CubeのAudiokineticバージョンのサウンドの多くは、オリジナルのゲームから変更されています。



### Note

**Wwiseプロジェクトファイルは以下のフォルダにあります:** <WwiseRoot>\Cube  
\WwiseProject。

以下のセクションで、Wwiseプロジェクトにおける設計上の決定をいくつか説明します。

### Actor-Mixer Hierarchyを構築する

Actor-Mixer Hierarchyは、プロジェクト内のサウンドアセットを整理する場所です。

Cubeプロジェクト用に以下のオーディオ構造を作成しました:

- 「Main」フォルダ内に4つのActor-Mixerを作成し、サウンドを論理カテゴリにまとめました: Items、Magic、Main Characters、Monsters。
- Actor-Mixer内でさまざまなRandom ContainerやSwitch Containerを採用することで、プロジェクトのサウンドに変化を持たせたり、固有のサウンドに仕上げたりしました。
- 一部の魔法と足音のボリュームプロパティとピッチプロパティにランダムマイザを適用しました。ランダムマイザは、オブジェクトが再生されるたびにプロパティ値をランダムに変化させます。
- 「Main」フォルダ内のすべてのサウンドが3Dサウンドで、ゲーム定義のポジショニングを使用し、距離に基づく標準的なアテニュエーション（減衰）がボリュームに適用されます。
- 「Maps」フォルダ内に、マップ「dcp\_the\_core」に関連するサウンド、ボイス、ミュージックがあります。このマップ内のボイスとミュージックはすべて2Dサウンドで、ハードドライブからストリーミングされます。

### Master-Mixer Hierarchyを構築する

Master-Mixer Hierarchyは、バスやAUXバス（補助バス）からなる別の階層構造で、プロジェクト内のさまざまなサウンド構造やミュージック構造をまとめ、出力に備えることができます。

Cubeプロジェクト用に以下のバス構造を作成しました:

- 3つの主要なコントロールバスを作成しました: Music、SFX、Voice。
- Environmentsバスに一連のAUXバスをまとめました。各AUXバスに、ゲーム内の部屋を表すリバーブエフェクトが挿入されています。どのAUXバスがランタイムに処理されるかは、ゲームのコードで決定されます。

- オートダッキングをVoiceバスに適用したので、Voiceの再生中はMusicがダッキングされます。
- RTPC（リアルタイムパラメータコントロール）を、3つの主要なコントロールバスのVolumeプロパティに適用しました。これらのボリュームコントロールは、ゲーム内のボリュームフェーダ（ゲームコンソールを使用）にマッピングしました。

## ゲームシンクを使用する

Wwiseでは、ゲームシンクを使用し、ゲーム内のアクションや条件の変化に関連する特定のオーディオ変化を効率的に処理します。Cubeプロジェクトで以下のゲームシンクを使用しました：

- **スイッチ** - ゲーム内に存在するさまざまな地面素材を管理するために「Material」というスイッチグループを作成しました。このグループ内に、コンクリートや草など、各地面素材用のスイッチを作成しました。これらのスイッチは、ゲーム中の地面の変化に合わせて、メインキャラクター用に異なる足音サウンドを定義するために使用されます。
- **ゲームパラメータ** - Cubeゲーム内のボリュームフェーダに対応する3つのゲームパラメータを作成しました。これらのゲームパラメータを、RTPCを使用して、3つの主要なバスのボリュームプロパティにマッピングしました。フェーダをボリュームプロパティにマッピングすることで、ゲームプレイヤーはゲーム内のサウンドとミュージックのボリュームを自分でコントロールできるようになります。

ゲームパラメータ「PlayerHealth」は、ゲーム中のプレイヤーの体力ゲージに関連付けられています。これをデフォルトで使用するサウンドはありません。このゲームパラメータは、サウンドオブジェクトのプロパティに関連付けられます。

TeleportのループするSynth Oneサウンドでは、ゲームパラメータ「Distance\_to\_Object」をシンセの「Base Frequency」パラメータに対して使用し、テレポーターに近づくに従い面白いピッチベンド効果を生み出します。このゲームパラメータは、ほかのサウンドオブジェクトに対して再利用できます。

## イベントを作成する

Wwiseではイベントを使用してゲーム内のサウンド、ミュージック、ダイアログを動作させます。イベントには1つ以上のアクションが含まれ、プロジェクト階層にあるさまざまなサウンド構造やミュージック構造に適用されます。Cubeプロジェクト用にイベントを作成し、Cubeがトリガーするサウンド名に合致する名前を付けました。

## SoundBankを理解する

SoundBankはそれまでの作業の成果物であり、ゲームの一部となる最終的なオーディオパッケージです。ゲーム内の特定の時点でゲームのプラットフォームメモリにロードされるイベント、Wwiseオブジェクト、変換されたオーディオファイルのグループが含まれます。Cubeプロジェクト用に以下の2つのSoundBankを作成しました：

- **Main** - すべてのマップで使用される、すべてのイベントのグループです。
- **dcp\_the\_core** - マップ「dcp\_the\_core」用のイベントのグループです。

---

## Chapter 5. サウンドデザインを変更する

以下のセクションで、オーディオをリアルタイムで微調整し、それらの変更をゲームに統合する方法を説明します。

### ゲーム中にオーディオをリアルタイムで微調整する

Cubeゲームに接続後、ゲームをプレイしながらリアルタイムで各サウンドのプロパティを変更し、オーディオを微調整できます。



#### Note

Wwise Authoringアプリケーションをゲームに接続する方法については、Wwiseヘルプの[ローカルまたはリモートで、ゲーム機に接続する](#)をご参照ください。

サウンドプロパティをリアルタイムで変更するには（Wwiseをゲームに接続後）：

1. Cubeで、左クリックして氷のジェムを発射し、現在のサウンドを聞く。
2. Wwiseで、IceGem\_Blastサウンドオブジェクトを選択し、Property EditorとTransport Controlにロードする。

IceGem\_Blastサウンドオブジェクトは、MainフォルダのMagic Actor-Mixer下にある。

3. IceGem\_Blastサウンドのボリュームとピッチのプロパティを変更する。
4. Playをクリックして新しいサウンドを聞く。
5. Cubeに戻り、もう一度氷のジェムを発射する。

新しいサウンドが自動的にゲーム内で再生される。

6. すばやく簡単にゲームオーディオを微調整できるので、ゲーム内のほかのサウンドでも試してみてください。

### 変更をゲームに統合する

ゲームに接続中に加えた変更を聞くことはできますが、これらは一時的な変更にすぎません。変更をゲームに統合するには、SoundBankを再生成して既存のものを置き換える必要があります。SoundBankを置き換えると、新しいサウンドがCubeゲームに統合されます。



#### Note

必ずCubeを閉じてから新しいSoundBankを生成してください。ゲームの実行中は、1つ以上のSoundBankがゲームによってロックされている可能性があるため、ディスク上のSoundBankを正常に上書きできない場合があります。

Cube用に新しいSoundBankを生成するには：

1. 以下のいずれかを行い、SoundBankレイアウトに切り替える。
  - メニューバーからLayouts > SoundBankをクリックする。
  - F7を押す。

このプロジェクト用に2つのSoundBankがある: DCP\_the\_core、Main。

2. SoundBanksリストで**Select All**をクリックする。
3. Platformsリストで、作業しているオペレーティングシステムに応じて**Windows**または**Mac**のオプションを選択する。
4. Languagesリストで**English (US)**のオプションを選択する。
5. メニューバーで、**Project > Project Settings**をクリックする。

Project Settingsダイアログボックスが開く。

6. **SoundBanks**タブに切り替える。
7. Post-Generation Stepグループボックスで、プロジェクトのストリーミングファイルをSoundBankの保存場所にコピーするためのコマンドラインがあることを確認する。デフォルトで、すべてのプロジェクトにこのコマンドラインが生成後のステップとして含まれる。
8. **OK**をクリックして、Project Settingsダイアログボックスを閉じる。
9. **Generate Checked**をクリックするとSoundBanksの生成が開始される。

Generating SoundBanksダイアログボックスが開き、SoundBank生成プロセスの進行状況が表示される。SoundBankの生成が完了すると、SoundBanks Generation - Completedダイアログボックスが開く。

- 10 **Close**をクリックする。

新しいSoundBankが生成され、Cubeゲームに統合されました。次にCubeをプレイした時には新しいサウンドがゲームに含まれています。

## 既知の問題

これはWwiseサウンドエンジンをゲームエンジンに統合する方法を示す事例に過ぎないため、以下の問題はこのリリースで解決されていません。

### SDLの移行に関する既知の問題

- フルスクリーンを利用できません。
- Gammaコマンドが無効です。
- 専用モードで実行時にサーバメッセージを表示できません。
- フレームレートが低いとマウスカーソルがゲームから「逃げ」ます。
- マウスホイールによる武器の選択が機能しません。

### サウンドに関する既知の問題

- オクルージョンの対象となるキャラクターの足音サウンドがありません。
- 水中で足音が聞こえます。
- インタラクティブミュージックがありません。



---

## Chapter 6. ヘルプを利用する

### ヘルプを利用する

Wwiseヘルプに、Wwiseの各項目に関する詳細情報が記載されています。

以下のいずれかの方法でWwiseからHelpを開くことができます:

- ビューやダイアログボックスのタイトルバーにあるHelpアイコン (?) をクリックします。
- メニューバーで**Help > Wwise Help**をクリックします。
- **F1** を押します。

### その他のリソース

CubeゲームとそのWwiseプロジェクトは、Wwiseでゲーム全体を開発するワークフローを理解するうえで役立ちます。Wwiseについてさらに詳しく学ぶため、ゲームオーディオの実装に関するさまざまなアプローチを知りたいという場合は、以下のリソースに目を通してみてください:

- **Limbo** - LimboはPlaydeadによって開発された複数の受賞歴を持つゲームで、Audiokinetic Launcherから関連するWwiseプロジェクトとともに入手できます。Limboには複雑なサウンド階層も含まれるため、Wwiseの上級ユーザーに最適なゲームと言えます。なお、LimboはAudiokinetic LauncherからMacにインストールできませんのでご注意ください。
- **ビデオチュートリアル** - Audiokineticのウェブサイトにある[Audiokinetic ビデオチャンネル](#)と[AudiokineticのYouTube](#)チャンネルでビデオチュートリアルを視聴できます。
- **Wwise技能検定** - [Wwise技能検定](#)オンラインコースは、Wwiseの基本的なアプローチと上級者向けの特定のトピックを学べるプログラムをいくつか提供しています。技能検定プログラムのWwise-101とWwise-201ではWwiseの学習用プラットフォームとしてCubeを使用しています。