

# FFmpeg for Unity

---

## 目次

1. [概要](#)
2. [対応環境](#)
3. [FFmpeg本体について](#)
4. [使用方法](#)
5. [各種設定](#)
6. [シーン詳細](#)
7. [コンポーネント詳細](#)
  - 7-1. [基本](#)
    - 7-1-1. [FfmpegCommand](#)
    - 7-1-2. [FfmpegPlayerCommand](#)
    - 7-1-3. [FfmpegCaptureCommand](#)
  - 7-2. [バイト処理](#)
    - 7-2-1. [FfmpegBytesCommand](#)
    - 7-2-2. [FfmpegBytesPlayerCommand](#)
    - 7-2-3. [FfmpegBytesCaptureCommand](#)
  - 7-3. [フレーム毎バッチ処理](#)
    - 7-3-1. [FfmpegGetTexturePerFrameCommand](#)
    - 7-3-2. [FfmpegWriteFromTexturesCommand](#)

## 1. 概要

これはUnityのエディタ上やアプリ上でFFmpegを使用できるアセットです。

以下のようなことが出来ます。

- 動画の再生(mp4, avi, mov etc.)
- ゲーム画面、もしくはゲーム内カメラからの録画
- 動画の変換
- [Youtube](#)へのライブ配信(rtmp etc.) など

## 2. 対応環境

- Unityエディタ(Windows/Mac/Linux)
- スタンドアローン(Windows/Mac/Linux)(Mono/IL2CPP)
- Android
- iOS

### 3. FFmpeg本体について

このアセットでは以下のスクリプトからビルドしたFFmpeg本体(Windows/Mac/Linux)、もしくはプラグイン(Windows/Android/iOS)を使用しております。

外部ライブラリは一部のみ（「ライセンス」を参照）にしておりますので、使用したい機能がない場合は手順に従って再ビルドし、それに置き換えてください。

- Windows: <https://github.com/NON906/ffmpeg-windows-build-helpers>
- Mac/Linux: <https://github.com/NON906/ffmpeg-build-script>
- Android/iOS: <https://github.com/NON906/ffmpeg-kit>

### 4. 使用方法

1. このアセットをインポートします。
2. 必要であれば、「FFmpeg本体について」にあるURLからFFmpegをビルドし、置き換えます。
3. 以下のように設定を変更します。
  - 3-1. メニューバーから'File -> Build Settings'を選択してください。
  - 3-2. 'Player Settings...'をクリックしてください。
  - 3-3. 'Other Settings -> Api Compatibility Level'を'.NET 4.x'に変更してください。
4. サンプルシーンを実行したい場合、以下を行います。

- `Assets/StreamingAssets`に再生（もしくは変換）したい動画を`sample.mp4`という名前で配置してください。
- PlayerVR, ListVR: 上記に加え、Package ManagerからXR Interaction Toolkitとそれに対応したPluginをインポートしてください。
- YoutubeLive: URLにある`[STREAM_KEY]`をYoutubeのライブ配信の画面から取得したストリームキーに置換してください。

NOTE: ver1.6から、`com.artenica.smartexception`を同梱するようになりました。

アップデート後にエラーが発生した場合は、`Assets/Plugins/Android/mainTemplate.gradle`の設定を削除してください。

## 5. 各種設定

メニューバーのTools→Ffmpeg for Unity→Open Setting Windowから設定を行えます。  
動作の不具合などがない場合は、デフォルト設定のまま使用するのを推奨します。

NOTE: WindowsでBuilt In BinaryもしくはInstalled Binaryを使用したい場合は、[こちら](#)からffmpeg本体をインポートする必要があります

- Library (Windowsのみ)

ライブラリ化したffmpegを使用します

- Built In Binary (Windows/Mac/Linux)

以下のパスに存在するffmpeg本体を使用します

アプリとしてビルドする場合は自動で組み込まれます

ver1.6.1以前でUse Built InをONにしたときと同じ動作になります

Windows: Assets/FfmpegUnity/Bin/Windows

Mac: Assets/FfmpegUnity/Bin/Mac

Linux: Assets/FfmpegUnity/Bin/Linux

- Installed Binary (Windows/Mac/Linux)

実行環境にインストールされているffmpeg本体を使用します

実行環境にffmpeg本体を事前にインストールしており、なおかつPATHが通っている必要があります

ver1.6.1以前でUse Built InをOFFにしたときと同じ動作になります

- Library Memory (Android/iOS)

Unityとffmpeg-kitの連携をメモリコピーで行います

- Pipe (Android/iOS)

Unityとffmpeg-kitの連携を名前付きパイプで行います

ver1.6.1以前の動作になります

## 6. シーン詳細

- List
- ListVR

使用できる機能の一覧を表示します  
後者はVR用です

- Convert

動画形式の変換のサンプルです

- Player
- PlayerVR

動画プレイヤーのサンプルです

- TexturePlayer
- TexturePlayerRender

動画をTextureに適用する方法のサンプルです

- Capture
- CaptureCamera
- CaptureRenderTexture

動画録画のサンプルです

それぞれで録画元が異なります

- SendStream
- TextureStreamPlayer
- TexturePlayerCompare

動画のストリーミング配信のサンプルです

- YoutubeLive

YoutubeLiveへの配信のサンプルです

- BytesInput
- BytesOutput
- ConvertBytes
- TextureBytesPlayer
- TextureBytesCaptureAndPlayer

バイトを直接入出力する場合のサンプルです

- BatchTexture

フレーム毎に処理する場合のサンプルです

## 7. コンポーネント詳細

### 7-1. 基本

#### 7-1-1. FfmpegCommand

設定されたFFmpegのコマンドを実行します。動画の変換など、表示が不要な処理に使用してください。

インスペクターの設定:

Execute On Start: シーン開始時にコマンドを実行します。使用しない場合は、`StartFfmpeg()`から実行してください。

Options: 実行したいコマンドのオプションを設定します。以下を記載した場合、対応するパスに変換されます。

```
{STREAMING_ASSETS_PATH} : Application.streamingAssetsPath
{PERSISTENT_DATA_PATH} : Application.persistentDataPath
{TEMPORARY_CACHE_PATH} : Application.temporaryCachePath
```

NOTE: Use Built Inは[設定ウィンドウ](#)に引っ越しました

## 7-1-2. FfmpegPlayerCommand

設定したFfmpegPlayerVideoTextureとAudioSourceに、映像や音声を設定します。

インスペクターの設定:

Execute On Start, Options: FfmpegCommandと同じです。

Input Options: 動画の指定する前のコマンドのオプションを設定します。

Default Path: 基準となるパスを指定します。以下の2つは同じです。

```
Default Path: STREAMING_ASSETS_PATH
Input Path: sample.mp4
```

```
Default Path: NONE
Input Path: {STREAMING_ASSETS_PATH}/sample.mp4
```

Input Path: 動画ファイルのパスを指定します。

Auto Settings: 自動で動画の大きさなどを設定します。基本的にONで構いませんが、一度設定を取得してから、再度実行することになるため、ストリーミングなどでは上手くいかない場合があります。

Video Textures: 設定したFfmpegPlayerVideoTextureの設定に応じて、以下の処理が行われます。

FfmpegPlayerVideoTextureのVideoTextureが空の場合: VideoTextureに新しくTexture2Dが設定されます。  
スクリプトでVideoTextureをRendererなどに設定してください。

FfmpegPlayerVideoTextureのVideoTextureにRenderTextureがある場合: 設定したRenderTextureに書き込みます。

Audio Sources: 設定したAudioSourceのclipに音声が設定されます。

### 7-1-3. FfmpegCaptureCommand

映像や音声を取得して、動画の保存や配信を行います。

NOTE: スマートフォンなどで処理が間に合わない場合は、映像の速度が異常に早くなることがあります。その場合、サイズなどの設定を行って、処理を軽減させてください。

インスペクターの設定:

Execute On Start, Options: FfmpegCommandと同じです。

Capture Sources: キャプチャしたい内容を設定します。Video Sizeが設定出来る場合に0以下の値を設定した場合、その値は自動で設定されます。

## 7-2. バイト処理

### 7-2-1. FfmpegBytesCommand

byte[]での入力と出力ができるFfmpegCommandの派生コンポーネントです。

`AddInputBytes(bytes, inputNo)`で入力ができます。

(`inputNo` は'Input Option'に対応したストリーム番号です)

`GetOutputBytes(outputNo)`で出力内容を取得できます。

(戻り値はbyte[]です)

(`outputNo`は'Output Option'に対応したストリーム番号です)

Input Options: 入力ストリームの数とその前に指定するオプションを設定します。

NOTE: このリストの数とストリームの数は一致します。 オプションを指定しない場合でも、（空欄で構わないので）必要なストリームの数を用意する必要があります。

Output Options: 出力ストリームの数とその前に指定するオプションを設定します。

NOTE: 'Input Options'と同様に一致させる必要があります。

### 7-2-2. FfmpegBytesPlayerCommand

byte[]での入力ができるFfmpegPlayerCommandの派生コンポーネントです。

詳細は[FfmpegPlayerCommand](#)と[FfmpegBytesCommand](#)を参考にしてください。

### 7-2-3. FfmpegBytesCaptureCommand

byte[]での出力ができるFfmpegCaptureCommandの派生コンポーネントです。

詳細は[FfmpegCaptureCommand](#)と[FfmpegBytesCommand](#)を参考にしてください。



## 7-3. フレーム毎バッチ処理

### 7-3-1. FfmpegGetTexturePerFrameCommand

フレーム毎に取得が可能なFfmpegPlayerCommandの派生コンポーネントです。

`GetNextFrame()` コルーチンを実行することで次のフレームに更新します。  
(実行しない間は次のフレームに進みません)

### 7-3-2. FfmpegWriteFromTexturesCommand

Textureの内容を動画に書き込むためのコンポーネントです。

`WriteTexture(inputTexture)` コルーチンで指定のTextureの内容を書き込むことが出来ます。