

# FFmpeg for Unity

---

## Table of Contents

1. [Overview](#)
2. [Correspondence Environment](#)
3. [Note for FFmpeg](#)
4. [How to Use](#)
5. [Settings](#)
6. [Scenes](#)
7. [Component Details](#)
  - 7-1. [Basic](#)
    - 7-1-1. [FfmpegCommand](#)
    - 7-1-2. [FfplayCommand \(New Player\)](#)
    - 7-1-3. [FfmpegPlayerCommand \(Old Player\)](#)
    - 7-1-4. [FfmpegCaptureCommand](#)
  - 7-2. [Bytes](#)
    - 7-2-1. [FfmpegBytesCommand](#)
    - 7-2-2. [FfmpegBytesPlayerCommand](#)
    - 7-2-3. [FfmpegBytesCaptureCommand](#)
  - 7-3. [Batch Per Frames](#)
    - 7-3-1. [FfmpegGetTexturePerFrameCommand](#)
    - 7-3-2. [FfmpegWriteFromTexturesCommand](#)

## 1. Overview

This assets is to execute [FFmpeg](#) on editor and application.

You can do below:

- Play Movies(mp4, avi, mov etc.)
- Capture Game View or Camera(In Game)
- Convert Video Files
- Live Streaming to [Youtube](#)(rtmp etc.)
- etc.

## 2. Correspondence Environment

- Unity Editor(Windows/Mac/Linux)
- Standalone(Windows/Mac/Linux)(Mono/IL2CPP)
- Android(IL2CPP)
- iOS

### 3. Note for FFmpeg

This assets uses FFmpeg binary(Windows/Mac/Linux) or plugins(Windows/Mac/Android/iOS) built below.

It using libraries is part only(please show Lisense). So, if libraries you want to use is not included, please rebuild and replace files.

For Windows, Mac (Library) and Linux(Library), replace generated 1. from 2. and execute it.

- Windows:
  1. <https://github.com/NON906/ffmpeg-windows-build-helpers>
  2. [https://github.com/NON906/fftools\\_lib](https://github.com/NON906/fftools_lib)
- Mac(Library):
  1. <https://github.com/NON906/ffmpeg-kit>
  2. <https://github.com/NON906/FfmpegUnityMacPlugin>
- Linux(Library):
  1. <https://github.com/NON906/ffmpeg-build-script>
  2. [https://github.com/NON906/fftools\\_lib](https://github.com/NON906/fftools_lib)
- Android/iOS: <https://github.com/NON906/ffmpeg-kit>
- Mac/Linux(Binary): <https://github.com/NON906/ffmpeg-build-script>

## 4. How to Use

1. Import this assets.
2. (Optional) Rebuild FFmpeg and replace files.
3. Edit Settings:
  - 3-1. Open 'File -> Build Settings' on the menu bar.
  - 3-2. Click 'Player Settings...'.
    - 3-3. For Unity2020, change to '.NET 4.x' in 'Other Settings -> Api Compatibility Level'.
    - 3-4. For Android, change to 'IL2CPP' in 'Other Settings -> Scripting Backend'.
    - 3-5. For Android, turn on 'ARM64' in 'Other Settings -> Target Architectures'.
4. If want to run sample scenes, do it:
  - Please place `sample.mp4` to `Assets/StreamingAssets`.
  - PlayerVR, ListVR: In addition to the above, import the 'XR Interaction Toolkit' and its corresponding Plugin from Package Manager.
  - YoutubeLive: Please replace stream key(from Youtube Live) to `[STREAM_KEY]`.
5. For iOS, If you want to check the output data, please set the following in xcode (Unity-iPhone -> Info) after building:
  - Application supports iTunes file sharing: YES
  - Supports opening documents in place: YES

NOTE: (For Android) From ver1.6, `com.athenica.smartexception` is included.

If you get an error after updating, please delete the setting of `Assets/Plugins/Android/mainTemplate.gradle`.

NOTE: Since ver2.7, apps built on Android may crash.

If this happens, try one of the following:

- Change Unity version to 2021.3.27f1 or later.
- Build with 'Export Project' turned on and build that project in Android Studio (Gradle needs to be updated).

## 5. Settings

You can set setting from Tools->Ffmpeg for Unity->Open Setting Window on the menu bar.  
If there are no malfunctions, I recommend using the default settings.

- Library

Use ffmpeg library.

- Built In Binary

Use the ffmpeg binary that exists in the following path.

When building, it will be installed automatically.

It will be the same operation as when Use Built In is turned ON in ver1.6.1 or earlier.

Windows: Assets/FfmpegUnity/Bin/Windows

Mac: Assets/FfmpegUnity/Bin/Mac

Linux: Assets/FfmpegUnity/Bin/Linux

- Installed Binary

Use the ffmpeg binary installed to OS.

You need to have ffmpeg installed in advance and have a PATH.

It will be the same operation as when Use Built In is turned OFF in ver1.6.1 or earlier.

NOTE:

- for Windows:

If you want to use 'Built In Binary' or 'Installed Binary' on Windows, please import [HERE](#).

- for Mac:

If you want to use 'Built In Binary' on Mac, please import [HERE](#).

Please note that this binary may not work on M1 Macs, so if applicable, prepare a corresponding binary or use something other than 'Built In Binary'.

## 6. Scenes

- List
- ListVR

Displays a list of available functions.

The latter is for VR.

- Convert
- ConvertProgress

There are sample of video format conversion.

- FfplayPlayer

There are sample of video player.

- FfplayTexture
- FfplayRenderTexture

There are sample of how to apply a video to Texture.

- Capture
- CaptureRenderTexture

There are sample of video capture.

The capturing source is different for each.

- SendStream

This is sample of video streaming distribution.

- FfplayStream

This is sample of playing video stream.

- YoutubeLive

This is a sample of delivery to Youtube Live.

- BytesInput
- BytesOutput
- ConvertBytes
- FfplayBytesTexture

There are sample for inputting / outputting bytes directly.

- BatchTexture
- BatchTextureWithSound

This is a sample for processing frame by frame.

## 7. Component Details

### 7-1. Basic

#### 7-1-1. FfmpegCommand

Execute FFmpeg command.

Inspectors:

Execute On Start: Execute the command at the beginning of the scene. If not uses, please call `StartFfmpeg()` when want to start command.

Options: FFmpeg commands option. If you set below, this replace to path when execute command.

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

NOTE: 'Use Built In' is moved to [Setting Window](#).

## 7-1-2. FfplayCommand (New Player)

Setting video to FfmpegPlayerVideoTexture, and audio to AudioSource.

Inspectors:

Execute On Start: Same as FfmpegCommand.

Options: FFplay commands option. NOTE: You must specify options for ffplay. NOT ffmpeg command.

Default Path: Standard path. Both are the same:

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

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

Input Path: Video File path.

Video Textures: Depending on the FfmpegPlayerVideoTexture settings, the following will be done:

```
FfmpegPlayerVideoTexture.VideoTexture is empty: Make Texture2D to VideoTexture.
Please set with scripts to Renderer etc.
```

```
FfmpegPlayerVideoTexture.VideoTexture is setted in RenderTexture: Write to
RenderTexture.
```

Audio Sources: Set audio to AudioSource's clip.

### 7-1-3. FfmpegPlayerCommand (Old Player)

Setting video to FfmpegPlayerVideoTexture, and audio to AudioSource.

Inspectors:

Execute On Start, Options: Same as FfmpegCommand.

Input Options: FFmpeg commands option before input streams.

Default Path: Standard path.

Input Path: Video File path.

Auto Settings: Auto Setting to video size etc. In most cases ON.

Video Textures: Depending on the FfmpegPlayerVideoTexture settings, the following will be done:

FfmpegPlayerVideoTexture.VideoTexture is empty: Make Texture2D to VideoTexture.  
Please set with scripts to Renderer etc.

FfmpegPlayerVideoTexture.VideoTexture is setted in RenderTexture: Write to  
RenderTexture.

Audio Sources: Set audio to AudioSource's clip.

Buffer Time: The delay time (seconds) for buffering.

It can be 0 unless the operation is unstable.

A negative value disables buffering.



#### **7-1-4. FfmpegCaptureCommand**

Capture video and audio, and save movies or streaming.

NOTE: When processing is not in time, video speed may be abnormally high. If that happens, please reduce the processing (change video size etc.).

NOTE: Video\_Camera cannot be used with URP or HDRP.

For VR devices, Video\_GameView and Video\_Camera may not be available.

Please use Video\_RenderTexture where applicable.

Execute On Start, Options: Same as FfmpegCommand.

Capture Sources: Input for capture. If Video Size is 0 or less, this value is auto setting.

## 7-2. Bytes

### 7-2-1. FfmpegBytesCommand

It is bytes input/output version of FfmpegCommand.

Set the input bytes to `AddInputBytes(bytes, inputNo)`.

(`inputNo` is input stream's number corresponding to 'Input Option'.)

Get the output bytes from `GetOutputBytes(outputNo)`.

(Return value is `byte[]`)

(`outputNo` is output stream's number corresponding to 'Output Option'.)

Input Options: Number of input streams, and FFmpeg commands option before input streams.

NOTE: The number of this list matches the number of streams.

Even if you don't need the options, you need to prepare items for the number of streams (can be blank).

Output Options: Number of output streams, and FFmpeg commands option before output streams.

NOTE: Note the same as 'Input Options'.

### 7-2-2. FfmpegBytesPlayerCommand

It is bytes input version of FfmpegPlayerCommand.

Please check [FfmpegPlayerCommand](#) and [FfmpegBytesCommand](#).

### 7-2-3. FfmpegBytesCaptureCommand

It is bytes output version of FfmpegCaptureCommand.

Please check [FfmpegCaptureCommand](#) and [FfmpegBytesCommand](#).

## 7-3. Batch Per Frames

### 7-3-1. FfmpegGetTexturePerFrameCommand

This is the version of FfmpegPlayerCommand that can be obtained for each frame.

Call `GetNextFrame()` coroutine to update to the next frame.

(Does not advance to the next frame while not calling)

### 7-3-2. FfmpegWriteFromTexturesCommand

A component for writing the contents of Texture to a video.

Call `WriteTexture(inputTexture)` coroutine to write the contents of the specified Texture.