Bink Plugin for Unreal Engine

Requirements

The Bink plugin for Unreal is built into Unreal 4.27+ (and eventually Unreal 5.0). All Unreal platforms (including NDA platforms) are supported.

Installation

1. **Turn on the Bink Plugin from the editor -** to do this, select Settings->Plugins in the editor. Then search for “Bink”, and check the “enabled” box.
2. **Copy your Bink video files** - copy your Bink video files into the “Contents/Movies” directory under your Unreal Project directory. You create Bink files using the “Bink 2 Encoder for Unreal” found in the Engine/Binaries/ThirdParty/Bink directory (Bink2ForUnreal.EXE).
3. **Load the project -** Double-click on the .uproject file of the project and a warning dialog should pop up and ask if you want to rebuild the project due to missing plugins. Allow the rebuild to execute and Unreal should start with the project loaded up.
4. **All done!** You should be able to play the Bink files in your Content/Movies directory.

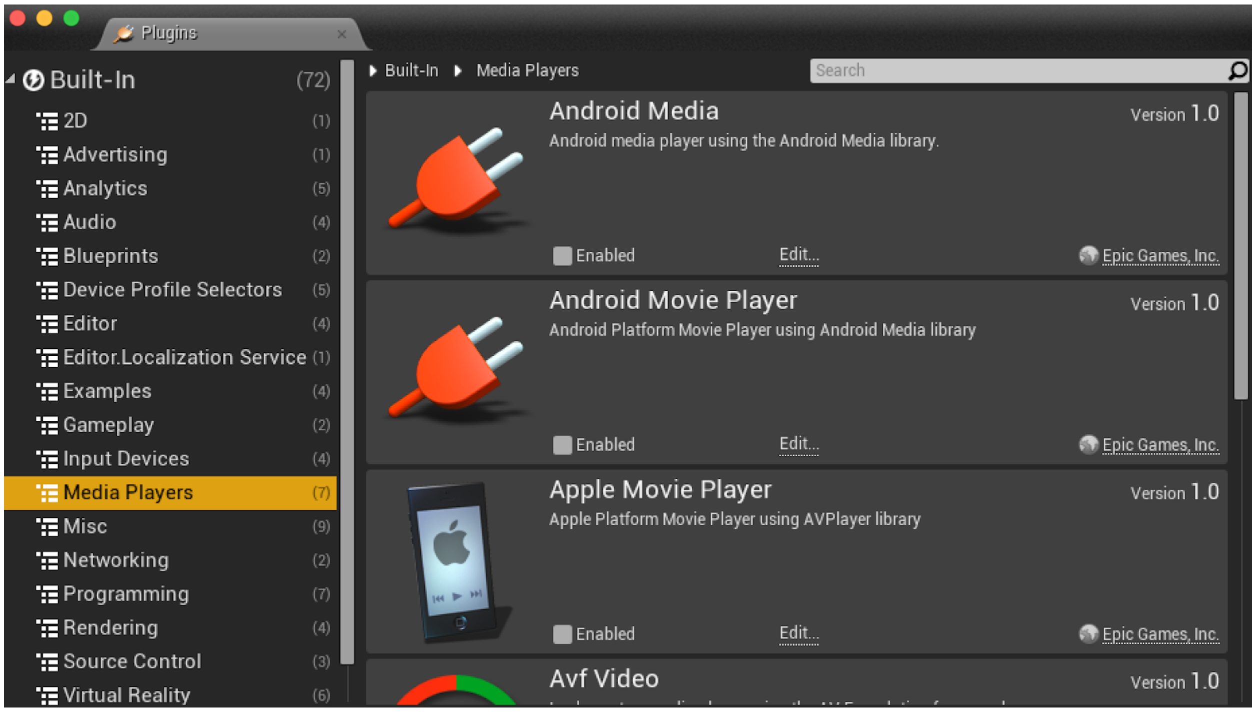
Bink for Unreal Movie Types

Unreal has two distinct types of movies - fullscreen startup movies, and in-game non-startup movies. You use each type a little differently - we describe each below.

1. Fullscreen Startup Movie Setup

First, you need to disable all other movie player plugins (otherwise the mpeg4 player will try to play the bink movie files which will silently fail).

To do this, go into Edit->Plugins:



Built-In	(72)
2D	(1)
Advertising	(1)
Analytics	(5)
Audio	(4)
Blueprints	(2)
Device Profile Selectors	(5)
Editor	(4)
Editor.Localization Service	(1)
Examples	(4)
Gameplay	(2)
Input Devices	(4)
Media Players	(7)
Misc	(9)
Networking	(2)
Programming	(7)
Rendering	(4)
Source Control	(3)
Virtual Reality	(6)

Android Media — Version 1.0
Android media player using the Android Media library.
Enabled — Edit... — Epic Games, Inc.

Android Movie Player — Version 1.0
Android Platform Movie Player using Android Media library
Enabled — Edit... — Epic Games, Inc.

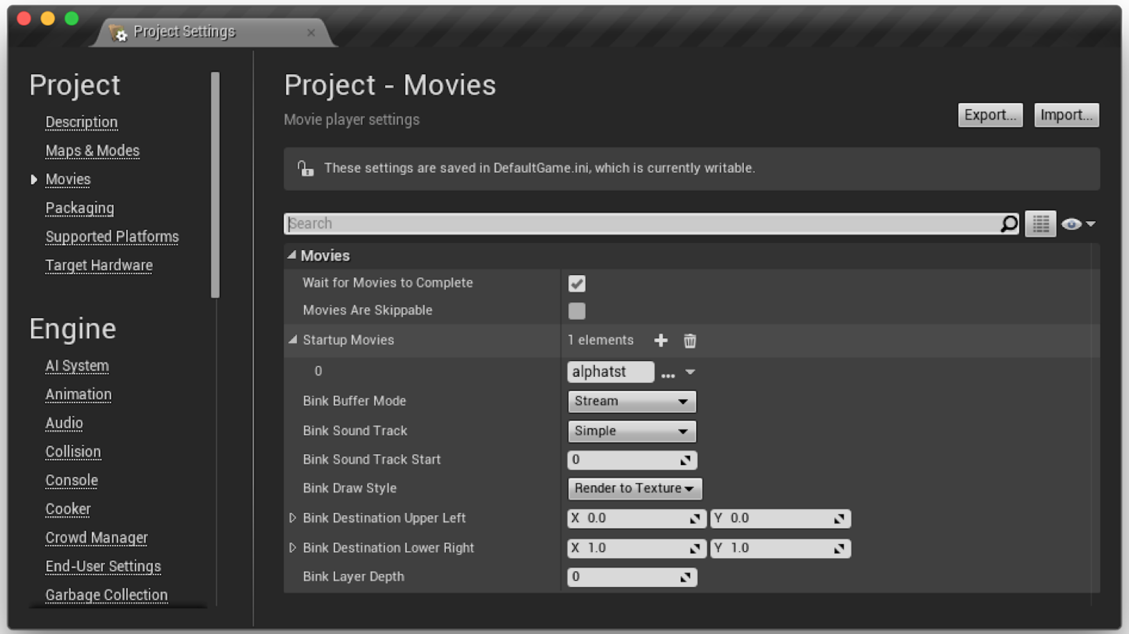
Apple Movie Player — Version 1.0
Apple Platform Movie Player using AVPlayer library
Enabled — Edit... — Epic Games, Inc.

Avf Video — Version 1.0

And disable all plugins other than the “Bink Movie Player” plugin.

Second, you need to configure some movies to play on startup.

Go into Edit->Project Settings



Project - Movies

Movie player settings

These settings are saved in DefaultGame.ini, which is currently writable.

Movies

Wait for Movies to Complete	✓
Movies Are Skippable	☐
Startup Movies	1 elements
0	alphatst
Bink Buffer Mode	Stream
Bink Sound Track	Simple
Bink Sound Track Start	0
Bink Draw Style	Render to Texture
Bink Destination Upper Left	X 0.0 Y 0.0
Bink Destination Lower Right	X 1.0 Y 1.0
Bink Layer Depth	0

Click the + next to “Startup Movies” and then click the > to the left to show the movies. You can play multiple movies in sequence. Then click the “...” to select a file to play.

There are some Bink specific options:

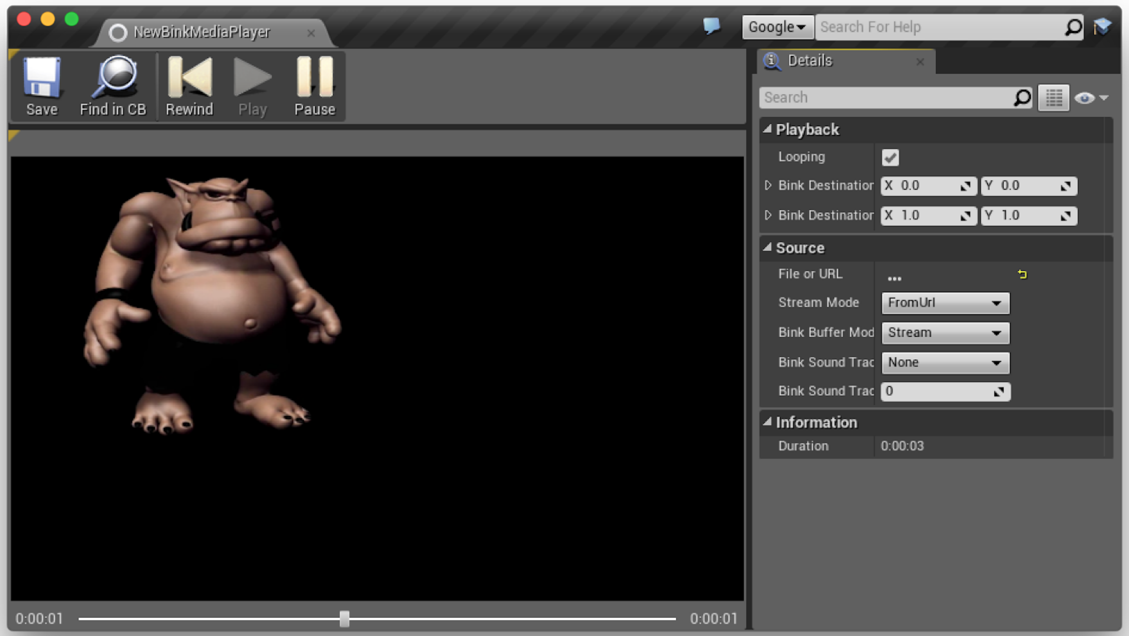
1. **Bink Buffer Mode** This controls whether or not you want to stream a portion of the movie from disk, pre-load the whole movie before playing, or don’t start streaming until the whole movie is resident.
2. **Bink Sound Track** This determines how you want to play the sound. There are a bunch of options:
    a. SndNone - don't play any sound in this Bink.
    b. "Snd Simple" (default) - this option tries to figure out what you want based on filename of the Bink file. If the filename ends in "_51", we use the "Snd 51" option. If it ends in "_51L", then "Snd 51 Language Override". If it ends in "_71", then "Snd 71". And if it ends in "_71L", then "Snd 71 Language Override". Otherwise, we just play the Bink track specified by the "Sound Track Start" value (by default, track 0).

   c. "Snd Language Override" - this option plays two tracks - the audio in track 0 (usually the background music/effects) and then also the track specified by the "Sound Track Start" value (which is usually the language track).
   d. "Snd 51" - this option plays the 6 mono tracks starting at the track specified by the "Sound Track Offset" value (by default, 0) into the systems. So, if you have fully localized 5.1 tracks for 4 languages, you would mix in 24 Bink tracks, and use "Sound Track Start" to specify the right range of audio tracks to play.
   e. "Snd 51 Language Override" - this option plays the 6 mono background/effect tracks starting at offset 0, and then the one mono track specified by the "Sound Track Start" value as the language track into the center channel. So, for 4 languages here, you would have 10 audio tracks - the 5.1 background music/effects tracks and then 4 different center language tracks.
   f. "Snd 71" - this option plays the 8 mono tracks starting at the track specified by the "Sound Track Offset" value (by default, 0) into the systems. So, if you have fully localized 7.1 tracks for 4 languages, you would mix in 28 Bink tracks, and use "Sound Track Offset" to specify the right range of audio tracks to play.
   g. "Snd 71 Language Override" - this option plays the 8 mono background/effect tracks starting at offset 0, and then the one mono track specified by the "Sound Track Start" value as the language track into the center channel. So, for 4 languages here, you would have 12 audio tracks - the 7.1 background music/effects tracks and then 4 different center language tracks.
3. **Bink Sound Track Start** A bink movie can contain multiple different sound tracks. With this option you can specify which sound track you want to play.
4. **Bink Destination Upper Left / Lower Right** This specifies the rectangle you want the movie to render to. You can use this to force a letter box for example.

2. In-game (Non-Startup) Movie Setup

You can also render movies during the game level itself. You can either render directly to a texture and then use that texture in game, or you can render directly to the screen (Bink will render after all the graphics but before UI (allowing you to draw subtitles on top of the movie).

To do this you right click in the “Content Browser” and under the “Miscellaneous” section add a new “Bink Media Player”.



Playback
Looping ✓
Bink Destinatior X 0.0 Y 0.0
Bink Destinatior X 1.0 Y 1.0

Source
File or URL ...
Stream Mode FromUrl
Bink Buffer Mod Stream
Bink Sound Trac None
Bink Sound Trac 0

Information
Duration 0:00:03

The Bink specific configuration options are:

1. **Bink Buffer Mode** This controls whether or not you want to stream a portion the movie from disk, pre-load the whole movie before playing, or don’t stop streaming until the whole movie is resident.
2. **Bink Sound Track** This determines how you want to play the sound. There are various options from simple stereo to 7.1 surround sound.
3. **Bink Sound Track Start** A bink movie can contain multiple different sound tracks. With this option you can specify which sound track you want to play.
4. **Bink Draw Style** With this you can override the default UE4 functionality of rendering to a texture and instead render directly to the screen bypassing UE4 rendering.

5. **Bink Destination Upper Left / Lower Right** This specifies the rectangle you want the movie to render to. You can use this to force a letter box for example.
6. **Bink Layer Depth** Allows you to render multiple videos at the same time and set the depth to control the order that they are drawn.

Important Bink Movie files (*.bk2) should be placed in the “MyProject/Content/Movies” folder. This will ensure that they are copied through to the final build properly in all configurations.

You can then right click on the BinkMediaPlayer in the Content Browser and select “Create Media Texture” to generate a texture from the player. You can then create a default material using this texture by right clicking the BinkMediaPlayer_Tex texture in the Content Browser and select “Create Material”. You can then use this material and texture like you would use any other material or texture in Unreal.

Other Bink for Unreal Notes

1. GPU-Assisted Video

The Bink plugin for Unreal can use GPU-assisted decoding - this is great in two ways: it's much faster (usually twice as fast), and it stresses the CPU less so it can be used for other tasks.

The bad news is that on Windows and Linux, GPU-assisted mode requires the very latest video drivers. We recommend it be turned off by default on Linux and Windows for this reason - Bink is already super fast on these platforms anyway. If you do enable it, have a way to turn it off in the UI for your end-users.

On NDA platforms, it is safe to rely on. Also, if you are in an embedded situation, you can control the GPU hardware and driver version to make sure it works. In other cases, just don't worry about it - Bink is plenty fast already.

You turn on GPU-assist mode by calling the BinkPluginTurnOnGPUAssist() in the BinkMediaPlayerModule.cpp file (after the BinkPluginInit call).

2. Multi-threaded Decoding

The Bink plugin for Unreal natively supports multithreaded decoding. It will use up to four threads during decompression, depending on the runtime CPU count.

By default, the Bink 2 encoder for Unreal will use 4 video slices for best multithreaded performance .

3. Seeking

You can set a new playback position by calling the Bink plugin function, BinkPluginGoto(), directly. Usually, you want to jump to a key frame or all the intervening frames will have to be decompressed along the way (the Bink plugin does this in the background for you). You control how much CPU Bink spends each frame seeking to the new frame with the ms_per_process parameter. Usually using 30 ms or so per frame, will seek relatively quickly.

4. Video Depth (sorting during drawing)

If you are drawing lots of video overlays with the non-startup movie option, you can control the order in which they are drawn with the "Depth" option. This lets your videos stack correctly regardless of the order that Unreal processes your videos.

For render targets, the "Depth" controls the order that the videos are drawn into the render target (but the render target is drawn into the scene normally by Unreal, of course).

5. Subtitles

It's easy to handle subtitles in Unreal on top of the video playback.

The first step is to load a text file containing the subtitles. This can be in any format you want, since it's up to you to take the Bink frame number and then look up the subtitle string. Usually you just want to load the file into a std:map or some other frame to string generic.

Once you have a function that, given a frame number, will return a string, you use Bink.GetInfo() to get the frame that will be displayed and find the string.

Now you have to draw the string after the Bink frame is displayed. For this, use a Unreal UI text field, and it will draw after the Bink is displayed.

6. Stereo Video

Bink is fast enough for great looking 3D video. To do this, use the render target path, and then draw to a screen aligned quad. For each eye, adjust the quad to fill the screen with either the top half of the quad or the bottom half of the video.

7. Bink Plugin API

There a bunch of other controls to the Bink plugin API as well. You can call any of the functions in binkplugin.h when you have special needs (seeking, pausing, etc).