

Media Framework Technical Reference

Technical information regarding Media Framework in Unreal Engine 5.



This page outlines the currently supported file extensions, URLs, and capture devices, as well as pointing out some troubleshooting tips on how to enable Media Framework with different platforms. There are some known platform-specific issues and limitations which we are currently investigating and trying to resolve, each of which has been identified in the Troubleshooting and Known Issues sections.

Supported Files, URLs & Capture Devices

The table below indicates the supported file extensions based on each media player platform.

Click the [link here](#) to download a `.xlsx` version of the supported files and URLs spreadsheets.

	AndroidMedia Player	AvfMedia Player	Electra Player	PS4Media Player	WmfMedia Player	MfMedia Player (XboxOne)	MfMedia Player (Windows 7+)
.3g2		Y			Y		Y
.3gp		Y			Y		Y
.3gp2		Y			Y		Y
.3gpp	Y	Y			Y		Y
.aac	Y				Y		
.ac3		Y					Y
.adts					Y		
.aif		Y					Y
.aifc		Y					Y
.aiff		Y					Y
.amr		Y					Y
.asf					Y		
.au		Y					Y
.avi					Y		
.bwf		Y					Y
.caf		Y					Y
.cdda		Y					Y
.m2ts					Y		
.m3u8	Y		Y	Y			
.m3d				Y			
.m4a		Y		Y	Y		Y
.m4v		Y		Y	Y		
.mp4v							Y
.mov		Y		Y	Y		Y
.mp3		Y			Y		Y
.mp4	Y	Y	Y	Y	Y	Y	Y
.qt		Y					Y
.sami					Y		
.sdv		Y					Y
.smi					Y		
.snd		Y					Y
.wav		Y			Y		Y
.wave		Y					Y
.wma					Y		
.wmv					Y		

The table below shows the supported URLs based on media player platform.

	AndroidMedia Player	AvfMedia Player	Electra Player	PS4Media Player	WmfMedia Player	MfMedia Player (XboxOne)	MfMedia Player (Windows 7+)
file://	Y	Y	Y	Y	Y	Y	Y
http://	Y		Y	Y	Y		Y
httpd://	Y				Y		Y
https://	Y		Y	Y	Y		Y
mms://	Y				Y		Y
rtsp://	Y				Y		Y
rtspt://	Y				Y		Y
rtspu://	Y				Y		Y

The list of supported file formats above is ongoing, and support for additional file extensions will come online in future engine updates. These have been updated for the **Electra Media Player**.



For the best compatibility and performance, it is recommended to use **H.264** encoded **MP4 (.mp4)** container files.

Capture Devices

Media Framework supports audio/video capture hardware on Windows (WmfMedia) and Android (AndroidCamera) as of 4.18 (consoles however, are not yet supported).



Please see the [Playing Live Video Captures](#) page for an example guide on working with capture devices.

WMF Media Configuration

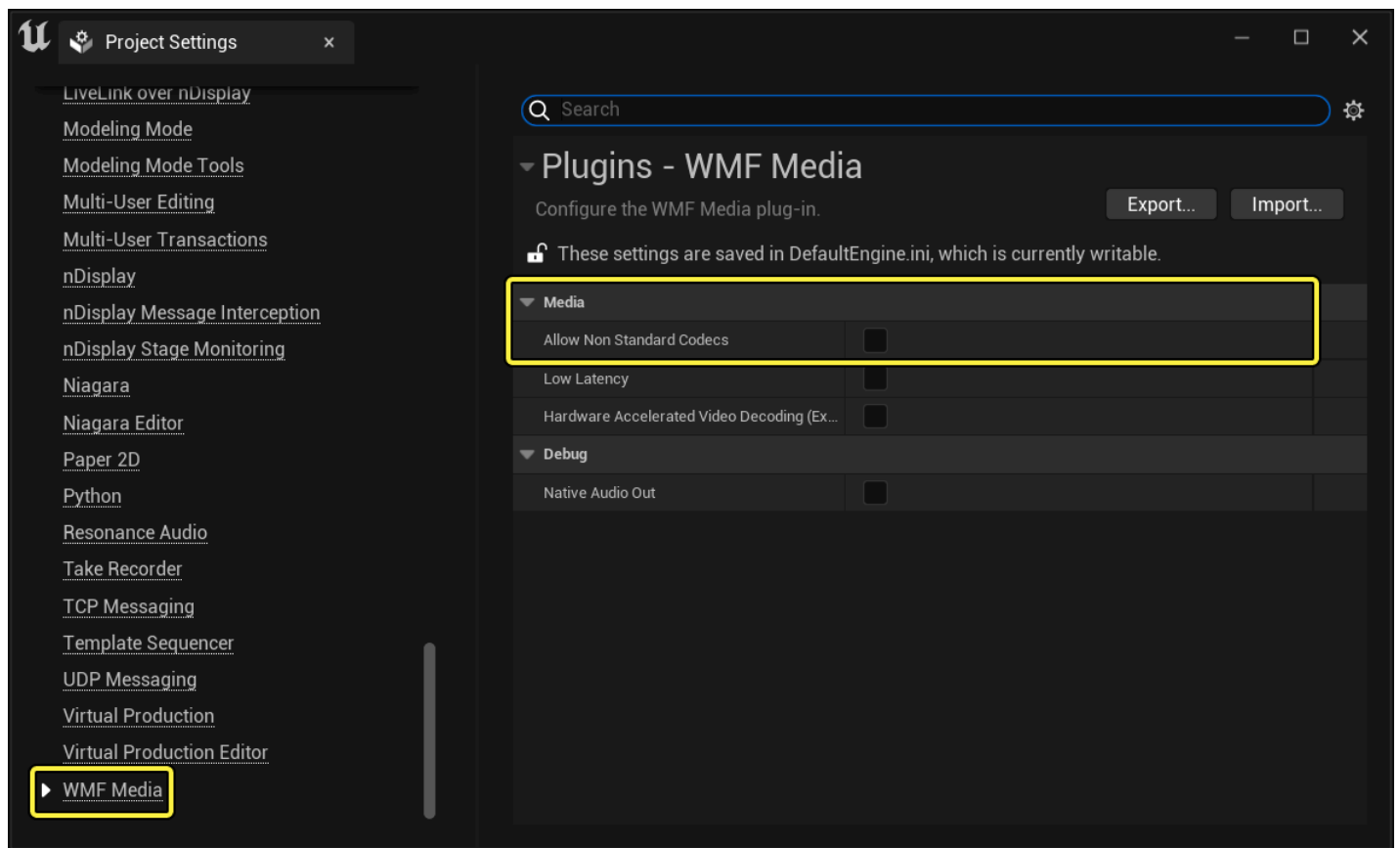
The following sections describe the configuration options that you can set for the WMF Media Plugin in your Project Settings.

Custom Codecs

Windows Media Foundation (WMF) handles standard audio/video playback and recording on the Windows Platform; however, WMF is also extensible. By default, WMF supports a number of different formats, encoding, and file containers, but it can also be extended with the use of optional codec packs that can be downloaded and installed from the Internet. What this means, however, is that certain users may not have a particular codec needed to decode and playback a media file.

In the past, Unreal Engine only supported default codecs that were supplied with WMF to ensure that everyone could handle and process media the same way. There may be times when you want to use a different method of encoding your media or your own proprietary media encoder that you want to ship your game with (for example, you ship your game with an installer that provides the needed encoder).

You can now enable and allow non-standard codecs for **WMF Media** from the **Plugins** section of **Project Settings**.



Click image for full size.

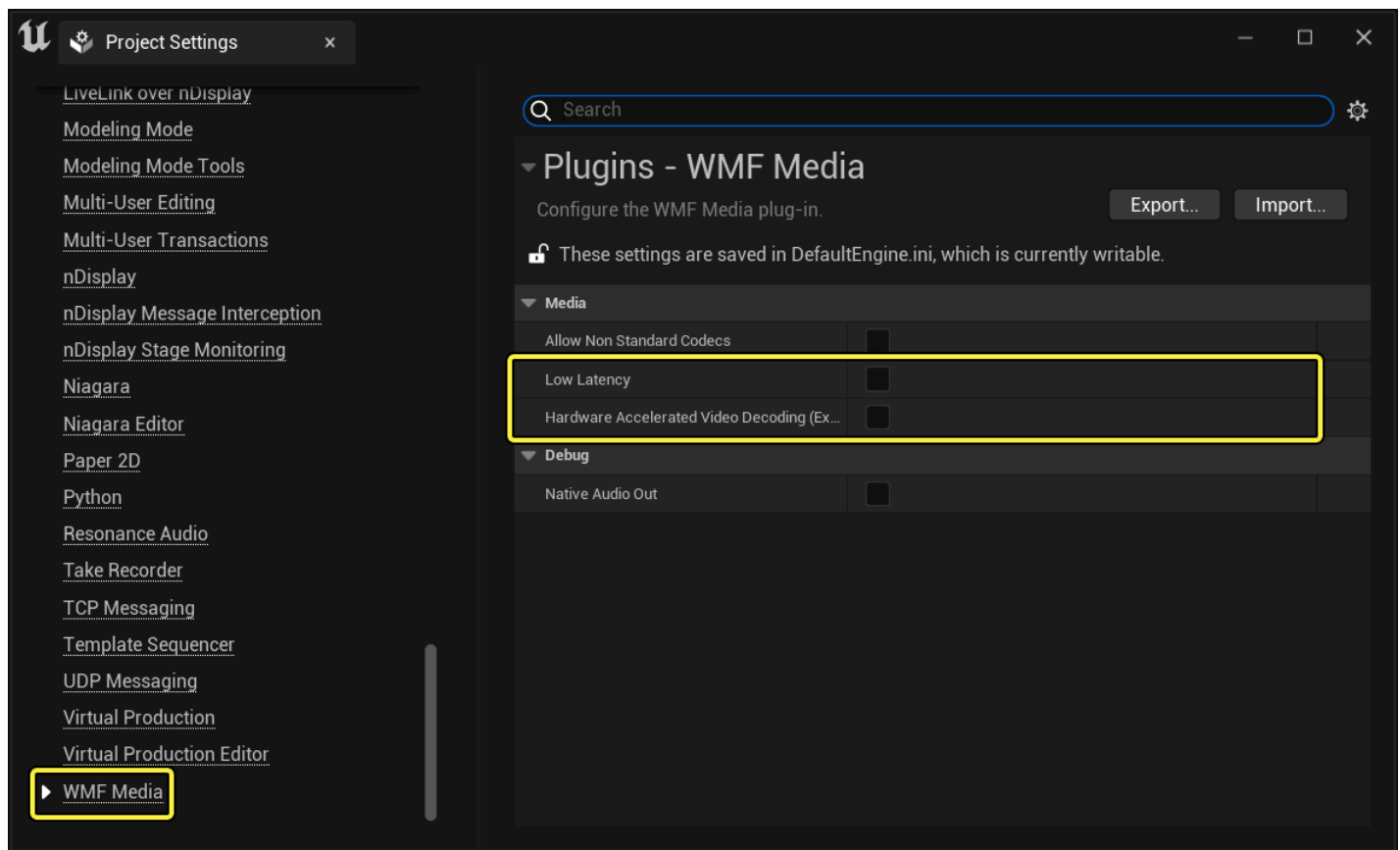
By default, the player plugin will attempt to detect audio and video codecs that are not supported by the operating system (OS), requiring the user to install additional codec packs. If the player plugin detects an unsupported format when accessing the media, a warning message is provided in the **Output Log**.

Additionally, you can see media asset information inside a **Media Player** asset in the **Info** panel.

When enabling **Allow non-supported codecs**, the Plugin player will skip the check and allow the usage of non-standard codecs.

Playback Optimization

The WMF Media Plugin offers some options that you can use to customize the way it decodes and plays back your media. If the Plugin doesn't play your media with the responsiveness, smoothness, or performance that you need, you can try enabling these options.



Click image for full size.

Setting	Description
Low Latency	This setting minimizes delays, or latency, in media playback. If your video playback tends to lag, this setting may help you achieve a faster response from the WMF Media Plugin. However, enabling this setting may adversely affect the quality of the video and/or audio. Only works on Windows 8 or later.
Hardware Accelerated Video Decoding (Experimental)	Uses the GPU instead of the CPU to decode video streams. If your CPU is a bottleneck on your video performance, this setting may improve the smoothness of your video playback. It may also enable you to use more media at the same time, with larger resolutions. Only works on Windows platforms, rendering with DirectX 11.

Setting	Description
	This option is considered Experimental . We do not recommend shipping projects with Experimental features.

HAP Codec Playback Support

HAP Codec Playback allows you to produce high quality media with low CPU/GPU processing requirements.

In UE5, you can play **1× 4K 60 FPS** movie or **2× 4K 30 FPS** movie, which can be stretched to **2× 4K 60 FPS** movies.

Specification	Supported Setting
Format	HAP, HAP Alpha, HAP Q, HAP Q Alpha
Frame Rates	10, 15, 23.976, 24, 25, 29.97, 30, 50, 59.94, 60
Resolutions	Up to 4K 3840×2160
Bit Depth	8bit (4:4:4:)
Pixel Aspect Ratio	Aspect Ratio Independent



As of 4.23, HAP playback does not support embedded audio or timecodes.

Benchmarking Test Case

Below is the benchmark test case the UE5 team used to determine these settings. You can use this test case as a comparison for your own project.

The current benchmarking process is meant to give a rough idea of system performance. We tested playback of the same movie file in 3 different softwares: Unreal Engine, Ventuz, and VLC. For each test, we monitored and logged CPU, GPU, and Disk usage with Windows Task Manager.

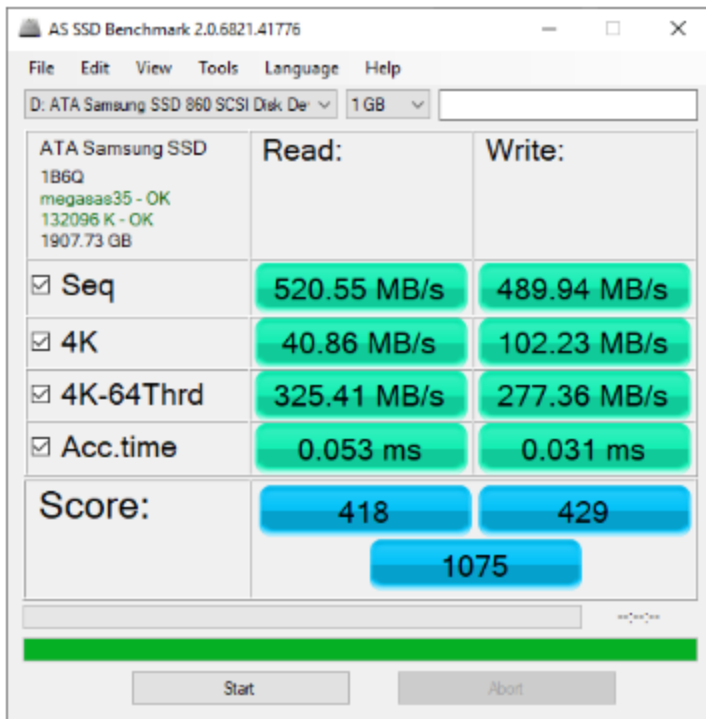
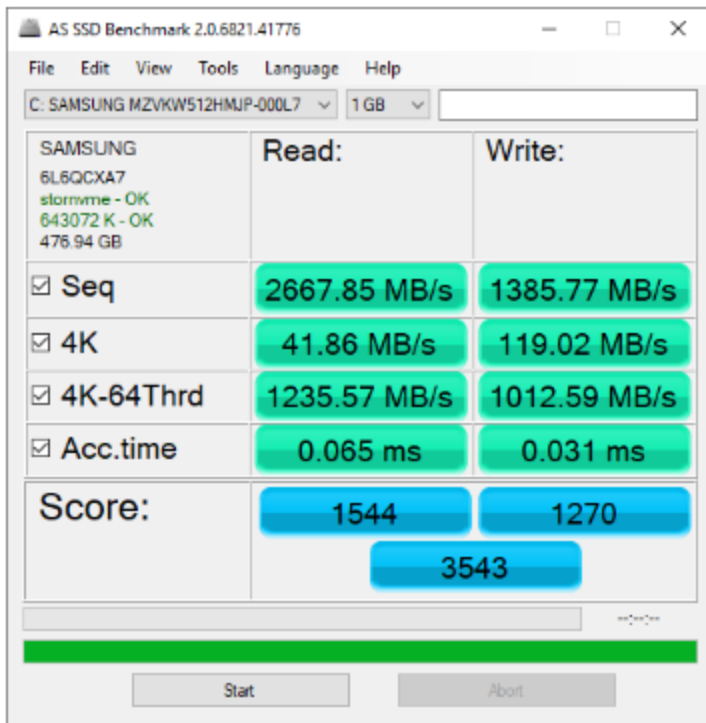
The movie file was an **11 second 4K 60 FPS movie** from an 8K 60 FPS source that we recompressed in 4K, 8K, and 16K using HAP. We discovered issues with WMF that cause anything above **4K 60 FPS** to fail on all three softwares. We also found that using two SSDs is significantly faster than using one SSD.

Here are some additional notes:

- We were unable to properly assess disk usage for Unreal and Ventuz applications.
- Note that a 60 FPS video clip is the equivalent of two 30 FPS video clips to decode.
- At this point, this is not a PRO Video Playback solution so there is no guarantee on synchronous playback in between files or machines.

Test System Description

- Xeon 3.39 Ghz
- 96 GB
- Samsung MZVKW512HMJP-000L7 (C:)
- ATA Samsung SSD 860 SCSI Disk (D:)



Troubleshooting

As of engine 4.18, most player Plugins have had extensive logging added, and will provide detailed logging of potential problems.

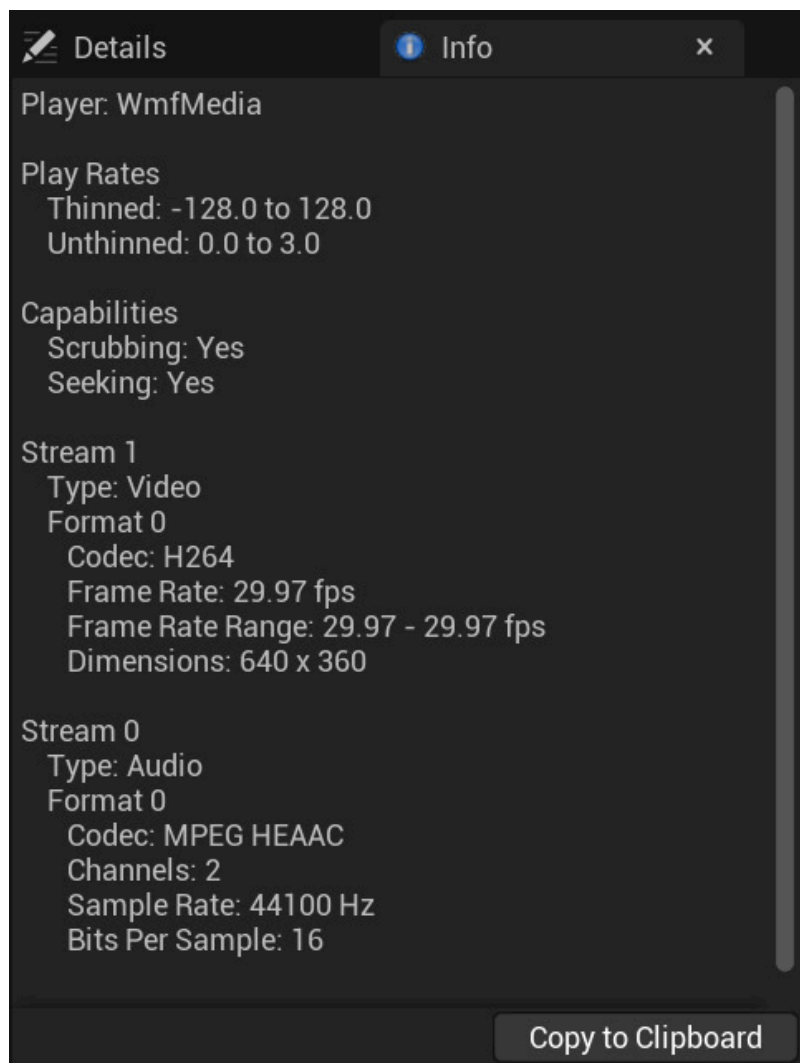
To enable logging, add the following to your project's **Engine.ini** file:


```
1 [Core.Log]
2 LogMedia=VeryVerbose
3 LogMediaAssets=VeryVerbose
4 LogAndroidMedia=VeryVerbose
5 LogAvfMedia=VeryVerbose
6 LogMfMedia=VeryVerbose
7 LogPS4Media=VeryVerbose
8 LogWmfMedia=VeryVerbose
9
```

 Copy full snippet

If a media source fails to open or play, check the **Output Log** for additional details.

You can also check information about a Media Source inside the Media Player asset and **Info** tab.



The screenshot shows the 'Info' tab of a media player asset inspector. The title bar has 'Details' (with a pencil icon), 'Info' (with an 'i' icon), and a close button 'x'. The main content area displays the following information:

- Player: WmfMedia
- Play Rates
 - Thinned: -128.0 to 128.0
 - Unthinned: 0.0 to 3.0
- Capabilities
 - Scrubbing: Yes
 - Seeking: Yes
- Stream 1
 - Type: Video
 - Format 0
 - Codec: H264
 - Frame Rate: 29.97 fps
 - Frame Rate Range: 29.97 - 29.97 fps
 - Dimensions: 640 x 360
- Stream 0
 - Type: Audio
 - Format 0
 - Codec: MPEG HEAAC
 - Channels: 2
 - Sample Rate: 44100 Hz
 - Bits Per Sample: 16

A 'Copy to Clipboard' button is located at the bottom right of the panel.

This will show what options are available for the media source, as well as the different Audio and Video streams and information about each stream.

Windows

The Windows Media Foundation (WMF) player Plugin WmfMedia has a number of limitations for MP4 containers.



See Microsoft's [MPEG-4 File Source](#) page for more information.

Known Issues & Limitations

Media Framework is still in development, and there are some known issues and limitations, as outlined below:

- **Android**

- Audio currently plays via the OS and cannot be piped through UE5's Sound System.

- **Editor**

- The Playlist UI inside the Media Player always has a playlist created internally.
- Playlists can be saved from the Media Player; however, there is no way currently to edit the playlist in the Media Player.

Workaround: Save a Playlist from the Media Player then open and edit the Media Playlist asset.

- **Sequencer**

- Media Playback will not render out properly when recording cinematics from Sequencer.
 - We have this scheduled to release in the near future as we know this is important.
- The ImgMedia Plugin will be synced with Sequencer.

- **Windows**

- QuickTime movie (.mov) support on Windows 7 and later is currently unreliable.

Workaround: This format is currently not recommended for use.

- **Linux**

- Currently, the WebM Media Player is the only available media player on Linux.