RENDERHEADS

AVPro Windows Media

Unity plugin for fast flexible video playback

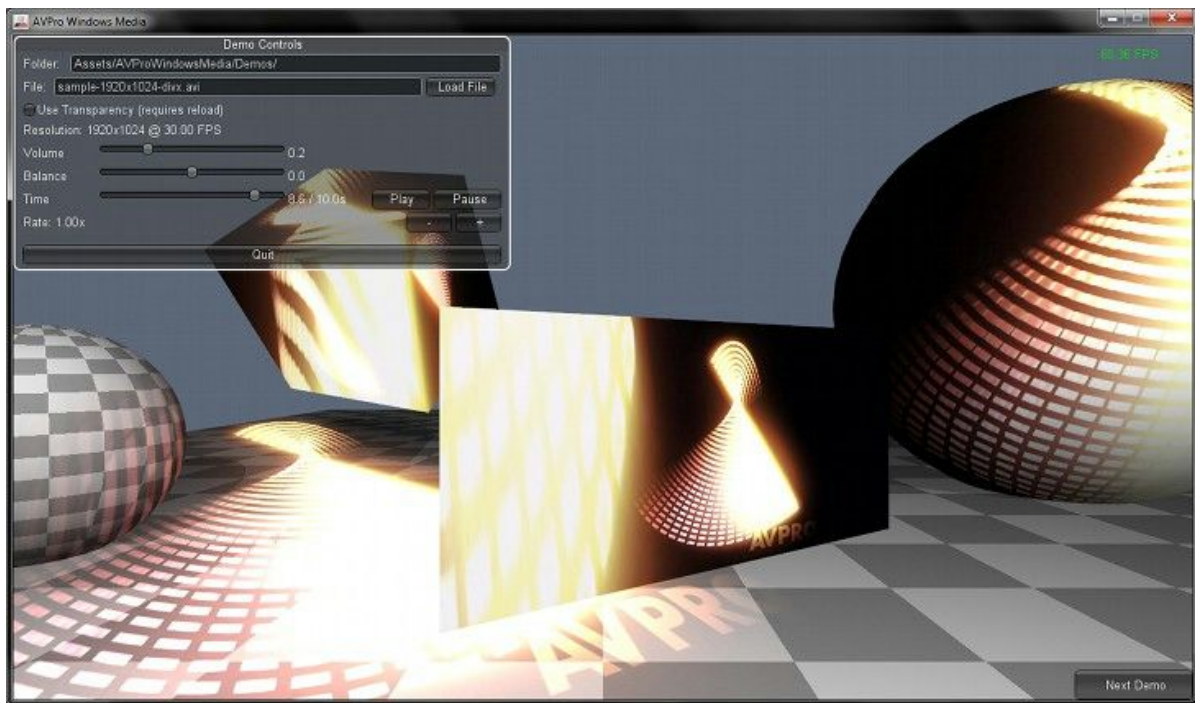*Version 2.90*

Released 24 February 2016

Contents

1. Introduction

"AVPro Windows Media" is a plugin for Unity that allows fast and easy playback of supported Microsoft DirectShow media.

The plugin is aimed at the high-end user group that require video playback features beyond Unity's built-in video support.

DirectShow is a trademark of Microsoft Inc., registered in the U.S. and other countries.

1.1 Features

- Unity 4.6 - 5.4 supported
- Both 32 and 64-bit friendly
- Free watermarked trial version available
- Years of professional use in the field
- Fast native Direct3D9, Direct3D11 and OpenGL texture updates
- Frame accurate seeking and scrubbing
- Supports video transparency
- Native Hap codec support (no plugin needed)
- Play videos from disk or from memory
- Supports internal clips
- 7.1 audio channel mixer
- AV sync offset control

- Unity 4.6 uGUI support.
- NGUI support.

Not ideal for 4K playback - unless using the Hap or H.263 codec (see 4K notes below)

1.2 Trial Version

We offer an unlimited trial version of AVPro Windows Media for download from our website. The trial version has no missing features or time restrictions but it does apply a watermark to the rendered output.   The watermarking shouldn't affect performance.

1.3 Media Credits

sample-sphere-1024x512-divx - (c) copyright Daniel Arnett, https://vimeo.com/97887646

Audio in sample video created by Citizen Nyx (http://ccmixter.org/people/nyx)

2. System Requirements

- Unity Pro 4.6 - 5.4
- Desktop Microsoft Windows platform (32-bit and 64-bit)
- Codecs for any video/audio you want to play

2.1 Platforms not Supported

- WebGL
- WebPlayer
- Mobile, Android, iOS, Windows Phone
- Mac
- Linux
- Windows Metro / Store Apps (Note: Windows Metro apps don't support DirectShow which this plugin is built upon. This plugin only supports Windows desktop apps)

*** Oculus Rift: Not officially supported (yet), but feel free to test it for yourself with the trial version. We have heard from people that have successfully used it.*

3. Installation

1. Import the **unitypackage** file into your Unity project.

2. Make sure you remove any old copies of the plugin DLLs (eg in your /Assets/Plugins folder).

3. Move the DLL plugin files to the appropriate folder:

   a. In Unity version 4.x move the **Assets/AVProWindowsMedia/Plugins** folder to **Assets/Plugins**.

   b. In Unity version 5.x the above can be done, or the **plugin inspector** can be used. Click on each DLL and assign the appropriate platform settings.

      For the 32-bit plugin (in the /Plugins/x86 folder), assign the Editor and Standalone platforms. For the Editor platform assign x86 and Windows only. For the Standalone platform assign x86 only:
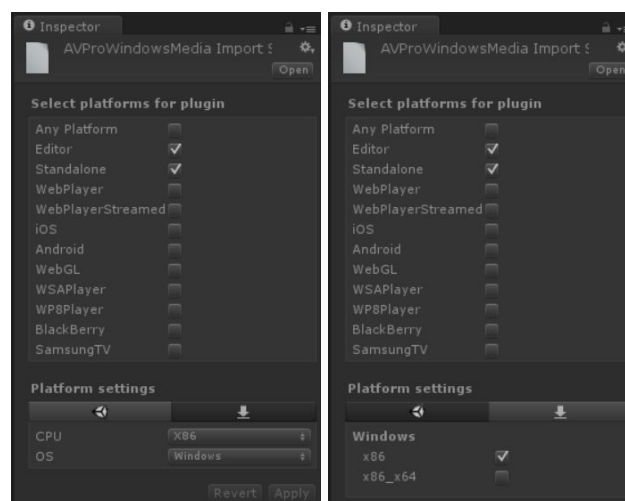
      For the 64-bit plugin (in the /Plugins/x86_64 folder), assign the Editor and Standalone platforms. For the Editor platform assign x86_64 and Windows only. For the Standalone platform assign x86_64 only:
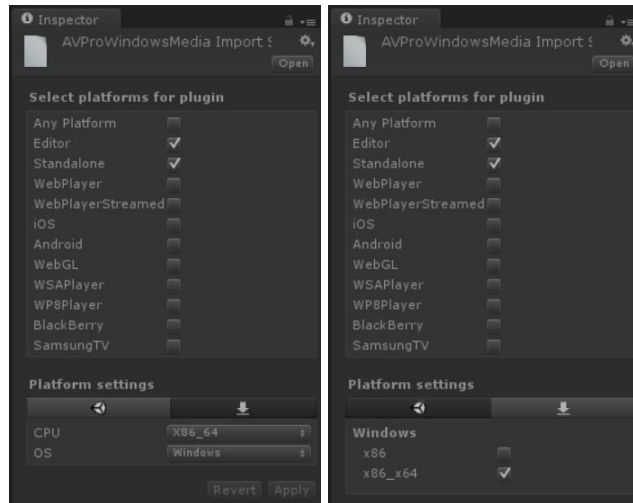
Make sure you only have a single copy of the plugins in your project.

4. Ensure you have the relevant **codecs installed** for the content you want to play.

3.1 Trial Version Upgrade Notes

If you are upgrading from the trial version, make sure you delete the old /Assets/Plugins folder as this contains the trial plugin and could conflict.

If you are using Unity 4.x you will also then need to copy the new /Assets/AVProWindowsMedia/Plugins folder contents to your /Assets/Plugins folder.

3.2 Video File Locations

The recommended place to put video files is in the **/Assets/StreamingAssets/** folder in your Unity project. This is a special folder that Unity copies to the build without processing, so you don't have to remember to copy over the video files to your build manually each time. However you are of course free to store your video files in any location.

You can use the BROWSE button to browse to a media file. Make sure you have the required codecs installed to play your video file.

This plugin is can be used to play video files from any location disk (normal folder + filename). You should place video files in a folder relative to the root of your project (the folder above /Assets). If you placed your videos in a MyVideos folder then you would set the 'folder' field to MyVideos and add the name of the video file to the 'filename' field. Ideally videos shouldn't be placed inside your /Assets folder as Unity will then try to import them.

When you make a build you need to make sure your build folder contains the same video folder structure. This is done either with a manual copy or a build script that copies the

folder over.

Another option is to put the video files in your Resources folder and use the AVProWindowsMediaMovieFromResources component. These videos will then be built directly into your build (no external files) and the videos will be loaded into memory. This approach can work well but not all video formats are supported and if the files are very large it can take a lot of memory.

4. Unity Components

This asset includes a number of Unity script components that allow use of the asset without any scripting. Naturally scripting can be used to control these components if necessary.

4.1 AVProWindowsMediaManager

There must always be exactly one **AVProWindowsMediaManager** in your scene when you use this plugin.   It is also important that this component starts before the other **AVProWindowMedia** components (controllable via Script Execution Order setting).

4.2 AVProWindowsMediaMovie
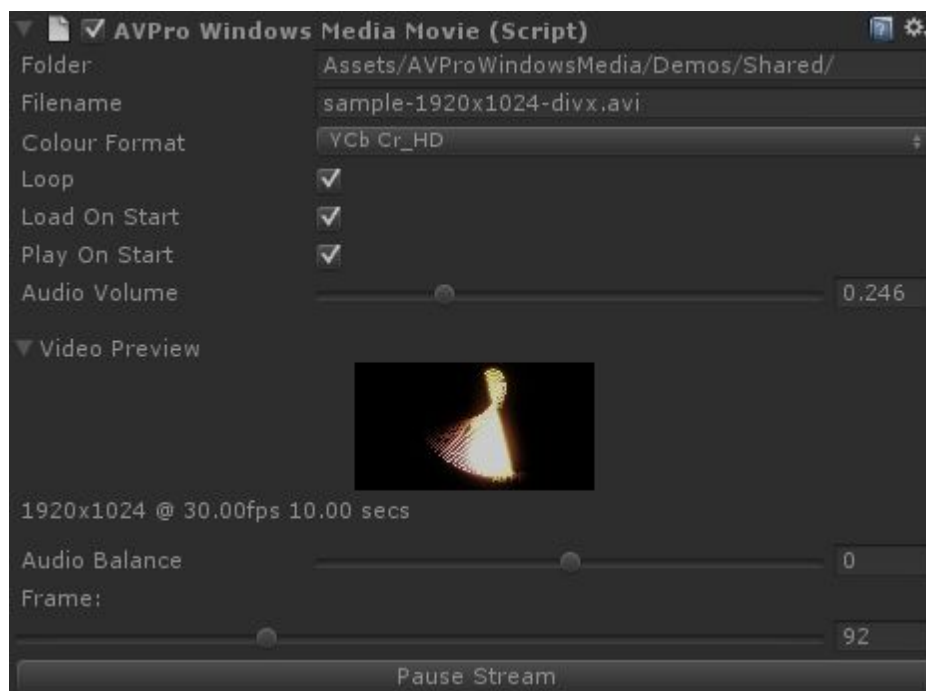
This component represents a single piece of media (video or audio) that can be loaded and played. The colour format is the internal format that is used to play the video and can affect quality and performance. the options are:

- "YCbCr_HD" (default): Fast playback using the Rec. 709 YUV colour conversion. This option is used for move videos, including all Hap videos.
- "YCbCr_SD": Fast playback using the Rec.601 YUV colour conversion.
- "RGBA32": This mode is slower but it allows for videos with alpha channel. Hap videos shouldn't use this setting, even if they have transparency.

The colour format cannot be changed once the video is playing.

When the editor is playing additional controls are displayed showing you the contents of the video which is useful during development. Note that having the video preview visible can affect frame rate as it forces update of the Unity UI.

This component simply plays the movie and doesn't display it on the screen. For display take a look at the components below.

4.3 AVProWindowsMediaGUIDisplay
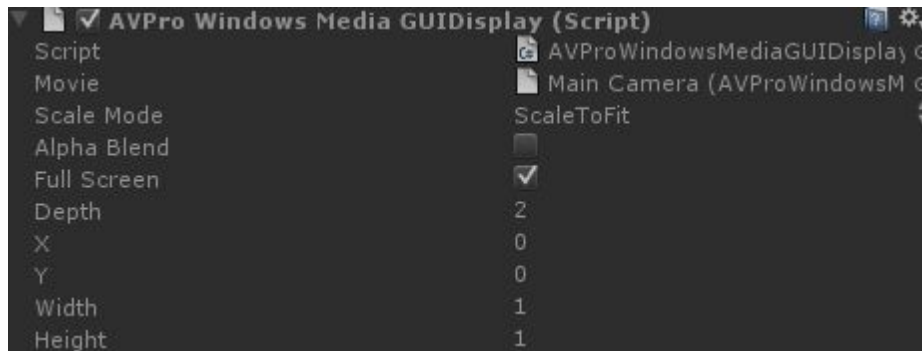
This component displays an AVProWindowsMediaMovie on the screen using Unity's IMGUI system. The IMGUI is always rendered last in Unity so use another component below if you want to display the movie in the main scene or using uGUI. To use, simply select the **AVProWindowsMediaMovie** component you want to display in the "Movie" option. Next you can set the placement of the item on the screen or use the fullscreen default.

4.4 AVProWindowsMediaUGUIComponent

This component displays an AVProWindowsMediaMovie on the screen using the new uGUI system introduced in Unity 4.6. Simply select the **AVProWindowsMediaMovie** component you want to display in the "Movie" option.

4.5 AVProWindowsMediaApplyUITextureNGUI

This component displays an AVProWindowsMediaMovie on the screen using the the NGUI library. Simply select the **AVProWindowsMediaMovie** component you want to display in the "Movie" option.

4.6 AVProWindowsMediaMeshApply

Use this component to apply an **AVProWindowMediaMovie** to all of the materials used by a mesh in

your scene.

4.7 AVProWindowsMediaMaterialApply

Use this component to apply an **AVProWindowMediaMovie** to a material in your scene. Optionally a texture name can be specified to override a specific texture slot in the material.

5. Use Case Notes

Below are some handy notes we have collected to help with the most common use cases of this plugin.

5.1 Transparent Video

There aren't many codecs that support transparent (alpha) video on the PC. Codecs we have used are:

- Uncompressed RGBA
- Lagarith
- Hap Alpha

There are also some codecs that look promising but we haven't had luck with:
- VP8/VP9 WebM - supports alpha but the DirectShow codec doesn't seem to.
- VP6 - popular Flash codec, looks great but we couldn't find a DirectShow codec.

All codecs we have found use straight alpha, not premultiplied alpha.

5.2 Video Clips

Often it useful to export a single video containing all of the sub-clips rather than having multiple video files. This uses much less memory and is often easier to manage. Our plugin supports specifying clips within a video and allows easy playback and looping of those clips.
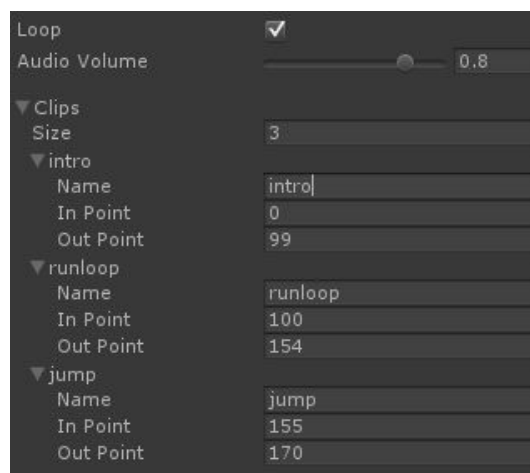
In the AVProWindowsMediaMovie component you can specify these clips and their frame ranges. Playback and looping of these clips can then be started by calling PlayClip() in the script. For best results videos with all key-frames should be used.

5.3 Video Scrubbing

Often people want to be able to rapidly scrub backwards and forwards through videos, even playing them at variable speeds. While this plugin does support this ability it isn't one of its strengths. Our other plugin AVPro QuickTime actually gives the best scrubbing results due to QuickTime having better support for this than DirectShow. DirectShow scrubbing is limited to about 15 frames per second.

Smooth scrubbing also only works best when using a video codec with only keyframes. We recommend a fast codec such as Hap for this.

This plugin contains functions to change the playback rate of a video, but it doesn't support negative (backwards) playback rates. If the audio is uncompressed then it will change pitch as the playback rate changes. If you pause the video and scrub through is manually (by rapidly seeking frames) this is possible but audio will not be played due to being paused).

5.4 Super High Resolution 4K / 8K Playback

For UHD playback we recommend using either XVid or Hap codec as both of these codecs are fast to decode and most other codecs require too much CPU power to decode at such high resolutions.

MP4 videos are usually unsuited for UHD playback as we do not have hardware H.264 decoding. However if a powerful machine is used then this could still work. If you must use H.264 then see our section on H.264 codec for tips on how to optimise H.264 for fast smooth playback. DirectShow also doesn't support H.264 decoding greater than 1080p by default, so you would need to install a 3rd party H.264 decoder and also change the default H.264 decoder in Windows via a Tool such as 'Codec Tweaker' (see notes in the FAQ section for details).

Hap is the best codec for this task as it plays extremely quickly. You'll need a lot of disk storage space though and a fast SSD drive to play from. See our section on Hap codec for more information.

We have found XVid / DivX to be a great codec for high resolution playback . It uses more CPU to decode than Hap, but much less than H.264. The files are also much smaller than Hap.

5.5 Low CPU Usage

Sometimes low CPU usage is important - for example when you need to play back multiple videos at the same time, or if you have something that is running constantly and want to keep hardware temperatures low.

For this we recommend using the Hap codec. See our notes on this for more information.

5.6 Multiple Audio Speakers

One use case requires the video to play its audio to different speakers and be able to change the speakers it plays to dynamically. An example of this is when multiple speakers are positioned a long a wall and only the speakers closest to the user are meant to play the audio. This plugin supports this ability through a 7.1 audio mixer. The volume levels of the 8 speakers can be set by using the function SetAudioChannelMatrix() with an array of 8 normalised float values.

5.7 Oculus Rift Virtual Reality

Playback of 3D videos in a virtual reality environment is a popular use case. We have not yet added direct support for this usage, however we have heard from people who have used this plugin successfully for this purpose.

The basic approach is to encode the left and right eye frames into a single video frame in a SBS (side-by-side) layout. A plane with the left side of the video mapped is created and tagged so that only the right eye camera can render it. The same is done for the right eye.

We hope to add more of our own support and samples for this in the future.

6. Playback Performance

6.1 Hardware

The first thing to check is your hardware to make sure it's suitable. You'll need a decent GPU and CPU with the specifications related to the resolution of video you're trying to play back and also the settings used during the encoding of the video. Use the Task Manager and Resource Monitor to help determine if there is too much load being placed on any one part of the system (HDD, CPU, GPU etc).

If the video file is very large or the video is a high frame rate then your hard drive speed could also be an issue. You should find out how many MB/s your drive can read and work out how many MB/s your video requires for playback (by using file size in MB / frames per second). For large videos (especially Uncompressed or Hap encoded videos) an SSD drive is recommended.

We have also had reports that systems with multiple GPUs (eg AMD CrossFire / NVidia SLI) can have jerky playback issues due to a phenomena known as micro-stutters. Currently we don't recommend such configurations for completely smooth playback.

6.2 Video Format

You should use the best suited video format for your intended use. Some codecs are great for compression, others are tuned for speed and most have many encoding options you can tweak to suit your needs.

If you don't mind using large videos then the Hap video codec is probably your best choice. Hap encoded videos use minimal CPU to decode and are compressed in a format that is suitable for direct upload to the GPU.

H.264 is usually one of the most CPU intensive codecs H.264 videos (commonly in a .mov or .mp4 container) are often highly compressed and this can make them slower to play back. The H.264 encoder has many many options that can be used to tune how the videos are compressed and how they play. It is important when encoding videos to select the best options for playback. One handy tool we use is FFMPEG.   FFMPEG can be used to convert videos for faster playback with the following command-line:

ffmpeg -i input.mp4 -c:v libx264 -pix_fmt yuv420p -preset veryslow -tune fastdecode -profile:v main -coder 0 -g 6 -crf 20 output.mp4

or even more aggressive:

ffmpeg -i input.mp4 -c:v libx264 -pix_fmt yuv420p -preset veryslow -tune fastdecode -profile:v main -crf 20 -refs 1 -coder 0 -g 30 -x264opts no-deblock -x264opts bframes=0 output.mp4

or even:

ffmpeg.exe -i input.mp4 -c:v libx264 -pix_fmt yuv420p -bf 0 -preset veryslow -tune fastdecode -profile:v main -coder 0 -g 25 -crf 10 -refs 3 -y output.mp4

The H.263 compatible XVid/DivX formats offer a good balance between speed, file size and quality.

6.3 Codec Tuning

Some codecs are configurable and can be tuned for better playback. The Lagarith codec for example has options for multi-threading which can be enabled. To view these options you can use software like Virtualdub to configure compression codecs.

6.4 Video Resolution

This is a minor point, but selecting the right resolution can impact on performance due to memory alignment for GPU texture updates. In order of preference:

1. Video width and height are both power-of-2 size (eg 256, 512, 1024, 2048, 4096).
2. Video width is power-of-2, height is any value.
3. Video width and height are multiples of 16 (perfectly divisible by 16).
4. Video width is a multiple of 16, height is any value.
5. Video width and height are a multiple of 4.
6. Video width is a multiple of 4, height is any value.
7. Video width and height are any value.

6.5 Multi-Threading

If your video plays smoothly in the editor but is jerky when you make a build then disabling Unity's multi-threaded rendering can help. Go to Player Settings, switch the Inspector to Debug disable and disable "MT Rendering":
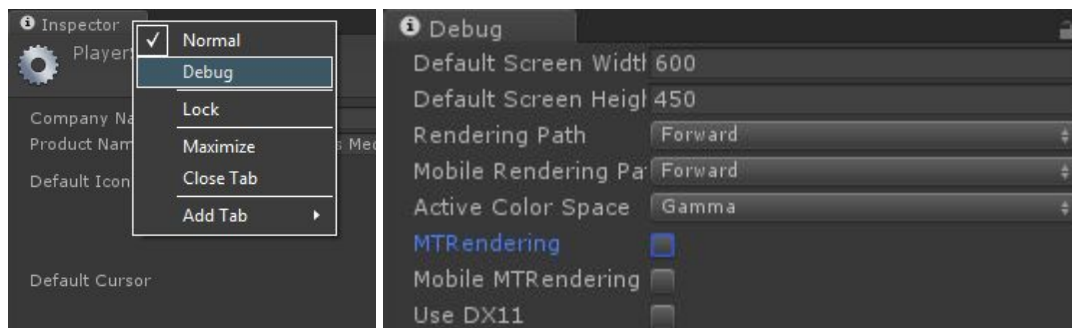
6.6 Display Frame-Rate Sync

Videos should be authored at a frame-rate that is a multiple of the device used to display them. For example a 30fps video can display smoothly on a 60hz monitor, but a 24fps video will not.

The AVProWindowsMediaMovie component has an option 'Display Sync' which attempts to give smoother playback by buffering frames internally and synchronising with the display monitor. Using this option results in a slight pause when starting or seeking videos due to the buffering, but this is offset by offering smoother playback. This option only works correctly when running full-screen and with vsync enabled. This option only works on fixed frame-rate videos. The full-screen mode must be set to EXCLUSIVE in the PlayerSettings.

6.7 Machine Setup

For best results the machine should have minimal background processes running as these can all affect application performance.

6.8 Garbage Collection

Garbage collection can cause a frame drop. Make sure your application isn't generating too much garbage, or follow Unity's guide for preparing the garbage heap so you can control collection better.

6.9 Latest version

You should also check the Asset Store to make sure you're using the latest version of this plugin, and using the latest supported version of Unity.

*If you have questions about your file formats etc you can contact us with your questions.

7. Codec Guide

Video is a complex topic so we hope the small guide below helps you get the most out of this plugin.

7.1 Formats

This plugin supports a variety of video formats via built-in native support but mostly via the use of the Microsoft DirectShow video system.

Videos are stored within container files which come in various formats. These container files contain data streams (audio, video etc) which are each encoded in various formats by codec (coder-decoder) software.

File Formats / Containers include:

- AVI
- MOV
- MP4
- MKV
- FLV
- OGG

For a more complete list: http://en.wikipedia.org/wiki/Comparison_of_container_formats

Different container formats have support for different number of data streams (eg multiple audio tracks), subtitle features etc. Within the container can be multiple streams (video, audio, text etc) and these are in different formats based on the codec used to encode/compress them.

Video codecs include:

- Uncompressed (raw)
- H.263 (MPEG-4 part 2 ASP)
- H.264 (MPEG-4 part 10 AVC)
- Hap
- Lagarith

For a more complete list: http://en.wikipedia.org/wiki/Comparison_of_video_codecs

A video data stream compressed with H.264 can be stored within an AVI files, or a MOV file, or a MP4, or an MKV file. The container file is merely a wrapper that contains the video data. So not all videos with the same extension are equal as the video steam they contain could be encoded in any format.

7.2 Format Support

In order for software to be able to play a video file it must be able to:
1. Read the container file format
2. Decode the video and audio stream

This can be achieved if the software has native support for that container / codec format, or if it supports a plugin architecture that allows 3rd party plugins to be installed which add support for these containers / codecs.

Popular software such as Windows Media Player, VLC and Media Player Classic have built-in support for many containers and codecs, but also support 3rd party plugins via plugin architectures like Windows Media Foundation, DirectShow and the older VFW/ICM system.

The Windows operating system comes with many MWF / DirectShow video 'filters' installed by default. These filters add support for other file formats and other codecs.

Filters that parse container files and produce the video/audio/subtitle stream data are called 'splitters'. The output of the splitters are then passed to decoder filters that decompress and decode the media steam and produce the raw output frames.

7.3 Container Format Support

This plugin relies on DirectShow filters/plugins for all container file format support. The following tables shows that Windows doesn't ship with many DirectShow container filters by default:

DirectShow support built in to Windows	Requires 3rd party DirectShow filters
AVI	MP4
WMV	MOV
MPG	MKV
	FLV
	WEBM
	OGG

To add DirectShow support for these other container formats 3rd party plugins must be installed. Non-commercial splitter plugins we have used in the past include:

Haali Media Splitter http://haali.su/mkv/
Adds support for opening of MP4, MKV, OGG/OGM files

LAV Filters https://github.com/Nevcairiel/LAVFilters/releases
Adds support for opening pretty much all of the above

GDCL MPEG-4 http://www.gdcl.co.uk/mpeg4/
Amongst other things adds support for opening of MP4 and MOV files. Requires command-line installation (see FAQ for steps).

Xiph.org http://xiph.org/dshow/
Adds support for opening OGG, WEBM and others

7.4 Codec Format Support

3rd party video codecs are sometimes also required as Windows has limited built-in support for codecs:

Directshow support built in to Windows	Requires 3rd party DirectShow filters
Uncompressed	
H.264 (Windows 7 and above, and only up to 1080p Main profile)	H.264 (larger than 1080p, or High profile)
Motion JPEG (only with MJPG 4cc)	H.265
H.263 (Xvid/DivX, not in some older Windows versions)	VP8 / VP9
WMV 7-9	Lagarith
MPEG-1 / MPEG-2 (DVD)	
Cinepak (very old)	
Microsoft RLE (very old)	
Microsoft Video 1 (very old)	
	Many more..

This plugin has native support for the Hap codec (including Hap Alpha and Hap Q flavours). All other video codec support is provided via either built-in or 3rd-party DirectShow filter plugins.

Non-commercial codec plugins we have used in the past include:

Lagarith http://lags.leetcode.net/codec.html

LAV Filters https://github.com/Nevcairiel/LAVFilters/releases
Adds support for pretty much all codecs, including larger than 1080p H.264 decoding. Note

that for overriding built-in OS codecs such as H.264, tweaking may be required (see other notes in FAQ section).

7.5 Keyframes

A video contains many frames, but not all frames are stored equally in the video stream. Selecting the best type of frame type during video encoding can help to ensure your videos play back smoothly, or are easily scrubbable, or are fast to seek - whatever your requirement.

The important elements are covered here, but for more details you can read:
http://en.wikipedia.org/wiki/Video_compression_picture_types

The summary is:

- The more keyframes you have in your video the faster it will be to seek/scrub through.
- If you set a keyframe interval and want to seek (eg to a contained clip), make sure that all of your clips start on a keyframe.
- For smooth playback it is often useful to avoid using B-frames as this can add to the decoder CPU usage.
- Codecs are highly tunable, use the right encoder settings for your use case.

7.5.1 Intra-Frame Compression (I-Frames)

Some compression techniques simply compress each frame and so each frame is completely independent from other frames. These frames are called I-Frames or Keyframes. This is a simply method of compression and because each frame is a pure frame it means that seeking to any part of the video is very fast. The only downside is that these videos tend to be larger and can sometimes be slower to decode depending on their content. Codecs with only I-Frame support include:

- Uncompressed
- Lagarith
- Hap
- QuickTime Animation
- Prores

Most other codecs (including H.264) have options to tune them to use I-frames only.

7.5.2 Inter-Frame Compression (P/B-Frames)

The above intra-frame compression doesn't take a very important feature of video into account: subsequent frames tend to be very similar. This means storing only changes that happen between a series of frames can result in higher compression.

This works by first storing an I-frame which is a complete frame, and then storing a series of P-frames (predictive frames) that only contain the changes to the I-frame. Typically like this:

I-P-P-P-P-P-P-P-P-P-P-P-P-P-P-P-I-P-P-P-P-P-P-P-P-P-P-P-P-P-P-P I-P-P-P-P-P...

P-frames are commonly used, but increasingly common are B-frames (bi-directional predictive frames) which not only use data from a previous I-frame, but also from the next I-frame, resulting in potentially more compression.

Using P/B-frames means more compression, but it can also makes the task of decoding the video trickier for the decoder. If a video seeks to a P/B-frame it isn't able to decode the picture immediately as it must first decode the I-frame and all of the P/B frames leading up to it. This makes seeking and scrubbing slow. B-frames also tend to be more expensive to decode than P-frames.

Almost all modern video codecs use a combination of intra and inter-frame compression. Codecs include:

- H.264 (MPEG-4 part 10 AVC)
- H.263 (MPEG-4 part 2 ASP)
- MPEG-2
- VP8 / VP9

These codecs are all highly tunable, and P/B frames can be turned off completely, or the rate of I frames can be set.

7.6 Fixed vs Variable Frame-Rate

Videos can be encoded with a fixed or variable frame rate. Variable frame rate is common when the video source doesn't generate frames at a fixed rate - eg recording from a webcam. Some codecs also have options that allow encoding to be done at a variable frame rate based on the contents of the media. For best results you should always use a fixed frame-rate.

7.7 Specific Codecs

7.7.1 H.264 (MPEG-4 part 10 AVC)

H.264 is probably the most common video compression in use today. It is a very powerful and flexible format with many parameters that allow the the video to be encoded for different uses.

H.264 is most useful when small file size is important.

We recommend encoding H.264 videos for fast playback. This means disabling CABAC, disabling B-frames and a few other options. Most H.264 codecs have presets like 'ultrafast'

or hints like 'fastdecode' that can be selected. The resulting file size will be slightly larger but the playback will be smoother.

The DirectShow H.264 decoder than comes with Windows has some limitations that one needs to be aware of: it doesn't support the HIGH profile and it doesn't support videos greater than 1080p. To get around this a 3rd party H.264 decoder needs to be installed and the Microsoft codec needs to be disabled. See the FAQ for more details on this.

7.7.2 H.265 / HEVC

H.265 is the next generation of video compression and is only supported by this plugin when using 3rd party codecs.

7.7.3 H.263 (MPEG-4 part 10 ASP) / DivX / XVid

H.263 is the previous popular generation of video compression and is still very useful. Codecs such as XVid are ideal for high resolution video displays as they offer multi-threaded decoding and require much less CPU than H.264. The compression isn't as good as H.264 so the file size will be bigger but it's quite a good balance of quality, file size and CPU usage. Another benefit is that most Windows versions supports decompression natively so no 3rd party codecs are required.

7.7.4 WMV

Personally we do not use WMV as we feel it is not maintained or as advanced as some of the other options. One advantage is that most Windows versions support it natively, so no 3rd party codecs are required.

7.7.5 Lagarith

Lagarith is a great lossless codec that also supports transparency. Be aware that the multi-threading option isn't enabled by default and should be enabled for best performance. It's home page URL is: http://lags.leetcode.net/codec.html

7.7.6 Hap

The Hap codec is great for fast smooth playback, however the file sizes can be very large. Hap also supports transparency.

The AVPro Windows Media plugin has native Hap support so no 3rd party codec is required. However to encode videos as Hap the codec is needed.

DirectShow Hap codec URL: http://www.renderheads.com/portfolio/HapDirectShow/

Hap video width and height should be multiple of a 4.
Hap Alpha requires straight not pre-multipled alpha.

8. Video Tools & Software

8.1 Virtualdub

Virtualdub is a free Swiss army knife for video tasks and is great for encoding to AVI files using DirectShow codecs.

8.2 Adobe After Effects / Premiere

Not much to say about these professional tools except to check the encoding options as the defaults may not be best suited for high performance playback.

8.2 FFmpeg

FFmpeg is a very handy command-line tool which can convert videos into almost any format and has hundreds of options for tuning video encoding.

8.3 LAV Filters

Though we generally don't recommend codec/filter packs as they often do more harm than good (it's better to just install the codecs you know you need to use), if we had to recommend one it would be LAV Filters. Installing these filters opens up many more codecs to DirectShow. LAV filters are built from the same codebase as FFmpeg and is very stable and well written. The LAV filters also include an H.264 decoder that supersedes the one included with Windows as it allows playback of videos using the HIGH profile and with a higher than 1080p resolution.

9. Support

If you are in need of support or have any comments/suggestions regarding this product please contact us.

Website: http://renderheads.com/product/av-pro-windows-media/
Forum: http://forum.unity3d.com/threads/released-avpro-windows-media.142396/
Email: unityplugins@renderheads.com

9.1 Bug Reporting

If you are reporting a bug, please include any relevant files and details so that we may remedy the problem as fast as possible:

- Details of the video file you're having problems with
    - Resolution
    - Codec
    - Frame rate
- Better still, include a link to the video file.
- The console/output log if relevant
- The version of Unity you are using.
- The version of AVPro Windows Media plugin you are using.
- The version of Windows you are using.
- Details of any codec packs you have installed.

10. About RenderHeads Ltd

RenderHeads Ltd is an award winning creative and technical company that has been designing and building cutting edge technology solutions since its formation in 2006. We specialise in creating unique interactive audio-visual software for installations at auto shows, museums, shows and expos.

10.1 Services

- Unity plugin development
- Unity game / interaction / augmented reality development
- Unity consulting

10.2 Our Unity Plugins

AVPro QuickTime

AVPro Windows Media

AVPro Movie Capture

AVPro Live Camera

Appendix A - FAQ (Frequently Asked Questions)

A1.1 Installation

1. **How do I fix the error: "DLLNotFoundException"?**

   You need to move/copy the "Plugins" folder from your "AVProWindowsMedia" folder into the root of your folder structure. This means the "Plugins" folder should be moved to your "Assets" folder.

2. **How do I fix the error: "DLLNotFoundException" where it seems to be looking for the 64-bit DLL?**

   Sometimes Unity gets confused and will try to load the 64-bit DLL in the 32-bit editor. We found that to fix this you must:

   a. Open Build Settings
   b. Select Web Player platform and press Switch Platform
   c. Select PC and Mac Standalone (with Target platform set to Windows not Windows 64-bit) and press Switch Platform

A1.2 Scripting

1. **I have compiled the scripts into a DLL and am now experiencing some unexpected behaviour.**

   Some of our scripts have Unity version-specific preprocessor defines which determine how they compile (eg UNITY_4_0). Usually when you build an external DLL these defines are missing and so the incorrect version of the code can be compiled. You need to add the appropriate compiler defines to your build.

2. **Frame seeking doesn't seem to be working properly?**

   To seek you need to set the PositionFrame property to the frame number you want to seek to. Frame seeking happens asynchronously and frames are buffered internally so there are actually 2 values for the current frame. The PositionFrame property is the frame that the internal playback system is currently decoding. The DisplayFrame property is the frame that is currently displaying. When you set PositionFrame it make take a few frames before DisplayFrame becomes equal to the frame number you seeked to.

A1.3 Codecs

1. **What codecs are needed to play MP4/MKV files?**

   Install Haali Media Splitter (http://haali.su/mkv/)

   or

   Install GDCL Mpeg-4 filter (http://www.gdcl.co.uk/mpeg4/)

2. **What codecs are needed to play MOV files?**

   Install GDCL Mpeg-4 filter (http://www.gdcl.co.uk/mpeg4/)
   It should allow playback of most MOV files, but some codecs (such as Hap) are not
   supported by default.

3. **How do I install GDCL MPEG-4 Filter?**

   a. Download the mpeg4.zip file from http://www.gdcl.co.uk/mpeg4/
   b. Extract the zip to somewhere on your system (eg C:\Tools\mpeg4)
   c. Create a file called register.bat and edit it in notepad and paste the following:

      regsvr32.exe C:\Tools\mpeg4\x86\mp4mux.dll
      regsvr32.exe C:\Tools\mpeg4\x86\mp4demux.dll
      regsvr32.exe C:\Tools\mpeg4\x64\mp4mux.dll
      regsvr32.exe C:\Tools\mpeg4\x64\mp4demux.dll
      pause

   d. Save the file and close notepad
   e. Right click on the register.bat file and choose "Run as administrator"
   f. The script should run and register all 4 DLLs to your system

4. **How can I play DVD VOB files?**

   By default Windows doesn't come with DirectShow filters to decode the audio and
   video of VOB files. Installing the LAV filter pack should solve this.

A1.4 Playback / Performance

1. **My video doesn't play smoothly, how can I make it play smoother?**

   See playback performance section of this document.

2. **Video playback is jerky on high-end GPU system, how can I fix it?**

If you have a top-end GPU setup there could be micro-stutter issues. We don't recommend using multiple GPU configurations (eg AMD CrossFire / NVidia SLI) with our plugin.

3. **How do I get transparent videos to play correctly?**

First verify that your video does indeed contain an alpha channel.   Set the "Color Format" field on the AVProWindowsMediaMovie component to RGBA32 (unless using the Hap Alpha codec, in which case leave it set to YCbCrHD). If you are rendering the video texture using the Unity GUI then make sure you have the "alphaBlend" parameter set to true. If you are rendering the video texture using a material in the 3D scene then make sure the material supports alpha blending.

3. **I'm trying to play a very high quality H.264 video and I only get a blank screen with audio playing, how can I fix it?**

Windows 7 and 8 have their own preferred list of codecs and (at least in Windows 7) the DirectShow video decoder is unable to play H.264 videos encoded with the "High" profile only handle "Simple" and "Main") or videos larger than 1080p.

1) Install LAV filters
2) Install Codec Tweaker
3) Run Codec Tweaker, go to "Preferred decoders" and change "H.264" from "Microsoft" to "LAV Video".

4. **I'm trying to play back a high resolution H.264 video but it doesn't display and reports an incorrect resolution, how can I fix it?**

The built-in DirectShow H.264 decoder in Windows is quite limited and doesn't support resolutions higher than 1080p. Follow the steps in the FAQ above should solve it.

A1.5 Other

1. **Which of your Unity video-playback plugin is more suitable for me: AVPro Windows Media or AVPro QuickTime?**

If you need easy cross-platform (PC-Mac) video support then AVPro QuickTime is the only way to go as AVPro Windows Media only supports the Windows platform. It is possible to use both plugins together and get the best of both worlds but this would require some scripting to create a basic wrapper to encapsulate both plugins.

If you are focusing on Windows PC only and need high performance then we recommend using AVPro Windows Media as we have found QuickTime support on PC to be lacking (especially in multi-threaded video codec performance) since the QuickTime engine for PC hasn't been updated in some time.

Using QuickTime though can be easier as it only requires a single install whereas on the PC you need to know what codecs you need to support and potential install multiple codecs.

You can contact us with your requirements if you're not sure. We also have downloadable demos on the website which you can use for testing. Here is a table to help you decide:

	AVPro QuickTime	AVPro Windows Media
Windows PC Support	Yes	Yes
Mac Support	Yes (not in Unity 5.x)	No
64-bit support	No	Yes
Good performance playing multiple HD videos on Windows PC	No	Yes
Scrubbing*	Very fast (codec dependent)	Fast (codec dependent)
Requires additional installs	Only on Windows PC - you must install the QuickTime player	Only if you want to support codecs that Windows doesn't natively expose to DirectShow
ProRes Codec	Yes	No
MP4 Container File	Supported	Must install a codec

*AVPro Windows Media is based on DirectShow which inherently only allows scrubbing at a maximum of 15-20fps even with the ideal video codecs. QuickTime is able to achieve at least 30fps during scrubbing when the ideal video codecs are used.

Appendix B - Version History

- **Version X**
  - Add subtitle support?
  - Add multiple soundtrack support?
  - Add support for 4:2:0 planar pixel formats?
  - Add support for Unity 5.0 YUY2 format?
  - Improve code documentation?
  - Add support or CLSID_CColorConvertDMO?
  - Add Media Foundation support?
  - Add support for loading WMV and MP4 from memory
  - Add mip-map generation support
  - Audio mixer to handle 48khz audio tracks and not crash
  - Support synchronised video playback
  - Support multi-stream MKV videos?
  - Add ffmpeg integration (ffmpeg generated file)
  - Add stop for clip playback and event for end of clip
  - Add more events
  - Add aspect-ratio correct uGUI component
  - Look at performance bug when UI is hidden
  - Pipe audio back to Unity for 3D VR audio rotation
  - Test clips not looping
  - Update FrameExtract demo to show that OutputTexture can be RenderTexture or Texture2D
    http://answers.unity3d.com/questions/1018096/casting-from-texture-to-texture2d-returns-null.html
  - ← Your suggestion here

- **Version 2.90 - Wednesday 24 February 2016**
  - Dropped support for Unity versions below 4.6, 4.6 is the new minimum
  - Added 360 degree video demo
  - Added two uGUI demos
  - Added background demo
  - Improved AVProWindowsMediaMovie component, it now allows browsing to files and can report problems with the setup such as missing files, sub-optimal performance settings or warning that video files will need to be copied for builds
  - All sample videos now moved to StreamingAssets folder
  - All shaders removed from Manager script and now loaded from resources
  - Fixed uGUI issues related to API changes in Unity 5.2
  - Fixed flipped videos issue related to changes in Graphics.Blit in Unity 5.3
  - Added handy prefabs
  - AVProWindowsMediaManager now added dynamically if missing

- **Version 2.80 - Monday 15 June 2015**

- ○ Added Unity 5.1 support
- ○ Audio can now be completely disabled
- ○ Can now preview videos in the editor without playing
- ○ Better support for 'StreamingAssets' path
- ○ Can now set the texture filter and wrap type
- ○ Improved looping when using DisplaySync
- ○ Fixed some off-by-one bugs related to last frame
- ○ Fixed watermarking crash bug in trial version

- **Version 2.70 - Friday 13 February 2015**
  - ○ Added Unity 4.6 uGUI support
  - ○ Added Unity 5.0 beta support
  - ○ Support for Unity 3.x dropped
  - ○ Removed lots of legacy code used to support older versions of Unity
  - ○ Added 'ignore flip' feature which improves performance for videos that don't require format conversion to RGBA32 and only require flip/orientation changes. Such videos formats include uncompressed RGBA32, Hap / Hap Alpha, Lagarith RGBA32.
  - ○ Improved performance of RGBA32 videos in Unity 5 DX11 due to not having to swap red-blue channels
  - ○ Improved preview with alpha channel view
  - ○ Improved manager with overview of all videos
  - ○ Trial watermark performance improvements
  - ○ Improved documentation

- **Version 2.64 - Thursday 1 May 2014**
  - ○ Fixed textures not being released in editor
  - ○ Added texture names
  - ○ Memory buffers changed to 16-byte alignment

- **Version 2.62 - Saturday 29 March 2014**
  - ○ Performance improvements
  - ○ Improved Hap codec support

- **Version 2.6 - Tuesday 25 March 2014**
  - ○ Added display sync support (BETA) allows smoother playback when running full-screen with vsync by synchronising the frame display rate with the monitor refresh rate.
  - ○ Added support for linear colour-space (BETA)
  - ○ Optimisation for Unity 4.3 allowing for less CPU usage during texture updates in DirectX9 and DirectX11 modes (BETA).
  - ○ Added demo for playing back videos from Resources.Load()
  - ○ Added native support for Hap video codec
  - ○ Made audio delay feature optional
  - ○ Fixed bug in variable frame-rate WMV files
  - ○ Fixed bug in frame-stepping
  - ○ Fixed handle recycling bug

- ○ Fixed thread leak bug

- **Version 2.5 - Monday 10 September 2013**
  - ○ Added audio delay support for perfect av-sync
  - ○ Added multi-channel audio mixing support and demo
  - ○ Added audio playback support (WAV, MP3 and MIDI)
  - ○ Fixed frame extract demo
  - ○ Fixed Unity 4.2 support
  - ○ Added support for upcoming Unity 4.3

- **Version 2.4 - Wednesday 30 May 2013**
  - ○ Added support for playing videos from memory
  - ○ Added new frame extract demo
  - ○ Improved internal frame buffering
  - ○ Improved instance handle recycling
  - ○ Fixed some bugs

- **Version 2.3 - Monday 29 April 2013**
  - ○ Added Unity 4.1 non-pow2 texture support - optimisation
  - ○ Added new demo that plays multiple videos simultaneously
  - ○ Added frame-rate display to inspector
  - ○ Added IsFinishedPlaying property
  - ○ Fixed lost texture bug when pausing and dragging window
  - ○ Fixed 64-bit crash bug when resizing window
  - ○ Minor bug fixes
  - ○ Added new FAQs to documentation

- **Version 2.22 Monday 18 March 2013**
  - ○ Added Unity 4.1 support
  - ○ Fixed some platform #if issues

- **Version 2.2 Monday 4 March 2013**
  - ○ Added a new demo that plays a queue of videos
  - ○ Improved AVProWindowsMediaManager
  - ○ Optimised and improved pixel format conversion
  - ○ Removed last traces of Overlay mode
  - ○ Added unique plugin ID for GL.IssuePluginEvent()
  - ○ Added path resolve for relative paths when working path isn't that of the EXE
  - ○ Fixed a rare null pointer bug in the plugin

- **Version 2.1 Thursday 20 December 2012**
  - ○ Fixed a bug in Unity 4.0 when playing multiple videos

- **Version 2.0.0 Wednesday 5 December 2012**

- ○ Added Unity 4.0 support
- ○ Added Unity 4.0 support for DX9 & DX11 textures updates
- ○ Improved speed and accuracy of pre-roll to get initial frame
- ○ Can now seek while paused
- ○ Fixed bug playing files > 4GB
- ○ Added better reporting for missing DLL

- **Version 1.50 - Wednesday 4 July 2012**
  - ○ Added 64-bit build.
  - ○ More powerful components for non-scripters.
  - ○ Improved names of components to make them unique.
  - ○ Improved demo scenes.
  - ○ Added in-editor preview of videos.
  - ○ Added support for YCbCr Rec.709 colour space.
  - ○ Frame seeking added.
  - ○ Pre-roll added.
  - ○ Improved memory cleanup.
  - ○ Fixed looped video playback bug.
  - ○ Documentation improved.

- **Version 1.40 - Wednesday 3 May 2012**
  - ○ Fixed OpenGL texture update in Unity3.5.
  - ○ Volume is now linear.

- **Version 1.35 - Saturday 10 March 2012**
  - ○ Fixed a thread leak in source filter.
  - ○ Fixed an audio handle leak.

- **Version 1.3 - Friday 17 February 2012**
  - ○ Fixed a memory leak in RenderTexture.
  - ○ Improved API for programmatically loading videos.
  - ○ Fixed blank video frame from appearing.
  - ○ Switching between videos of the same dimensions is now faster.

- **Version 1.2 - Wednesday 11 January 2012**
  - ○ Added playback rate control, including reverse.
  - ○ Added audio balance control.

- **Version 1.0 - Thursday 22 December 2011**
  - ○ Initial release submitted to Asset Store.