VIDEO

View online: https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/video

The **Video** object can play a video inside a project. The video renders into the project canvas itself, allowing other objects to appear on top of it and effects to be applied, unlike with HTML elements.

Video formats

The supported video formats varies depending on the browser, operating system, and sometimes hardware. The most widely supported video format is probably MP4 H.264 but as a patent-encumbered format it is sometimes not possible to support in open source platforms. As of 2024 the modern and open codec AV1 is increasingly widely supported, but may not be supported on some older devices.

As a result to guarantee that video playback will work on all browsers and on all platforms, it may be necessary to encode your videos in two different formats. The Video plugin allows you to set two sources for a video: a **primary source** which is played if supported, otherwise it falls back to the **secondary source** (which is also only played if supported, but two formats should be enough to achieve universal support). This also allows choosing a modern efficient format which isn't universally supported as a primary source, and an older and less efficient but universally supported format as a secondary source, ensuring optimal playback on modern devices while still supporting older devices.

Identifying video formats

Video files actually encompass multiple different sub-formats. These include:

- **Container format:** this is usually WebM or MPEG-4 (MP4).
- **Video codec:** the codec used to compress video data, such as VP9, H.264 or AV1.
- **Audio codec:** if the video has an audio track, then this is the codec used to compress audio data, such as Opus or AAC.

Use a third-party tool to identify the container, video codec and audio codec used by a video file. For example VLC media player provides codec information via the menu Tools▶Codec information while playing a video.

Importing video

Be sure to import video files to the **Videos project folder**. If video files are added in any other project folder, e.g. *Files*, they may be exported to a different folder and fail to load.

Due to the complexities of video compression and the patent-encumbrance of some codecs like H.264, Construct does not provide a video importer like it does with audio. You must encode your video files yourself, and then import them as project files. WebM and its codecs VP8, VP9 and AV1 are open formats and you should be able to find free encoders. However H.264/H.265 encoders may involve a fee.

When publishing a project using video playback to the web, be sure that your server has the correct MIME types set up otherwise video playback may fail after export.

Playback restrictions

In some cases, video playback cannot begin unless triggered by a user input event. The *Play* action will work in a user input trigger like *On touch start*, but if done outside of that it cannot play right away. To work around this the video plugin will wait until the next touch event to start playing the video. This also applies to autoplaying videos: it will not start until the first touch.

This restriction generally only applies to web browsers. Usually if you publish an app, the restriction is able to be removed because the app can adjust the permissions.

Video properties

Primary source

Primary format

The name of the project file to play if the primary format is supported. Make sure *Primary format* matches the actual container and codec used by the specified project file. See *Video formats* above for more details.

Secondary source

Secondary format

The name of the project file to play, if the primary format is *not* supported. Make sure *Secondary format* matches the actual container and codec used by the specified project file. See *Video formats* above for more details.

Autoplay

The autoplay or preload mode. This can be:

- **No:** nothing is done until the video is requested to be played.
- **Preload:** on startup the video will start downloading the video data, but will not start playing it yet. This can allow video playback to start more quickly when requested. Some platforms (e.g. mobile devices on cellular data connections) may ignore this.

- **Yes:** on startup the video will start downloading the video data, and also start playing it as soon as it determines the progress and transfer rate are sufficient to play through to the end without stalling for buffering. Some mobile platforms will not start playing until the first touch event - see *Compatibility* for more information.

Play in background

If disabled, then switching browser tab, minimising the browser window, switching to a different mobile app, or otherwise hiding the window will pause the video and resume it when switching back. This is intended to avoid annoying the user with continued audio playback when deciding to do something else, and it also helps save battery on mobile devices. However for some types of app it may be desirable to keep playing in the background, in which case enabling this allows continued playback even when in the background.

Initially visibile

Whether the video is initially visible or invisible. Note that if it is invisible, audio playback may still be heard when playing, so it may be desirable to also mute the video.

Video conditions

Has ended

True if the video playback has reached the end of the video and stopped.

Is muted

True if the audio playback from the video has been muted.

Is paused

True if the video playback has been paused.

Is playing

True if the video playback is actively playing.

On playback event

Triggers when a playback event occurs. This can be one of:

- *Can play:* triggered when enough data is available to play at least a couple of frames, but there may not be enough data to play through to the end.

- *Can play through:* triggered when the browser determines that the load progress and transfer rate are sufficient for playback through to the end without stalling for buffering.

However this is not a guarantee, since the transfer rate could drop or be cut off completely.

- *Ended:* triggered when playback reaches the end of the video.
- *Error:* triggered if an error occurs during video loading, decoding or playback.
- *Started loading:* triggered when the browser begins loading video data.
- *Played:* triggered when playback begins.
- *Paused:* triggered upon pausing the video playback.
- *Stalled:* triggered if the video download rate is too slow to keep up the current playback rate. This will cause the video to pause while it finishes loading the rest of the video, also known as buffering.

Video actions

Pause

Pause the video playback if it is currently playing.

Play

Start playing the video. On some platforms this can only happen in a user input event. For more information, see the section on *Compatibility*.

Set looping

Set whether the video is looping, so that it restarts from the beginning when it reaches the end.

Set muted

Set whether the audio playback from the video is muted (inaudible) or unmuted.

Set playback rate

Set how fast the video playback proceeds, as a multiplier of its original speed. That means 1 is the original speed, 2 is twice as fast, 0.5 is half as fast, etc.

Set playback time

Set the video playback time to a specific time in seconds (i.e. seek to the given time). Due to the way video encoding technologies work, the video may only be able to seek close to but not exactly on the specified time.

Set source

Set a different video file to play. As with the object properties, different primary and secondary formats can be specified. Setting the source does not automatically start playing the video; use the *Play* action to start it after changing the source.

Set volume

Set the volume of the audio playback from the video, in decibels attenuation. 0 is full volume, -10 dB is approximately half as loud, etc. The audio cannot be amplified: positive volume values will be treated as 0.

Video expressions

Duration

The video duration in seconds, if the video has loaded enough for this to be determined.

PlaybackRate

The current playback rate as set by the *Set playback rate* action, as a multiplier of the original rate (e.g. 1 is original speed, 2 is twice as fast, etc).

PlaybackTime

The current playback time in seconds.

VideoWidth

VideoHeight

The dimensions of the source video, in pixels.

Volume

The current audio playback volume in dB attenuation.