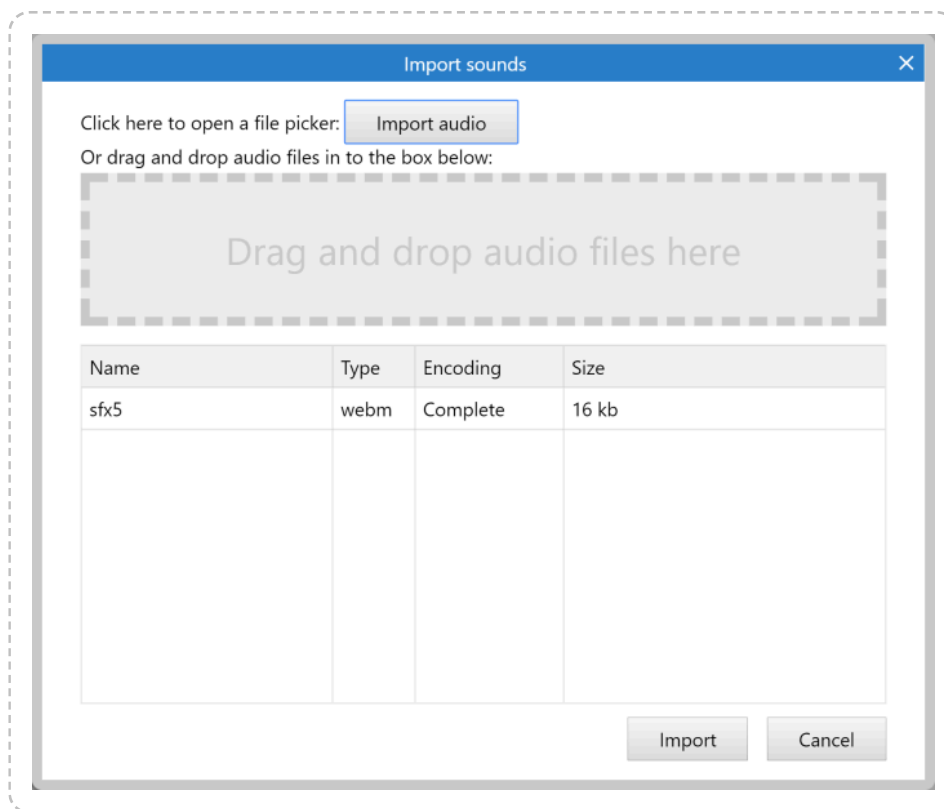


THE IMPORT AUDIO DIALOG

View online: <https://www.construct.net/en/make-games/manuals/construct-3/interface/dialogs/import-audio>

The **Import Audio dialog** allows audio files to be added to the project from disk. It is accessed by selecting Import Sounds or Import Music from the right-click menu of the *Sounds* or *Music* folders in the **Project Bar**. Be sure to add the Audio object to your project to play back sounds and music.



There are two ways to import audio files:

- 1 Click the **Import audio** button, which opens a file picker to choose some local files to import
- 2 **Drag-and-drop** some local files in to the area that says *Drag and drop audio files here*

As soon as you choose some files or drop them in, Construct will encode them to **WebM Opus** for best compatibility across platforms. It is recommended to import **16-bit PCM WAV** or **FLAC** files to ensure Construct can encode them without any quality loss. If you import files which are already WebM Opus, they will simply be copied to the project.

Construct will use all available CPU cores to encode imported audio, which is useful if you need to import a lot of audio files. The progress of encoding is displayed in the table of files. Once all encoding is done, you can click **Import** to add them to the project.

Categorise audio files correctly

It is important to organise audio files appropriately. Audio files in the **Sounds** project folder are loaded completely before playing, but files in the **Music** folder are streamed. This means if a music track is accidentally put in the *Sounds* folder, it would have to load completely before it can start playing. However, audio in the *Music* folder can start playing much more quickly since it is streamed, and also uses less memory for long tracks.

Platform compatibility

There is not one audio format that can play across all browsers and platforms with built-in support. To avoid having to use multiple audio formats, Construct uses its own WebM Opus decoder for Safari & iOS, which are the last platforms left that do not have built-in support for WebM Opus.

In most browsers Construct can transcode AAC or MP3 audio for you. However if the browser does not support decoding these formats, Construct may be unable to transcode these formats and may show a warning. It is strongly recommended to use WebM Opus for all audio in Construct projects, as it is the only format that is guaranteed to play on all platforms. You will need to use other software to encode your audio to WebM Opus, and then import the .webm files to your project. Construct will simply copy these files in to your project if they are already WebM Opus.

Import formats

The following formats can be imported to Construct. PCM WAV or FLAC files are recommended.

- **PCM WAV** (.wav): converted to WebM Opus
- **FLAC** (.flac): converted to WebM Opus
- **WebM** (.webm): copied to project if Opus, else converted to Opus
- **MPEG-4 AAC** (.m4a): converted to WebM Opus in most browsers (note this is lossy)
- **MP3** (.mp3): converted to WebM Opus in most browsers (note this is lossy)
- **Ogg Vorbis** (.ogg): converted to WebM Opus (note this is lossy)
- **Ogg Opus** (.opus): converted to WebM Opus (note this is lossy)

If you need to import a different format, you'll need to find third-party software to convert it. If you can, convert it directly to WebM Opus, as that is the preferred format for Construct.

Playback preference

Projects only need to use WebM Opus files to support all platforms. However in some cases, especially with older projects, audio files may be available in multiple formats in the project. In that case Construct uses the following order of preference to pick which to play at runtime. The first format in the list that has built-in support is used. If none of the available formats has built-in support, and there is a WebM Opus file available, then Construct falls back to the software decoder last of all.

- 1 WebM Opus
- 2 Ogg Vorbis (used by Construct 2)
- 3 MPEG-4 AAC
- 4 MP3
- 5 Software decode WebM Opus