

AUDIO SCRIPT INTERFACE

View online: <https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/plugin-interfaces/audio>

The `IAudioObjectType` interface derives from `IObjectClass` to add APIs specific to the `Audio plugin`.

The script interface essentially just provides access to the underlying `AudioContext` (part of the `Web Audio API`) that the `Audio` plugin uses internally for audio playback. However this is sufficient to provide complete control over audio playback, including setting up complex graphs of audio processing nodes. It is also convenient, since Construct manages some awkward details such as ensuring playback is enabled as soon as possible given most browsers impose autoplay restrictions. Adding the `Audio` object and using its script interface saves you from having to re-implement these details yourself.

Note this class derives from the object class interface, not the instance interface. Typically it is used through `runtime.objects.Audio`.

Since the Web Audio API is not available in Web Workers, the `Audio` object's script interface can only be used in DOM mode, i.e. with the project Use worker option turned off.

Examples

The following examples demonstrate using the `Web Audio API` for audio playback.

- [Audio scripting](#) demonstrates loading audio files and playing them
- [Sound synthesis](#) demonstrates generating sounds with code

Audio APIs

audioContext

The `Audio` plugin's internal `AudioContext` used for audio playback.

destinationNode

The destination node to connect any additional audio nodes to.

While `AudioContext` has its own `destination` property, in some cases Construct redirects the audio output to another destination, such as when recording playback. To achieve this it creates its own destination node which it can redirect the output from. By

connecting your own nodes to this destination your script's audio output will properly integrate with other Construct features like recording.

isSilent

Set or get a boolean indicating whether *Silent mode* is enabled. In silent mode all currently playing sounds are muted and no new sounds will play.

masterVolume

Set or get the current master volume, which is applied to all audio playback, in dB.

stopAll()

Stop all currently playing sounds.