

# roAudioPlayer

## ON THIS PAGE

- [ifMessagePort](#)
  - [SetPort\(port As roMessagePort\) As Void](#)
- [ifUserData](#)
  - [SetUserData\(user\\_data As Object\)](#)
  - [GetUserData\(\) As Object](#)
- [IfIdentity](#)
  - [GetIdentity\(\) As Integer](#)
- [ifMediaTransport](#)
- [ifAudioControl](#)
  - [SetPcmAudioOutputs\(outputs As roArray\) As Boolean](#)
  - [SetCompressedAudioOutputs\(outputs As roArray\) As Boolean](#)
  - [SetMultichannelAudioOutputs\(array As Object\) As Boolean](#)
  - [SetAudioOutput\(audio\\_output As Integer\) As Boolean](#)
  - [SetAudioMode\(audio\\_mode As Integer\) As Boolean](#)
  - [MapStereoOutput\(mapping As Integer\) As Boolean](#)
  - [MapDigitalOutput\(mapping As Integer\) As Boolean](#)
  - [SetVolume\(volume As Dynamic\) As Boolean](#)
  - [SetChannelVolumes\(channel\\_mask As Integer, volume As Integer\) As Boolean](#)
  - [SetPreferredAudio\(description As String\) As Boolean](#)
  - [SetUsbAudioPort\(a As Integer\) As Boolean](#)
  - [SetSpdifMute\(a As Boolean\) As Boolean](#)
  - [SetStereoMappingSpan\(a As Integer\) As Boolean](#)
  - [ConfigureAudioResources\(\) As Boolean](#)
  - [SetAudioStream\(stream\\_index As Integer\) As Boolean](#)
  - [SetAudioDelay\(delay\\_in\\_milliseconds As Integer\) As Boolean](#)
  - [SetVideoDelay\(delay\\_in\\_milliseconds As Integer\) As Boolean](#)
  - [StoreEncryptionKey\(a As String, b As String\) As Boolean](#)
  - [StoreObfuscatedEncryptionKey\(a As String, b As String\) As Boolean](#)
  - [SetAudioOutputAux\(audio\\_output As Integer\) As Boolean](#)
  - [SetAudioModeAux\(audio\\_mode As Integer\) As Boolean](#)
  - [MapStereoOutputAux\(mapping As Integer\) As Boolean](#)
  - [SetVolumeAux\(volume As Integer\) As Boolean](#)
  - [SetChannelVolumesAux\(channel\\_mask As Integer, volume As Integer\) As Boolean](#)
  - [SetAudioStreamAux\(stream\\_index As Integer\) As Boolean](#)
- [Configuring Audio Outputs](#)
- [Playing Multiple Audio Files Simultaneously](#)

## OS8

- [OS8](#)
- [Version 7.1](#)
- [Version 7.0](#)
- [Version 6.2](#)
- [Version 6.1](#)
- [Previous Versions](#)

An audio player is used to play back audio files using the generic *ifMediaTransport* interface. If the message port is set, the object will send events of the type *roAudioEvent*. All object calls are asynchronous. In other words, audio playback is handled in a different thread from the script. The script may continue to run while audio is playing.

## ifMessagePort

**SetPort(port As roMessagePort) As Void**

Posts messages of type *roAudioEvent* to the attached message port.

## ifUserData

### SetUserData(user\_data As Object)

Sets the user data that will be returned when events are raised.

### GetUserData() As Object

Returns the user data that has previously been set via *SetUserData()*. It will return *Invalid* if no data has been set.

## IfIdentity

### GetIdentity() As Integer

#### Note

The *ifIdentity* interface has been deprecated. We recommend using the *ifUserData* interface instead.

## ifMediaTransport

See *roVideoPlayer* for a description of *ifMediaTransport* methods.

## ifAudioControl

### SetPcmAudioOutputs(outputs As roArray) As Boolean

Specifies which audio connectors should output PCM audio. This method accepts one or more outputs in the form of an *roArray* of *roAudioOutput* instances.

### SetCompressedAudioOutputs(outputs As roArray) As Boolean

Specifies which audio connectors should output compressed audio (e.g. Dolby AC3 encoded audio). This method accepts one or more outputs in the form of an *roArray* of *roAudioOutput* instances. When one or both of the above output methods are called, they will override the settings of the following *ifAudioControl* methods: *SetAudioOutput()*, *MapStereoOutput()*, *SetUsbAudioPort()*, *MapDigitalOutput()*.

### SetMultichannelAudioOutputs(array As Object) As Boolean

### SetAudioOutput(audio\_output As Integer) As Boolean

Configures the audio output of the *roAudioPlayer* object. This method accepts the following values:

- 0: Analog audio
- 1: USB audio
- 2: Digital audio, stereo PCM
- 3: Digital audio, raw AC3
- 4: Onboard analog audio with HDMI mirroring raw AC3

### SetAudioMode(audio\_mode As Integer) As Boolean

Sets the audio mode of the *roAudioPlayer* object. This method accepts the following values:

- 0: AC3 Surround
- 1: AC3 mixed down to stereo
- 2: No audio
- 3: Left
- 4: Right

#### Note

Options 0 and 1 only apply to video files, while options 2, 3, and 4 apply to all audio sources.

### MapStereoOutput(mapping As Integer) As Boolean

Determines which output to use when the output is set to the analog audio (i.e. `SetAudioOutput(0)`).

- 0: Stereo audio is mapped to onboard analog output.
- 1: Stereo audio is mapped to left output of the expansion module.
- 2: Stereo audio is mapped to middle output of the expansion module.
- 3: Stereo audio is mapped to right output of the expansion module.

### MapDigitalOutput(mapping As Integer) As Boolean

Maps the digital audio output from the `roAudioPlayer` object. This method accepts the following values:

- 0: Onboard HDMI
- 1: SPDIF from expansion module

### SetVolume(volume As Dynamic) As Boolean

Specifies the volume of the audio output as either a percentage or decibel amount. To use a percentage measurement, pass an integer value between 0 and 100. To use a decibel measurement, pass an `roAssociativeArray` containing a single `{db:<value As Float>}` parameter. The decibel measurement is an absolute value: passing 0 specifies no change to the audio output, and the effective range of measurements is from approximately -80 to 20 decibels.

The volume value is clipped prior to use (i.e. `SetVolume(101)` will set the volume to 100 and return `True`). The volume is the same for all mapped outputs and USB/SPDIF/analog.

Separate volume levels are stored for `roAudioPlayer` and `roVideoPlayer`.

### SetChannelVolumes(channel\_mask As Integer, volume As Integer) As Boolean

You can control the volume of individual audio channels. This volume command takes a hex channel mask, which determines the channels to apply the volume to, and a level, which is a percentage of the full scale. The volume control works according to audio channel rather than the output. The channel mask is a bit mask with the following bits for MP3 output:

- `&H01`: Left
- `&H02`: Right
- `&H03`: Both left and right

### SetPreferredAudio(description As String) As Boolean

Selects which audio track to use if there are multiple audio tracks in the stream. See the [Preferred Streams](#) section on the `roVideoPlayer` page for more details.

### SetUsbAudioPort(a As Integer) As Boolean

### SetSpdifMute(a As Boolean) As Boolean

### SetStereoMappingSpan(a As Integer) As Boolean

### ConfigureAudioResources() As Boolean

### SetAudioStream(stream\_index As Integer) As Boolean

### **SetAudioDelay(delay\_in\_milliseconds As Integer) As Boolean**

Adds a presentation time stamp (PTS) offset to the audio. This makes it possible to adjust for file multiplexing differences. Delays are limited to 150ms or less. Currently, the system software only supports positive delays; therefore, if you need to set the audio ahead of the video, you will need to use `SetVideoDelay()` instead.

### **SetVideoDelay(delay\_in\_milliseconds As Integer) As Boolean**

Adds a presentation time stamp (PTS) offset to the video. This makes it possible to adjust for file multiplexing differences. Delays are limited to 150ms or less.

### **StoreEncryptionKey(a As String, b As String) As Boolean**

### **StoreObfuscatedEncryptionKey(a As String, b As String) As Boolean**

**HD2000 Only:**

### **SetAudioOutputAux(audio\_output As Integer) As Boolean**

### **SetAudioModeAux(audio\_mode As Integer) As Boolean**

### **MapStereoOutputAux(mapping As Integer) As Boolean** **SetVolumeAux(volume As Integer) As Boolean**

### **SetChannelVolumesAux(channel\_mask As Integer, volume As Integer) As Boolean**

### **SetAudioStreamAux(stream\_index As Integer) As Boolean**

## **Configuring Audio Outputs**

If a audio/video file is playing or has played, you need to call `Stop()` before changing the audio output.

The following script shows how to use the `SetPcmAudioOutputs()` and `SetCompressedAudioOutputs()` methods in conjunction with *roAudioOutput*. The audio/video player is configured to output decoded audio to the analog output or compressed audio to the HDMI and SPDIF outputs.

```
a1=CreateObject("roAudioOutput", "Analog")
a2=CreateObject("roAudioOutput", "HDMI")
a3=CreateObject("roAudioOutput", "SPDIF")

a1=CreateObject("roAudioPlayer")
a1.SetPcmAudioOutputs(a1)

ar = CreateObject("roArray", 2, true)
ar[0] = a2
ar[1] = a3
a1.SetCompressedAudioOutputs(ar)
```

#### **Important**

In most cases, rerouting audio outputs during audio/video playback will cause playback to stop. The system software will still be responsive, so you can use commands to exit playback during or after an audio output modification.

This code sets audio output to the rightmost expansion module audio port:

```
video = CreateObject("roVideoPlayer")
video.SetAudioOutput(0)
video.MapStereoOutput(3)
```

This code sets the volume level for individual channels:

```
audio = CreateObject("roAudioPlayer")
audio.SetChannelVolumes(&H01, 60)      'left channel to 60%
audio.SetChannelVolumes(&H02, 75)      'right channel to 75%
audio.SetChannelVolumes(&H03, 65)      'all channels to 65%
```

## Playing Multiple Audio Files Simultaneously

Multiple MP3 files, as well as the audio track of a video file, can be played to any combination of the following:

- Analog outputs
- SPDIF / HDMI
- USB

Only a single file can be sent to an output at any given time. For example, two *roAudioPlayers* cannot simultaneously play to the SPDIF output. The second one to attempt a *PlayFile* will get an error. To free an output, the audio or video stream must be stopped (using the *ifMediaTransport* *Stop* or *StopClear* calls).

Note the following about multiple audio-file functionality:

- The onboard analog audio output and HDMI output are clocked by the same sample-rate clock. Therefore, if different content is being played out of each, the content must have the same sample rate.
- Currently, only a single set of USB speakers is supported.
- Each audio and video stream played consumes some of the finite CPU resources. The amount consumed depends on the bitrates of the streams. Testing is the only way to determine whether a given set of audio files can be played at the same time as a video. The maximum recommended usage is a 16Mbps video file with three simultaneous MP3 160kbps streams.

This code plays a video with audio over HDMI and an MP3 file to the onboard analog port.

```
video=CreateObject("roVideoPlayer")

video.SetAudioOutput(3)
video.PlayFile("video.mpg")

audio=CreateObject("roAudioPlayer")

audio.MapStereoOutput(0)
audio.PlayFile("audio.mp3")
```